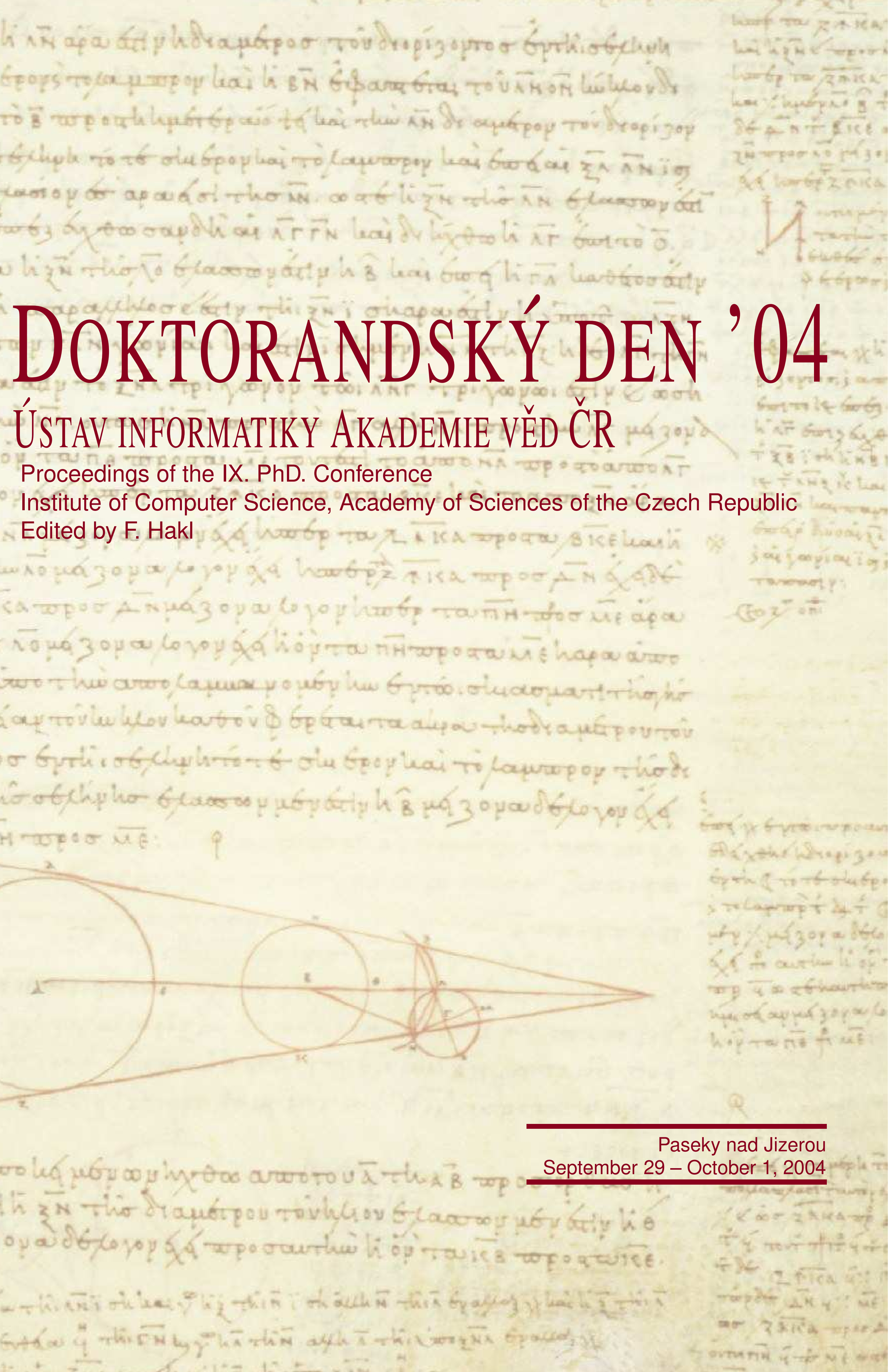


DOKTORANDSKÝ DEN ’04

ÚSTAV INFORMATIKY AKADEMIE VĚD ČR

Proceedings of the IX. PhD. Conference
Institute of Computer Science, Academy of Sciences of the Czech Republic
Edited by F. Hakl

Paseky nad Jizerou
September 29 – October 1, 2004

Doktorandský den ’04

Ústav informatiky
Akademie věd České republiky

Paseky nad Jizerou, 29. září – 1. říjen 2004

matfyzpress
vydavatelství Matematicko-fyzikální fakulty
University Karlovy v Praze

Publikaci ”Doktorandský den ’04” sestavil a připravil k tisku František Hakl
Ústav Informatiky AV ČR, Pod Vodárenskou věží 2, 182 07 Praha 8

ISBN 80-86732-30-4

Obsah

Libor Běhounek: **Formal Semantics for Fuzzy Yes-No Questions** **0**

Lubomír Bulej, Tomáš Bureš: **Connectors in the Context of OMG D&C Specification** **7**

Petr Cintula: **Functional Representation for Product Logic** **22**

Jakub Dvořák: **Změkčování hran v rozhodovacích stromech** **29**

Roman Kalous: **Evolutionary operators on ICodes** **35**

Zdeněk Konfršt: **Strong Definition of Performance Metrics and Parallel Genetic Algorithms** **42**

Emil Kotrč: **Leaf Confidences for Random Forests** **49**

Pavel Krušina: **Models of Multi-Agent Systems** **58**

Petra Kudová: **Kernel Based Regularization Networks and RBF Networks** **59**

Zdeňka Linková: **Integrace dat a sémantický web** **66**

Radim Nedbal: **Relational Databases with Ordered Relations** **75**

Petra Přečková: **Digitální knihovny, biomedicínská data a znalosti** **84**

Petr Rálek: **Modelling of Piezoelectric Materials** **91**

Jindra Reissigová: **Estimations of Cardiovascular Disease RiskA survey of our results from 2004** **101**

Patrícia Rexová: **Item Analysis of Educational Tests in System ExaME** **107**

Martin Řimnáč: **Rekonstrukce databázového modelu na základě dat (studie proveditelnosti)** **113**

Milan Rydvan: **Alternative Target Functions for Multilayer Neural Networks** **121**

Martin Saturka: **Short Survey on Bioinformatics with Fuzzy Logic** **129**

Terezie Šidlofová: **Kernel Based Regularization and Neural Networks** **134**

Milan Šimůnek: **Automatická tvorba analytického popisu systému** **141**

Roman Špánek: **Security in Mobile Environment** **149**

Josef Špidlen: **MUDRLite - Health Record Tailored to Your Needs** **156**

Filip Zámek: **Probabilistic Clinical Decision Support Systems** **164**

In Greek, Tenth century

This is the oldest and best manuscript of a collection of early Greek astronomical works, mostly elementary, by Autolycus, Euclid, Aristarchus, Hypsicles, and Theodosius, as well as mathematical works. The most interesting, really curious, of these is Aristarchus's "On the Distances and Sizes of the Sun and Moon", in which he shows that the sun is between 18 and 20 times the distance of the moon. Shown here is Proposition 13, with many scholia, concerned with the ratio to the diameters of the moon and sun of the line subtending the arc dividing the light and dark portions of the moon in a lunar eclipse.

Vat. gr. 204 fol. 116 recto math06 NS.02

Doktorandský den Ústavu informatiky Akademie věd České republiky se koná již po osmé, nepřetržitě od roku 1996. Tento seminář poskytuje doktorandům, podílejícím se na odborných aktivitách Ústavu informatiky prezentační možnosti pro výsledky jejich odborného studia a následného směrování jejich výzkumu. Současně poskytuje prostor pro oponentní připomínky k přednášené tematice a použité metodologii práce, ze strany přítomné odborné komunity.

Z jiného úhlu pohledu, toto setkání doktorandů podává průřezovou informaci o odborném rozsahu výzkumu, který je realizován na pracovištích či za spoluúčasti Ústavu informatiky AV ČR.

Od počátku organizování doktorandského dne byl soubor přednesených příspěvků publikován formou technických zpráv ÚI, které jsou dostupné elektronicky na adrese `www.cs.cas.cz`. V tomto ročníku se již podruhé přistoupilo k vydání příspěvků v knižní formě. Jednotlivé příspěvky v publikaci jsou uspořádány podle jmen autorů, neboť uspořádání podle tematického zaměření nepovažujeme za účelné, vzhledem k rozmanitosti jednotlivých témat.

Vedení ÚI a vědecká rada ÚI jakožto organizátor tohoto odborného setkání věří, že toto setkání mladých doktorandů, jejich školitelů a ostatní odborné veřejnosti povede ke zkvalitnění celého procesu doktorandského studia zajišťovaného v součinnosti s Ústavem informatiky, k odborné a pedagogické sebereflexi jak na straně doktorandů tak i na straně školitelů, a v neposlední řadě k navázání a vyhledání nových odborných kontaktů.

František Hakl
editor
1. září 2004

Formal Semantics for Fuzzy Yes-No Questions

Post-Graduate Student:
MGR. LIBOR BĚHOUNEK
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
behounek@cs.cas.cz

Supervisor:
DOC. PHDR. PETR JIRKŮ, CSC.
Filosofická fakulta
Universita Karlova
Celetná 20
Praha 1
petr.jirku@ff.cuni.cz

Field of Study:
Logics
Classification: 6101 V

Abstract

The paper is a short overview of the generalization of Groenendijk-Stokhof's system of erotetic logic (also known as the partition semantics of questions) to fuzzy questions. Fuzzy intensional semantics, necessary for Groenendijk-Stokhof's system, is developed within Henkin-style second-order fuzzy logic, which is introduced first. Our attention is restricted to fuzzy yes-no questions.

Acknowledgments

This paper is an overview of my paper [1] on fuzzy semantics of yes-no questions. It employs the formalism and results of [2], which is my joint work with Petr Cintula. The work on this paper was supported by the grant of the Grant Agency of the Czech Republic No. GA ČR DG 401/03/H047 *Logical foundations of semantics and knowledge representation.* The co-advisor for my research in the area of fuzzy logic is Prof. RNDr. Petr Hájek, DrSc.

1. Introduction

Fuzzy logic is a kind of many-valued logics aimed at capturing the laws of inference under a certain kind of vagueness, explicable in terms of degrees of truth. Recent advances in metamathematics of fuzzy logic (esp. [3]) provided a solid background for an axiomatic development of other areas of fuzzy logic in broad sense. One of the fields in which formal fuzzy logic can fruitfully be applied is erotetic logic, or the logic of questions. The importance of fuzzy erotetic logic is seen from the fact than many questions in natural language ask for information about fuzzy predicates. Furthermore, questionnaires often employ scaled answers (e.g., yes, rather yes, rather no, no), which can be handled by fuzzy logic.

The paper [1] develops a fuzzy generalization of Groenendijk-Stokhof's system of erotetic logic, described in [4] and [5]. Since Groenendijk-Stokhof's system (also known as the *partition semantics of questions*) is based on intensional semantics of classical logic, fuzzy intensional semantics is developed first, within the framework of Henkin-style second-order fuzzy logic, described in [2]. Our attention is restricted to fuzzy yes-no questions.

Section 2 describes axiomatic elementary fuzzy set theory developed within Henkin-style second-order fuzzy logic. Section 3 gives an overview of classical intensional semantics and Groenendijk-Stokhof logic of yes-no questions. Section 4 describes formal fuzzy intensional semantics, which is then employed for formal semantics of fuzzy questions in Section 5 .

2. Fuzzy theory of classes

Fuzzy theory of classes FCT, developed in [2], is an axiomatization of Zadeh's notion of fuzzy set by means of Henkin-style second-order fuzzy logic ŁΠ. It is capable of providing a framework for various branches of fuzzy mathematics, incl. fuzzy intensional semantics. This section repeats basic definitions and some of the results of [2]. For details on (zeroth- and first-order) logic ŁΠ see [6] and [7].

Convention 2.1 *Unless stated otherwise, the expression $\varphi(x_1, \ldots, x_n)$ implies that all free variables of φ are among $x_1, \ldots, x_n$.*

Convention 2.2 *Let $\varphi(p_1, \ldots, p_n)$ be some propositional formula and $\psi_1, \ldots, \psi_n$ any formulae. By $\varphi(\psi_1, \ldots, \psi_n)$ we denote the formula φ in which all the occurrences of p_i are replaced by ψ_i.*

Convention 2.3 *We omit indices of defined t-norm connectives of logic* ŁΠ *whenever they do not matter, i.e., whenever $\Delta(\varphi \rightarrow_* \psi) \leftrightarrow_{\mathrm{L}} \Delta(\varphi \rightarrow_\diamond \psi)$ is provable in (propositional or predicate)* ŁΠ *for arbitrary t-norms $*$ and $\diamond$ expressible in* ŁΠ.

Class theory FCT is a theory over logic ŁΠ∀ with two sorts of variables: *object variables,* denoted by lowercase letters $x, y, \ldots$, and *class variables,* denoted by uppercase letters $X, Y, \ldots$. The only primitive predicate of FCT is the binary membership predicate $\in$ between objects and classes. The principal axioms of FCT are the instances of the *class comprehension scheme*

$$(\exists X)\, \Delta\, (\forall x)\, (x \in X \leftrightarrow \varphi(x))$$

where φ can contain any class or object parameters except for X. The Skolem functions of comprehension axioms are (eliminably) introduced as comprehension terms $\{x \mid \varphi(x)\}$ with axioms

$$y \in \{x \mid \varphi(x)\} \leftrightarrow \varphi(y)$$

(φ in a comprehension term may be allowed to contain other comprehension terms). The intended notion of fuzzy class is extensional, therefore we require the *axiom of extensionality* which identifies classes with their membership functions:

$$(\forall x)\, \Delta(x \in X \leftrightarrow x \in Y) \rightarrow X = Y$$

The *axiom of fuzziness* $c \in C \leftrightarrow \neg_{\mathrm{L}} c \in C$ guarantees the existence of non-crisp sets. FCT is further enriched in the obvious way by functions and axioms for handling with tuples of objects. The intended models interpret classes as all functions from the domain of object variables to a suitable ŁΠ algebra (standardly, $[0, 1]$). The following definitions show that FCT contains the apparatus of elementary fuzzy set theory.

Definition 2.1 (Class operations and relations) *Let $\varphi(p_1, \ldots, p_n)$ be a propositional formula. We define the n-ary* class operation generated by φ *as*

$$\mathrm{Op}_\varphi(X_1, \ldots, X_n) =_{\mathrm{df}} \{x \mid \varphi(x \in X_1, \ldots, x \in X_n)\}.$$

The n-ary uniform relation *between $X_1, \ldots, X_n$ generated by φ is defined as*

$$\mathrm{Rel}^{\forall}_\varphi(X_1, \ldots, X_n) \equiv_{\mathrm{df}} (\forall x)\, \varphi(x \in X_1, \ldots, x \in X_n).$$

The n-ary supremal relation *between $X_1, \ldots, X_n$ generated by φ is defined as*

$$\mathrm{Rel}^{\exists}_\varphi(X_1, \ldots, X_n) \equiv_{\mathrm{df}} (\exists x)\, \varphi(x \in X_1, \ldots, x \in X_n).$$

We use the notational abbreviations of common relations and operations, summarized in Tables 1 and 2. The following (meta)theorems effectively reduce elementary fuzzy set theory to fuzzy propositional calculus.

Table 1: Class operations

φ	$\mathrm{Op}_\varphi(X_1,\dots,X_n)$	Name
0	$\emptyset$	empty class
1	V	universal class
$\Delta(\alpha \to p)$	X_α	α-cut
$\Delta(\alpha \leftrightarrow p)$	$X_{=\alpha}$	α-level
$\neg_{\mathrm{G}} p$	$\backslash X$	strict complement
$\neg_{\mathrm{L}} p$	$-X$	involutive complement
$\neg_{\mathrm{G}}\neg_{\mathrm{L}} p$ (or Δp)	$\mathrm{Ker}(X)$	kernel
$\neg\neg_{\mathrm{G}} p$ (or $\neg\Delta\neg_{\mathrm{L}} p$)	$\mathrm{Supp}(X)$	support
$p \mathbin{\&_*} q$	$X \cap_* Y$	$*$-intersection
$p \vee q$	$X \cup Y$	union
$p \oplus q$	$X \uplus Y$	strong union
$p \mathbin{\&_*} \neg_{\mathrm{G}} q$	$X \setminus_* Y$	strict $*$-difference
$p \mathbin{\&_*} \neg_{\mathrm{L}} q$	$X -_* Y$	involutive $*$-difference

Table 2: Class properties and relations

Relation	Notation	Name
$\mathrm{Rel}^{\exists}_{p}(X)$	$\mathrm{Hgt}(X)$	height
$\mathrm{Rel}^{\exists}_{\Delta p}(X)$	$\mathrm{Norm}(X)$	normality
$\mathrm{Rel}^{\forall}_{\Delta(p \vee \neg p)}(X)$	$\mathrm{Crisp}(X)$	crispness
$\mathrm{Rel}^{\exists}_{\neg\Delta(p \vee \neg p)}(X)$	$\mathrm{Fuzzy}(X)$	fuzziness
$\mathrm{Rel}^{\forall}_{p \to_* q}(X,Y)$	$X \subseteq_* Y$	$*$-inclusion
$\mathrm{Rel}^{\forall}_{p \leftrightarrow_* q}(X,Y)$	$X \approx_* Y$	$*$-equality
$\mathrm{Rel}^{\exists}_{p \mathbin{\&_*} q}(X,Y)$	$X \Vert_* Y$	$*$-compatibility

Theorem 2.4 *Let $\varphi, \psi_1, \dots, \psi_n$ be propositional formulae.*

Then $\vdash \varphi(\psi_1, \dots, \psi_n)$

$$\begin{aligned} \textit{iff} \quad & \vdash \mathrm{Rel}^{\forall}_{\varphi}(\mathrm{Op}_{\psi_1}(X_{1,1},\dots,X_{1,k_1}),\dots,\mathrm{Op}_{\psi_n}(X_{n,1},\dots,X_{n,k_n})) \\ \textit{iff} \quad & \vdash \mathrm{Rel}^{\exists}_{\varphi}(\mathrm{Op}_{\psi_1}(X_{1,1},\dots,X_{1,k_1}),\dots,\mathrm{Op}_{\psi_n}(X_{n,1},\dots,X_{n,k_n})) \end{aligned}$$

Theorem 2.5 *Let $\varphi_i, \varphi'_i, \psi_{i,j}, \psi'_{i,j}$ be propositional formulae. Then*

$$\vdash \mathop{\&_*}_{i=1}^{k} \varphi_i(\psi_{i,1},\dots,\psi_{i,n_i}) \to \bigwedge_{i=1}^{k'} \varphi'_i(\psi'_{i,1},\dots,\psi'_{i,n'_i})$$

iff

$$\vdash \mathop{\&_*}_{i=1}^{k} \mathrm{Rel}^{\forall}_{\varphi_i}\left(\mathrm{Op}_{\psi_{i,1}}(\vec{X}),\dots,\mathrm{Op}_{\psi_{i,n_i}}(\vec{X})\right) \to \bigwedge_{i=1}^{k'} \mathrm{Rel}^{\forall}_{\varphi'_i}\left(\mathrm{Op}_{\psi'_{i,1}}(\vec{X}),\dots,\mathrm{Op}_{\psi'_{i,n'_i}}(\vec{X})\right)$$

Theorem 2.6 *Let $\varphi_i, \varphi'_i, \psi_{i,j}, \psi'_{i,j}$ be propositional formulae. Then*

$$\vdash \mathop{\&_*}_{i=1}^{k} \varphi_i(\psi_{i,1},\dots,\psi_{i,n_i}) \to \bigvee_{i=1}^{k'} \varphi'_i(\psi'_{i,1},\dots,\psi'_{i,n'_i})$$

iff

$$\vdash \mathop{\&_*}_{i=1}^{k-1} \mathrm{Rel}^{\forall}_{\varphi_i}\left(\mathrm{Op}_{\psi_{i,1}}(\vec{X}),\dots,\mathrm{Op}_{\psi_{i,n_i}}(\vec{X})\right) \&_* \mathrm{Rel}^{\exists}_{\varphi_k}\left(\mathrm{Op}_{\psi_{k,1}}(\vec{X}),\dots,\mathrm{Op}_{\psi_{k,n_k}}(\vec{X})\right) \to$$
$$\to \bigvee_{i=1}^{k'} \mathrm{Rel}^{\exists}_{\varphi'_i}\left(\mathrm{Op}_{\psi'_{i,1}}(\vec{X}),\dots,\mathrm{Op}_{\psi'_{i,n'_i}}(\vec{X})\right)$$

Further theorems of [2] show that any classical formal theory can be reproduced within FCT. Furthermore, there is a method of the natural fuzzification of the concepts of any such theory, as well as a method of controlled 'defuzzification' if some concepts are to be kept crisp.

3. Classical intensional semantics of propositions and yes-no questions

In this section we repeat the basic definitions of intensional semantics for classical propositional logic and classical Groenendijk-Stokhof semantics of yes-no questions (denoted by GS). For details, see [4] and [5].

Definition 3.1 (Classical intensional semantics) *Let W be a non-empty set. A* valuation in W *is a function $\|\cdot\|$ taking formulae to subsets of W, such that*

$$\begin{aligned} \|\neg\varphi\| &= W - \|\varphi\| \\ \|\varphi \,\&\, \psi\| &= \|\varphi\| \cap \|\psi\| \\ \|\varphi \vee \psi\| &= \|\varphi\| \cup \|\psi\| \\ \|\varphi \to \psi\| &= (W - \|\varphi\|) \cup \|\psi\| \end{aligned}$$

The pair $\mathcal{W} = \langle W, \|\cdot\|\rangle$ is called a logical space, *the elements of W are called* indices *or* possible worlds, *the subsets of W* propositions. *The proposition $\|\varphi\|$ is called the* intension *of φ (in $\mathcal{W}$). The truth value of $w \in \|\varphi\|$ for $w \in W$ is called the* extension of φ in w *and denoted by $\|\varphi\|_w$.*

A formula φ holds *in a logical space $\mathcal{W} = \langle W, \|\cdot\|\rangle$ (written $\mathcal{W} \models \varphi$) iff $\|\varphi\| = W$. A formula φ is a* tautology *(written $\models \varphi$) iff it holds in any logical space. A formula φ* entails *a formula ψ* in $\langle W, \|\cdot\|\rangle$ *iff $\|\varphi\| \subseteq \|\psi\|$. A formula φ* entails *a formula ψ (written $\varphi \models \psi$) iff φ entails ψ in any logical space.*

Theorem 3.1 (Adequacy of classical intensional semantics) *A formula is provable in classical propositional calculus iff it is a tautology of intensional semantics.*

GS extends this semantics to interrogative formulae $?\varphi$ (read *whether* φ), where φ is any propositional formula.

Definition 3.2 (Semantics of interrogative formulae) *Let $\mathcal{W} = \langle W, \|\cdot\|\rangle$ be a logical space.*

The extension $\|?\varphi\|_w$ *of $?\varphi$ in $w \in W$ is the proposition $\{w' \in W \mid \|\varphi\|_{w'} = \|\varphi\|_w\}$.*

The intension $\|?\varphi\|$ *of $?\varphi$ in $\mathcal{W}$ is the equivalence relation $\{\langle w, w'\rangle \in W^2 \mid \|\varphi\|_w = \|\varphi\|_{w'}\}$. The partition of W induced by this equivalence relation will be denoted by $W/\,\|?\varphi\|$.*

Definition 3.3 (Answerhood and interrogative entailment) *Let $\mathcal{W} = \langle W, \|\cdot\|\rangle$ be a logical space.*

We say that ψ is a direct answer *to $?\varphi$ in $\mathcal{W}$ iff $\|\psi\| \in W/\,\|?\varphi\|$. We say that ψ is an* answer *to $?\varphi$ in $\mathcal{W}$ (written $\psi \models^{\mathcal{W}} ?\varphi$) iff $\|\psi\|$ entails a direct answer to $?\varphi$ in $\mathcal{W}$.*

We say that $?\psi$ entails *$?\varphi$ in $\mathcal{W}$ (written $?\psi \models^{\mathcal{W}} ?\varphi$) iff every answer to $?\psi$ is an answer to $?\varphi$ in $\mathcal{W}$. We say that $?\psi$ and $?\varphi$ are* equivalent *in $\mathcal{W}$ (written $?\psi \equiv^{\mathcal{W}} ?\varphi$) iff $?\psi$ entails $?\varphi$ in $\mathcal{W}$ and vice versa.*

We say that these relations hold generally iff they hold in any logical space.

Theorem 3.2

$$\begin{array}{rcl} ?\psi \models^{\langle W, \|\cdot\| \rangle} ?\varphi & \text{iff} & W/\,\|?\psi\| \subseteq W/\,\|?\varphi\| \\ ?\psi \equiv^{\langle W, \|\cdot\| \rangle} ?\varphi & \text{iff} & W/\,\|?\psi\| = W/\,\|?\varphi\| \end{array}$$

Corollary 3.3

$$\begin{array}{rcl} ?\varphi \models ?\psi & \text{iff} & \varphi \equiv \psi, \text{ or } \varphi \equiv \neg\psi, \text{ or } \psi \equiv \bot, \text{ or } \psi \equiv \top \\ ?\varphi \equiv ?\psi & \text{iff} & \varphi \equiv \psi, \text{ or } \varphi \equiv \neg\psi \end{array}$$

4. Fuzzy intensional semantics

In this section we generalize classical intensional semantics to fuzzy intensional semantics, i.e., we allow propositions to be fuzzy sets. We define the semantical notions of intensional semantics in FCT, in order to be able to prove theorems on entailment axiomatically.

Definition 4.1 (Formal fuzzy intensional semantics) *The translation $\|\cdot\|$ of the formulae of propositional* ŁΠ *to* FCT *is defined as follows:*

- *The translation $\|p_i\|$ of an atomic formula p_i is a class variable A_i.*
- *The translation of a complex formula $\varphi(p_1, \dots, p_n)$ is*

$$\|\varphi(p_1, \dots, p_n)\| =_{\mathrm{df}} \mathrm{Op}_{\varphi}(\|p_1\|, \dots, \|p_n\|).$$

The semantic notions of tautologicity, entailment, and logical equivalence (relative to an ŁΠ*-definable t-norm $*$) are defined as the following formulae of* FCT*:*

$$\begin{array}{rcl} \models_* \varphi & \equiv_{\mathrm{df}} & W \subseteq_* \|\varphi\| \\ \varphi \models_* \psi & \equiv_{\mathrm{df}} & W \cap_* \|\varphi\| \subseteq_* \|\psi\| \\ (\varphi \equiv_* \psi) & \equiv_{\mathrm{df}} & (\varphi \models_* \psi) \,\&_*\, (\psi \models_* \varphi) \end{array}$$

The notation can be generalized to any class terms of FCT*, writing $\models_* A$ for $W \subseteq_* A$, etc.*

Notice that the defined semantic notions generally need not be crisp.

Theorem 4.1 (Adequacy of formal fuzzy intensional semantics)

$$\begin{array}{rcl} \text{ŁΠ} \vdash \varphi & \text{iff} & \mathrm{FCT} \vdash (\models \varphi) \\ \text{ŁΠ} \vdash \varphi \to \psi & \text{iff} & \mathrm{FCT} \vdash (\varphi \models \psi) \end{array}$$

The precondition $\Delta(W \subseteq W \cap_* W)$ of all of the following theorems is automatically satisfied for crisp W, or any W if $*$ is G. Individual statements hold even under weaker preconditions (e.g. the first under $W \subseteq_* W \cap_* W \cap_* W$).

Theorem 4.2 (Properties of fuzzy entailment) *It is provable in* FCT *that* $\Delta(W \subseteq W \cap_* W)$ *implies*

$$\begin{aligned}
[(A \models_* B)\ \&_*\ (B \models_* C)] &\quad\rightarrow\quad (A \models_* C)\\
[(A \equiv_* B)\ \&_*\ (B \equiv_* C)] &\quad\rightarrow\quad (A \equiv_* C)\\
[(A \equiv_* A')\ \&_*\ (B \equiv_* B')] &\quad\rightarrow\quad [(A \models_* B) \leftrightarrow_* (A' \models_* B')]\\
(\varphi \models_* \psi) &\quad\rightarrow\quad (\neg_*\psi \models_* \neg_*\varphi)
\end{aligned}$$

Unlike in classical logic, the converse of the last implication does not generally hold. (It nevertheless holds if $*$ is Ł, as well as for crisp φ, ψ.)

5. Fuzzy logic of yes-no questions

In this section we extend intensional semantics to interrogative formulae $?\varphi$. The yes-no question 'Is it the case that φ?' is answered by a proposition A iff A either entails φ (then it is an affirmative answer) or entails $\neg\varphi$ (a negative answer). Propositions that entail neither φ nor ψ do not solve the question, and thus correspond to 'I do not know' answers.

Definition 5.1 *A proposition A is an* $*$-affirmative answer *to $?\varphi$ iff $A \models_* \varphi$. It is a* $*$-negative answer *to $?\varphi$ iff $A \models_* \neg_*\varphi$. It is a* $*$-yes-no answer *(in symbols, $A \models_* ?\varphi$) iff it is a $*$-affirmative answer or a $*$-negative answer:*

$$A \models_* ?\varphi \quad \equiv_{\mathrm{df}} \quad (A \models_* \varphi) \vee (A \models_* \neg_*\varphi)$$

Theorem 5.1 FCT *proves that* $\Delta(W \subseteq W \cap_* W)$ *implies*

$$\begin{aligned}
(A \models_* B) &\quad\rightarrow\quad [(B \models_* ?\varphi) \rightarrow_* (A \models_* ?\varphi)]\\
(A \equiv_* B) &\quad\rightarrow\quad [(B \models_* ?\varphi) \leftrightarrow_* (A \models_* ?\varphi)]\\
(\varphi \equiv_* \psi) &\quad\rightarrow\quad [(A \models_* ?\varphi) \rightarrow_* (A \models_* ?\psi)]
\end{aligned}$$

Theorem 5.2 FCT *proves that if* $\Delta(W \subseteq W \cap_* W)$, *then $*$-affirmative and $*$-negative answers $*$-exclude one another, i.e.,*

$$((\psi_1 \models_* \varphi) \&_* (\psi_2 \models_* \neg_*\varphi)) \rightarrow (\models_* \neg_*(\psi_1 \&_* \psi_2))$$

Definition 5.2 (Yes-no $*$-entailment and $*$-equivalence of questions)

$$\begin{aligned}
?\varphi \models_* ?\psi &\quad\equiv_{\mathrm{df}}\quad (\forall A)\,((A \models_* ?\varphi) \rightarrow_* (A \models_* ?\psi))\\
?\varphi \equiv_* ?\psi &\quad\equiv_{\mathrm{df}}\quad (?\varphi \models_* ?\psi) \&_* (?\psi \models_* ?\varphi)
\end{aligned}$$

Theorem 5.3 FCT *proves that* $\Delta(W \subseteq W \cap_* W)$ *implies*

$$\begin{aligned}
(?\varphi \models_* ?\psi) &\quad\rightarrow\quad ((A \models_* ?\varphi) \rightarrow_* (A \models_* ?\psi))\\
(?\varphi \equiv_* ?\psi) &\quad\rightarrow\quad ((A \models_* ?\varphi) \leftrightarrow_* (A \models_* ?\psi))\\
(?\varphi \models_* ?\psi) &\quad\rightarrow\quad ((?\psi \models_* ?\chi) \rightarrow_* (?\varphi \models_* ?\chi))\\
(?\varphi \equiv_* ?\psi) &\quad\rightarrow\quad ((?\psi \equiv_* ?\chi) \rightarrow_* (?\varphi \equiv_* ?\chi))\\
(\varphi \equiv_* \varphi') &\quad\rightarrow\quad ((?\varphi \models_* ?\psi) \rightarrow_* (?\varphi' \models_* ?\psi))\\
(\psi \equiv_* \psi') &\quad\rightarrow\quad ((?\varphi \models_* ?\psi) \rightarrow_* (?\varphi \models_* ?\psi'))\\
(\varphi \equiv_* \psi) &\quad\rightarrow\quad (?\varphi \models_* ?\psi)\\
(?\varphi \models_* ?\psi) &\quad\rightarrow\quad (?\varphi \models_* ?\neg_*\psi)
\end{aligned}$$

Corollary 5.4 FCT *proves that* $\Delta(W \subseteq W \cap_* W)$ *implies* $?\varphi \models_* ?\neg_*\varphi$.

Unlike in classical logic, the converse $?\neg_*\varphi \models_* ?\varphi$ does not hold generally (though it does if $*$ is Ł, or for crisp propositions). Examples from natural language show that negative fuzzy questions may indeed be weaker than positive ones.

References

[1] L. Běhounek, "Fuzzification of Groenendijk-Stokhof propositional erotetic logic." Submitted, 2004.

[2] L. Běhounek and P. Cintula, "Fuzzy class theory." Submitted, 2004.

[3] P. Hájek, *Metamathematics of Fuzzy Logic*, vol. 4 of *Trends in Logic*. Dordercht: Kluwer, 1998.

[4] J. Groenendijk, M. Stokhof, "Partitioning logical space." 2nd ESSLLI Annotated Handout, 1990.

[5] J. Groenendijk, M. Stokhof, "Questions," in *Handbook of Logic and Language* (J. van Benthem and A. ter Meulen, eds.), pp. 1055–1124, Elsevier/MIT Press, 1997.

[6] F. Esteva, L. Godo and F. Montagna, "The ŁΠ and ŁΠ$\frac{1}{2}$ logics: Two complete fuzzy systems joining Łukasiewicz and product logics," *Archive for Mathematical Logic*, vol. 40, pp. 39–67, 2001.

[7] P. Cintula, "The ŁΠ and ŁΠ$\frac{1}{2}$ propositional and predicate logics," *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 21–34, 2001.

Connectors in the Context of OMG D&C Specification

Post-Graduate Student:
LUBOMÍR BULEJ, TOMÁŠ BUREŠ
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
bulej@cs.cas.cz, bures@cs.cas.cz

Supervisor:
PROF. ING. FRANTIŠEK PLÁŠIL, DRSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
plasil@cs.cas.cz

Field of Study:
Software systems
Classification: I-2

Abstract

The OMG Deployment and Configuration specification is an attempt at standardizing the deployment process of component-based applications in distributed environment. Software connector is an abstraction capturing interaction among components. Apart from middleware independence, connectors provide additional services (e.g. adaptation, monitoring, etc.) and benefits, especially in the area of integration of heterogeneous component-based applications. This paper presents an approach for using connectors in the context of deployment process defined by the OMG Deployment and Configuration specification.

1. Introduction and motivation

Component-based software engineering is a paradigm advancing a view of constructing software from reusable building blocks, components. A component is typically a black box with a well defined interface, performing a known function. The concept builds on the techniques well known from modular programming, which encourage the developers to split a large and complex system into smaller and better manageable functional blocks and attempt to minimize dependencies between those blocks.

Pursuing the vision of building software from reusable components, the component-based software engineering paradigm puts a strong emphasis on design and modeling of software architectures, which allows for reuse of both implementation and application design. The high level abstractions employed in architecture modeling often lack support in the existing technology, so an emphasis is put also on developing support for runtime binding of components, flexible communication mechanisms, or deployment of component applications in distributed environment.

Some of the ideas have been embraced by the software development industry and as a result, there are now several component models, which are extensively used for production of complex software systems. The well-known models include Enterprise Java Beans [24] by Sun Microsystems, CORBA Component Model [18] by OMG, and .Net [14] by Microsoft.

There is a large number of other component models, designed and used mainly by the academic community. While most of the academic component models lack the maturity of their industrial counterparts, they aim higher with respect to fulfilling the vision of the component-based software engineering paradigm. This is mainly reflected in support for advanced modeling features, such as component nesting, or connector support. Of those we are familiar with, we should probably mention SOFA [22][17], Fractal [16], and Darwin [13].

One of possibly several common problems of most component models is deployment of component-based applications. Most of the component models available have attempted to address the issue in some way, but the differences between various component models have made it difficult to arrive at a common solution. The differences comprise mainly component packaging and deployment, communication middleware, hierarchical composition, component instantiation, or lifecycle management. As a result, integration and maintenance of component applications implemented in different component models is very difficult.

The deployment process generally consists of several steps, which have to be performed in order to successfully launch an application, and is typically component technology and vendor specific. That means that even applications written with specific technology in mind have to be deployed with vendor specific tools an in vendor specific way.

The specification by Object Management Group [20] aims to lay foundation for an industrial standard for deployment and configuration of component-based distributed applications. Since it does not explicitly address deployment and configuration of heterogeneous component applications, as a part of our research, we are attempting to design and develop tools compatible with the OMG D&C specification that would allow for deployment of heterogeneous component applications. To demonstrate the problems associated with deployment of heterogeneous component applications as well as the feasibility of our approach, we aim to support deployment of component applications with components written in SOFA, Fractal, and EJB.

One of the main problems inherent to deployment of heterogeneous component applications is related to interconnection of components from different component models. The problem arises mainly due to 1) different middleware used by the component models to achieve distribution, and 2) different ways of instantiating components and accessing their interfaces.

Of the three mentioned component models, SOFA offers the most freedom in the choice of middleware, as it has native support for software connectors, which allow using almost arbitrary middleware for communication. Fractal, on the other hand, supports distribution with its own middleware based on serialization defined by RMI [26]. The middleware is, however, not compatible with classic SUN RMI. Finally, EJB uses SUN RMI to achieve distribution.

Regarding the component instantiation mechanisms, the SOFA and the Fractal component models are quite similar. Both employ the concept of factory (component builder in SOFA, generic factory in Fractal) for creating component instances, yet they differ substantially in the way a component structure is described. The SOFA model describes the structure statically, using SOFA-specific ADL called Component Definition Language. In Fractal, the description of the structure is dynamic, passed as a parameter to the generic factory.

The EJB component model, on the other hand, bears very little similarity to either of the discussed models. The EJB component model supports four different kinds of components, beans: a) entity beans, stateful, the state is persistent and usually stored in a database, b) stateful session beans, the state of which is preserved for the duration of a session, c) stateless session beans, which are quite similar to libraries, and d) message-driven beans, which are similar to stateless session beans, except they lack the classic business interface, and instead process incoming requests in a message loop. Every component has a business interface and a home interface. The home interface of a bean is used to instantiate components of a specific kind, and in case of entity beans, restore component state from the database. Bean home interfaces can be obtained through naming service. Prior to any request to the naming service, a bean has to be first deployed into an EJB container, using implementation-specific deployment tools. Unlike the SOFA and the Fractal models, EJB does not support component nesting.

To overcome the differences between these models, we have decided to use software connectors to facilitate the bridging between these technologies. Software connectors encapsulate all communication among components and are typically responsible for 1) distribution (employing a communication middleware), 2) adaptation (hiding changes in method names, and order of arguments, or performing more complex transformation), and 3) additional services (e.g. encryption, communication monitoring, etc.). Being such a flexible concept, the connectors fit very well in our approach.

2. Goals and structure of the text

Although a connector based approach appears to be very promising for our project, we cannot use connectors directly, because the OMG D&C specification does not support connectors natively. Instead, the components are meant to be directly interconnected. To preserve compatibility with the OMG D&C specification, we want to avoid substantial changes to the OMG D&C specification, rather we want to show how to map the connectors into concepts already present in the specification. With regard to the discussion above, the goal of this paper is to show how to use connectors in a deployment framework compatible with the OMG D&C specification.

The paper is organized as follows: To introduce the context of our work, Section 3 gives an overview of the relevant parts of the OMG D&C specification. Section 4 then presents an overview and key features of our connector model and explains what a deployment of a component-based application with connectors looks like. Having explained the related topics, Section 5 shows how to utilize connectors in the scope of OMG D&C specification. We discuss related work in Section 6 and conclude the paper with summary and future work in Section 7 and Section 8, respectively.

3. Overview of OMG D&C Specification

The deployment and configuration of component-based distributed application describes the relation between three major abstractions. First, there is a component-based application, which consists of other components, the application itself being a component considered independently useful. Then there is a target environment, termed domain, which provides computational resources for execution of component-based applications. And finally, there is a deployment process, which takes a component-based application and a target environment as an input and produces an instance of the application running in the target environment as a result.

Given enough information about the application and the target environment, the deployment process is expected to be reasonably generic, especially at higher levels of abstraction. The required information is made available to the process in form of detailed description with a standardized data model. To allow for specialization at lower levels of abstraction, the OMG specification is compliant with the Model Driven Architecture (MDA) [21], also defined by OMG. The core of the specification defines a set of concepts and classes relevant for the implementation of the specification, which forms a platform independent model (PIM) of the specification. The model can be the transformed to platform specific models (PSM), which can capture the specifics of particular component middleware technology, programming language, or information formatting technology.

The component model defined by the core specification is explicitly independent of distributed component middleware technology such as CORBA CCM [18] or EJB [24]. Components can be either implemented directly (a monolithic implementation), or by an assembly of other components. The hierachical composition allows for capturing logical structure of an application and a configuration of an assembly of components. Ultimately, though, every application can be decomposed into a set of components with monolithic implementation, which is required for deployment.

The target environment, a domain, consists of nodes, interconnects and bridges. Of these, only the nodes provide computational resources, while interconnects group nodes that are able to communicate directly within a domain. A situation where the nodes cannot, for some reason (e.g. a firewall, an application proxy), communicate directly is modeled by grouping the nodes in different interconnects. Bridges are then used to facilitate communication between nodes in different interconnects.

The deployment process consists of five stages, termed *installation*, *configuration*, *planning*, *preparation*, and *launch*. Prior to deployment, the application must be packaged and made available by the producer. The package has to contain all relevant meta data describing the application as well as binary code and data, required to run the application.

To minimize the amount of interdependencies and to lower the overall complexity of the platform independent model, the specification defines two dimensions for segmenting the model into modules. The first dimension provides a distinction between a *data model* of the descriptive information and a *management model* of runtime entities, that process the information. The second dimension takes into account the role of the models in the deployment process, and distinguishes among component, target, and execution models.

Since giving a complete overview of the whole specification is far beyond the scope of this paper, we have selected only the parts required to understand the context of the presented work. Of the modules mentioned earlier, we will only describe the component and execution data models, and provide a brief description of the deployment process with emphasis on the planning stage.

3.1. Component Data Model

The component data and management models are mainly concerned with description of and manipulation with component software packages. The description specifies requirements that have to be satisfied for successful deployment, most of which are independent of a particular target system. Both the application metadata and code artifacts are expected to be stored and configured in a repository during the installation and configuration stages of the deployment. The information will be then accessed and used during the planning, and preparation stages.

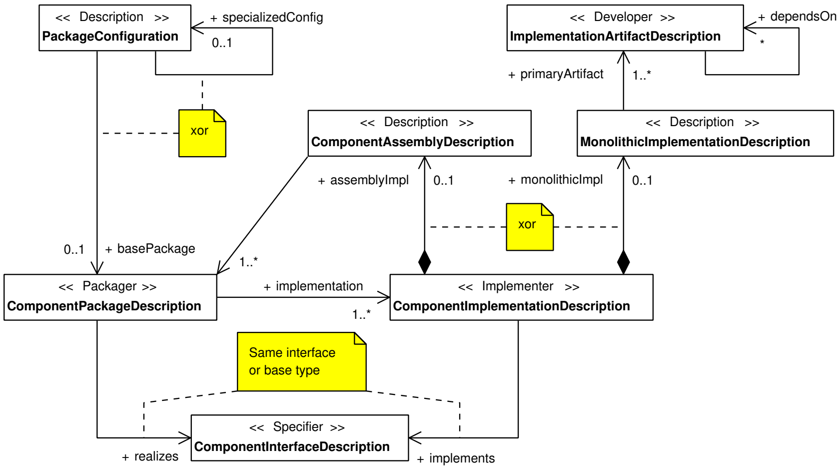


Figure 1: An overview of component data model

Figure 1 shows a high level overview of the component data model. The key concept here is a component package, which contains the configuration and implementation of a component. If a component has multiple implementations, the configuration should specify selection requirements, which influence deployment decisions by matching the requirements to capabilities of individual implementations.

Each component package realizes a component interface, which is implemented by possibly multiple component implementations. Figure 2 shows a detailed view of a component interface description. A component interface is a collection of ports, which can participate as endpoints in connections among components. A collection of properties carries component interface configuration.

As shown in Figure 1, an implementation of a component can be either monolithic, or an assembly of other components. In case of monolithic implementation, the description of the implementation consists of a list of implementation artifacts that make up the implementation. The artifacts can depend on each other and, which is not shown on the figure, can carry a set of deployment requirements and execution parameters. The requirements have to be satisfied before an artifact can be deployed on a node.

A component implementation that is not monolithic is defined as an assembly of other components. Fig-

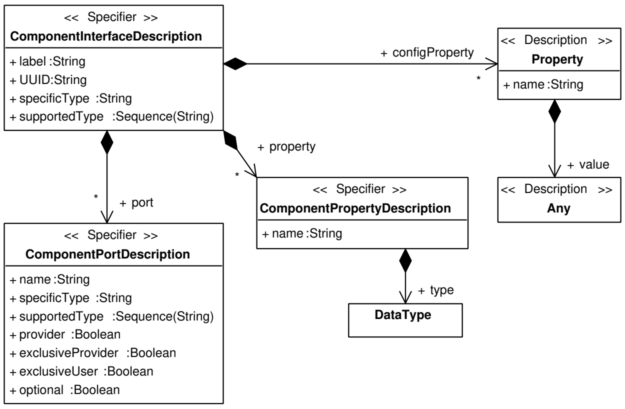


Figure 2: Detailed view of component interface description

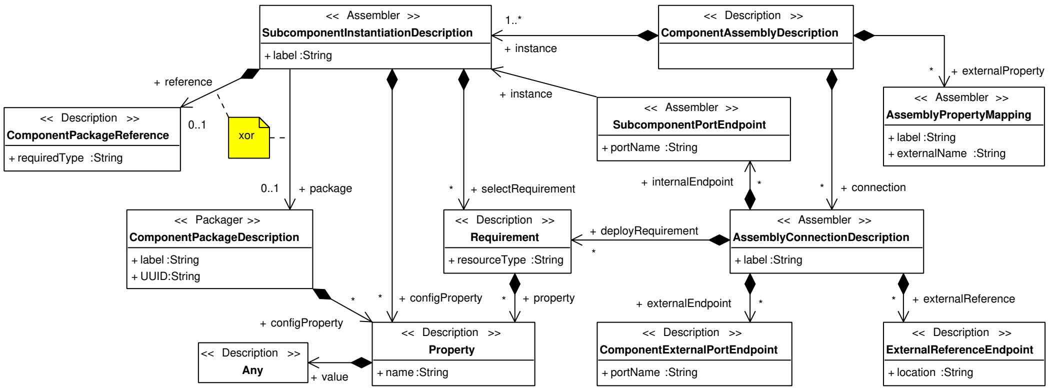


Figure 3: Detailed view of component assembly description

ure 3 shows a detailed view of a component assembly description. An assembly describes instances of subcomponents and connections among them. A subcomponent instance can reference a component package both directly and indirectly. Indirect package reference contains a specification of component interface the package has to realize and is expected to be resolved before deployment.

A set of selection requirements is part of an instance description and serves in choosing an implementation when a component package contains multiple implementations. Since the configuration of an assembly needs to be delegated to the configuration of its subcomponents, the description of an assembly contains a mapping of its configuration properties to configuration properties of its component instances.

The instances of components inside the assemblies can be connected using connections. A connection description contains a set of endpoints and deployment requirements for the connection. The endpoints can be of three kinds: a port of subcomponent's component interface, an external port of the assembly, or an external reference.

3.2. Execution Data Model

The execution data model is used for holding the result of combining the component software models with target models. The combining takes place during the planning stage of the deployment process, and the result captures how the application will execute in the target environment, i.e. what component implementation instance will run on which node. The information, termed *deployment plan*, held by the execution data model is used by execution management entities during preparation and launch stages of the deployment process.

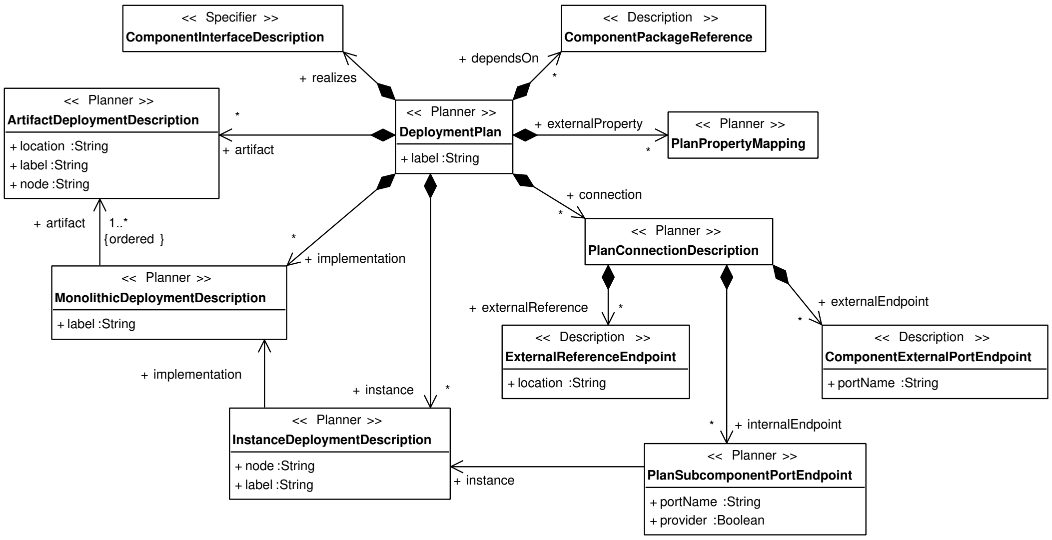


Figure 4: An overview of execution data model

Figure 4 shows a high level overview of the execution data model with additional details exposed in some of the classes. The deployment plan is analogous to the description of component assembly, and in fact contains a flattened view of the top level component assembly which represents the whole application. Of the original logical structure of the application, only the information required to create component instances and connections is retained. There is not, however, a direct mapping between all the classes in the component and execution data models.

The classes capturing the composition of individual artifacts into component implementations, instantiation of components and the connections among components are similar to those of the component data model, but not identical. This adds a significant amount of flexibility to the deployment process. If e.g. the component data model is extended to support other, possibly higher level, abstractions for which code can be automatically generated, the planner tool performing the transformation from the component data model to the execution data model can also generate the required code (or have other application do it on demand) and augment the resulting deployment plan so that it reflects the higher level abstractions in implementation.

3.3. Deployment Process

The deployment process as defined by the OMG specification consists of five stages. Prior to deployment, the software must be developed, packaged, and published by the provider and obtained by the user. The target environment in which the software is to run consists of nodes, interconnects and bridges, and contains a repository, in which the software package can be stored.

Installation During the installation stage, the software package is put into a repository, where it will be accessible from other stages of deployment. The location of the repository is not related to the domain the software will execute in, and the installation also does not involve any copying of files to individual nodes in a domain.

Configuration When the software is in the installation repository, its functionality can be configured by the deployer. The software can be configured multiple times for different configurations, but the configuration should not concern any deployment related decisions or requirements. The configuration stage is meant solely for functional configuration of the software.

Planning After a software package has been installed into a repository and configured, the deployer can start planning the deployment of the application. The process of planning involves selection of computational nodes the software will run on, the resources it will require for execution, deciding which implementations will be used for instantiation of components, etc. The planning does not have any immediate effect on the environment.

The planning stage of deployment is probably the most powerful concept of the specification. The result of planning is a deployment plan, which is specific to the target environment and the software being deployed. The plan is produced by transforming the information from the component data model into execution data model. Higher level abstractions in the component data model can be interpreted by the planner tool and transformed into deployment primitives of the deployment plan. In this stage, the planner or planner plugins can generate additional code artifacts, resolve indirect artifact or component package references, and transform the logical view of the component application into the physical view of the application, which is required for deployment.

An example of such a higher level abstraction are software connectors. While the original specification intends the connections among component interface ports to be direct, indirect communication can be achieved by modifying the planner to interpret requirements of individual endpoints in a connection and synthesize a connector implementation with desired features. The original component model can be then automatically adjusted to reflect the use of connectors for communication among components. The resulting component model is then transformed into deployment plan, which will describe the newly created artifacts and connections.

Preparation Unlike planning, the preparation stage involves performing work in the target environment in order to prepare the environment for execution of the software. If a software is to be executed more than once according to the same plan, the work performed during the preparation stage is reusable. The actual moving of files to computational nodes in the domain can be postponed until the launch of the application.

Launch The application is brought to the executing state during the launch stage. As planned, instances of components are created and configured on target nodes and the connections among the instances are established. The application runs until it is terminated.

4. Software Connectors

Software connectors are first class entities capturing communication among components (see Figure 5 showing an example of component-based application utilizing connectors). In our approach we use a connector model developed in our group [6][5]. This section briefly describes its key features.

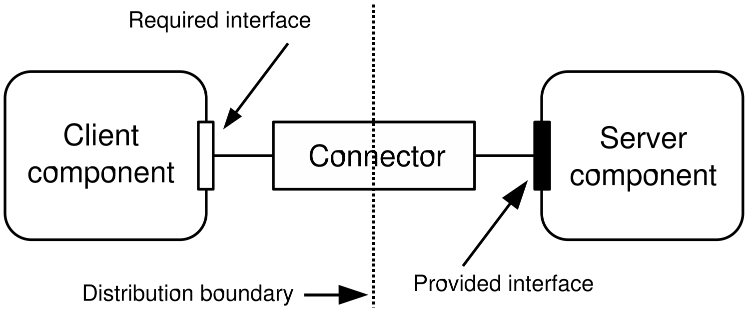


Figure 5: Components connected via a connector

In principle, our connector model captures two main levels of abstraction – a specification of connector requirements and a generated connector. On the level of requirement specification a deployer (a person driving a deployment process) defines features desired in a connector in terms of a communication style and non-functional properties (NFPs).

A communication style expresses the nature of the realized communication. So far, we have identified four basic communication styles: a) procedure call (local or remote; e.g. CORBA [19], RMI [26], DCE RPC

[28], SOAP [29], etc.), b) messaging (asynchronous message delivery; e.g. JMS [25], MQSeries [9], etc.), c) streaming (uni- or bi-directional stream of data; e.g. TCP/IP, RTP [1], unix pipe, etc.), and d) blackboard (distributed shared memory; e.g. JavaSpaces [27], Linda, [7], Bonita [23], etc.).

Non-functional properties define additional features or behavior that are related to a selected communication style. They allow specification of requirements such as that a realized connection must be secure (e.g. when transmitting sensitive data), monitored (e.g. for benchmarking purposes), that an adaptation should take place (e.g. in case of interconnecting incompatible interfaces or technologies), etc.

The information capturing the connector requirements is then passed to a connector generator (a computer program), which finds out how to assemble a connector with the desired functionality. At runtime, the generated connector is instantiated and bound to components that participate in a connection. Every inter-component link is realized by a unique instance of connector (more precisely by a unique instance of a connector unit as explained later in this section).

The connector generator relies on two basic concepts – connector and element architectures [2] and primitive element templates [5].

A connector architecture describes a top-level connector structure. The model of connectors as a set of interconnected elements is very similar to a model of components (see Figure 6 for an example of connector architecture for the procedure call communication style). Connector elements are responsible for particular features found in a connector. In Figure 6, the stub and skeleton elements are responsible for distribution. The interceptor element monitors calls performed on the server, and the adaptor element translates the calls between incompatible interfaces.

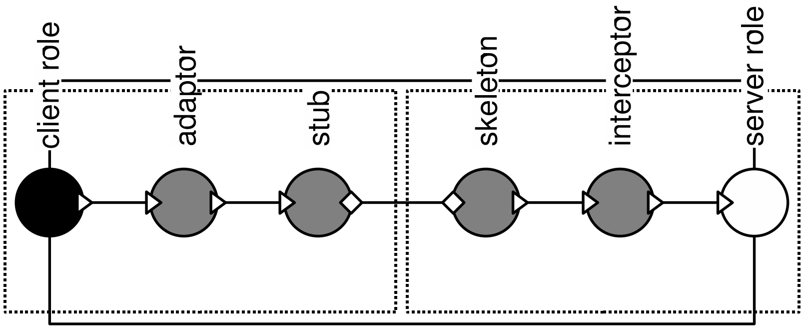


Figure 6: A connector architecture for procedure call communication style

An element in connector architecture is, however, just a black box. The element has to be assigned an implementation, which can either be another architecture (composite element) a or code implementing the required functionality (primitive element). The process is recursively applied until there are no elements without an implementation assigned.

The dotted line in Figure 6 marks the boundary of a connector unit (i.e. a distribution boundary). A connector unit describes elements that will be linked to a particular component. The division of a connector into connector units is only performed on the top-level connector architecture, which prevents composite connector elements from spanning multiple connector units. At runtime, inter-element links inside a connector unit are realized by a local procedure call. Links crossing the unit boundary are realized by stubs and skeletons in a proprietary way, depending on the middleware technology used.

The connector generator assembles a connector implementation based on the information found in a repository of connector and element architectures and a repository of primitive element implementations. Such a connector implementation has yet to be adapted to the component interface it is going to mediate. To make the necessary adaptation possible, each of the primitive elements in the assembled connector is implemented as a template. The templates are then expanded to provide an implementation of a primitive element with the required component interface. The connector is instantiated at runtime by instantiating the connector elements and binding them together according to the architecture.

The matter becomes a bit more complicated at runtime where we would like to access all connectors units in a uniform way. Consider for example the following function: `bindToRemoteReference(Reference ref)`. The function is responsible for establishing a remote connection between client and server connector units. Although the processing of this function call is in most cases delegated to a stub element, we cannot rely on it. We have to access a connector unit as a black-box. Therefore, a connector unit has to implement these control functions and (with knowledge of its own structure) delegate them to appropriate connector elements (i.e. the stub element in our example).

We implement the required control functions by adding a special element (called element manager) to each connector unit and composite element. The control interface it exposes is subsumed to the ”frame” of encapsulating connector unit or composite element. The element manager knows the structure of the connector unit/composite element it resides in and delegates the control function calls to corresponding elements (in fact its neighbors). Since the services realized by the element manager are mostly an implementation detail, we do not reflect this element in connector unit/composite element architectures.

5. Solution

The OMG D&C specification is very comprehensive, but also fairly complex. To allow use of connectors for mediating communication among components, we only have to deal with parts of it. Most of the work concerned with generation of connectors will be done at planning stage. The specification itself requires some modifications to the component data model to allow for specification of desired connector features for every connection among components (using communication style and non-functional properties), a way of transforming the modified specification to the base component and execution data models, and a way of ensuring correct instantiation of connectors and establishing of connections at application launch.

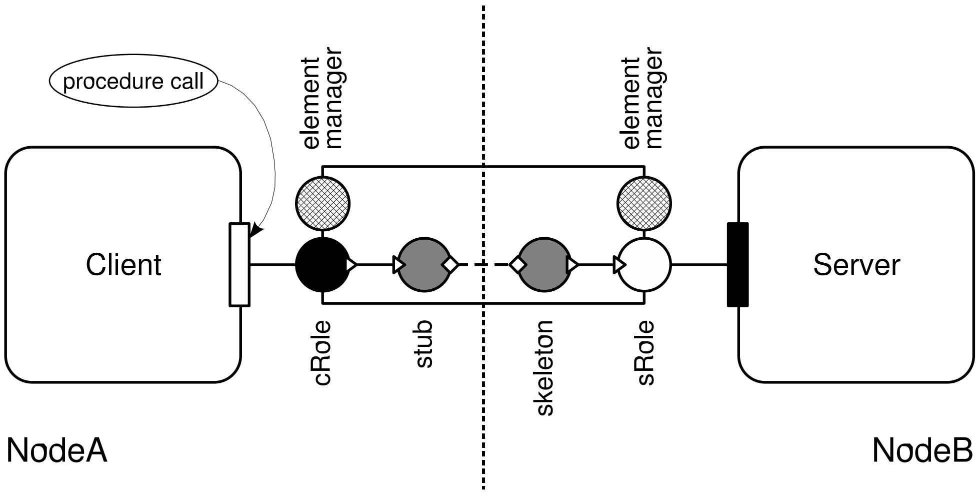


Figure 7: Example of a component application using a connector

Figure 7 shows an example of a simple component application using a connector to mediate the communication between the Client and the Server components. The example will be used throughout this section to demonstrate the approach we have chosen.

5.1. Specification of connector features

To generate a connector, a connector generator needs to have enough information concerning the requirements for the communication the connector is expected to mediate. The specification of connector features has a form of communication style and non-functional properties. Each connection among instances of components in an assembly can have different requirements.

The original OMG D&C platform independent component data model requires a minor extension to allow for specification of connector features. We have added another association, identical to that of `deployRequirement`, but named `connectionRequirement` to the AssemblyConnectionDescription class. The reason for not using the existing `deployRequirement` is to avoid overloading the semantics of the `deployRequirement` association, the contents of which are matched against requirement satisfiers describing resources available on the nodes in a domain.

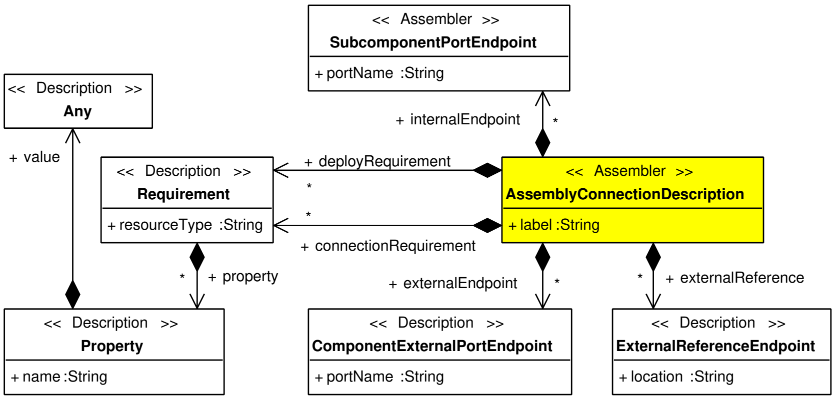


Figure 8: A modification of AssemblyConnectionDescription class

Figure 8 shows the modified AssemblyConnectionDescription class with the new `connectionRequirement` association. The XML fragments in Figure 9 and 10 are parts of the component data model description of the simple application depicted in Figure 7. The `connectionRequirement` element contains a description a connection requirements.

```
<componentPackageDescription>
  <label>Example Application</label>
  <realizes fileinarchive="ExampleApplicationInterface.cfd"/>
  <implementation fileinarchive="ExampleApplicationImpl.cid"/>
</componentPackageDescription>
```

Figure 9: ExampleApplication.cpd

5.2. Transformation of the component application description

During the planning stage of the deployment process, a planning tool aware of the connection requirements communicates with a connector generator [6] and provides it with information necessary to build a connector for each connection in the application. In addition to the connection requirements specified in the description of the component application, the tool can also provide information on assignment of connection endpoints to individual nodes in a domain as well as information on resources available on each of the nodes. The connector generator creates the necessary connector code and the connector-aware part of the planning tool transforms the original application description specifying connection requirements into a new description, which reflects the changes required to deploy connectors along with the original components.

The transformation adds instances of connector units into the application description and decomposes the original connections so that for each endpoint of the original connection, a new connection is created, connecting the component endpoint to an endpoint of connector unit instance. The original connection is then replaced with a new connection connecting the connector units together. The resulting description of component application adheres to the original OMG D&C specification of component data model, with connectors represented by regular components. This description can be then transformed to deployment

```
<componentImplementationDescription>
  <label>Example Application Implementation</label>
  <assemblyImpl>
    <instance id="1">
      <label>Client Component</label>
      <package fileinarchive="ClientComponent.cpd"/>
    </instance>

    <instance id="2">
      <label>Server Component</label>
      <package fileinarchive="ServerComponent.cpd"/>
    </instance>

    <connection>
      <label>Connection between Client and Server components</label>
      <internalEndpoint instanceRef="1">
        <portName>ClientComponentInterfacePort</portName>
      </internalEndpoint>
      <internalEndpoint instanceRef="2">
        <portName>ServerComponentInterfacePort</portName>
      </internalEndpoint>

      <connectionRequirement>
        <resourceType>connectionType</resourceType>
        <property>
          <name>allow-remote</name><value>true</value>
        </property>
        <property>
          <name>comm-style</name><value>procedure call</value>
        </property>
      </connectionRequirement>
    </connection>
  </assemblyImpl>

  <implements fileinarchive="ExampleApplicationInterface.cfd"/>
</componentImplementationDescription>
```

Figure 10: ExampleApplicationImpl.cid

plan by flattening the logical structure of the application description.

The XML fragment in Figure 11 describes the implementation of the simple application depicted in Figure 7 after the connectors have been integrated into the original description. Note the new component instances and connections.

5.3. Instantiation of connectors

A connector has to be instantiated from top to bottom, starting with a connector unit and the corresponding element manager. Then the elements on the next level are instantiated. In case of composite connector elements, the process has to be applied recursively until all the primitive elements are reached. Since the OMG D&C specification does not support ordering of instantiation of individual components, decomposing the internal structure of connectors into components and connections so as to let the execution management entities instantiate the connector elements would not produce the expected result.

Instead of modification of the OMG D&C specification of the deployment process, the instantiation of a connector is a responsibility of an element/unit factory. For that it needs to know the internal structure of the connector. Since the connector code is generated, a code for instantiating a specific connector architecture

```xml
<componentImplementationDescription>
  <label>Example Application Implementation</label>
  <assemblyImpl>
    <instance id="1">
      <label>Client Component</label>
      <package fileinarchive="ClientComponent.cpd"/>
    </instance>

    <instance id="2">
      <label>Server Component</label>
      <package fileinarchive="ServerComponent.cpd"/>
    </instance>

    <instance id="3">
      <label>ClientComponentInterfacePort Unit</label>
      <package href="access-method://ClientComponentInterfacePortUnit.cpd"/>
    </instance>

    <instance id="4">
      <label>ServerComponentInterfacePort Unit</label>
      <package href="access-method://ServerComponentInterfacePortUnit.cpd"/>
    </instance>

    <connection>
      <label>Connection between Client component and a client unit</label>
      <internalEndpoint instanceRef="1">
        <portName>ClientComponentInterfacePort</portName>
      </internalEndpoint>
      <internalEndpoint instanceRef="3">
        <portName>ServerComponentInterfacePort</portName>
      </internalEndpoint>
    </connection>

    <connection>
      <label>Connection between Server component and a server unit</label>
      <internalEndpoint instanceRef="2">
        <portName>ServerComponentInterfacePort</portName>
      </internalEndpoint>
      <internalEndpoint instanceRef="4">
        <portName>ClientComponentInterfacePort</portName>
      </internalEndpoint>
    </connection>

    <connection>
      <label>Connection between client and server units</label>
      <internalEndpoint instanceRef="3">
        <portName>ClientComponentInterfacePort</portName>
      </internalEndpoint>
      <internalEndpoint instanceRef="4">
        <portName>ServerComponentInterfacePort</portName>
      </internalEndpoint>
    </connection>
  </assemblyImpl>

  <implements fileinarchive="ExampleApplicationInterface.cfd"/>
</componentImplementationDescription>
```

Figure 11: ExampleApplicationImpl.cid

can be generated as well. A more flexible solution though, is to pass a description of connector structure to a generic element/unit factory through execution parameters in the description of connector implementation.

6. Evaluation and related work

In this paper, we have presented an approach which allows using software connectors in the context of OMG D&C specification. The original platform independent component data model assumes direct communication among component endpoints in a connection. This assumption requires that a connection to be described at a lower level of abstraction than e.g. the structure of the component application, because it has to connect ports provided by specific artifacts. As a consequence, the description of the component application cannot abstract from e.g. a middleware technology used for communication in distributed environment.

Enhancing the description of a connection among components with the specification of communication style and non-functional properties allows e.g. the selection of communication middleware to be postponed until the planning stage of the deployment, or introduction of logging, monitoring, or encryption facilities into communication without changing the description of the component application.

To our knowledge, there is no other work concerning use of connectors in the OMG D&C specification or other deployment orchestration framework. There is, however, a number of mature business solutions for interconnecting the leading business component models such as EJB [24], CCM [18], and .NET [14]. A common denominator of these models is the lack of certain features (e.g. component nesting), which makes the problem of their interconnection a matter of middleware bridging. Each of those component models has a native middleware for communication in distributed environment (RMI [26] in case of EJB, CORBA [19] in case of CCM, and .NET remoting in case of .NET). A middleware bridge is usually realized as a ”bridge” component translating one middleware protocol to another. A list of leading middleware bridges comprises:

- *Borland Janeva* [4] — Allows for interconnection of .NET applications with CORBA objects. It uses CORBA IIOP natively and provides a tool for generating .NET proxies. The proxies are then added into the resulting .NET assembly, thus allowing for easier deployment of the .NET part.
- *ObjectWeb DotNetJ* [15] — Allows to call Java classes or even EJB components from .NET applications. Starts a dedicated JVM with class implementing the .NET remoting protocol. Remotely called Java classes are loaded directly to the JVM, calls to EJB components (residing in another JVM) are transformed to RMI calls.
- *Intrinsyc J-Integra for .NET* [10] — Works in a way similar to DotNetJ. Uses .NET remoting as a native protocol and allows to bridge .NET and EJB technologies. Unlike DotNetJ, allows for calls in both directions.
- *BEA WebLogic* [3] — More a middleware suite than just a middleware bridge. Allows accessing CORBA servers from EJB via a designated bridge.
- *IONA Orbix* [11] — Also a rather comprehensive middleware suite, similar to BEA WebLogic. Builds on CORBA infrastructure. Provides bridges for EJB, COM and .NET clients allowing them to access CORBA objects. All of them are based on deploying a ”bridge” component into respective component technology.
- *IONA Artix* [12] — A full-fledged SOAP middleware. Provides ”bridge” classes for EJB and CORBA technologies that accept SOAP calls and delegate them to appropriate components/objects. The bridging is only done in one direction.
- *K2* [8] — An implementation of CCM container. Allows for integration with EJB via an EJB bridge. Natively supports also SOAP [29], thus allowing for seamless integration with Web Services.

Even though these bridges represent mature software products, they do not provide a standardized approach. All the listed bridges are proprietary solutions of individual vendors. Usually, they were either created to achieve a specific goal of connecting two particular platforms (e.g. DotNetJ) or they were originally created as an ORB middleware and evolved into a more robust solution later, allowing to accommodate other platforms. Nevertheless, the bridging only works in one direction (e.g. .NET, EJB clients accessing CORBA object in case of Orbix, or EJB component accessing CORBA services in case of WebLogic).

In our approach, we address heterogeneity from the very beginning. We use the platform independent component data model defined in the OMG D&C specification to describe a component application. Our extension of the component data model also does not introduce any platform or language dependencies. Component interconnections are modeled by connectors, which are created for a specific platform and language during the planning stage of the deployment process. A connector is generated for each of the connections, which allows for adaptation of the connector to the platforms on which the connection endpoints reside.

The adaptation of the connector to the connection endpoint's platform results in no or minimal overhead for local connections, small overhead for connections between identical platforms (e.g. using RMI locally for connecting Java to Java), and moderate overhead when connecting originally incompatible platforms (e.g. Java to .NET using SOAP). All connections are two-way, without any specialized code in the component implementation, which allows for smooth building of heterogeneous component applications.

7. Summary

We have presented an approach for using software connectors in deployment frameworks compatible with OMG D&C specification. The use of connectors eases the deployment and interconnection of heterogeneous component-based distributed applications, the components of which can be implemented in different component models. We have only needed to introduce a very minor change into the specification for it to support specification of connection requirements.

The description and implementation of connectors is mapped into already present concepts and classes (i.e. component packages, monolithic component implementation, implementation artifact). The presented solution is generic, described in a platform independent way, and allows mapping to different component models.

8. Future work

The presented solution relies on a connector generator capable of creating connectors with respect to a high-level specification and generate their implementation for different component models and middleware technologies. We currently have a prototype implementation of a connector generator for the SOFA component system. The generator needs to be redesigned to allow for more flexibility and support for the Fractal and EJB component models needs to be written. Moreover, since the Fractal and EJB models have no connector support, it is also necessary to develop a runtime infrastructure for connectors in these two component models.

References

[1] Audio-Video Transport Working Group, “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, Jan 1996

[2] D. Bálek, F. Plášil, “Software Connectors and Their Role in Component Deployment”, Proceedings of DAIS'01, Krakow, Kluwer, Sep 2001

[3] BEA WebLogic 8.1, http://www.bea.com, 2003

[4] Borland Janeva 6.0, http://www.borland.com/janeva, 2004

[5] L. Bulej, T. Bureš, “A Connector Model Suitable for Automatic Generation of Connectors”, Tech. Report No. 2003/1, Dep. of SW Engineering, Charles University, Prague, Jan 2003

[6] T. Bureš, F. Plášil, “Communication Style Driven Connector Configurations”, Copyright Springer-Verlag, Berlin, LNCS3026, ISBN: 3-540-21975-7, ISSN: 0302-9743, pp. 102-116, 2004

[7] D. Gelernter, “Generative communication in Linda”, ACM Transactions on Programming Languages and Systems, pp. 80-112, Jul 1985

[8] iCMG, K2 Component Server 1.5, http://www.icmgworld.com/corp/k2/k2.overview.asp, 2003

[9] IBM Corporation, WebSphere MQ family, http://www-306.ibm.com/software/integration/mqfamily/, Apr 2002

[10] Intrinsyc, J-Integra for .NET, http://j-integra.intrinsyc.com/, 2004

[11] IONA Technologies PLC, Orbix 6.1, http://www.iona.com/products/orbix.htm, 2004

[12] IONA Technologies PLC, Artix 2.1, http://www.iona.com/products/artix, 2003

[13] J. Magee, N. Dulay and J. Kramer, “Regis: A Constructive Development Environment for Distributed Programs”, In IEE/IOP/BCS Distributed Systems Engineering, 1(5), pp. 304-312, Sep 1994

[14] Microsoft Corporation, .NET, http://www.microsoft.com/net, 2004

[15] ObjectWeb Consortium, DotNetJ, http://dotnetj.objectweb.org, 2003

[16] ObjectWeb Consortium, Fractal Component Model, http://fractal.objectweb.org, 2004

[17] ObjectWeb Consortium, SOFA Component Model, http://sofa.objectweb.org, 2004

[18] Object Management Group, “Corba Components, version 3.0”, http://www.omg.org/docs/formal/02-06-65.pdf,Jun 2002

[19] Object Management Group, “Common Object Request Broker Architecture: Core Specification, version 3.0.3”, http://www.omg.org/docs/formal/04-03-12.pdf, Mar 2004

[20] Object Management Group, “Deployment and Configuration of Component-based Distributed Applications Specification”, http://www.omg.org/docs/ptc/03-07-02.pdf, Jun 2003

[21] Object Management Group, “Model Driven Architecture”, http://www.omg.org/docs/ormsc/01-07-01.pdf, Jul 2001

[22] F. Plášil, D. Bálek and R. Janeček, “SOFA/DCUP: Architecture for Component Trading and Dynamic Updating”, Proceedings of ICCDS’98, Annapolis, Maryland, USA, IEEE CS Press, May 1998

[23] A. Rowstron, A. Wood, “BONITA: A set of tuple space primitives for distributed coordination”, Proceedings of the 30th Annual Hawaii International Conference on System Sciences. Published by the IEEE Computer Society Press, 1997

[24] Sun Microsystems, Inc., “Java 2 Platform Enterprise Edition Specification, version 1.4”, http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf, Nov 2003

[25] Sun Microsystems, Inc., “Java Message Service Specification, version 1.2”, http://java.sun.com/products/jms/docs.html, Apr 2002

[26] Sun Microsystems, Inc., “Java Remote Method Invocation Specification”, ftp://ftp.java.sun.com/docs/j2se1.4/rmi-spec-1.4.pdf, 2001

[27] Sun Microsystems, Inc., “Java Spaces Service Specification, version 2.0”, http://wwws.sun.com/software/jini/specs/js2_0.pdf, Jun 2003

[28] The Open Group, Inc., “DCE 1.1: Remote Procedure Call”, Aug 1997

[29] World Wide Web Consortium, “SOAP, version 1.2”, http://www.w3.org/2000/xp/Group/, Jun 2003

Functional Representation for Product Logic

Post-Graduate Student:
ING. PETR CINTULA
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

cintula@cs.cas.cz

Supervisor:
PROF. RNDR. PETR HÁJEK, DRSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

hajek@cs.cas.cz

Field of Study:
Logics
Classification: P3913

Abstract

By McNaughton theorem (see [7]), the class of functions representable by formulas of Łukasiewicz logic is the class of piecewise linear functions with integer coefficients. The first goal of this work to find an analogy of the McNaughton result for *product logic* (see [2], [6] and [5]). The second goal is to define a Conjunctive and Disjunctive semi-normal form (CsNF, DsNF) of the formulas of product logic (these forms are a syntactical counterpart of the piecewise monomial functions). These results show us how the functions expressible by the formulas of product logic look like.

Acknowledgement

This paper is an overview of a joint work with Brunella Gerla from the University of Salerno, Italy. For the full text see [3]

1. Preliminaries

Formulas of product are built in the usual way from a denumerable set of variables VAR, from two basic connectives of strong conjunction ($\&$) and implication ($\rightarrow$), and from the constant $\overline{0}$. In [5] Hájek defined an axiomatic system for product logic showed that these axiomatic is sound and complete with respect to the algebraic structure induced by product t-norm.

In product logic it is possible to define the following derived connectives $\neg, \wedge, \vee, \equiv$. Let *Form* denote the set of all formulas of product logic. Given a formula φ let VAR_φ denote the set of all variables occurring in φ. Interpretation of connectives of product logic is given by the following definition.

Definition 1.1 *An* evaluation *is a function* $e : Form \rightarrow [0,1]$ *such that* $e(\overline{0}) = 0$ *and*

- $e(\varphi \& \psi) = e(\varphi) \cdot e(\psi)$ *usual product of real numbers;*
- $e(\varphi \rightarrow \psi) = \begin{cases} 1 & \text{if } e(\varphi) \leq e(\psi) \\ \frac{e(\psi)}{e(\varphi)} & \text{otherwise} \end{cases} = \begin{cases} 1 & \text{if } e(\varphi) = 0 \\ \frac{e(\psi)}{e(\varphi)} \wedge 1 & \text{otherwise.} \end{cases}$

Note that for each evaluation e,

$$e(\neg\varphi) = \begin{cases} 1 & \text{if } e(\varphi) = 0 \\ 0 & \text{otherwise,} \end{cases}$$

$e(\varphi \wedge \psi) = \min(e(\varphi), e(\psi))$ and $e(\varphi \vee \psi) = \max(e(\varphi), e(\psi))$.

The notion of tautology, proof, provability and theorem are defined as usual. The standard completeness theorem says that a formula φ is a theorem of product logic iff φ is a tautology.

For a conjunction with n equal arguments φ, we use the abbreviation φ^n. A conjunction of zero formulas (also written φ^0) is considered as equal to $\overline{1}$. The set $\{1, \dots, n\}$ will be denoted by $\hat{n}$.

Definition 1.2 *Let φ be a formula and let $V \subseteq VAR_\varphi$. Then:*

- *an evaluation e is called* (V, φ)-positive *if for each $v \in VAR_\varphi$ holds: $e(v) > 0$ iff $v \in V$;*
- *an evaluation is called* φ-positive *if it is (VAR_φ, φ)-positive;*

2. Semi-Normal Forms and Normal Forms

In this section we define a Conjunctive and Disjunctive semi-normal forms (CsNF, DsNF). We start with the definition of a literal, which is more complex than in the classical logic, and then we build a CsNF (DsNF) just like in classical logic. We will continue with the proof of the *partial* equivalence of the formulas of product logic with formulas in CsNF (DsNF). The reason we will prove only partial equivalences lies in the fact that semantics of the product implication is not continuous in the point (0,0). We will develop a machinery to help us overcome this problem.

In the second part of this section we define Conjunctive and Disjunctive Normal Forms (CNF, DNF). Furthermore, we prove that each formula of can be equivalently written in CNF (DNF).

Definition 2.1 *A normal literal (or literal for short) is a formula in form:*

$$v_1^{k_1} v_2^{k_2} \dots v_l^{k_l} \to v_{l+1}^{k_{l+1}} v_{l+2}^{k_{l+2}} \dots v_m^{k_m},$$

where k_i are natural numbers and v_i arbitrary pairwise distinct propositional variables. Let I and J_i for $i \in I$ be finite sets and for every $i \in I$ and $j \in J_i$ let $\alpha_{i,j}$ be literals. The formula φ is said to be in a Conjunctive semi-normal form (CsNF) if

$$\varphi = \bigwedge_{i \in I} \bigvee_{j \in J_i} \alpha_{i,j}$$

The formula φ is said to be in a Disjunctive semi-normal form (DsNF) if

$$\varphi = \bigvee_{i \in I} \bigwedge_{j \in J_i} \alpha_{i,j}$$

Furthermore, we define that truth constant $\overline{0}$ is in both CsNF and DsNF

Definition 2.2 *Let φ be a formula, V a subset of VAR_φ. Let χ be a characteristic function of V. Let us define $\neg^0\varphi = \neg\varphi$ and $\neg^1\varphi = \neg\neg\varphi$. Then the following formula is called* (V, φ)-evaluator*:*

$$\nu^{(V,\,\varphi)} = \bigwedge_{v \in VAR_\varphi} (\neg^{\chi(v)} v)$$

Lemma 2.3 *Let φ be a formula, V a subset of VAR_φ and e an evaluation. Then holds:*

$$e(\nu^{(V,\,\varphi)}) = \begin{cases} 1 & \textit{if e is } (V, \varphi)\textit{-positive} \\ 0 & \textit{otherwise} \end{cases}$$

The following theorem can be proven using syntactical manipulation with formulas, i.e., we give an algorithm which for given formula φ and given set V produces a corresponding formulas in the conjunctive(disjunctive) semi-normal form. By corresponding we mean that each (V, φ)-positive evaluation assigned both of them the same truth value.

Theorem 2.4 *Let φ be a formula, V a subset of VAR_φ. Then:*

- *there is formula φ^D_V in DsNF such that $\nu^{(V,\,\varphi)} \to (\varphi \equiv \varphi^D_V)$ is a theorem;*
- *there is formula φ^C_V in CsNF such that $\nu^{(V,\,\varphi)} \to (\varphi \equiv \varphi^C_V)$ is a theorem.*

The formula φ^C_V (φ^D_V) is called a V-Conjunctive (V-Disjunctive) semi-normal form of formula φ, the formula $\varphi^C_{Var_\varphi}$ ($\varphi^D_{Var_\varphi}$) is called a Conjunctive (Disjunctive) semi-normal form of formula φ.

Notice that both V-Conjunctive and V-Disjunctive semi normal forms are not unique. In the next section we will show how to find a "simpler" form to given formula in CsNF (DsNF). Now we use our results to define a conjunctive and disjunctive normal forms.

Theorem 2.5 *Let φ be a formula. Then for each $V \subseteq VAR_\varphi$, there are formulas φ^D_V in DsNF and φ^C_V in CsNF such that*

$$\varphi \equiv \bigvee_{V \subseteq VAR_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \varphi^D_V\right) \equiv \bigvee_{V \subseteq VAR_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \bigvee_{i \in I^V} \bigwedge_{j \in J^V_i} (\alpha^V_{i,j})\right) \tag{1}$$

$$\varphi \equiv \bigvee_{V \subseteq VAR_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \varphi^C_V\right) \equiv \bigvee_{V \subseteq VAR_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \bigwedge_{i \in I^V} \bigvee_{j \in J^V_i} (\alpha^V_{i,j})\right) \tag{2}$$

Expression (1) is called Disjunctive normal form (DNF) of φ and expression (2) is called Conjunctive normal form (CNF) of φ.

3. Simplification of formulas in CsNF and DsNF

In this section, we show how to simplify formulas in semi-normal form. We formalize this notion in the following definition. We will work with CsNF in this section (results for DsNF are analogous).

Definition 3.1 *Let φ be a formula in CsNF. A formula ψ, resulting from formula φ by omitting some literals or conjuncts or disjuncts is called a* simplification *of φ iff formula $\nu^{(V,\,\varphi)} \to (\varphi \equiv \psi)$ is a theorem.*

Notice that if φ a V-Conjunctive semi normal form of χ and ψ is a simplification of φ, then also ψ is a V-Conjunctive semi normal form of χ.

If we deal with a finite set L of literals we can fix an enumeration on the set $W = \{v_1, v_2, \dots, v_m\}$ of all propositional variables occurring in literals in L. Then each normal literal $\alpha \in L$ is uniquely determined by an m-tuple $\mathbf{q}^\alpha = (q^\alpha_1, \dots, q^\alpha_m)$ of integers. A positive component q^α_i is considered as power for variable v_i in consequent and negation $-q^\alpha_j$ of a negative component is considered as power for variable v_j in antecedent. This can be expressed with a permutation π on $\hat{m}$ and an index l such that $q^\alpha_{\pi(i)} \leq 0$ for $i \leq l$, $q^\alpha_{\pi(i)} > 0$ for $i > l$ and

$$\alpha = v_{\pi(1)}^{-q^\alpha_{\pi(1)}} \& v_{\pi(2)}^{-q^\alpha_{\pi(2)}} \& \dots \& v_{\pi(l)}^{-q^\alpha_{\pi(l)}} \to v_{\pi(l+1)}^{q^\alpha_{\pi(l+1)}} \& v_{\pi(l+2)}^{q^\alpha_{\pi(l+2)}} \& \dots \& v_{\pi(m)}^{q^\alpha_{\pi(m)}}. \tag{3}$$

If $l = 0$ (respectively $l = m$) we understand the antecedent (respectively consequent) as truth constant $\overline{1}$. Also v_i^0 is considered as $\overline{1}$.

Let α and β be literals. The question is if there is a simple way to find out that $\alpha \to \beta$ is a theorem. Indeed, if both α and β are conjuncts in one conjunction, knowing that $\alpha \to \beta$ is a theorem would allow to omit β and obtain a simplification of our conjunction. In order to do that we define an order on tuples.

Definition 3.2 *Let* $\mathbf{a} = (a_i)_{i \leq m}$ *and* $\mathbf{b} = (b_i)_{i \leq m}$ *be tuples. Then* $\mathbf{a} \preceq \mathbf{b}$ *iff* $a_i \geq b_i$ *for all* $i \leq m$*.*

Now we may finally formulate lemma on simplification of formulas in CsNF.

Lemma 3.3 *Let* φ *be a formula in* **CsNF***, i.e.* $\varphi = \bigwedge\limits_{i \in I} \bigvee\limits_{j \in J_i} \alpha_{i,j}$*. Then the formula resulting from* φ *after processing the following four steps is a simplification of* φ*:*

(1) We replace all literals normal literals.

(2a) If there are indexes i,k such that $\mathbf{0} \preceq \mathbf{q}^{\alpha_{i,k}}$ and $I \neq \{i\}$ then we omit the conjunct $\bigvee\limits_{j \in J_i} \alpha_{i,j}$

(2b) If there are indexes i,k such that $\mathbf{0} \preceq \mathbf{q}^{\alpha_{i,k}}$ and $I = \{i\}$ then we replace formula φ by $\overline{1}$.

(3) If there are indexes i, j and j', $j \neq j'$ such that $\mathbf{q}^{\alpha_{i,j}} \preceq \mathbf{q}^{\alpha_{i,j'}}$ then we omit the literal $\alpha_{i,j}$.

(4) If there are indexes i, i', $i \neq i'$ and for each index $k \in J_i$ there is index $k' \in J_{i'}$ such that $\mathbf{q}^{\alpha_{i,k}} \preceq \mathbf{q}^{\alpha_{i',k'}}$ then we omit the conjunct $\bigvee\limits_{j \in J_{i'}} \alpha_{i',j}$.

4. Theorem proving algorithm

In this section we will use result from the previous sections to define an algorithm, which can be used to check whether a formula is a theorem or not. We will use the standard completeness theorem and Theorem 2.5. We start with a definition:

Definition 4.1 *Let* S *be a set of indexes of literals of* m *variables and* n *be the cardinality of* S*. The* $n \times m$ *matrix* $\mathbf{A}^S$ *is the matrix with rows* $\mathbf{q}^{\alpha_i}$ *for each* $i \in S$*.*

We want to describe what has to hold in order to the formula

$$\varphi \equiv \bigvee_{V \subseteq V_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \varphi_V^C \right) \equiv \bigvee_{V \subseteq V_\varphi} \left(\nu^{(V,\,\varphi)} \wedge \bigwedge_{i \in I^V} \bigvee_{j \in J_i^V} (\alpha_{i,j}^V) \right)$$

not being a tautology. If φ is not a tautology, then there is an evaluation e such that $e(\varphi) < 1$. Recall that for each evaluation e there is a unique set V, such that e is (V, φ)-positive. Thus $e(\varphi) < 1$ means that $e(\varphi_V^C) < 1$. Observe that if $e(\varphi_V^C) < 1$ then there is an index $i \in I^V$ such that $e(\bigvee\limits_{j \in J_i^V} (\alpha_{i,j}^V)) < 1$. This hold iff $e(\alpha_{i,j}^V) < 1$ for each $j \in J_i^V$. Which can be equivalently written as:

Theorem 4.2 *Let* φ *be a formula in CNF. Then* φ *is not a theorem iff there are a set* V *and an index* $i \in I^V$ *such that the matrix inequality* $\mathbf{A}^{J_i^V} \mathbf{x}^T < \mathbf{0}$ *has a non-negative solution.*

The problem is hence reduced to a problem of effectively solving an integer matrix inequality, that can be solved by means of integer linear programming. Using all our previous results we can formalize an algorithm to prove formulas of product logic. This algorithm is very inefficient-due to its exponential nature-however the average complexity seems to be much better. Anyway, we do not pursuit the problem of complexity in this paper.

We have a formula φ with m propositional variables. Let us define M as the set of already processed subsets of VAR_φ and K as the set of already processed indexes. In the beginning K and M are empty.

(1) If $M = \mathcal{P}(VAR_\varphi)^1$ GOTO (+), ELSE generate a set $V \in \mathcal{P}(VAR) \setminus M$, with smallest cardinality. Add V into M. Empty the set K.

(2) Using the proof of Lemma 2.4 find a formula φ_V^C.

(3) Simplify formula φ_V^C using Lemma 3.3 to a formula $\psi = \bigwedge_{i \in I^V} \bigvee_{j \in J_i^V} (\alpha_{i,j}^V)$

(4) If $K = I^V$ GOTO (1) ELSE add index i ($i \in I^V$, $i \notin K$) into the set K.

(5) If inequality $\mathbf{A}^{J_i^V} \mathbf{x}^T < \mathbf{0}$ has a non-negative solution GOTO (-) ELSE GOTO (4).

(+) A formula φ is a theorem of the product logic.

(-) A formula φ is not a theorem of the product logic.

5. Functional representation

In this section we give a characterization of the class of functions represented by formulas of product logic, analogously of what McNaughton theorem expresses for Łukasiewicz logic ([7]).

Definition 5.1 *Let $\mathcal{C}$ be an arbitrary function from $(0,1]^n$ into $[0,1]$ and let φ be an arbitrary formula with $VAR_\varphi = \{v_1, \ldots, v_n\}$. We say the function $\mathcal{C}$ is:*

- represented by the formula φ *(φ is a representation of $\mathcal{C}$) if* $e(\varphi) = \mathcal{C}(\, e(v_1),\, e(v_2),\, \ldots,\, e(v_m))$, *where e is an arbitrary evaluation.*
- positively represented by the formula φ *(φ is a positive representation of $\mathcal{C}$) if* $e(\varphi) = \mathcal{C}(e(v_1), e(v_2), \ldots, e(v_m))$, *where e is an φ-positive evaluation.*

Definition 5.2 *An integral monomial of m variables is a function $f : (0,1]^m \rightarrow (0,1]$ such that $f(x_1, \ldots, x_m) = x_1^{k_1} x_2^{k_2} \ldots x_m^{k_m}$, with $k_m \in \mathbf{Z}$.*

Now we give a McNaughton-like functional representation (c.f. [7]). Just as Łukasiewicz formulas are in correspondence with continuous piecewise linear functions, we are going to describe the class of functions in correspondence with product formulas.

---

[1] By $\mathcal{P}(S)$ we denote the powerset of S

Definition 5.3 *A* piecewise monomial function *of n variables is a continuous function f from $(0,1]^n$ into $[0,1]$ which is either identically equal to 0 on $(0,1]^n$, or there exist finitely many integer monomials $p_1 \dots, p_u$ and regions $D_1, \dots, D_u$ of $(0,1]^n$ such that for every $\mathbf{x} \in D_i$, $f(\mathbf{x}) = p_i$.*

Theorem 5.4 *Each piecewise monomial function is positively representable by some formula. Each formula is a positive representation of some piecewise monomial function.*

One part of this theorem is an obvious consequence of the Theorem 2.4, the second is proven by analogous methods as in Łukasiewicz case. Next we use Theorem 2.5 to extend this result to the full functional characterization of product fuzzy logic. Before we do se we need some additional definitions.

Definition 5.5 *Let n be a natural number and let M be a subset of $\{i \mid 1 \leq i \leq n\}$. Then the (M,n)-region of positivity $Pos^{M,\,n}$ is defined as*

$$Pos^{M,\,n} = \{(x_1, \dots, x_n) \in [0,1]^n \mid x_i > 0 \textit{ iff } i \in M\}$$

Example 5.6 *For $n = 2$ we have four regions of positivity:*

$$\begin{aligned}
Pos^{\emptyset,\,2} &= \{(0,0)\}\\
Pos^{\{1\},\,2} &= \{(x_1,0) \mid x_1 \in (0,1]\}\\
Pos^{\{2\},\,2} &= \{(0,x_2) \mid x_2 \in (0,1]\}\\
Pos^{\{1,2\},\,2} &= \{(x_1,x_2) \mid x_1, x_2 \in (0,1]\}
\end{aligned}$$

Lemma 5.7 *Let φ be a formula, n the cardinality of VAR_φ, $VAR_\varphi = \{v_i \mid i \leq n\}$. Let V be a subset of VAR_φ and set $M = \{i \mid v_i \in V\}$. Then the (V,φ)-evaluator $\nu^{(V,\,\varphi)}$ is a representation of the characteristic function of $Pos^{M,\,n}$.*

Now we can finally give the full description of functions represented by formulas of product logic. In fact we also give a functional interpretation of the description of free product algebras given in [1].

Definition 5.8 *A function $\mathcal{C} : [0,1]^n \to [0,1]$ is a* product function *if for every $M \subseteq \hat{n}$ the restriction of $\mathcal{C}$ to $Pos^{M,n}$ is a piecewise monomial function.*

Theorem 5.9 *Each product function is representable by some formula and, vice-versa, each formula is a representation of some product function.*

The algebraic counterpart of Product logic are the Product algebras, see [5]. Due to the standard completeness theorem, giving a complete characterization of functions associated with Product formulas with n variables is equivalent to give a description of the free Product algebra over n generators.

6. The references

References

[1] R. Cignoli, A. Torrens, An algebraic analysis of product logic. *Multiple-valued Logic*, vol. 5, pp. 45–65, 2000.

[2] P. Cintula, About axiomatic systems of product fuzzy logic. *Soft Computing*, vol. 5, pp. 243–244, 2001.

[3] P. Cintula, B. Gerla, Semi-normal Forms and Functional Representation of Product Fuzzy Logic. *Fuzzy Sets and Systems*, vol. 143, pp. 89–110, 2004.

[4] B. Gerla, *Many-valued logics of continuous t-norms and their functional representation*. PhD thesis, University of Milano, Italy, 2002.

[5] P. Hájek, *Metamathematics of Fuzzy Logic*. Trends in Logic. Kluwer, Dordrecht, 1998.

[6] P. Hájek, L. Godo and F. Esteva, A complete many-valued logic with product conjunction. *Archive for Mathematical Logic*, vol. 35, pp. 191–208, 1996.

[7] R. McNaughton, A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic*, vol. 16, pp. 1–13, 1951.

Změkčování hran v rozhodovacích stromech

doktorand:
MGR. JAKUB DVOŘÁK

Ústav informatiky AV ČR
Pod Vodárenskou věží 2

182 07 Praha 8

dvorak@cs.cas.cz

školitel:
RNDR. PETR SAVICKÝ, CSC.

Ústav informatiky AV ČR
Pod Vodárenskou věží 2

182 07 Praha 8

savicky@cs.cas.cz

obor studia:
Teoretická informatika
číselné označení: I1

Abstrakt

V tomto článku je popsána technika změkčování hran v rozhodovacích stromech sloužící ke zlepšení predikce metod strojového učení založených na stromech. Jedná se o způsob postprocesingu stromů získaných z některých běžných metod. Je zde vysvětlen princip změkčování hran a ukázány jeho základní efekty.

1. Úvod

Konstrukce rozhodovacích stromů je jednou z úspěšných technik strojového učení. Mezi její hlavní výhody patří použitelnost na objekty s atributy různých typů (číselné, kategoriální), dále jednoduchost a srozumitelnost použité struktury a s tím související možnost interpretace získaného stromu jako posloupnosti pravidel.

Ačkoliv nalezení rozhodovacího stromu zvolené velikosti, který by nejlépe klasifikoval trénovací data, je pro obvyklé úlohy výpočetně příliš náročné, jsou známy úspěšné heuristické metody — ke klasickým patří CART [1], C4.5 [3] a její varianta C5.0. Jmenované metody konstruují takové rozhodovací stromy, že rozhodovací pravidlo v libovolném vnitřním uzlu závisí pouze na jednom atributu klasifikovaného objektu. Navíc je-li tento atribut číselný, má rozhodovací pravidlo podobu porovnání hodnoty atributu s hodnotou prahu příslušného k danému uzlu. Listy stromu přiřazují odhad pravděpodobností příslušnosti objektu k jednotlivým třídám klasifikace — tento odhad je stejný pro všechny objekty (vzory), které projdou rozhodovacím stromem do téhož listu. Z uvedeného vyplývá, že máme-li pouze číselné atributy, pak takovýto rozhodovací strom definuje rozdělení vstupního prostoru na hyperkvádry (které mohou být v některých směrech nekonečné) a všechny vzory v témž hyperkvádru jsou klasifikovány totožně.

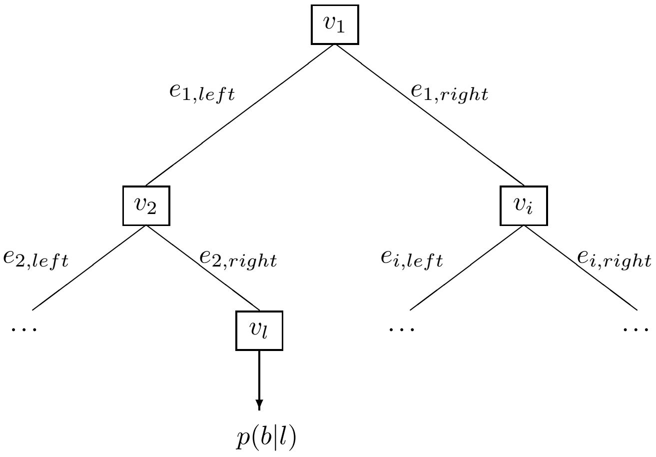
Protože není vždy žádoucí, aby malá změna hodnoty číselného atributu vedla ke zcela jiné klasifikaci, neboli aby hrany (hranice) hyperkvádrů byly ostře určené, umožňuje program C4.5 (a též C5.0) použít tzv. měkké (pravděpodobnostní) prahy, což vede k tzv. změkčení hran mezi hyperkvádry určenými rozhodovacím stromem. Princip spočívá v tom, že je-li hodnota testovaného atributu blízko hodnotě prahu, jsou prozkoumány obě větve stromu a výsledky zkombinovány podle vzdálenosti hodnoty od prahu.

2. Rozhodovací stromy v klasifikátorech

Dále se budeme zabývat případem, kdy vzory, na nichž chceme naučit klasifikátor, mají pouze číselné atributy. Máme tedy doménu $\Xi \subset \mathbb{R}^n$, každý vzor $\mathbf{x} = (x_1, \ldots, x_n)$ z této domény patří do jedné z tříd $\mathcal{C}_1, \ldots, \mathcal{C}_c$. Mějme trénovací množinu $T = \{(\mathbf{x}^i, b^i) | i = 1, \ldots, t\}$, v níž je pro každý vzor $\mathbf{x}^i$ uvedeno označení třídy, do níž vzor patří, tedy je-li $(\mathbf{x}^j, b^j) \in T$, znamená to, že vzor $\mathbf{x}^j$ patří do třídy $\mathcal{C}_{b^j}$.

Trénovací množina T je vstupem učícího algoritmu, který vyprodukuje klasifikátor. Klasifikátor po předložení vzoru $\mathbf{x} \in \Xi$ určí pro každou třídu klasifikace $\mathcal{C}_b$ odhad $P(b|\mathbf{x})$ pravděpodobnosti, že vzor $\mathbf{x}$ patří do třídy $\mathcal{C}_b$.

V případě bez změkčování hran je výsledkem učícího algoritmu — jako je CART, C5.0 apod.— binární rozhodovací strom, jaký ukazuje obrázek 1. Každému vnitřnímu uzlu v_i je přiřazen index testovaného atributu $a_i \in \{1, 2, \ldots, n\}$ a práh $split_i \in \mathbb{R}$, v každém listu v_l je uložen stochastický vektor $(p(b|l)); b = 1, \ldots, c$.



Obrázek 1: Rozhodovací strom

Při použití klasifikátoru na vzor $\mathbf{x} = (x_1, \ldots, x_n)$ se prochází rozhodovací strom na základě atributů předloženého vzoru: Na začátku je aktuálním uzlem kořen v_1 — proměnná j má hodnotu 1. Dokud aktuální uzel není listem stromu, provede se test

$$x_{a_j} \leq split_j \tag{1}$$

a v případě, že je nerovnost (1) splněna, přejde se do nového uzlu po hraně $e_{j,left}$, není-li splněna, po hraně $e_{j,right}$, aktuálním uzlem se stane uzel na konci této hrany — do proměnné j se uloží nový index aktuálního uzlu. Když je již aktuálním uzlem list v_j, pak výsledným odhadem $P(b|\mathbf{x})$ pravděpodobnosti, že vzor $\mathbf{x}$ patří do třídy $\mathcal{C}_b$, je hodnota $p(b|j)$.

3. Změkčování hran

Při změkčování hran se vychází z hotového rozhodovacího stromu, jako je na obrázku 1. Každé hraně $e_{j,d}; d \in \{left, right\}$ je přiřazena funkce $f_{j,d}(x) : \mathbb{R} \to \langle 0, 1 \rangle$, tak že pro každé j platí

$$\forall \mathbf{x} \in \Xi \quad f_{j,left}(\mathbf{x}) + f_{j,right}(\mathbf{x}) = 1 \tag{2}$$

S použitím hodnot $p(b|l)$ známých v listech v_l definujeme hodnoty $p(b|j, \mathbf{x})$ ve všech uzlech v_j induktivně: Je-li v_j list, potom

$$p(b|j, \mathbf{x}) = p(b|j) \quad \text{pro libovolné } \mathbf{x} \in \Xi$$

Jinak nechť z uzlu v_j vede hrana $e_{j,left}$ do uzlu v_p a hrana $e_{j,right}$ do uzlu v_q. Potom

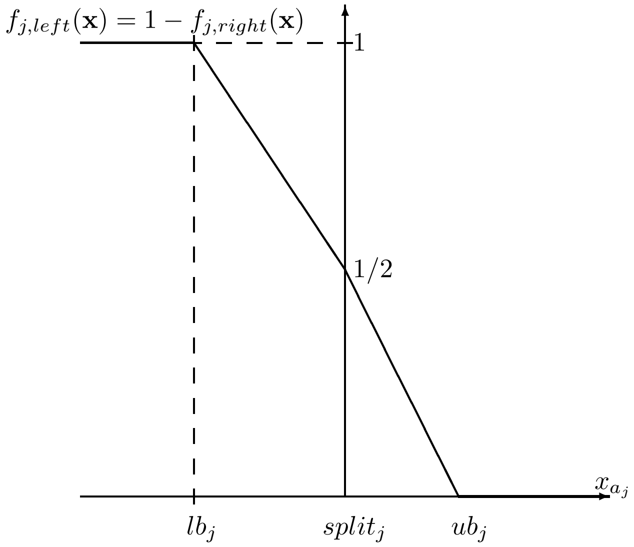
$$p(b|j, \mathbf{x}) = f_{j,left}(\mathbf{x}) p(b|p, \mathbf{x}) + f_{j,right}(\mathbf{x}) p(b|q, \mathbf{x})$$

Výsledným odhadem pravděpodobnosti příslušnosti vzoru $\mathbf{x}$ do třídy $\mathcal{C}_b$ je tato hodnota v kořeni stromu, tedy $p(b|1,\mathbf{x})$. Jinak to lze také vyjádřit vztahem

$$P(b|\mathbf{x}) = \sum_{v_l \in Leaves} p(b|l) \prod_{e_{i,d} \in Path(v_l)} f_{i,d}(\mathbf{x})$$

kde $Leaves$ označuje množinu všech listů a $Path(v_j)$ je množina všech hran na cestě z kořene v_1 do uzlu v_j.

V programu C4.5 jsou jakožto funkce $f_{j,left}(\mathbf{x})$ použity tzv. změkčující křivky. Funkce $f_{j,right}(\mathbf{x})$ jsou určeny z $f_{j,left}(\mathbf{x})$ podle vztahu (2). Změkčující křivka $f_{j,left}(\mathbf{x})$ je v každém vnitřním uzlu v_j spojitá po částech lineární funkce závisející pouze na atributu x_{a_j}, tedy na tom atributu, na kterém se ve stromu bez změkčení v tomto uzlu prováděl test (1). Změkčující křivka $f_{j,left}(\mathbf{x})$ (viz obrázek 2) je parametrizována indexem atributu a_j, hodnotou $split_j$, která je též známa z původního rozhodovacího stromu, a potom dvěma dalšími hodnotami $lb_j, ub_j \in \mathbb{R}$ (lower bound, upper bound). Povšimněme si, že zvolíme-li $lb_j = split_j = ub_j$ pro všechny vnitřní uzly v_j, potom klasifikátor dává výsledky totožné s původním stromem (bez změkčení hran).



Obrázek 2: Změkčující křivka

Program C4.5 nastavuje parametry lb_j, ub_j následovně: Nechť $T_j \subset T$ obsahuje ty trénovací vzory, při jejichž klasifikaci se provádí test v uzlu v_j. Kdyby se v uzlu v_j při testování (1) místo hodnoty $split_j$ použila jiná hodnota $split'_j$, některé vzory z T_j by byly klasifikovány odlišně od původní klasifikace, tedy byl by jiný počet chybně klasifikovaných trénovacích vzorů. Nechť E_j je počet chyb na T_j. Potom směrodatná odchylka počtu chyb v uzlu v_j je určena:

$$S_j = \sqrt{\frac{(E_j + \frac{1}{2})(|T_j| - E_j - \frac{1}{2})}{|T_j|}}$$

Hodnoty lb_j, ub_j jsou nastaveny na takové hodnoty $split'_j$, při kterých je počet chyb na T_j nejblíže $E_j + S_j$.

4. Experimentální algoritmus

Naše pokusy ukázaly, že změkčování hran v metodách C4.5 a C5.0 má ještě rezervy — pomocí stejného tvaru změkčovací křivky, jen jiným nastavením parametrů, by bylo možné dosahovat lepších výsledků.

Pro nastavení parametrů změkčení, tedy hodnot lb_j, ub_j pro všechny vnitřní uzly v_j, jsme implementovali jednoduchý algoritmus, který ani není efektivní, ani nenalézá optimální řešení — má sloužit pouze k experimentálnímu prozkoumání možností a vlastností změkčování hran. Je založen na náhodném prohledávání okolí dosud nejlepšího nalezeného řešení.

Úlohu hledání parametrů změkčení ve stromu můžeme formulovat tak, že hledáme vektor $\mathbf{v}$ sestávající z $2m$ reálných čísel, kde m je počet vnitřních uzlů stromu — $\mathbf{v}$ kóduje parametry lb_j, ub_j, pro všechny vnitřní uzly v_j. Parametry kódované vektorem $\mathbf{v}$ musí splňovat

$$\forall j \quad lb_j \leq split_j \leq ub_j \tag{3}$$

a snažíme se nalézt takový vektor $\mathbf{v}$, že klasifikátor se změkčeními, jejichž parametry jsou kódovány vektorem $\mathbf{v}$, má co nejmenší chybu na trénovací množině T.

```
**function** LearnBounds($stop, step, stepcount$)
    $v \leftarrow$ kód takových parametrů, že $\forall j \quad lb_j = ub_j = split_j$
    $bestval \leftarrow v$
    $besterr \leftarrow$ počet chyb klasifikátoru bez změkčení na trénovací množině $T$
    $unsuccess \leftarrow 0$
    while $unsuccess < stop$ do
        $d \leftarrow$ nenulový náhodný vektor z rovnoměrného rozdělení na intervalu $\langle -1, 1\rangle^{2m}$
        $delta \leftarrow d \cdot step \,/\, \|d\|$
        $success \leftarrow$ false
        for all $s \in \{1, \ldots, stepcount\}$ do
            $v \leftarrow v + delta$
            Složky vektoru $v$, které porušují podmínku (3), nastav tak,
            aby byla podmínka (3) splněna s rovností.
            $err \leftarrow$ počet chyb klasifikátoru s parametry změkčení kódovanými
                    vektorem $v$ na trénovací množině $T$
            if $err < besterr$ then
              $bestval \leftarrow v$
              $besterr \leftarrow err$
              $success \leftarrow$ true
            end if
        done for
        $v \leftarrow bestval$
        if $success$ then
          $unsuccess \leftarrow 0$
        else
          $unsuccess \leftarrow unsuccess + 1$
        end if
    done while
    return $v$
```

Obrázek 3: Experimentální algoritmus

V každém cyklu našeho experimentálního algoritmu (viz obrázek 3) je náhodně zvolen směr, potom je dosud nejlepší nalezené řešení pozměňováno v tomto směru s pravidelným krokem $step$ až do zvolené vzdálenosti dané argumentem $stepcount$ (počet kroků). V každém takto posunutém vektoru parametrů je vypočtena chyba na trénovací množině a nejlepší hodnota je uchována jako výchozí pro další cyklus. Algoritmus končí po sekvenci cyklů, v nichž nedošlo ke zlepšení, jejíž délka je určena argumentem $stop$.

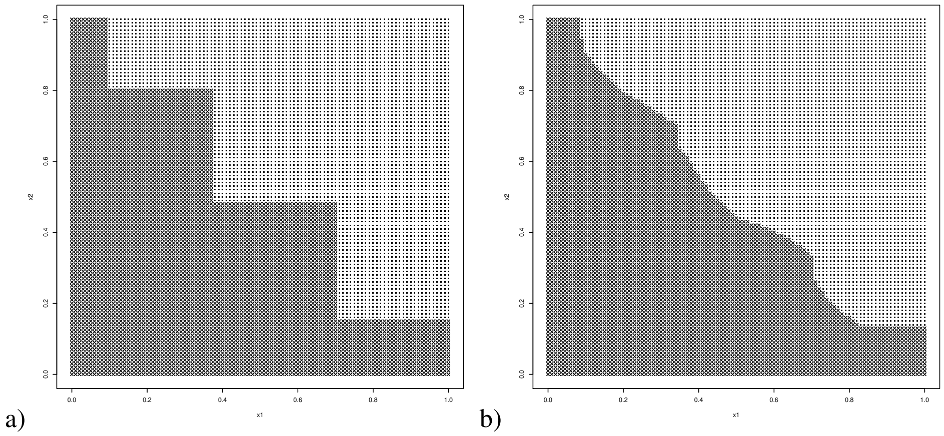
5. Efekt změkčování hran

Při změkčování hran popsaným způsobem se mění odhad pravděpodobnosti příslušnosti předloženého vzoru $\mathbf{x}$ do třídy $\mathcal{C}_b$ pouze pro vzory v blízkosti hranic hyperkvádrů - takové vzory, jejichž hodnota testovaného atributu x_{a_j} leží mezi lb_j a ub_j pro některé j. Chceme-li předloženému vzoru přiřadit index třídy, do níž nejspíše přísluší, použije se

$$\arg \max_{b=1,\ldots,c} P(b|\mathbf{x})$$

Je-li nyní jediný atribut vzoru $\mathbf{x}$ v blízkosti hranice hyperkvádrů, potom se takováto klasifikace nezmění. Aby došlo ke změně, musí se složit dohromady změkčení aspoň ve dvou uzlech stromu. Tedy klasifikace se může měnit v blízkosti rohů hyperkvádrů.

Viz Obrázek 4 Tato vlastnost je vidět na obrázku 4, který porovnává klasifikaci bez změkčení hran a



Obrázek 4: Výsledek klasifikace rozhodovacím stromem a) před změkčením hran b) po změkčení hran

se změkčením na umělých datech — dvou třídách oddělených diagonálou. Zde na intervalu $\langle 0, 1\rangle$ byly náhodně rovnoměrně vygenerovány trénovací vzory

$$\mathbf{x}^k = (x_1^k, x_2^k) \qquad k = 1, \ldots, 1000$$

Pokud bylo $x_1^k + x_2^k \leq 1$, byl vzor $\mathbf{x}^k$ zařazen do jedné třídy, jinak do druhé. Na základě této množiny byl vytvořen rozhodovací strom, jemuž byla ke klasifikaci předložena množina vzorů ležících v pravidelné mřížce. Výsledek je v levé části obrázku 4. Potom byly pro tento strom nalezeny parametry změkčení pomocí našeho experimentálního algoritmu a výsledek klasifikace se změkčením je v pravé části obrázku 4.

6. Závěr

Experimenty naznačují, že změkčování hran by mohla být perspektivní cesta ke zlepšení vlastností klasifikátorů založených na rozhodovacích stromech, protože je možné takto získat lepší výsledek klasifikace (menší počet chyb), přitom zůstává zachována většina výhod, jež klasifikátory s rozhodovacími stromy mají: Použitelnost na objekty s atributy různých typů zůstává — ačkoliv jsme uvedli, že se zabýváme pouze klasifikátory objektů s číselnými atributy, přesto máme-li strom pro klasifikaci objektů s některými atributy kategoriálními, je možné změkčení aplikovat pouze v uzlech s testy na číselné atributy, v ostatních uzlech ponechat strom beze změny. Stejně jako strom bez změkčených hran lze interpretovat jako posloupnost pravidel, strom se změkčením určuje posloupnost fuzzy pravidel.

Předmětem další práce bude hledání vhodného algoritmu, který by nalezl dostatečně dobré nastavení parametrů změkčení a přitom byl použitelný z hlediska výpočetních nároků. Přitom by mohlo pomoci prozkoumání souvislosti s modelem Hierarchical Mixtures of Experts (viz např. [2], kap. 9.5), který je též založen na rozhodovacích stromech, ovšem je komplikovanější, než zde zmiňované metody.

Literatura

[1] L. Breiman, J. H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Belmont CA: Wadsworth, 1993

[2] T. Hastie, R. Tibishirani and J. Friedman, *The Elements of Statistical Learning*, Springer, 2001

[3] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo–California, 1993

Evolutionary operators on ICodes

Post-Graduate Student:
ING. ROMAN KALOUS

FNSPE, Department of Mathematics,
Czech Technical University

Prague, Czech Republic

kalous@kmlinux.fjfi.cvut.cz

Supervisor:
ING. FRANTIŠEK HAKL, CSC.

Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

hakl@cs.cas.cz

Field of Study:
Stochastic optimisation
Classification:

Abstract

ICodes are used as representations of neural networks architectures in an evolutionary optimisation process. The evolutionary algorithm based on ICodes contains large amount of parameters, moreover, the corresponding search space itself if huge. To get deeper the insight into the transitional behaviour of the process, the mathematical properties of ICodes are explored a studied. This article gives initial definitions and propositions about structure of ICodes and about the corresponding evolutionary operators.

1. Introduction

The general purpose of ICodes is effective representation of *neural networks* architecture (the topology of the network along with additional parameters). Briefly, the neural network topologies are *acyclic oriented graphs*; the graphs, then, are represented via *cellular encoding* (CE)[1] due to F. Gruau. Those CE are represented via ordinary integer series – *read's codes* (RC) due to D. Read. Since the CE / RC represent only basic informations about the neural network, ie. topology, the additional instructions/informations/parameters are added in form of next entry of RC . This leads to an ICode : ICode represents the neural network topology and some of its other parameters which gives the architecture.

The RC (and so the ICode) can be defined independently on CE using so called *level-property*. This allows to narrow down on the properties of ICodes as mathematical objects. The main concern is put on the following

- state the cardinality of the set of all ICodes of given length, and even cardinality of the set of all ICodes – gives information about the search space,
- describe the evolutionary operators on ICodes using some binary operator on the set of ICodes – algebraic structure provides better formalism when discussing properties of evolution (stability, convergence, etc.),
- study further the transitional behaviour of such operators using the notation mentioned.

In the first section the ICodes are introduced using the level-property, the term of $\mathrm{SubICode}$ is described and showed in example, and the operator of *left-addition* $\oplus_k$ is defined. In the second section the evolutionary operators are described and rewritten in terms of the mapping $\oplus_k$. Finally, a little comparison

---

[1] This representation provides very convenient properties, on the other hand, it's only 'one-way'. That means, that only a graph can be built according to a representation; then, the representation of a general graph is not trivially constructed. One always has to consider the graphs as being built from some representation.

with the usual binary case evolutionary algorithm is provided in the last section, the yet found facts are summarized and future plans are reviewed.

The CE was introduced by F. Gruau in [Gruau94]. The RC was introduced by D. Read in [Read72]. For the ICodes , the approach with closer context with CE / RC is given in [HaKal04].

2. ICodes

In this article, the ICodes will be defined without explicit knowledge of CE (like it is in [HaHlaKal03] and [HaKal04]), as the basic term is stated the RC . The definition is due to a level-property – a set of inequations and one equation. The general RC consists of all non-negative numbers, still, in this article the RC is defined as only two-valued, which is for the ICodes enough to handle the neural network architectures.

Definition 1 *Let $N \in \mathbb{N}$. Define respectively*

1. *level-property: $\{a_j\}_{j=1}^{N},\ a_j \in \{0\} \cup \mathbb{N}, j \in \{1,\ldots,N\}$ fulfills a* level-property *if*

$$\sum_{j=1}^{k} a_j > k-1,\ k \in \{1,\ldots,N-1\},\ \sum_{j=1}^{N} a_j = N-1, \tag{1}$$

2. *for $N \in \{2k-1 \mid k \in \mathbb{N}\}$ the set of read's codes of length N:*

$$\mathbf{RC}(N) \overset{def}{=} \Big\{ \{a\}_{j=1}^{N} \ \Big|\ a_j \in \{0,2\},\ \{a\}_{j=1}^{N} \textit{ fulfills a level-property} \Big\}, \tag{2}$$

3. *the set of all read's codes:*

$$\mathbf{RC} \overset{def}{=} \bigcup_{\substack{N=2k-1\\ k\in\mathbb{N}}} \mathbf{RC}(N). \tag{3}$$

The level-property mentioned in previous proposition is of importance. First, the level-property provides easy random generating of an RC – starting from the first entry of a series, the next are generated at random and fulfilling the level-property. Second, the level-property defines the length of an RC subpart at any position $k \in \{1,\ldots,N\}$ – this will be described in the following paragraph. Third, the level-property can be helpfull when looking for lower/upper bounds of the cardinality of the set of ICodes .

The next important term is a *subcode* of an RC. Consider $N \in \{2k-1 \mid k \in \mathbb{N}\}$, $\{\mathrm{a}_j\}_{j=1}^{N} \in \mathbf{RC}(N)$. For each position $k \in \{1,\ldots,N\}$ there exist $N-k+1$ sequences consisting of values $\{\mathrm{a}_j\}_{j=k}^{l}$, $l = k,\ldots,N$. Then, only one of these is uniquely defined as fulfilling the level-property. This subsequence is called a subcode on position k. The subcode is found quite easily using level-property: it is started from position k and the sums $\sum_{j=1}^{l} \mathrm{a}_{k-1+j}$ are evaluated for $l \geq 1$; the seeking stops after the level-property is reached while the appropriate subcode of length l is found. Note that subcode on first position is the whole RC ; on the other hand any subcode on position k for which $\mathrm{a}_k = 0$ gives subcode of length 1. These situations are shown in table 1.

Position		**Original code:**												
		2	0	2	2	2	0	0	2	0	2	0	0	0
$k = 1$	$l - 1$	0	1	2	3	4	5	6	7	8	9	10	11	12
	$\sum_{j=1}^{l} a_{k-1+j}$	2	2	4	6	8	8	8	10	10	12	12	12	12
	subcode	2	0	2	2	2	0	0	2	0	2	0	0	0
$k = 7$	$l - 1$							0						
	$\sum_{j=1}^{l} a_{k-1+j}$							0						
	subcode							0						
$k = 10$	$l - 1$										0	1	2	
	$\sum_{j=1}^{l} a_{k-1+j}$										2	2	2	
	subcode										2	0	0	

Table 1. Subcode construction. For each position k the first row shows value of $l - 1$ which is the right side of the (in)equations in level-property, the second row shows the sum of a_{k-1+j} which is the left side of level-property, and the third row shows the subcode being built.

Following the last equation of level-property in (1) we gain (the term $[expr]$ is an indicator; if $expr$ may be true of false, the $[expr]$ returns 1 if $expr$ is true, and 0 otherwise)

$$\sum_{j=1}^{N} \mathrm{a}_j = \sum_{j=1}^{N} 2\,[\mathrm{a}_j > 0] = \\ = 2\left|\{a_j \mid a_j > 0,\ j \in \{1, \ldots, N\}\}\right| = N - 1$$

which gives the number of non-zero parts of an RC :

$$\left|\{a_j \mid a_j > 0,\ j \in \{1, \ldots, N\}\}\right| = \frac{N-1}{2}. \tag{4}$$

This number is an integer since N is odd. The number of zero parts of an RC is equal to $(N + 1)/2$.

To provide wider representation power, additional instruction entries are added. This structure is, then, called an *instruction code*, ICode . The values of the instructions are of two types according to the non-zero and zero entries of an ICode . They are called *building* and *terminating* instructions and the sets are assigned as **BI** and **TI** . The terms of length and subcode are intuitively adopted from read's codes.

Definition 2 *Let* $N \in \{2k - 1 \mid k \in \mathbb{N}\}$, $\{\mathrm{a_j}\}_{j=1}^{N} \in \mathbf{RC}(N)$. *The* ICode $\mathscr{P}$ *of length N is defined as*

$$\mathscr{P} \overset{def}{=} \{\mathrm{a}_j, \alpha_j\}_{j=1}^{N}, \tag{5}$$

where $(\forall j \in \{1, \ldots, N\})\ ((\mathrm{a}_j = 2 \Rightarrow \alpha_j \in \mathbf{BI}) \wedge (\mathrm{a}_j = 0 \Rightarrow \alpha_j \in \mathbf{TI}))$. *Set of all* ICodes *of length N is assigned as* $\mathbf{ICodes}(N)$. *The set of all* ICodes *is defined as*

$$\mathbf{ICodes} \overset{def}{=} \bigcup_{\substack{N=2k-1 \\ k \in \mathbb{N}}} \mathbf{ICodes}(N). \tag{6}$$

Let $\mathscr{P} \in \mathbf{ICodes}$, $\mathscr{P} = \{\mathrm{a}_j, \alpha_j\}_{j=1}^{N}$. *Next define the length of* $\mathscr{P}$

$$\left|\mathscr{P}\right| \overset{def}{=} N, \tag{7}$$

and a SubICode *on position* $k \in \{1, \ldots, \left|\mathscr{P}\right|\}$

$$\mathscr{Q} = sub\{\mathscr{P}, k\} = \{\mathrm{a}_j, \alpha_j\}, \tag{8}$$

such that $\{\mathrm{a}_j\}$ *is subcode of* $\{\mathrm{a}_j\}_{j=1}^{N}$ *on position k.*

Example 3 *Consider an* ICode

$$\mathscr{P} = \{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\{2,2\},\{0,11\},\{0,11\},\\ \{2,2\},\{0,11\},\{2,1\},\{0,11\},\{0,11\},\{0,11\}\} \tag{9}$$

Clearly, the length is equal to $\left|\mathscr{P}\right| = 13$. SubICode *on position* 7 $\mathscr{Q}_1 = \{\{0,11\}\}$ *is an* ICode *of length* 1, SubICode *on position* 4 $\mathscr{Q}_2 = \{\{2,2\},\{2,2\},\{0,11\},\{0,11\},\{2,2\},\{0,11\},\{2,1\},\{0,11\},\{0,11\}\}$ *is an* ICode *of length* 9.

The operator defined in the following definition just generalizes the most frequent operation performed with RC (with CE) which consists in replacing SubIcode on given position with another.

Definition 4 *Let* $\mathscr{P} \in \mathbf{ICodes}$, $\mathscr{P} = \{\mathrm{a}_j, \alpha_j\}_{j=1}^{N}$ *and* $\mathscr{Q} \in \mathbf{ICodes}$, $\mathscr{Q} = \{\mathrm{b}_i, \beta_i\}_{i=1}^{M}$, $k \in \{1,\ldots,\left|\mathscr{P}\right|\}$, $N_{sub} = \left|sub\{\mathscr{P},k\}\right|$. *The* left-addition $\oplus_k$ *is defined as follows.*

$$\mathscr{P} \oplus_k \mathscr{Q} = \{\mathrm{c}_l, \gamma_l\}_{l=1}^{P},$$

where

$$P = N - N_{sub} + M, \quad and$$

$$\{\mathrm{c}_l, \gamma_l\} = \begin{cases} \{\mathrm{a}_l, \alpha_l\}, & l = 1,\ldots,k-1 \\ \{\mathrm{b}_{l-k+1}, \beta_{l-k+1}\}, & l = k,\ldots,k+M-1 \\ \{\mathrm{a}_{l-M+N_{sub}}, \alpha_{l-M+N_{sub}}\}, & l = k+M,\ldots,P \end{cases} \tag{10}$$

Proposition 5 *For any* $\mathscr{P} \in \mathbf{ICodes}$, $\mathscr{P} = \{\mathrm{a}_j, \alpha_j\}_{j=1}^{N}$, *any* $\mathscr{Q} \in \mathbf{ICodes}$, $\mathscr{Q} = \{\mathrm{b}_i, \beta_i\}_{i=1}^{M}$, *and any* $k \in \{1,\ldots,N\}$, *the sequence* $\{\mathrm{c}_l, \gamma_l\}_{l=1}^{P} = \mathscr{P} \oplus_k \mathscr{Q}$ *is an* ICode *of length* $N - \left|sub\{\mathscr{P},k\}\right| + M$, *ie.*

$$\mathscr{P} \oplus_k \mathscr{Q} \in \mathbf{ICodes}(N - \left|sub\{\mathscr{P},k\}\right| + M).$$

As for the description of $\mathbf{ICodes}(N)$, one of the first concerns is the cardinality of this set. $\oplus_k$ helps among other to formalize the upper bound. For the first two values of N hold the following equations

$$\left|\mathbf{ICodes(1)}\right| = 1, \quad \left|\mathbf{ICodes(3)}\right| = 1. \tag{11}$$

Consider $\mathscr{P} \in \mathbf{ICodes}(N-2)$, $k \in \{1,\ldots,N-2\}$ such that $\mathrm{a}_k = 0$. Next, let $\mathscr{Q} = \{\{2,\beta_1\}, \{0,\beta_2\}, \{0,\beta_3\}\}$, $\beta_1 \in \mathbf{BI}$, $\beta_2, \beta_3 \in \mathbf{TI}$. Since $\left|\mathrm{sub}\{\mathscr{P},k\}\right| = 1$ and $\left|\mathscr{Q}\right| = 3$, the ICode $\mathscr{P} \oplus_k \mathscr{Q}$ is member of $\mathbf{ICodes}(N)$. Applying this operation to every $\mathscr{P} \in \mathbf{ICodes(N-2)}$ for every $k \in \{1,\ldots,N-2\}$, $\mathrm{a}_k = 0$, the set $\mathbf{ICodes}(N)$ is reached. Because different ICodes lead after this step to the identical ICodes (symmetry), the recursive schema gives only upper bound. Next, since instructions at every position are evaluated in $\mathbf{BI}$ or $\mathbf{TI}$, the cardinality is influenced by the cardinalities $\left|\mathbf{BI}\right|$ and $\left|\mathbf{TI}\right|$.

$$\begin{aligned} &\left|\mathbf{ICodes(N)}\right| \leq \\ &\leq \mathbf{BI}^{\frac{N-1}{2}}.\mathbf{TI}^{\frac{N+1}{2}}.\tfrac{(N-2)+1}{2}.\left|\mathbf{ICodes(N-2)}\right| \leq \mathbf{BI}^{\frac{N-1}{2}}.\mathbf{TI}^{\frac{N+1}{2}}.\tfrac{(N-2)+1}{2}.\ldots.\tfrac{3+1}{2}.\tfrac{1+1}{2} = \\ &= \mathbf{BI}^{\frac{N-1}{2}}.\mathbf{TI}^{\frac{N+1}{2}}.\left(\tfrac{N-1}{2}\right)! \end{aligned} \tag{12}$$

This upper bound of the cardinality grows as $\left(\frac{N-1}{2}\right)^{\frac{N-2}{2}}$ (using Stirling's formula), on the other hand the duplicities caused by the symmetry may lower it significantly. The recursive schema for the exact number of symmetries in $\mathbf{ICodes}(N)$ wasn't found yet.

3. Evolutionary Operators Defined on ICodes

The operators play a vital role in the model of evolutionary algorithms while their definition determines the transitional behaviour of the system. The operators defined for ICodes are intuitively adopted from the CE theory; the operations on subtrees in CE are due to level-property equivalently defined on SubICodes.

3.1. Mutation

Mutation is usually described as randomly changing the subparts of a representation. This approach is kept, mutation is considered as a function $\mathcal{M}$ mapping an ICode $\in \mathbf{ICodes}$ onto another ICode , formally

$$\mathcal{M} : \mathbf{ICodes} \to \mathbf{ICodes}. \tag{13}$$

Technically, the mutation is realized as a change of SubICode on randomly chosen position. Let $\mathscr{P} \in \mathbf{ICodes}$. The mutation $\mathscr{Q} = \mathcal{M}(\mathscr{P})$ proceeds as follows:

1. Choose randomly position $k \in \{1, \dots, |\mathscr{P}|\}$. This can be done according to an arbitrary distribution on $\{1, \dots, |\mathscr{P}|\}$, e.g. uniform.
2. Generate randomly ICode $\mathscr{Q}_{tmp}$.
3. Substitute the part of $\mathscr{P}$ corresponding to a SubICode on position k with $\mathscr{Q}_{tmp}$.

Using notation of $\oplus_k$ mutation can be formally written as

$$\mathcal{M}(\mathscr{P}) = \mathscr{P} \oplus_k \mathscr{Q}_{tmp}, \tag{14}$$

where $k \in \{1, \dots, |\mathscr{P}|\}$ and $\mathscr{Q}_{tmp} \in \mathbf{ICodes}$ are random.

The length of $\mathcal{M}(\mathscr{P})$ is bounded as $|\mathscr{Q}| \leq |\mathcal{M}(\mathscr{P})| \leq |\mathscr{Q}| + |\mathscr{P}| - 1$. In case the first position is chosen at step 1 of the mutation mechanism, it actually means that the whole ICode is interchanged with new random ICode (the lower bound is reached). On the other hand picking up the positions with non-zero entries means growing the ICode (the upper bound is reached, the resulting ICode is of length that is greater or equal of the mutated one).

Example 6 *Let* $\mathscr{P}$ *be the* ICode *as in example 3,* 8 *the randomly picked position, and* $\mathscr{Q}_{tmp} = \{\{2,2\}, \{0,11\}, \{0,11\}\}$ *the random* ICode. *The resulting* ICode *is created as (the interchanged* SubICodes *are emphasised):*

$$\begin{aligned}
&\mathcal{M}(\mathscr{P}) = \\
&= \mathcal{M}\big(\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\{2,2\},\{0,11\},\{0,11\},\textit{\{2,2\},\{0,11\},\{2,1\},\{0,11\},\{0,11\}},\{0,11\}\}\big) = \\
&= \big\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\{2,2\},\{0,11\},\{0,11\},\textit{\{2,2\},\{0,11\},\{0,11\}},\{0,11\}\big\} = \\
&= \mathscr{Q}.
\end{aligned} \tag{15}$$

In this case, the mutation maps member of $\mathbf{ICodes}(13)$ *to a member of* $\mathbf{ICodes}(11)$.

3.2. Crossover

Crossover is the operator that recombines the subparts of its operands. It is considered as mapping

$$\mathcal{C} : \mathbf{ICodes} \times \mathbf{ICodes} \to \mathbf{ICodes} \times \mathbf{ICodes}. \tag{16}$$

Realization of crossover is given as interchange of SubICodes. Let $\mathscr{P}_1, \mathscr{P}_2 \in \mathbf{ICodes}$. The crossover $(\mathscr{Q}_1, \mathscr{Q}_2) = \mathcal{C}(\mathscr{P}_1, \mathscr{P}_2)$ proceeds as follows:

1. Choose randomly positions $k_1 \in \{1, \dots, |\mathscr{P}_1|\}$, $k_2 \in \{1, \dots, |\mathscr{P}_2|\}$. This can be done according to an arbitrary distributions on $\{1, \dots, |\mathscr{P}_1|\}$, $\{1, \dots, |\mathscr{P}_2|\}$, e.g. uniform.
2. Create ICode $\mathscr{Q}_1$ substituting the part of $\mathscr{P}_1$ corresponding to a SubICode on position k_1 with part of $\mathscr{P}_2$ corresponding to a SubICode on position k_2.
3. Create ICode $\mathscr{Q}_2$ substituting the part of $\mathscr{P}_2$ corresponding to a SubICode on position k_2 with part of $\mathscr{P}_1$ corresponding to a SubICode on position k_1.

Using notation of $\oplus_k$ crossover can be formally written as

$$\mathcal{C}(\mathscr{P}_1, \mathscr{P}_2) = (\mathscr{P}_1 \oplus_{k_1} \text{sub}\{\mathscr{P}_2, k_2\}, \mathscr{P}_2 \oplus_{k_2} \text{sub}\{\mathscr{P}_1, k_1\}). \tag{17}$$

where $k_1 \in \{1, \dots, |\mathscr{P}_1|\}$ and $k_2 \in \{1, \dots, |\mathscr{P}_2|\}$ are random.

Example 7 *Let* $\mathscr{P}_1 = \mathscr{P}$*,* $\mathscr{P}_2 = \mathscr{Q}$ *from the mutation example,* 8 *the randomly picked position in* $\mathscr{P}_1$*, and* 5 *the randomly picked position in* $\mathscr{P}_2$*. The crossover proceeds as (the* $\mathrm{SubICodes}$ *are emphasised):*

$$\begin{aligned}
&\mathcal{C}(\mathscr{P}_1, \mathscr{P}_2) = \\
&= \mathcal{C}\big(\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\{2,2\},\{0,11\},\{0,11\},\textit{\{2,2\},\{0,11\},\{2,1\},\{0,11\},\{0,11\}},\{0,11\}\}, \\
&\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\textit{\{2,2\},\{0,11\},\{0,11\}},\{2,2\},\{0,11\},\{0,11\},\{0,11\}\}\big) = \\
&= \big(\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\{2,2\},\{0,11\},\{0,11\},\textit{\{2,2\},\{0,11\},\{0,11\}},\{0,11\}\}, \\
&\{\{2,2\},\{2,1\},\{0,11\},\{2,2\},\textit{\{2,2\},\{0,11\},\{2,1\},\{0,11\},\{0,11\}},\{0,11\}\}\big) = \\
&= (\mathscr{Q}_1, \mathscr{Q}_2).
\end{aligned} \tag{18}$$

4. Conclusion

This article summarizes the basic terms and facts about the evolutionary algorithm based on ICodes which is used to optimize neural networks architecture. Since the algorithm is already defined, and even implemented (see [HlaKal03]), the model is currently undergoing the testing phase. In this sense, the monitoring of such highly parametrized evolutionary algorithm would be very helpful tool.

In [Vose99], the population model of the binary (and general c-ary) case of evolutionary algorithm is introduced. The population model approach of the evolutionary algorithm monitoring works with population as a basic entity, and controls its transitional behaviour. This requests sufficient algebraic structure of the search space, and explicit notation of the evolutionary operators. It is of interest to state these terms for the $\mathbf{ICodes}$ set. Currently, the upper bound of the cardinality is stated, the operator realizing evolutionary operators is explicitly described. The comparison of the binary case of evolutionary algorithm (the most discussed in the theory of evolutionary algorithm) is shown in table 2.

As for the future plans, the structure of $\mathbf{ICodes}(N)$ and $\mathbf{ICodes}$ will be further studied to get more precise algebraical description, or at least some sufficient approximation. The analysis of the results emerged from testing runs of the actuall implementation along with the next testing is supposed.

	Binary case	**ICodes**
Search space Ω	$(\mathbb{Z}_2)^l$	$\mathbf{ICodes}(N)$
Cardinality of Ω	2^l	$\leq \mathbf{BI}^{\frac{N-1}{2}}.\mathbf{TI}^{\frac{N+1}{2}}.\left(\frac{N-1}{2}\right)!$
Algebraic structure	finite field $\oplus \dots$ logical XOR $\otimes \dots$ logical AND	*not known* $\oplus_k \dots$ left-addition
Random generating	entry-wise, binomial ($p \in (0,1)$), independent for each entry	using level-property
Evolutionary operators	Crossover with mask $c \in \Omega$: $\mathcal{C}(i,j) = (k,l)$ $k = (i \otimes c) \oplus (j \otimes \bar{c})$ $l = (i \otimes \bar{c}) \oplus (j \otimes c)$ Mutation with mask $m \in \Omega$: $\mathcal{M}(j) = k = j \oplus m$	Crossover on position $k \in \{1, \dots, min\{\lvert\mathscr{P}_1\rvert, \lvert\mathscr{P}_2\rvert\}\}$: $\mathcal{C}(\mathscr{P}_1, \mathscr{P}_2) = (\mathscr{Q}_1, \mathscr{Q}_2)$ $\mathscr{Q}_1 = \mathscr{P}_1 \oplus_{k_1} \text{sub}\{\mathscr{P}_2, k_2\}$ $\mathscr{Q}_2 = \mathscr{P}_2 \oplus_{k_2} \text{sub}\{\mathscr{P}_1, k_1\}$ Mutation on position $k \in \{1, \dots, \lvert\mathscr{P}\rvert\}$: $\mathcal{M}(\mathscr{P}) = \mathscr{P} \oplus_k \mathscr{Q}_{tmp}$

Table 2. Comparison of the properties of the evolutionary algorithm based on ICodes with the binary case.

References

[Gruau94] Gruau F., *Neural Network Synthesis using Cellular Encoding and The Genetic Algorithm*, Doctor Thesis, Lyon, 1994.

[HaHlaKal02] Hakl F., Hlaváček M., Kalous R., *Application of Neural Networks Optimized by Genetic Algorithms to Higgs Boson Search*, In: Computational Science, pp. 554–563, Ed: (Sloot P.M.A., Tan C.J.K., Dongarra J.J., Hoekstra A.G.), Vol: 3. Workshop Papers, Berlin, Springer 2002, ISBN: 3-540-43594-8, ISSN: 0302-9743, Lecture Notes in Computer Science; 2331, Held: ICCS 2002. International Conference, Amsterdam, NL, 02.04.21-02.04.24, Grant: GA MPO(CZ)RP-4210/69/97; GA MŠk(CZ)LN00B096

[HaHlaKal03] Hakl F., Hlaváček M., Kalous R., *Application of Neural Networks to Higgs Boson Search*, Nuclear Instruments and Methods in Physics Research A, Vol. 502, 2003, pp. 489–491, ISSN: 0168-9002, Grant: GA MPO(CZ)RP-4210/69/97

[HlaKal03] Hlaváček M., Kalous R., *Structured Neural Networks*, In: PhD Conference, pp. 25–33, Ed: Hakl F., MatFyzPRESS Praha, 2003, ISBN: 80-86732-16-9. In Czech.

[HaKal04] Hakl F., Kalous R., *Evolutionary operators on DAG representation*, Proccedings of the International Conference on Computing, Communications and Control Technologies: CCCT '04, August 14-17, Austin, Texas, USA, 2004.

[Read72] Read R. C., *The coding of various kinds of unlabeled trees*, In: Graph theory and computing, pp. 153–182, Ed.: Read R. C., Academic Press 1972.

[Vose99] Vose M. D., *The Simple Genetic Alorithm, Foundations and Theory*, MIT Press, London, 1999.

Strong Definition of Performance Metrics and Parallel Genetic Algorithms

Post-Graduate Student:
ING. ZDENĚK KONFRŠT

Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2

Praha 6

konfrst@zikova.cvut.cz

Supervisor:
ING. MARCEL JIŘINA, DRSC.

Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

marcel@cs.cas.cz

Field of Study:
Artificial intelligence and biocybernetics
Classification: 3902V035

Abstract

As in many research works of parallel genetic algorithms (PGAs), claims of a super-linear speedup (super-linearity) have become so regular that some clarification is usually needed. This paper focuses on “the estimation” of computation characteristics from parallel computing. PGAs are stochastic based algorithms, so the application rules from parallel computing is not straight forward. We derive total (parallel) run times from population sizing, the estimation of selection intensity and convergence time. The flawless calculation of total run is essential for obtaining the characteristics such as speedup ($S(n,p)$) and others. However, the process of derivation such characteristics is not simple, it is possible as it is presented in the paper.

1. Introduction

In [6], the primary purpose of parallel computation is to reduce the computing time, it takes to reach a solution of a particular problem. By adding more processing elements (PE or processors p) to the computing system, the computing time of a parallel algorithm decreases by the number of processors. The improvement, the total parallel time with respect to the total time of a serial algorithm, is called as the parallel speedup of the algorithm.

Computing the speedup of a parallel algorithm is a well-accepted way of measuring its efficiency [9]. Although speedup is very common in the deterministic parallel algorithms field, it has been adopted [1, 2] in the parallel evolutionary algorithms field in a different flavors, not all of them with a clear meaning. Several definitions of speedup have been described to gather parallel genetic algorithms into the definition. According to [3], they are two types of speedups basically: *strong* and *weak* speedups.

I. Strong speedup	
II. Weak speedup	
	IIA. Speedup with solution-stop
	II.Aa. Versus panmixia
	II.Ab. Orthodox
	IIB. Speedup with predefined effort

Table 1: Taxonomy of Speedup measures.

1.1. Strong definition

Strong definition follows the meaning of speedup as it is in parallel computation. To avoid stochasticity of algorithms, We operate with average of independent runs in order to have representative time values. Tomassini and Alba [3] claimed that some practical problems arise with this type of the definition. And they give two reasons. First, it is difficult to decide whether or not a sequential EA (evolutionary algorithm) is the best algorithm. Second, that the researcher has to be aware of the faster algorithm solving any of the problems being tackled. These two reasons, they found hard to solve.

1.2. Weak definitions

Therefore they propose *weak definition* of speedup as the extend to which it is possible that a different algorithm exists (probably not an EA) that solves the problem faster in sequential mode. This definition helps to compare our PEA (parallel evolutionary algorithm) against well-known sequential EAs. The important point relating a weak definition is the stopping criterion. Speedup could be studied by imposing a *predefined global number of iterations* both to the sequential and to the PEA. This measure is called "Speedup with predefined effort", marked as II.B in Table 1. The measure compares two algorithms that are working out solutions of different fitness (quality), this breaking the fundamental statement of being "solving" the same problem with the "same" precision.

An *orthodox weak* definition, type II.Ab. in Table 1., uses termination criterion when a solution of the same quality had been found, optimal solution. One important consideration is the composition of the sequential EA. We could compare a panmictic (sequential single population) EA [7] with multi-deme EA of d demes (islands), each running on a different processor. This case is called *versus panmixia weak* comparison (Table 1., IIAa). The algorithm running on one processor is panmictic in this case, while the d islands that are using d processors represent a distributed migration model whose algorithmic behaviour is quite different from the panmictic one. This could provoke a very different result for the numerical effort needed to locate solution, and thus very different search times can be obtained (faster search for the distributed version).

In order to have a fair and meaningful speedup definition for PEAs, we need to consider exactly the same algorithm and then only change the number of processors, from 1 to d, in order to measure the speedup (*strong* or *weak orthodox*). In any case, the speedup measure should be close to the traditional definition of speedup as possible.

In this paper, we provide a methodology how to obtain *strong speedup* [5, 6] in parallel genetic algorithms. The methodology provide a step-by-step manual how to achieve the "closeness" to the traditional definition of speedup for parallel genetic algorithms.

2. Background

Apart from **selected measures** proposed in [6], we try to construct **speedup** $S(N, p)$ and **efficiency** $E(N, p)$. To get proper speedup, we need to obtain adequate procedure how to get run times of (parallel) genetic algorithms. In our opinion, the run times are based on a **good population sizing**, **the estimation of selection intensity**, **convergence time** and **calculation of total run**. In the next part, we pick the above mentioned topics one by one.

2.1. Population sizing for (P)GA

As it has been shown [5], the total number of individuals N' in the demes decreases (increases) based on a topology δ of a parallel genetic algorithm while still reaching a solution of the same quality. We describe it in a better way. Let's introduce a genetic algorithm. The genetic algorithm uses a population size N, runs on 1 processor and optimises function $f(.)$. Similarly, in a parallel genetic algorithm, the population size is N', it runs on p processors, processors are connected to δ topology and it optimises function $f(.)$.

As it has appeared in many studies, the number of individuals is the same for both genetic and parallel genetic versions, instead of scaling N' based on a topology δ. Using the same population sizes ($N = N'$), parallel versions are reaching the optimal solutions far quicker and it is a primary source of super-linearity.

Genetic algorithm Output of Gamler's ruin model operates with equations (1) and (2). The equation (1) gives the probability of failure in regards with an average of $\widehat{Q}$ partitions correct and the equation (2) is a population sizing for GA, where k order of BBs, σ_{bb}^2 is the average BB variance, $m' = m - 1$ is the number of partitions and d is difference between the best and second best BBs. For more details, see [5].

$$\alpha = 1 - \frac{\widehat{Q}}{m}, \tag{1}$$

$$n = -2^{k-1} ln(\alpha) \frac{\sigma_{bb}\sqrt{\pi m'}}{d}. \tag{2}$$

Parallel genetic algorithm with isolated demes The equation (3) shows the required probability per deme is relaxed as more demes are used, where $\mu_{r:r} \approx \sqrt{\sqrt{2}\ln r}$ and the number of demes is r. This equation (3) leads to the population sizing equations (4) and (5).

$$\widehat{P} = \frac{\widehat{Q}}{m} - \frac{\mu_{r:r}}{2\sqrt{m}}, \tag{3}$$

$$n_{isol} = 2^k \ln(1 - \widehat{P}) \frac{\sigma_{bb}\sqrt{\pi m'}}{2d}. \tag{4}$$

Parallel genetic algorithm with maximum connections

$$n_{cg} = \sqrt{-2^k \ln(1 - \widehat{P})} \frac{\sigma_{bb}\sqrt{\pi m'}}{2d}. \tag{5}$$

Parallel genetic algorithm with sparse connections

There, we want to construct an equation for demes, which are not isolated nor maximally connected. First, we take out from the equation above parts which are the same. That is $\frac{\sigma_{bb}}{2d}.\sqrt{\pi m'}$. Second, the term $[2^k \ln(1 - \widehat{P})]^\beta$, where β has to be defined. Surely, $\beta \in < 0.5; 1 >$, where the borders of the interval are from isolated and maximum connected demes. The numbers inside of the interval are for sparsely connected demes. Third, we have to defined how to change j based on the number of demes (vertices V) and connections (edges E) between them (as the graph $G(V, E)$).

$$j = f(V, E) \begin{cases} V, E = 0 & j = 1 \\ V, E & j = 1 + \frac{\binom{V}{2} - (\binom{V}{2} - E)}{\binom{V}{2}} \\ V, E = \binom{V}{2} & j = 2 \end{cases} \tag{6}$$

$$\text{the number of decimal points in } j \rightarrow d, \quad \beta = (\frac{10}{10})^d \frac{1}{j}, \tag{7}$$

$$n_{sp} = \frac{\sigma_{bb}}{2d} \sqrt{\pi m'} (-2^k \ln(1 - \widehat{P}))^\beta. \tag{8}$$

2.2. Selection intensity

OneMax problem is often used a test function. It is defined as $f_{OneMax} = \sum_{i=1}^{l} xi$, where l is the number of BBs and x_i the value of ith gene.

The mean and variance of fitness of fitness of the population can be approximate as a normal distribution with mean μ_t and variance σ_t^2. Therefore, $\mu_t = l.p_t$ and $\sigma_t^2 = l.p_t(1 - p_t)$ and p_t represents the proportion of correct BBs of the population in generation t. Mülenbein and Schlierkamp [8] proposed a convergence equation for the problem and ordinary selection schemes as follows

$$\mu_{t+1} = \mu_t + I_c \sigma_t. \tag{9}$$

There, I_c is the complete selection intensity that is defined as the expected increase in the average fitness of a population after selection operation.

Selection intensities are not the same for different selection schemes. In the equation (10), tournament based selection intensity I_s is given by [4], where Φ and ϕ are the normal distribution function and normal density function with zero mean and unit variance. Other selection intensities for other selection schemes present in the Table 1.

Selection schemes	Parameters	I_s
Tournament	s	$\mu_{s:s}$
(μ, λ)	(μ, λ)	$\frac{1}{\mu}\sum_{i=\lambda-\mu+1}^{\lambda} \mu_{i:\lambda}$
Linear Ranking	n^+	$(n^+ - 1)\frac{1}{\sqrt{\pi}}$
Proportional	σ_t, μ_t	σ_t/μ_t

Table 2: Selection intensities for various selection schemes.

$$I_s = \mu_{s:s} = s \int_{\infty}^{-\infty} x\phi(x)(\Phi(x))^{s-1} dx. \tag{10}$$

When we use parallel genetic algorithm, the complete intensity I_c is sum of I_s and I_m like

$$I_c = I_s + I_m. \tag{11}$$

For simple GA, it holds $I_m = 0$ and $I_c = I_s$. Migration intensity I_m is a sum of the selection intensity caused by selecting the best individuals to emigrate I_e and the replacement intensity replacing the worst individuals I_r:

$$I_m = I_e + I_r = \delta\phi(\Phi^{-1}(1-\rho)) + \phi(\Phi^{-1}(1-\delta\rho)). \tag{12}$$

The definition of the complete selection intensity is used in the next when we compute convergence times of different parallel genetic algorithms.

2.3. Convergence time

From equation (9), it leads to the equation (13) as presented in [8].

$$p_t = \frac{1}{2}[1 + sin(t.\frac{I_c}{\sqrt{l}} + arcsin(2p_o - 1))] \tag{13}$$

When we put $p_t = 1$, we derive a convergence time. The convergence time G(t) is the number of generations before convergence occurs as (14). As p_o stands for the initial proportion of bits correct, in the case of OneMax is often 0.5.

$$G(t) = (\frac{\pi}{2} - arcsin(2p_o - 1))\frac{\sqrt{l}}{I_c}$$
$$\approx (p_o = 0.5) \approx \frac{\pi}{2}\frac{\sqrt{l}}{I_c}. \qquad (14)$$

The simplified equation (14) gives the number of generation until convergence as a fraction of square root of l-bit and selection intensity I_c.

2.4. The size of input data for PGA

When the total parallel runtime is measured, we trace the number of processors and the size of input data. What is the size of input data in PGA? The size of input data is the size of total population N as well as the representation of an individual. That signify the sum of population over all demes, when the population is divided into many demes (sub-populations) and it is not stored globally. The increase of a population size makes the total runtime longer and vice versa. In next, the size of input data is the total population size N.

2.5. Construction of total runtimes for PGAs

The total parallel run time $T_{tot}(N, p)$ is sum of an evaluation time T_{eval} and a communication time T_c. The algorithm does not converge in one epoch, but many, so the number of epoch to convergence τ is there. The total parallel run time is $T_{tot}(N, p) = \tau(T_{eval} + T_c)$.

The evaluation time T_f is a time for evaluation of one individual. To get the appropriate estimate for calculation of the evaluation time, the number of individuals n has to be added and the convergence time $G(t)$. Then the evaluation time for the whole deme is $G(t).n.T_f$.

$$T_{tot}(N, p) = \tau(G(t).n.T_f + \delta T_c). \qquad (15)$$

The communication time T_c is a time, which is spent in communication with other demes. It depends on migration rate ρ, the number of individuals in deme (subdeme) n, topology of the demes δ and a length of individual (in case of OneMax() l. Also, the underlying network connection has to be represented somehow, therefore the latency of communication canal $T_{latency}$ was added.

$$T_c = T_{latency} + f(\rho, \delta, l, n) = T_{latency} + \frac{L_k}{B}$$
$$= T_{latency} + C.l.\rho.n. \qquad (16)$$

Variables are defined as: ρ - migration rate, δ - topology, L_k - length of transmitted message, B - communication canal bandwith, C - constant, l - length of individual (in case of OneMax(), BB), n - the number of individuals in deme (subdeme) and τ - the number of epochs to convergence.

3. Total parallel run times of PGAs

In Table 3 (below), the total parallel run times of various multi-deme parallel genetic algorithms are summarized. Table columns are GA type, the number of processors used, the total parallel run time and the communication time. The calculated values are employed further when the calculation of computing characteristics is needed. It is important to note that master-slave PGA, in Table row 2, is a different type of PGA, so the total parallel run time is not similar or close to the other ones at all.

The estimation of parallel run times of PGA is a starting point to determine the performance metrics of any parallel algorithm on any parallel system.

#	{P}GA	p	$T_{tot}(N,p)/\tau$	T_c
1	GA	1	$G(t).n.T_f$	0
2	PGA (M-S)	p	$n.T_f/p + k(p-1)T_c$	l.n/r
3	PGA (isl)	p	$G(t).n_{isl}.T_f$	0
4	PGA (sp1)	p	$G(t).n_{d1}.T_f + T_c$	T_{c4}
5	PGA (sp2)	p	$G(t).n_{d2}.T_f + T_c$	T_{c5}
6	PGA (cg)	p	$G(t).n_{cg}.T_f + T_c$	T_{cm}

Table 3: The type of {parallel} genetic algorithms, the number of available processors, the total run times $T(N,p)$ and the communication times T_c ($T_c : 0 < T_{c4} < T_{c5} < T_{cm}$) are summarized.

4. Basic performance metrics of PGAs

Compared to sequential algorithms, one additional dimension has to be considered while analyzing complexity of parallel algorithms: the number of processors. The dream of parallel computing community is to achieve linear speedup in solving problems. If the number of processors increases k times, we would like to see the time of the solution to decrease k times as well. Unfortunately, this is very hard to achieve in many cases and we need additional performance metrics to evaluate the quality of parallel algorithms.

A simple algorithm (below) shows how to find simple performance metrics as the speedup $S(n,p)$ and the efficiency $E(n,p)$. The algorithm follows the track of so called "strong definition" of performance metrics (in this case, speedup $S(n,p)$).

4.1. Definitions of performance metrics

In this part, we define two common performance metrics: the speedup $S(n,p)$ and the efficiency $E(n,p)$.

Definition of the speedup $S(n,p)$ is

$$S(n,p) = \frac{T_{tot}(GA)}{T_{tot}(PGA)}. \tag{17}$$

Definition of the efficiency $E(n,p)$ is

$$E(n,p) = \frac{S(n,p)}{p}. \tag{18}$$

4.2. Algorithm

The algorithm, how to obtain the theoretical speedup and efficiency, goes as follows. We want to achieve the speedup and the efficiency of parallel genetic algorithm. First, we need to find a version of simple genetic algorithm, which could be easily changed into a parallel version. For simple genetic algorithm, we calculate population size n, selection intensity I_s and the convergence time $G(t)$.

```
1. find SGA (simple genetic algorithm)
2. SGA $\rightarrow n, I_s, G(t)$
3. find a parallel version of SGA (PGA)
4. $n, I_s, G(t) \rightarrow d, n_d, I_c, G_p(t)$
5. calculate $T_{tot}(n,1)$ and $T_{tot}(n,p)$
6. calculate $S(n,p)$
7. calculate $S(n,p), p \rightarrow E(n,p)$
```

Table 4: Algorithm how to get $S(n,p)$ and $E(n,p)$ for PGA.

In the previous sections, there were mentioned steps and methods how to obtained population size n, selection intensity I_s and teh convergence time $G(t)$. Second, we construct a parallel multi-deme version of simple genetic algorithm. For the parallel version, we obtain the number of deme d, the size of deme n_d, selection intensity I_c and the convergence time $G_p(t)$. As far as we have those variables, we can calculate the estimations of parallel $T_{tot}(n,p)$ run times. Based on these ($T_{tot}(GA)$ and $T_{tot}(PGA)$), we can calculate the theoretical speedup $S(n,p)$ and efficiency $E(n,p)$.

5. Discussion

We have presented no sound validations of the proposed algorithm nor theoretical comparisons between types of PGAs. We proposed how to get performance characteristics in the way of strong definition, but the broader investigation and comparisons between theoretical estimations and real runs of parallel genetic algorithms are underway.

6. Conclusion

The paper shows how to obtain the theoretical speedup $S(n,p)$ and efficiency $E(n,p)$ for PGAs. The measures are based on strong definition of performance characteristics while using population sizing, the estimation of selection intensity and convergence time. The derivation of successive steps is not quite simple to get the result and it might be simplified soon.

References

[1] E. Alba, “Parallel evolutionary algorithms can achieve super-linear performance”, *Information Processing Letters*, vol. 82, 1, pp. 7–13, 2002.

[2] E. Alba, J. M. Troya, “Improving flexibility and efficiency by adding parallelism to genetic algorithms”, *Statistics and Computing* , vol. 12, pp. 91–114, 2002.

[3] E. Alba, M. Tomassini, “Parallelism and Evolutionary Algorithms”,*IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 443–462, 2002.

[4] T. Bäck, “Generalized convergence models for tournament- and (μ, λ)-selection”, *Proc. 7th Int. Conf. Genetic Algorithms*, pp. 152–159, 1997.

[5] E. Cantú-Paz, “*Efficient and Accurate Parallel Genetic Algorithms*”, p. 162, 2000.

[6] Z. Konfršt, “On Super-linearity in Parallel Genetic Algorithms”, *2nd IASTED Int. Conf. on Neural Networks and Computational Intelligence 2004*, CD-ROM, 2004.

[7] Z. Konfršt, J. Lažanský, “Extended Issues of PGAs based on One Population”, *Neuro Fuzzy Technologies ’2002*, pp. 71–78, 2002.

[8] H. Mühlenbein, D. Schlierkamp-Voosen, “Predictive models for the breeder genetic algorithm: I.Continuous paramater optimization”, *Evol. Compu.*, vol .1, no. 1, pp. 25–49, 1993.

[9] P. Tvrdík, “*Parallel Systems and Algorithms*”, CTU Publishing House, p. 167, 1997.

Leaf Confidences for Random Forests

Post-Graduate Student:
ING. EMIL KOTRČ
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
kotrc@cs.cas.cz

Supervisor:
RNDR. PETR SAVICKÝ, CSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
savicky@cs.cas.cz

Field of Study:
Mathematical engineering
Classification: P3913

Abstract

Decision trees belong to the basic classification methods of machine learning. Usually, to achieve smaller generalization error, ensembles of trees (decision forests) are used instead of one single tree. Several methods of growing such ensembles exist at present, namely bagging, boosting and randomization of internal nodes. Boosting using methods are based on a strong classifiers and usually use some confidences for each tree in the ensemble. Random Forests (Breiman 2001) is quite new method for building decision forests which adopted bagging and randomization of internal nodes for growing procedure. Random Forests technique differs from other forest based methods by combining weak classifiers with the same weights (confidences). Current paper shows that appropriately chosen leaf confidences may improve prediction of Random Forests of limited size. As an accessory product an useful statistical model is presented for better understanding the technique of Random Forests.

1. Introduction

This paper is concerned with *two class* classification problem and with the method of constructing classifiers in the form of an ensemble of decision trees called *Random Forests*(RF). We will briefly introduce methods of growing decision trees and forests and we will explain main differences between them. The main goal of this paper is to show a simple extension of RF which can improve the prediction of RF method under specific conditions.

At first we will give some basic notations and definitions. Assume a domain space $X \subseteq R^p$ with p numerical predictors (variables) and a label set of two classes $C = \{0, 1\}$. Our task is to build a classifier, which will be able to assign classes (labels 0, 1) to unknown cases from the domain space X. Formally, classifier is a function $h : X \rightarrow C$. To be able to construct such classifier h we will need a *training (learning)* set of cases with the known classification. Let $L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be our training set, where $x_i \in X$ is a case and $y_i \in C$ is its class, $i \in \hat{n}$. This approach to building classifiers with using a learning set is known as *supervised learning*. A *testing* set is used to estimate accuracy of a classifier and to compare various classifiers. The testing set is similarly as the training set a set of cases with the known classification, denote $K = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ such testing set of m cases.

2. Decision trees and forests

Decision trees and forests belong to the basic statistical method for creating classifiers, for other see [8]. These methods became very popular because of a simple and fast learning scheme, for their easy interpretability and a good accuracy. At first we will briefly describe decision trees and after that we will speak about ensembles of trees called decision forests.

Decision tree is a rooted tree which consists of internal nodes called *decision nodes*, of *final leaves* and of *branches* (ways). Decision nodes contain some specific tests (splits) and final leaves gives the classification. The unknown (unseen) case starts at the root and passes through decision nodes, where a branch to other node is chosen according to the test, until the case encounters a final leaf which assigns a class to that case. Current methods for growing decision trees (CART [2], C4.5, C5.0 [10]) usually use two kinds of tests in internal nodes - ordinary binary splits and linear combinations of input variables, for details and other possible tests see [9]. We will use only the ordinary binary splits in this paper. These tests have the form $X_k \leq a$, where X_k is some of p input variable and a is some threshold. Decision nodes with these tests have only two links (branches) to other nodes (childs). The unseen case goes to the left child if it success the condition else it goes to the right child. These kinds of splits are then called *binary splits*. Some methods (CRUISE [15]) uses also more generalized scheme called *multiway splits* with more than two ways, but as you can see in [8] this approach is not very recommended. The exception is C4.5 which uses multiway splits only for categorical (values from a finite set) variables.

The growing procedure of all current methods is based on recursive partitioning of the learning set. Methods differ in the best split selection in decision nodes. CART and C4.5/5.0 are based on *impurity* of the node and the best split is chosen according *gini* or *entrophy* criterion, for details see [2], [10], [9], [8]. Methods CRUISE and QUEST use different approach in split selection based on ANOVA F-test, see [16], [15]. After a tree is grown all methods usually *prune* the decision tree to avoid *overfitting*, because perfectly trained tree may be very accurate on the training set but it has not to be so accurate on other data. This effect may be caused for example by noise in the learning set. There exist two basic pruning methods at present - *cost complexity* used in CART and *error based* pruning used in C4.5 and C5.0.

In current paper we will use the CART methodology for growing decision trees. In other words, we will consider only ordinary binary splits and the cost complexity pruning.

More than one tree can be used for classification to make the prediction better. An *ensemble of trees* (decision forest) is a set of several decision trees and a rule for combining their predictions, for example the majority voting scheme. Let $F = \{T_1, \ldots, T_N\}$ be the decision forest and let the function $G : X \to C$ be the prediction of the whole ensemble. The decision forest has to be a set of different trees. So the main problem is how to build several diverse trees by standard methodology from one training set. There exist three basic methods (and their modifications)

- Boosting - based on adaptive reweightening of training cases, see [8], [7]
- Bagging - based on random samples from the training set, see [3]
- Randomization of internal nodes, see [12], [11]

The paper [12] is concerned with experimental comparison of these three methods and it implies some interesting facts. It shows that bagging and randomization construct very similar classifiers, and that boosting is unusable when training set is affected by noise, or when mixture of classes occurs. Bagging and randomization are better approaches in such conditions.

Random Forest [4] is a method of growing decision forests which combines bagging and randomization of test selection in internal nodes. For each tree in the forest new training set (bootstrap sample) is drawn randomly with replacement from the original training set and in each node a random subset of input variables is selected to split on. The best split among these selected variables become the test in decision node. This procedure is called *random input selection*, you can find more randomization techniques in [11]. The criterion for best split selection is based on the CART methodology and the final tree is not pruned on contrary to CART. To summary, RF is method for growing an ensemble of weak classifiers. When RF is used for classification each tree in forest assigns a class to an unseen case and the final prediction is given according to majority of all votes. More generalized voting scheme based on leaf confidences for RF is described in the following section.

3. Leaf confidences

To define leaf confidences we will need more formalized and generalized two class voting scheme instead of a majority voting as pure RF does. Let $T_j(x)$ be a leaf reached by case x in the tree T_j. As v we will denote final leaf. Furthermore, we will assign to each leaf v its *confidence level*, given by some appropriate function c. Its range are real values, positive values mean preference for the class 1 and negative for the second class 0. The higher absolute value implies the higher confidence level. With the given leaf level of confidence c we can define the prediction of the whole ensemble of N trees as

$$G_{N,t}(x) = \begin{cases} 1 & \text{if } F_N(x) \geq t \\ 0 & \text{otherwise} \end{cases}$$

where

$$F_N(x) = \sum_{j=1}^{N} c(T_j(x))$$

is the sum of the leaf confidences of all leaves reached by the case x in the forest and t is a threshold. For example the simple majority voting scheme uses $t = N/2$.

Suggestions to leaf confidences came from papers [7] and [6]. Shapire and Singer [7] introduce some ideas for leaf confidences by minimizing exponential loss function. Quinlan describes in [6] an ad hoc function called *Laplace correction* for leaf confidences in C4.5/5.0.
Furthermore, in current paper leaf confidences are functions of two statistics pos and neg. Let $pos(v)$ be the number of positive (with the class label equal to 1) cases from the training set which encountered the leaf v, and $neg(v)$ be the number of negative cases (class label 0). Random Forests grows trees until pure nodes (containing cases only from one class) are reached. These nodes are pure only on a random subsample (RF uses bagging) and do not have to be pure on the whole training set. So for each leaf v in each tree we are able to get couple of statistics $(pos(v), neg(v))$. Using these statistics we define leaf confidences as $c(v) = w(pos(v), neg(v))$ in current paper, where $w : N \times N \to R$ is an appropriate function. We have tested several functions w and some of them are described bellow.
At first we define leaf confidences

$$rf(pos, neg) \overset{def}{=} sign(pos - neg)$$

which simulates the original voting scheme from pure Random Forests. These rf confidences are based on the whole training set in opposite to pure Random Forests which assigns leaf confidences ($c \in \{0, 1\}$) only on the basis of a bootstrap subsample. In our experiments rf confidences reached approximately the same results as the pure RF. These rf confidences we have used mainly in our experiments with statistical model described in the next section. The differences between pure RF and rf confidences are caused by bagging because of omitting same cases.
As written above the paper [7] suggested some ideas for leaf confidences by minimizing the exponential loss function. Let us denote the simplest derived confidences with smoothing parameter ε as

$$q_{(\varepsilon)}(pos, neg) \overset{def}{=} \frac{1}{2} \ln \frac{pos + \varepsilon}{neg + \varepsilon}$$

These q weights (confidences) are very similar to the Quinlan's notation, see [6]. We tried to use both of these confidences but as none of them led to satisfactory results, our later work was dedicated to look for more accurate weights (confidences). The work seems to be partially successful because of new leaf confidences called *normalized difference* parameterized by h and α

$$nd_{(\alpha,h)}(pos, neg) \overset{def}{=} \frac{pos - neg - h}{(pos + neg)^{\alpha}}$$

The parameter h is appropriate for smoothing of unwanted effect of bagging when leaves are small and contain nearly the same amount of pos and neg cases. Results differed with the set and with the size of a forest.

Last confidences we describe in current paper are influenced by rf weights and sigmoidal functions. We were looking for a function with the similar behavior as the $signum$ function but more smoother and continuous. Sigmoidal functions have exactly these properties we wanted and we picked one of them $s(x) = (1 - e^{-x})/(1 + e^{-x})$ with the values in $(-1, 1)$. Leaf confidences we use with the function s are parameterized by k and are defined as

$$\sigma_{(k)}(pos, neg) \overset{def}{=} s\left(\frac{pos - neg}{k(pos + neg)}\right)$$

4. Statistical model for RF

For comparison of accuracy described leaf confidences on forests of various sizes we use ROC (Receiver Operator Characteristic) curves also known as signal versus background acceptance. The ROC curve is a set of points in $[0, 1]^2$. Each point $[a, b]$ on such curve expresses two probabilities - a is a probability that a case from the class 0 is classified as signal from class 1 (background acceptance), and b is a probability that the case from the class 1 is classified correctly (signal acceptance). The optimal point is clearly seen as the point $[0, 1]$, i.e. no noise is classified as signal and all signal particles are classified correctly. To generate more points on ROC curve we have to parameterize the classifier. In this paper we use the parameterized function $G_{N,t}$ defined above as the default classifier. We can define more precisely the ROC curve for our purpose as the set of points

$$\left[\frac{1}{|K_0|}\sum_{x_i \in K_0} G_{N,t}(x_i), \frac{1}{|K_1|}\sum_{x_i \in K_1} G_{N,t}(x_i)\right]$$

where $K_l = \{(x, y) \in K | y = l\}, l \in \{0, 1\}$. To obtain the ROC curve for given leaf confidences and forest with N trees we have to use all possible threshold parameter t. For example a curve for forest of N trees with rf confidences (values of rf are in $\{-1, 0, 1\}$) consists of $2N + 1$ points which is the amount of all possible values of threshold t. Parameter t for pure RF is from $\{0, 1, \ldots, N\}$.

Since growing a random forest is a random process we derived a statistical model in order to obtain an average behavior of a forest. For each case $x_i \in K$ and for each tested confidence c we estimated the mean value μ_i and standard deviation σ_i^2 of the random variable $c(\tilde{T}(x_i))$ $(= F_1(x_i))$, where $\tilde{T}$ represents a single random tree in a forest. These estimates are based on a random sample of 500 trees $c(T_1(x_i)), \ldots, c(T_{500}(x_i))$. Using estimated parameters μ_i and σ_i^2 we are able to express the expected value $E_F[G_{N,t}(x_i)]$ taken over the distribution F of random forests of size N using normal distribution $\mathrm{N}(N \cdot \mu_i, N \cdot \sigma_i^2)$. To get a point on the ROC curve the expected value of $G_{N,t}(x_i)$ over $x_i \in K$ is needed and is computed as an average over x_i of expected values $E_F[G_{N,t}(x_i)]$. This procedure is done for each threshold t and for both classes separately to get a ROC curve.

This statistical model implied from previous experiments that leaf confidences improve the prediction of a forest of limited size in a domain where background (noise) acceptance is low. This result was then verified by averaging ROC curves of 20 real forests with added leaf confidences. Since results from the model and from true forests are very close, this statistical model can be an useful tool for understanding of RF technique in further work.

5. Data sets

We have tested described leaf confidences on three data sets, with the same result - leaf confidences improve prediction of random forest of limited size.

5.1. MAGIC data set

This data set is generated by Monte Carlo code described in [5] and it simulates the detection of gamma (signal) and hadrons particles (noise) by MAGIC telescope[1]. The task is to separate these two kinds of par-

---

[1] http://hegra1.mppmu.mpg.de/MAGICWeb/

ticles. Various methods were used for classification, for more see the comparison study [1]. RF technique appeared to be one of the best compared methods. Since a mixture of gammas and hadrons, boosting decision trees is not usable. In opposite bagging and randomization seems to be a good choice as discussed in [12].

Data in MAGIC data set are described by 10 numerical predictors and consists of 12679 training cases and of 6341 testing cases, i.e. approximation in ratio 2:1.

5.2. Gaussian data set

The Gaussian data set tries to simulate similar behavior as the MAGIC data set, i.e. the mixture of signal and noise particles. Signal particles were generated as vectors of 5 variables each from $\mathrm{N}(0, 1)$ distribution restricted to the interval $[-5, 5]$. Noise particles were generated uniformly from the hypercube $[-5, 5]^5$. 5000 examples were used for the training procedure and the same amount for testing.

5.3. Friedman data set

Various authors use Friedman data set for testing their methods. This data set you can find for example in *mlbench* package for R [18]. For our purpose we use the Friedman2 data generator which generates data with four independent variables uniformly distributed over the ranges $X_1 \in \langle 0, 100\rangle$, $X_2 \in \langle 40\pi, 560\pi\rangle$, $X_3 \in \langle 0, 1\rangle$, $X_4 \in \langle 1, 11\rangle$ and with outcome given by formula $y = (X_1^2 + (X_2 \cdot X_3 - (1/(X_2 \cdot X_4)))^2)^{0.5} + e$, where e is from $\mathrm{N}(0, s^2)$ distribution. Since Friedman2 data are primarily used for regression we had to separate these data into two subsets giving two classes 0 and 1. It is done by the threshold parameter y_t equal to median of response of Friedman2 data with standard deviation s^2 equal to 0. The case belongs to the class 1 if the response y is greater than threshold y_t, otherwise it is put into the class 0. For training and for testing 10000 different cases were generated with default s^2.

6. Experimental results

In this paper we present some results from real RF with added leaf confidences instead of results from statistical model. Results from the model you can find in [17]. For each leaf confidence (or for pure RF without any confidences) 20 forests with N trees ($N \in \{20, 40, 80\}$) were grown and averaged. Results are shown in a table form. For each data set the table is created for all tested forest sizes (N) and for described leaf confidences (conf.t.), without rf confidences which are substituted by pure RF (labelled as $pureRF$). Values in each column represent signal acceptances (values on y-axis) at the fixed level of background acceptance (from 1% to 5% on x-axis of the ROC curve). Methods are then sorted in ascending order according to the average value (labelled as $aver.$) of these five signal acceptances. For each data set we also present the figure (1) which compares pure RF and the best leaf confidence on the forests with 20 and 80 trees.

As follows from tables, leaf confidences reached best results on Friedman2 and on MAGIC data set. The improvement on the Gaussian data is negligible, as you can see in figure (1), there is in fact "nothing" to improve because the ROC curves are near the upper border which is equal to 1. Regardless $\sigma_{(1)}$ and $nd_{(1,0)}$ are better than pure RF in average. On the other hand all shown leaf confidences appear to improve prediction for all tested forest sizes on the Friedman2 data set, as you can see in the table (3). On this data set a forest with 20 trees with leaf confidences is so accurate and sometimes better than the pure RF with 80 trees.

The situation is not so simple on MAGIC data, as the size of the forest is increasing the order of methods differs. For the forest with 20 trees confidences $nd_{(0.9,0)}$ and $nd_{(0.9,2)}$ reach the best results, but on the forest with 80 trees the best results are gained by $\sigma_{(1)}$ and $nd_{(1,0)}$. A problem occured with pure RF with 20 trees on MAGIC data set at 1% of background acceptance. Because no results at this acceptance level were reached no results of other methods are included. This problem is caused by RF which produces too few points (only 21) on the ROC curve for 20 trees. It can be smoothed out by rf confidences (see [17]), but as mentioned above rf confidences are not the same as the pure RF voting. That is why only pure RF is included in tables for comparison RF with leaf confidences.

N	conf.t.	1%	2%	3%	4%	5%	aver.
20	$pureRF$	NA	0,359798	0,455183	0,520818	0,577965	0,478441
20	$q_{(0.01)}$	–	0,378697	0,460204	0,526818	0,579506	0,486307
20	$nd_{(1,0)}$	–	0,390854	0,474155	0,536786	0,595172	0,499241
20	$\sigma_{(1)}$	–	0,39075	0,474167	0,536792	0,595536	0,499311
20	$nd_{(0.9,2)}$	–	0,403704	0,480054	0,539571	0,59387	0,504299
20	$nd_{(0.9,0)}$	–	0,402737	0,48113	0,544539	0,597659	0,506516
40	$pureRF$	0,264407	0,37407	0,460703	0,533453	0,59553	0,445632
40	$q_{(0.01)}$	0,289066	0,397446	0,477791	0,545147	0,595761	0,461042
40	$nd_{(0.9,2)}$	0,312698	0,410448	0,48819	0,551843	0,604184	0,473472
40	$nd_{(0.9,0)}$	0,311524	0,410229	0,490039	0,550724	0,607504	0,474004
40	$\sigma_{(1)}$	0,30627	0,411986	0,490629	0,555668	0,611828	0,475276
40	$nd_{(1,0)}$	0,30616	0,414911	0,490562	0,555881	0,612461	0,475995
80	$q_{(0.01)}$	0,289741	0,411129	0,490975	0,550979	0,603272	0,469219
80	$nd_{(0.9,0)}$	0,319867	0,410587	0,495281	0,557845	0,608441	0,478404
80	$nd_{(0.9,2)}$	0,321844	0,412564	0,493457	0,559189	0,606708	0,478752
80	$pureRF$	0,312004	0,411846	0,494624	0,565191	0,623614	0,481456
80	$nd_{(1,0)}$	0,326228	0,42511	0,49938	0,559529	0,615124	0,485074
80	$\sigma_{(1)}$	0,32593	0,427402	0,500456	0,559888	0,613987	0,485533

Table 1: Results on MAGIC data set

N	conf.t.	1%	2%	3%	4%	5%	aver.
20	$pureRF$	0,7852	0,93011	0,96466	0,97931	0,98714	0,929284
20	$q_{(0.01)}$	0,81653	0,92358	0,96327	0,98012	0,98764	0,934228
20	$\sigma_{(1)}$	0,8276	0,92554	0,96367	0,97972	0,98768	0,936842
20	$nd_{(1,0)}$	0,82776	0,92522	0,96372	0,97984	0,98774	0,936856
20	$nd_{(0.9,0)}$	0,83104	0,92692	0,96423	0,98006	0,98768	0,937986
20	$nd_{(0.9,2)}$	0,83392	0,92741	0,96406	0,98034	0,98822	0,93879
40	$q_{(0.01)}$	0,82626	0,92726	0,96538	0,9804	0,98847	0,937554
40	$nd_{(0.9,0)}$	0,83979	0,93038	0,96546	0,98028	0,98896	0,940974
40	$nd_{(0.9,2)}$	0,84095	0,93088	0,96572	0,9808	0,98928	0,941526
40	$pureRF$	0,84003	0,93262	0,96636	0,97981	0,98963	0,94169
40	$nd_{(1,0)}$	0,84479	0,92965	0,96578	0,98034	0,98902	0,941916
40	$\sigma_{(1)}$	0,84504	0,92932	0,96585	0,98038	0,98902	0,941922
80	$q_{(0.01)}$	0,83872	0,92797	0,96555	0,98067	0,98906	0,940394
80	$nd_{(0.9,0)}$	0,84145	0,9299	0,96574	0,98023	0,9893	0,941324
80	$nd_{(0.9,2)}$	0,84306	0,93126	0,96612	0,9812	0,98982	0,942292
80	$pureRF$	0,84302	0,93391	0,9673	0,98143	0,98948	0,943028
80	$\sigma_{(1)}$	0,84924	0,93066	0,96596	0,98025	0,98969	0,94316
80	$nd_{(1,0)}$	0,84954	0,93047	0,96592	0,98031	0,9897	0,943188

Table 2: Results on Gaussian data set

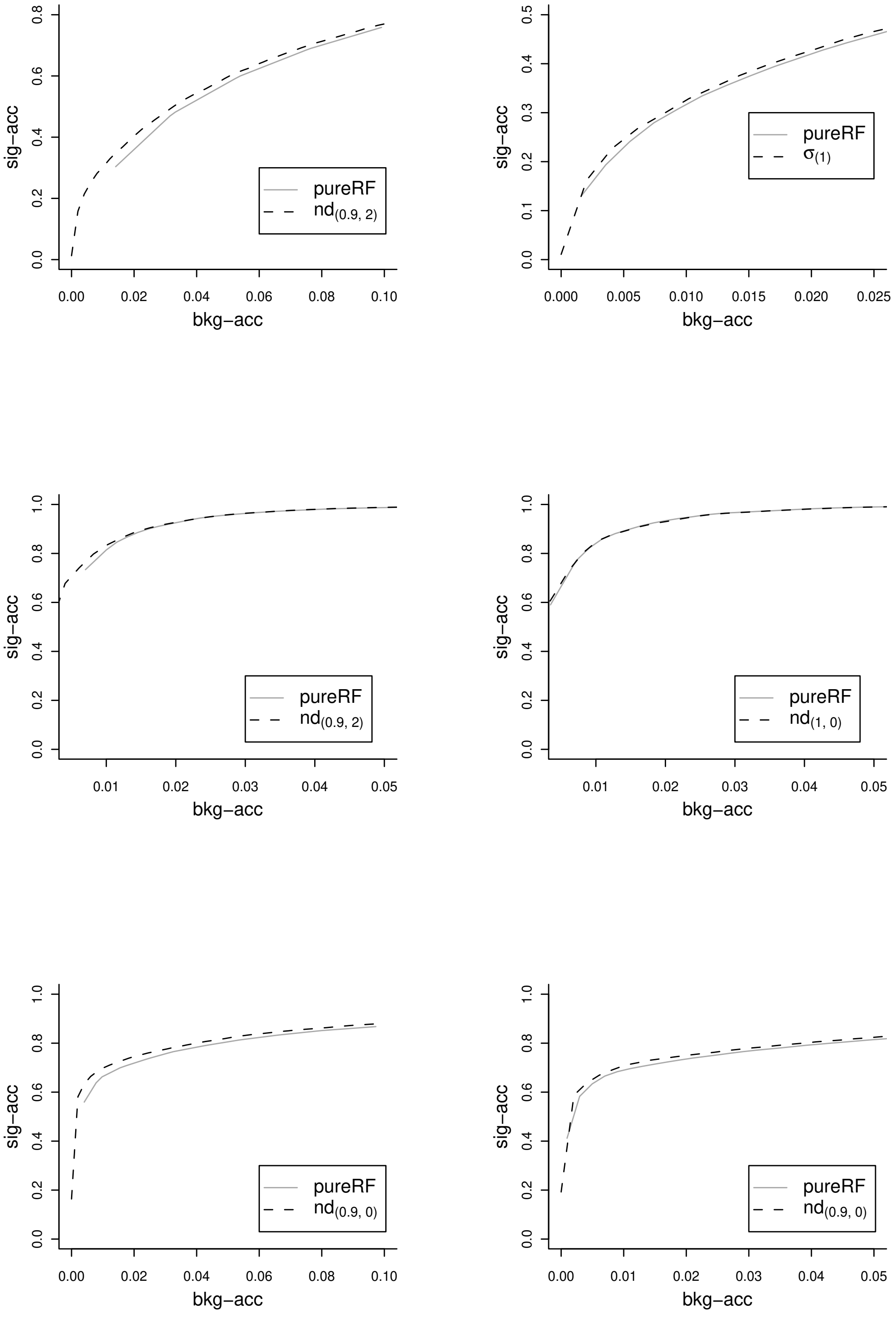


Figure 1: ROC curves for each tested data sets. The first row of graphs is on MAGIC, the second on Gaussian, and the third on Friedman2 data set. First column contains ROC curves for 20 trees and the second for 80 trees.

N	conf.t.	1%	2%	3%	4%	5%	aver.
20	$pureRF$	0,678443	0,718494	0,750332	0,777595	0,801844	0,745341
20	$\sigma_{(1)}$	0,686383	0,735935	0,770074	0,795669	0,81706	0,761024
20	$nd_{(1,0)}$	0,687437	0,736637	0,770386	0,79565	0,816846	0,761391
20	$q_{(0.01)}$	0,693952	0,742353	0,772903	0,795708	0,816572	0,764298
20	$nd_{(0.9,2)}$	0,695279	0,742626	0,773176	0,798654	0,819528	0,765853
20	$nd_{(0.9,0)}$	0,697347	0,745494	0,774756	0,799903	0,821323	0,767764
40	$pureRF$	0,677536	0,73373	0,764807	0,791251	0,814348	0,756334
40	$\sigma_{(1)}$	0,698791	0,742333	0,773586	0,79801	0,820025	0,766549
40	$nd_{(1,0)}$	0,6992	0,74286	0,773703	0,79803	0,819947	0,766748
40	$q_{(0.01)}$	0,704799	0,747132	0,776629	0,798127	0,81943	0,769224
40	$nd_{(0.9,2)}$	0,702146	0,745045	0,775322	0,801346	0,823381	0,769448
40	$nd_{(0.9,0)}$	0,703726	0,749005	0,777546	0,801697	0,822796	0,770954
80	$pureRF$	0,68849	0,73293	0,765675	0,792772	0,814739	0,758921
80	$\sigma_{(1)}$	0,70279	0,743679	0,775419	0,800468	0,820894	0,76865
80	$nd_{(1,0)}$	0,702829	0,74403	0,77581	0,800449	0,821069	0,768837
80	$nd_{(0.9,2)}$	0,704428	0,74645	0,776005	0,802926	0,823869	0,770735
80	$q_{(0.01)}$	0,709559	0,749785	0,778677	0,800956	0,820952	0,771986
80	$nd_{(0.9,0)}$	0,709013	0,750059	0,779087	0,803473	0,824561	0,773238

Table 3: Results on Friedman2 data set

7. Discussion and conclusion

Suggestions to the leaf confidences came from [7] and [6] and as shown in current paper on three data sets confidences may improve prediction of RF on the forests of limited size and when low background acceptance is needed. In cases where the mixture of classes occurs bagging appears to reach better results than boosting so bagging in RF is a good method. At present bagging is still not fully understood and it seems that leaf confidences improve the prediction because of so called leverage effect, see [14].

Let us have a briefly look at bagging. In a random subsample in bagging only 63% $\approx 1-\left(1-\frac{1}{|L|}\right)^{|L|}$ of training cases are used and a lot of them is chosen more than once to get the bootstrap sample of the same size as the training set. In Breiman's notation each case in a forest with N trees is approximately $N * 0.37$ times out of bag. This phenomenon may bring some troubles with bagging, as written in [14] - the main trouble with bagging does not lie in multi occurrence of some cases but in absence of some "important" cases. Leaf confidences may repair this effect of bagging by reweightening individual leaves in trees. The second conclusion from [14] implies that bagging can be less accurate when the influence of particles is the same, as seen on the Gaussian data.

These effects we have studied just on the Gaussian data, which have a simple structure and can be easy understood. Signal in Gaussian data is in fact a sphere inside a hypercube in a space of 5 dimensions and the noise is everywhere in hypercube because of uniform distribution. By easy calculation it can be proved that the radius of our sphere is 3.7 and we supposed to have problems with classification near this border because the amount of signal and noise particles became nearly the same. Since RF (and other tree based methods) divide the domain space (in Gaussian data a hypercube $[-5,5]^5$) into a rectangular areas, we have studied the effect of voting in these hyperrectangles. We constructed two forests of twenty trees to verify our theoretical assumptions. One forest was pure RF and the second one used $nd_{(0.9,2)}$ leaf confidences. All hyperrectangles defined by recursive partitioning were then separated into two sets in dependence of correctly/incorrectly classified testing cases. If the voting with leaf confidences improved a prediction of a case, all hyperrectangles containing this case will became members of a set marked as 1. Other hyperrectangles are members of the second set 0. So on the basis of the testing set we have separated all rectangles into two sets - one contains rectangles which improved classification by $nd_{(0.9,2)}$ and the second set contains the rest of them. This separation into two sets of hyperrectangles implied our theoretical assumptions. All rectangles labeled 1 contained cases near the border of the sphere and we found an interesting fact that the majority of them contained more than 14.5 positive (with the class label 1) cases.

As voting of RF is still not fully understood the further work will be concerned with this aspect. Some new ideas brings the paper [11] which gives another point of view to RF. RF can be seen as weighted PNN (potentially nearest neighbor) and this leads us very close to tree based method called k-NN (Nearest Neighbor) Tree, see [13]. For further work also our statistical model can be very useful to show relations between various randomization techniques.

References

[1] Bock R.K, Chilingarian A., Gaug M., Hakl F., Hengstebeck T., Jiřina M., Klaschka J., Kotrč E., Savický P., Towers S., Vaicilius A., Wittek W., “Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope”, *Nuclear Instr. Methods*, vol. 516, pp. 511–528, 2004.

[2] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., “Classification and regression trees”, *Belmont CA:Wadsworth*, 1984

[3] Breiman L., “Bagging predictors”, *Machine Learning*, vol. 24, pp. 123–140, 1996

[4] Breiman L., “Random forests”, *Machine Learning*, vol. 45, pp. 5–32, 2001

[5] Heck D. et al., “CORSICA, A Monte Carlo code to simulate extensive air showers”, *Forschungszentrum Karlsruhe FZKA 6019*, 1998

[6] Quinlan J.R., “Bagging, boosting and C4.5”, *Proceedings of the Thirteenth National Conference in Artificial Intelligence*, pp. 725–730, 1996

[7] Shapire R.E., Singer Y. “Improved boosting algorithms using confidence rated presictions”, *Machine Learning*, vol. 37, pp. 297–336, 1999

[8] Hastie T., Tibshirani R., Friedman J. H., “The Elements of Statistical learning”, *Springer-Verlag*, 2001

[9] Devroye L., Gyorfi L., Lugosi G., “A Probabilistic Theory of Pattern Recognition”, *Springer-Verlag*, 1996

[10] Quinlan J.R., “C4.5: Programs for Machine Learning: Book and Software Package ”, *Elsevier*, 1999

[11] Yi Lin, Yongho Jeon, “Random Forest and Adaptive Nearest Neighbors”, *Department of Statistics*, Technical Report No.1055, 2002

[12] Dietterich T.G., “An Experimental Comparison of Tree Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization”, *Machine Learning*, vol.40, pp. 139–157, 2000

[13] Buttrey S.E., Karo C., “Using k-nearest-neighbor Clasification in the Leaves of a Tree”, *Computational Statistics & Data Analysis*, vol.40, pp. 27–37, 2002

[14] Grandvalet Y., “Bagging Equalizes Influence”, *Machine Learning*, vol.55, pp. 251–270, 2004

[15] Kim H., Loh W., “Classification trees with unbiased multiway splits”, *Journal of the American Statistical Association*, vol. 96, p. 589-604, 2001

[16] Loh W.Y., Shih Y.S., “Split selection methods for classification trees”, *Statistica Sinica*, vol. 7, p. 815-840, 1997

[17] Savický P., Kotrč E., “Experimental study of leaf confidences for Random Forests”, *Proceedings of COMPSTAT*, 2004, in print

[18] R Development Core Team, “R: A language and environment for statistical computing”, *R Foundation for Statistical Computing*, www.r-project.org, 2004

Models of Multi-Agent Systems

Post-Graduate Student:
PAVEL KRUŠINA
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

krusina@cs.cas.cz

Supervisor:
MGR. ROMAN NERUDA, CSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

roman@cs.cas.cz

Field of Study:
Theoretical computer science
Classification: I-1

Abstract

Multi-agent systems typically utilize a non-blocking asynchronous communication in order to achieve required flexibility and adaptability. High performance computing techniques exploit the current hardware ability of overlapping asynchronous communication with computation to load the available computer resources efficiently. On the contrary, widely used parallel processes modeling methodologies do not often allow for an asynchronous communication description. At the same time those models do not allow their user to select the granularity level and provide only a fixed set of machine and algorithm description quantities. In this work[1] we addressed this issue and designed a new parallel processes modeling methodology. Its main features include an open set of atomic operations that are calculated and predicted for the algorithm in question, and the computer aided semi-automatic measuring of operation counts and approximation of cost functions. This allows not only for tuning the model granularity as well as accuracy according to user needs, but also to reach a such description complexity that would be very difficult to obtain without any computer aid. We demonstrated that our approach gives good results on the parallel implementation of a selected generalized genetic algorithm. A model was constructed and its predictions compared with the reality on various computer architectures, including one parallel cluster machine. We also designed and implemented an open multi-agent system suitable for the above mentioned experiments and many others. This system synthesizes the areas of high performance computing, multi-agent systems and computational intelligence into an efficient and flexible means of running experiments.

---

[1] The work was partially supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) “Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realisation”.

Kernel Based Regularization Networks and RBF Networks

Post-Graduate Student:
MGR. PETRA KUDOVÁ
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
petra@cs.cas.cz

Supervisor:
MGR. ROMAN NERUDA, CSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
roman@cs.cas.cz

Field of Study:
Theoretical computer science
Classification: I1

Abstract

We discuss two approaches to supervised learning, namely regularization networks and RBF networks, and demonstrate their performance on experiments, including benchmark, real and simulated data. We show that the performance of these two models is comparable, so the RBF networks can be used as an alternative to regularization networks. Their advantage is lower model complexity, while the regularization networks grows with the size of the training set.

1. Introduction

The problem of *learning from examples* (also called *supervised learning*) is a subject of great interest. Systems with the ability to autonomously learn a given task, would be very useful in many real life applications, namely those involving prediction, classification, control, etc.

The problem can be formulated as follows. We are given a set of examples $\{(\vec{x}_i, y_i) \in R^d \times R\}_{i=1}^m$ that was obtained by random sampling of some real function f, generally in the presence of noise. To this set we refer as *a training set*. Our goal is to recover the function f from data, or find the best estimate of it. It is not necessary that the function exactly interpolates all the given data points, but we need a function with good generalisation. That is a function that gives relevant outputs also for the data not included in the training set.

The learning problem can be handled by artificial neural networks. There is a good supply of network architectures and corresponding supervised learning algorithms (see [1]). In this case the model, that is a particular type of neural network, is chosen in advance and its parameters are tuned during learning so as to fit the given data.

The supervised learning of neural networks can be viewed as a function approximation problem. Given the data set, we are looking for the function that approximate the unknown function f. It is usually done by *Empirical Risk Minimization*, i.e. minimizing the functional $H[f] = \frac{1}{m}\sum_{i=1}^{N}(f(\vec{x}_i) - y_i)^2$ over a chosen *hypothesis space*.

Since this problem is ill-posed, we have to add some a priori knowledge about the function. We usually assume that the function is *smooth*, in the sense that two similar inputs corresponds to two similar outputs and the function does not oscillate too much. This is the main idea of the regularization theory, where the solution is found by minimizing the functional (1) containing both the data and smoothness information.

$$H[f] = \frac{1}{m}\sum_{i=1}^{N}(f(\vec{x}_i) - y_i)^2 + \gamma\Phi[f], \tag{1}$$

where Φ is called a *stabilizer* and $\gamma > 0$ is *the regularization parameter* controlling the trade off between the closeness to data and the smoothness of the solution. The regularization scheme (1) was first introduced by Tikhonov [2] and therefore it is called a Tikhonov regularization.

The regularization approach has good theoretical background, it was shown that for a wide class of stabilizers the solution has a form of feed-forward neural network with one hidden layer, called *regularization network*, and that different types of stabilizers lead to different types of regularization networks [3, 4].

However the theoretically proved existence and uniqueness of the solution does not necessarily mean that it is numerically feasible to find the solution. So it is also desirable to gain insight about practical applicability of the methods. This makes experimental evaluations very important.

In this work we discuss learning using the regularization network (RN) and RBF neural network. We compare their performance on experiments, including both benchmark, simulated and real learning tasks.

2. Approximation via regularization network

Poggio and Smale in [4] proposed a learning algorithm (Fig. 1) derived from the regularization scheme (1). They choose the hypothesis space as a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_K$ defined by an explicitly chosen, symmetric, positive-definite function $K_{\vec{x}}(\vec{x}') = K(\vec{x}, \vec{x}')$. As a stabilizer the norm of the function in $\mathcal{H}_K$ is taken. Having a training set $\{(\vec{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ we get

$$H[f] = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(\vec{x}_i))^2 + \gamma||f||_K^2. \tag{2}$$

The solution of (2) is unique and has the form

$$f(\vec{x}) = \sum_{i=1}^{m} c_i K_{\vec{x}_i}(\vec{x}) \qquad c_i = \frac{y_i - f(\vec{x}_i)}{m\gamma}. \tag{3}$$

The most commonly used kernel function is Gaussian $K(\vec{x}, \vec{x}') = e^{-\left(\frac{\|\vec{x}-\vec{x}'\|}{d}\right)^2}$.

The power of the algorithm (Fig. 1) is in its simplicity and effectiveness. On the other hand, it also has some drawbacks. First of all, the size of the model (that is a number of kernel functions) corresponds to the size of the training set and so the tasks with huge data sets lead to solutions of implausible size.

Then there are the parameters γ and d, which are supposed to be fixed. Let us describe how they influence the solution. Once they are fixed, the algorithm reduces to the problem of solving linear system of equations (4).

Since the system has m variables, m equations, K is positive-definite and $(m\gamma I + K)$ is strictly positive, it is well-posed. But we would also like it to be well-conditioned, i.e. insensitive to small perturbations of the data. In other words, we would like the condition number of the matrix $(m\gamma I + K)$ to be small, which is fulfilled if $m\gamma$ is large. Note that we are not entirely free to choose γ, because with too large γ we loose the closeness to data. See figure 6b.

The second parameter d determines the width of the Gaussians. Suppose that the distances between the data points are high or the widths are small, than the matrix K has 1s on diagonal and small numbers everywhere else and therefore is well-conditioned. If the widths are too large, all elements of the matrix K are close to 1 and its condition number tends to be high.

So the real performance of the algorithm depends significantly on the choice of parameters γ and d. Unfortunately the optimal choice depends on a particular data set.

Input: `Data set` $\{\vec{x}_i, y_i\}_{i=1}^m \subseteq X \times Y$ **Output:** `Function` f.

1. `Choose a symmetric, positive-definite function` $K_{\vec{x}}(\vec{x}')$, `continuous on` $X \times X$.
2. `Create` $f : X \to Y$ `as` $f(\vec{x}) = \sum_{i=1}^m c_i K_{\vec{x}_i}(\vec{x})$ `and compute` $\vec{c} = (c_1, \dots, c_m)$ `by solving`

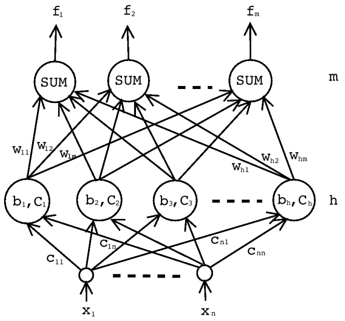
$$(m\gamma I + K)\vec{c} = \vec{y}, \tag{4}$$

   `where` I `is the identity matrix, K is the matrix` $K_{i,j} = K(\vec{x}_i, \vec{x}_j)$, `and` $\vec{y} = (y_1, \dots, y_m)$, $\gamma > 0$ `is real number.`

Figure 1: RN algorithm

3. RBF neural networks

An RBF neural network (RBF network) represents a relatively new model of neural network. On the contrary to classical models it is a network with local units which was motivated by the presence of many local response units in human brain. Other motivation came from numerical mathematics, radial basis functions (RBF) were first introduced in the solution of real multivariate problems [5].



$$y(\vec{x}) = \varphi\left(\frac{\| \vec{x} - \vec{c} \|_C}{b}\right) \tag{5}$$

$$f_s(\vec{x}) = \sum_{j=1}^{h} w_{js} \varphi\left(\frac{\| \vec{x} - \vec{c}_j \|_{C_j}}{b_j}\right) \tag{6}$$

Figure 2: a) RBF network architecture b) RBF network function

An RBF network is a standard feed-forward neural network with one hidden layer of RBF units and linear output layer (fig. 2). By an RBF unit we mean a neuron with n real inputs and one real output, realising a radial basis function (5), usually Gaussian. Instead of the Euclidean norm we use the *weighted norm* $|| \cdot ||_C$, where $\| \vec{x} \|_C^2 = (C\vec{x})^T(C\vec{x}) = \vec{x}^T C^T C\vec{x}$.

The network computes a function $\vec{f} = (f_1, \dots, f_m)$ as linear combination of outputs of the hidden layer (see (6)).

The goal of RBF network learning is to find the parameters (i.e. centers $\vec{c}$, widths b, norm matrices C and weights w) so as the network function approximates the function given by the training set $\{(\vec{x}_i, \vec{y}_i) \in R^n \times R^m\}_{i=1}^N$.

There is a variety of algorithms for RBF networks learning, in our past work we studied their behaviour and possibilities of their combinations [6, 7].

The two most significant algorithms,*Three step learning* and *Gradient learning*, are sketched in figure 3 and 4. See [6] for details.

```
Input: Data set $\{\vec{x}_i, \vec{y}_i\}_{i=1}^N$        Output: $\{\vec{c}_i, b_i, C_i, w_{ij}\}_{i=1..h}^{j=1..m}$
1. Set the centers $\vec{c}_i$ by a vector quantization algorithm.
2. Set the widths $b_i$ and matrices $C_i$.
3. Set the weights $w_{ij}$ by solving $\Phi W = D$.
   $D_{ij} = \sum_{t=1}^{N} \vec{y}_{tj} e^{-\left(\frac{\|\vec{x}_t - c_i\|_{C_i}}{b_i}\right)^2}, \Phi_{qr} = \sum_{t=1}^{N} e^{-\left(\frac{\|\vec{x}_t - c_q\|_{C_q}}{b_q}\right)^2} e^{-\left(\frac{\|\vec{x}_t - c_r\|_{C_r}}{b_r}\right)^2}$
```

Figure 3: Three step algorithm

```
Input: Data set $\{\vec{x}_i, \vec{y}_i\}_{i=1}^N$        Output: $\{\vec{c}_i, b_i, C_i, w_{ij}\}_{i=1..h}^{j=1..m}$
1. Put the small part of data aside as an evaluation set $ES$,
   keep the rest as a training set $TS$ .
2. $\forall j$ $\vec{c}_j(i) \leftarrow$ random sample from $TS_1$, $\forall j$ $b_j(i), \Sigma_j^{-1}(i) \leftarrow$ small
   random value, $i \leftarrow 0$
3. $\forall j, p(i)$ in $\vec{c}_j(i), b_j(i), \Sigma_j^{-1}(i)$:
   $\Delta p(i) \leftarrow -\epsilon \frac{\delta E_1}{\delta p} + \alpha \Delta p(i-1)$,        $p(i) \leftarrow p(i) + \Delta p(i)$
4. $E_1 \leftarrow \sum_{\vec{x} \in TS_1} (f(\vec{x}) - y_i)^2$, $E_2 \leftarrow \sum_{\vec{x} \in TS_2} (f(\vec{x}) - y_i)^2$
5. If $E_1$ and $E_2$ are decreasing, $i \leftarrow i + 1$, go to 3, else STOP. If $E_2$
   started to increase, STOP.
```

Figure 4: Gradient algorithm

4. Experiments

We tested the described methods on three experiments. All of them were run on nodes of Linux cluster with AMD Athlon(tm) XP 2100+ processors.

RN with Gaussian kernel function and RBF networks with Gaussian units were used. Linear systems were solved using the LAPACK library [8].

We always use two disjunct data sets, a training set for training of the network and a testing set for evaluating an error of results. In all experiments we use the normalized error (7):

$$E_{ts} = 100 \frac{1}{N} \sum_{i=1}^{N} ||\vec{y}_i - f(\vec{x}_i)||^2 \qquad \begin{array}{ll} N & \text{number of examples in } \{(\vec{x}_i, \vec{y}_i)_{i=1}^N\} \\ f & \text{network output} \end{array} \tag{7}$$

As an example of a practical task we have chosen the prediction of flow rate on the river Ploucnice. We have two data sets named *ploucnice1* and *ploucnice2*, for the prediction based on information from previous one and two days, respectively.

Table 2 shows the resulting error of a RN and an RBF network with 15 units. The parameters γ and d of the RN algorithm (Fig. 1) were estimated by cross-validation (the training set was divided to 10 parts, each run was one part put aside as an evaluation set). The RBF network was trained by the Three step algorithm (Fig. 3) and the computation was run 50 times. Mean errors and its standard deviation are listed. In figure 7 you can see the prediction by both the RN and the RBF network. Time of one run of the Three step algorithm (RBF) was approximately 28 seconds, one run of the RN algorithm lasted 55 seconds. Note that it is necessary to run the RN algorithm many times during cross-validation.

In order to compare an accuracy of RN and an RBF networks, we have selected three benchmark problems from the Proben1 database, the *Cancer* and the *Glass* classification tasks, and the *Heart* approximation problem. Moreover, each of the Proben1 data sets is available in three different ordering defining different data partitions for training and testing. These are referred to, eg. as glass1, glass2, and glass3 in the original report [9].

Table 1 and graphs in figure 5 compare the performance of RN, RBF and in addition MLP (multilayer perceptron) on Proben1 tasks. RBF networks were trained by the gradient algorithm (Fig. 4) (5000 iterations). The results for MLP are taken from [9]. We can see that the RN and the RBF network achieved almost the same rate of accuracy. The time requirements are following: 9 seconds (1 second, 22 seconds) for one run of the RN algorithm on *cancer* (*glass, hearta* respectively) data set, 100 iterations of the gradient algorithm took 14 seconds, 9 seconds, 74 seconds, respectively. Figure 6 shows the value of the resulting error with respect to γ and d.

For the last experiment we used simulated data sets. We have two data sets obtained by uniform sampling of the function $sin(6.28y) \cdot sin(6.28y)$ in the interval $(-1.0, 0) \times (-1.0, 0)$. The first *sin1* contains 121 samples, the second *sin2* 2500 samples. As the testing set we used uniform samples that did not coincide with the training samples.

The purpose was to show the relation between the optimal value of the parameter d of RN and the density of data points. The table 2 shows the values of parameters for which the error was minimal. The time needed for one computation was approximately 20 minutes 40 seconds for *sin1* and 1 second for *sin2*.

	RN			**RBF**			**MLP**		
	E_{ts}	d	γ	E_{ts}	std	arch	E_{ts}	std	arch
cancer1	1.60	1.0	0.0002	1.64	0.16	20	1.60	0.41	4+2
cancer2	2.99	1.4	0.0002	2.89	0.07	20	3.40	0.33	8+4
cancer3	2.76	1.3	0.0005	2.74	0.20	20	2.57	0.24	16+8
cancer	2.45			2.42			2.52		
glass1	6.75	0.3	0.0008	6.59	0.32	15	9.75	0.41	16+8
glass2	7.28	0.3	0.0014	7.85	0.43	15	10.27	0.40	16+8
glass3	6.48	0.2	0.0017	6.95	0.26	15	10.91	0.48	16+8
glass	6.84			7.13			10.31		
hearta1	4.44	1.9	0.0008	4.84	0.25	30	4.76	1.14	32+0
hearta2	4.32	1.9	0.0012	4.66	0.08	30	4.52	1.10	16+0
hearta3	4.45	1.9	0.0008	4.54	0.06	30	4.81	0.87	32+0
hearta	4.40			4.68			4.70		

Table 1: Comparison of Regularization Network (RN), RBF network (RBF) and multilayer perceptron (MLP).

		ploucnice1	ploucnice2
	E_{ts}	0.246	0.452
RBF	std	0.15	0.12
	h	15	15
	E_{ts}	0.056	0.121
RN	γ	1.48e-05	1.3e-05
	d	0.5	1.8

data	width	γ	error
sin1	0.1	1e-17	4.31e-12
sin2	0.6	1e-18	8.51e-11

Table 2: a) Comparison of the errors on the tasks ploucnice1 and ploucnice2. b) Winning values of parameters and error for RN on sin1 and sin2 data set.

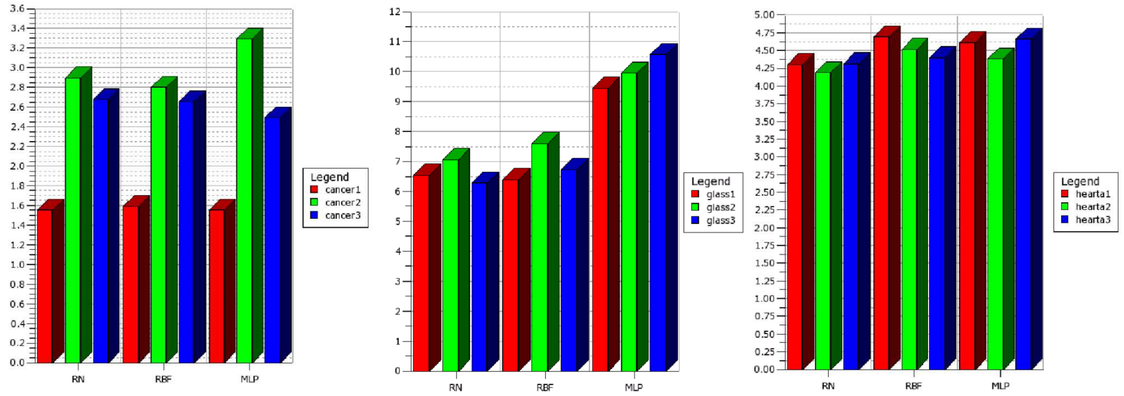


Figure 5: Comparison of RN, RBF, MLP: test set error.

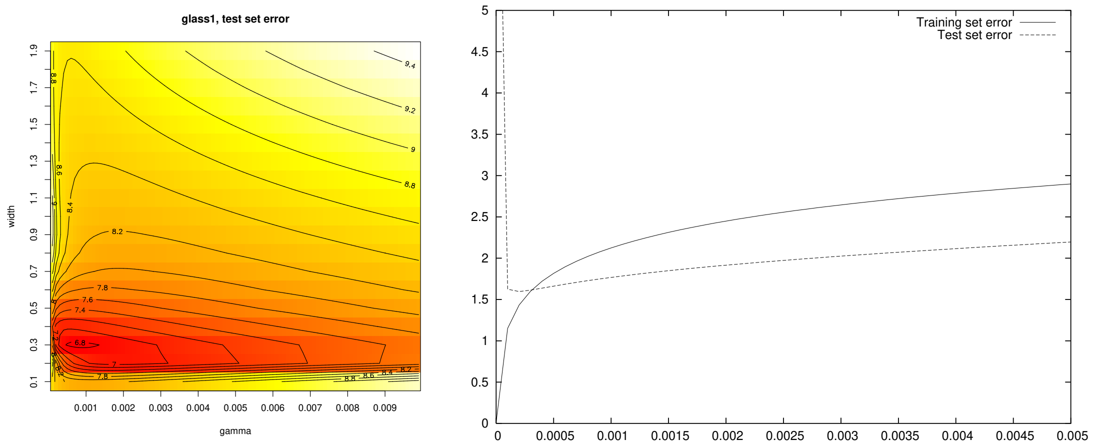


Figure 6: a) The dependence of the error function (computed on test set) on parameters γ a d. b) The relation between γ and training and testing error.

Task	Type	n	m	Train. set size	Test set size
Cancer	Class.	9	2	525	174
Glass	Class.	9	6	161	54
Heart	Approx.	35	1	690	230

Table 3: Overview of data sets from Proben1 database.

5. Conclusion

In this work we discussed two approaches to the learning task – regularization network and RBF network. We demonstrated their behaviour and performance on experiments, including benchmark, real and simulated data sets. We showed that the models are comparable, so the RBF network can be used as an alternative to RN in situations where the lower model complexity is desirable.

Both algorithms for RN and RBF networks suffer from the presence of extra parameters that have to be set explicitly. They are the regularization parameter γ and the width d in case of RN, and the number of hidden units h in case of RBF. However, while the estimation of parameters of RN by cross validation is quite time consuming, it is usually sufficient to try several values of h in the case of RBF network.

In our future work we will concentrate on the improvements of these algorithms, especially the ways of estimating the additional parameters.

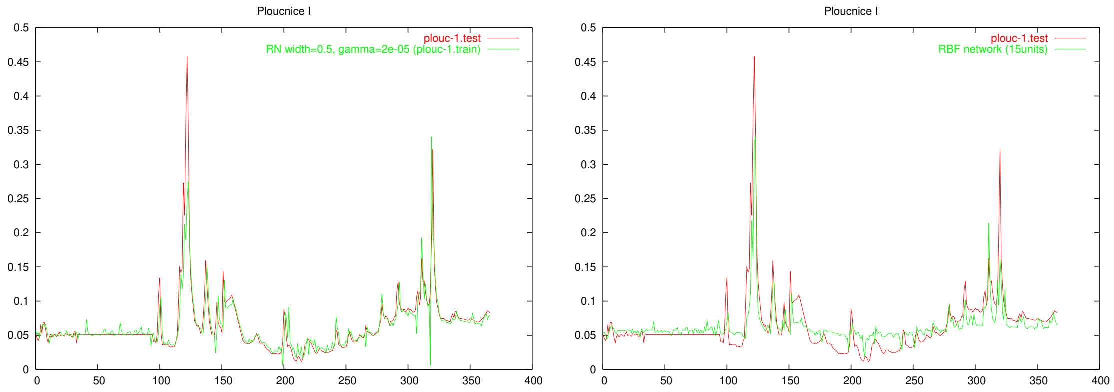


Figure 7: Prediction of the flow rate on the river Ploucnice: a) by RN b) by RBF

References

[1] S. Haykin, *Neural Networks: a comprehensive foundation*. Tom Robins, 2nd ed., 1999.

[2] A. Tikhonov, V. Arsenin, *Solutions of Ill-posed Problems.* W.H. Winston, Washington, D.C, 1977.

[3] F. Girosi, M. Jones and T. Poggio, "Regularization theory and Neural Networks architectures," *Neural Computation*, vol. 7,No.2, pp. 219–269, 1995.

[4] T. Poggio, S. Smale, "The mathematics of learning: Dealing with data," *Notices of the AMS*, vol. 50, No.5, pp. 537–544, 2003.

[5] M. Powel, "Radial basis functions for multivariable interpolation: A review," in *IMA Conference on Algorithms for the Approximation of Functions and Data*, (RMCS, Shrivenham, England), pp. 143–167, 1985.

[6] R. Neruda, P. Kudová, "Hybrid learning of RBF networks," *Neural Networks World*, vol. 12, no. 6, pp. 573–585, 2002.

[7] R. Neruda, P. Kudová, "Learning methods for RBF neural networks," *Future Generations of Computer Systems*, 2004. In press.

[8] "Lapack library." `http://www.netlib.org/lapack/`.

[9] L. Prechelt, "Proben1 – a set of benchmarks and benchmarking rules for neural network training algorithms," Tech. Rep. 21/94, Universitaet Karlsruhe, 9 1994.

Integrace dat a sémantický web

doktorand:
ING. ZDEŇKA LINKOVÁ

Katedra matematiky
Fakulta jaderného a fyzikálního inženýrství
ČVUT

Praha

zdena@centrum.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.

Ústav informatiky AV ČR
Pod Vodárenskou věží 2

182 07 Praha 8

stuller@cs.cas.cz

obor studia:
Matematické inženýrství
číselné označení: 39-10-9

Abstrakt

World Wide Web obsahuje data, která jsou pro počítačové programy nesrozumitelná. Následkem toho je na něm obtížné některé věci zautomatizovat. Nedostatky současného webu by měl odstranit sémantický web, ve kterém data budou mít přesně popsaný význam. Zlepšení může přinést také v oblasti integrace, která je v případě dat pocházejících z webu velmi obtížná. Tento článek[1] se zabývá integrací webových dat. Zaměřuje se na relační data ve formátu XML a navrhuje postupy základních integračních operací.

1. Úvod

World Wide Web (WWW) obsahuje obrovské množství informací vytvořených mnoha různými organizacemi, společnostmi i jednotlivci z mnoha různých důvodů. Současný web je však více přátelský k lidskému uživateli než k počítačovým programům. Vše na něm je pro počítačové aplikace sice čitelné, avšak nesrozumitelné. Následkem velkého množství různorodých dat je obtížné udržování, aktualizace a vyhledávání informací. To ztěžuje snahu věci na webu zautomatizovat. Přitom je hodně způsobů, jak by programy mohly obsahu webu využívat, jen kdyby mu rozuměly.

Web může plně dosáhnout svých možností pouze tehdy, jestliže se stane místem, kde mohou být data sdílena a zpracovávána automatizovanými nástroji stejně jako lidmi. Toho chce dosáhnout sémantický web. Je založen na myšlence mít data na webu nejen uložena, ale také definována a spojena takovým způsobem, aby mohla být programy užívána nejen k zobrazení, ale též pro navigaci, integraci, automatizaci a opakované použití napříč různými aplikacemi.

2. Sémantický web

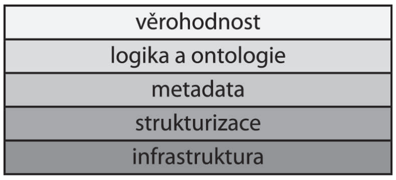
Sémantický web [1], [2] je zamýšlen jako rozšíření současného World Wide Webu sestávající z počítačově čitelných, srozumitelných a smysluplně zpracovávatelných dat. Základem je zavedení sémantiky – spolu s uloženými daty bude k dispozici i jejich popis. To znamená, že data budou mít definovány svůj význam.

Ačkoli v současnosti již některé črty existují, je sémantický web zatím pouze vize. Je to cíl, ke kterému se směřuje. Založen by měl být na standardech, které jsou průběžně definovány. O definici těchto standardů usiluje W3C (WWW Consorcium [3]). Ovšem stejně jako jiné oblasti výpočetní techniky se i tato neustále vyvíjí a dle potřeb mohou vznikat standardy nové. Toto pojetí rozšíření WWW je tedy ve fázi neustálého vývoje.

---

[1] Práce byla částečně podpořena projektem 1ET100300419 programu Informační společnost (Tématického programu II Národního programu výzkumu v ČR): Inteligentní modely, algoritmy, metody a nástroje pro vytváření sémantického webu.

Principy sémantického webu jsou implementovány v jednotlivých vrstvách webových technologií a standardů [4]. Vrstvy jsou zobrazeny na následujícím obrázku.



Obrázek 1: Vrstvy sémantického webu

Vrstva infrastruktury poskytuje možnost identifikace, lokalizace a transformace zdrojů. *Vrstva strukturizace*, *vrstva metadat* a *vrstva logiky a ontologie* jsou nezbytné k vyjádření obsahu webu, který ze zdrojů získáváme. *Vrstva věrohodnosti* je již záležitostí konkrétních aplikací. Týká se ověřování a důvěryhodnosti získané informace - ne vše umístěné na webu totiž musí být pravdivé. Aplikace musí rozhodnout, zda se na informaci spolehne, na základě nějakého podaného důkazu věrohodnosti zdroje.

2.1. Infrastruktura

Sémantický web bude sestávat z propojených zdrojů – bude obsahovat zdroje a odkazy mezi nimi. Objekt bude možné identifikovat (stejně jako na současném webu) užíváním identifikátorů: přímý odkaz vznikne vytvořením a přiřazení URI (Universal Resource Identifier) danému objektu. Sémantický web bude také samozřejmě decentralizovaný, jak jen to bude možné. Decentralizace však vyžaduje kompromisy, třeba v nutnosti tolerovat neúplnou či chybějící informaci v podobě odkazu na neexistující zdroj.

Sémantický web bude obsahovat nejen klasické (mediální) zdroje (stránky, texty, obrázky, audio klipy), ale mnohem více – bude obsahovat zdroje představující lidi, místa, organizace a události. Navíc přinese také možnost specifikace typů zdrojů i typů odkazů. Bude obsahovat mnoho různých druhů vztahů mezi různými typy zdrojů. Díky tomu budou moci aplikace zjistit druh vztahu mezi daty.

2.2. Vyjádření datového obsahu

Důležitým požadavkem počítačově zpracovatelné informace je *strukturování* dat. Na webu je hlavní strukturovací technikou značkování dokumentů pomocí tzv. tagů, což je určitá část textu, která obsahuje informace udávající role a vlastnosti obsahu dokumentu. V současné době je standardním mechanizmem ke strukturování dat jazyk XML (eXtensible Markup Language) [5]. Tento jazyk poskytuje datový formát pro strukturované dokumenty a umožňuje běžnou syntaxi pro počítačově čitelná data.

Samo XML ovšem k popisu dat nestačí. Pomocí tagů lze vytvářet strukturu, ale jejich použití neříká nic o tom, co daná struktura znamená. Technologií k určení druhu a významu informace je základ pro zpracování *metadat* – RDF (Resource Description Framework) [6], který je mechanismem, jak říci něco o datech. Představuje jednoduchý mechanizmus reprezentace znalostí pro webové zdroje. Datový model RDF poskytuje abstraktní, konceptuální rámec pro definici a použití metadat. Pro účely vytváření a výměny těchto metadat je však třeba konkrétní syntaxe. RDF k tomuto účelu používá kódování pomocí XML [7].

Prostředkem k definici termínů použitých k vyjádření metadat jsou ontologie, které tak zajišťují prostředek pro sdílení termínů při spolupráci aplikací. Myšlenka sémantického webu též zahrnuje přidání logiky na web - ve smyslu používání pravidel k tvoření závěrů a podobně. Ontologie [8] označuje ucelenou kolekci termínů, vztahů a vyvozovacích pravidel. V kontextu webových technologií je ontologií dokument nebo soubor, který formálně definuje vztahy mezi termíny. Slovník ontologie lze chápat jako jakýsi výkladový slovník pojmů. Pomocí odvozovacích pravidel můžeme nad pojmy činit různé závěry a slovník se tak může dále vyvíjet.

2.3. Provozování aplikací

Skutečná síla sémantického webu se projeví, jestliže lidé vytvoří mnoho programů, které budou shromažďovat webový obsah z různých zdrojů, zpracovávat informace a vyměňovat si výsledky s ostatními programy. Efektivita takovýchto tzv. *softwarových agentů* [9] se zvýší tím více, čím více bude obsah webu srozumitelnější pro počítače a čím přístupnější budou automatizované služby (zahrnující ostatní agenty).

Sémantický web bude moci poskytovat základ a strukturu k realizovatelnosti dalších technologií – kromě něj budou někteří agenti využívat umělé inteligence, například při automatickém vytváření složitých kolekcí hodnot, kdy se na výsledku podílí celý soubor specializovaných agentů.

3. Integrace dat pocházejících z webu

I když lze z webu získat mnoho informací, nejsou všechny informace poskytovány jediným zdrojem – jsou roztroušeny. K uspokojení konkrétního požadavku je často třeba pracovat s daty z více zdrojů, které jednotlivé dílčí části nabízejí. Výsledkem je pak ovšem více oddělených částí a nikoli požadovaná kompletní informace. Data je proto potřeba integrovat, tzn. z několika původních zdrojových informací vytvořit jediný informační zdroj, ať už materializovaný, či virtuální.

Integrace informací ze současného WWW je však velice obtížná. Na webu vznikají jakési ostrovy souvisejících dat, každý pochází od jiného poskytovatele. Poskytovatelé publikují data nezávisle, což vede k odlišnému užívání termínů a k používání odlišných nebo dokonce žádných schémat. Jednou z motivací tvorby sémantického webu je i usnadnění takové operace, jako je integrace.

3.1. Relační databáze a XML

V rámci diplomové práce[10] jsem pracovala na vývoji systému integrujícího zadané zdroje z webu. Integrační postup je založen na vstupním formátu XML, který se na webu stává de facto standardem. Formát zpracovávaných dat byl dále omezen na strukturu, kterou lze reprezentovat jako tabulku v relační databázi (viz následující obrázek). Jedním z důvodů je fakt, že integrace je dlouho uznávaným problémem databázových systémů a existují nástroje pro integraci databázových tabulek, je tedy možné oba přístupy porovnávat.

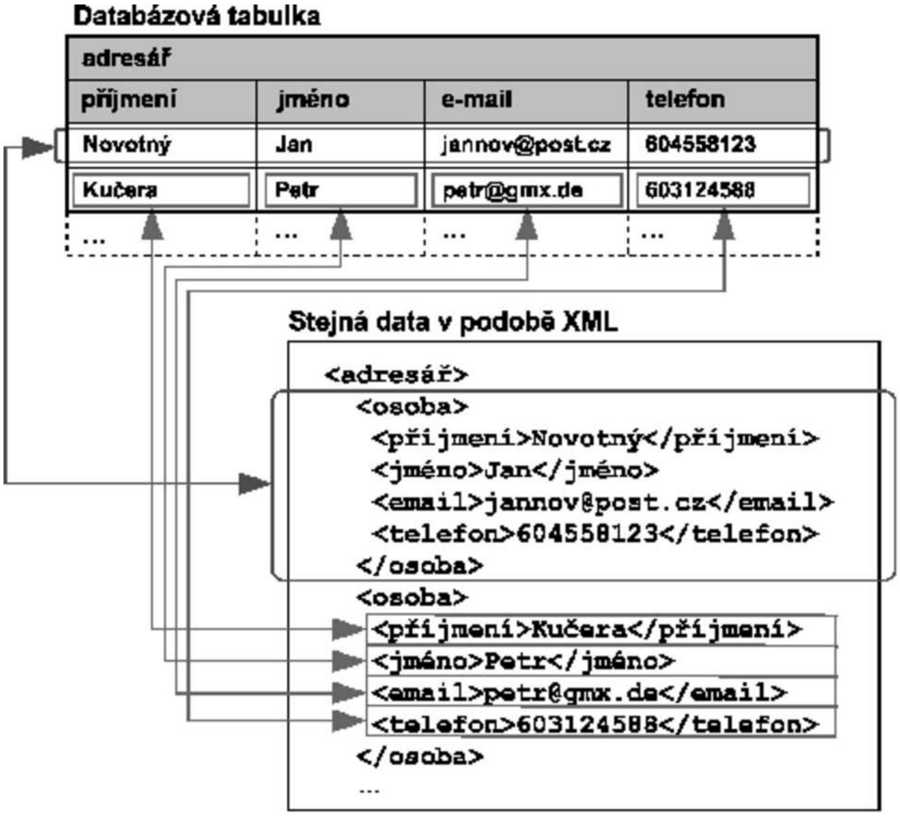
3.2. Model integrace

Při návrhu integračního systému byl použit přístup pomocí modelu stromu. Hierarchii struktury vnořených elementů v XML dokumentu lze skutečně nahlížet jako stromovou strukturu. Ze stromové struktury vychází také DOM – standardní API pro přístup k obsahu dokumentu XML.

DOM (Document Object Model) [11] je objektový model dokumentu. Dokument je prezentován jako stromová hierarchická struktura. Každému elementu (textový element, komentář, instrukce pro zpracování atd.) odpovídá jeden uzel stromu. DOM tvoří mnoho rozhraní obsahujících funkce, které umožňují celý strom dokumentu procházet a modifikovat jednotlivé uzly.

V návrhu je také použito označení inspirované názvoslovím modelu stromu:

- Zdroje jsou očíslovány a označeny: *zdroj1* a *zdroj2*.
- Uzel, který analogicky odpovídá řádku tabulky, je označen *řádek*.
  Uzel, který analogicky odpovídá sloupci tabulky, je označen *sloupec*.
  Jméno uzlu označuje *jméno*.
- Počet uzlů *řádek* ve *zdroj1* označuje *počet_řádků1* a *počet_řádků2* označuje počet uzlů *řádek* ve *zdroj2*.
  Počet uzlů *sloupec* ve *zdroj1* označuje *počet_sloupců1* a *počet_sloupců2* označuje počet uzlů *sloupec ve zdroj2*.
  Jednotlivé uzly popořadě jsou očíslovány a je použito označení *řádek(1), řádek(2),* . . . , resp. *sloupec(1), sloupec(2),* . . . .
- Uzel, který je následovníkem uzlu *sloupec* (je tedy typu text a obsahuje hodnotu), označuje *text*.



Obrázek 2: Formát XML a databázová data

- V přístupu je použita tečková notace.
- Je-li dán jakýkoli konkrétní uzel, je tím míněn uzel včetně svých potomků, tedy celý podstrom.

3.3. Postup integrace

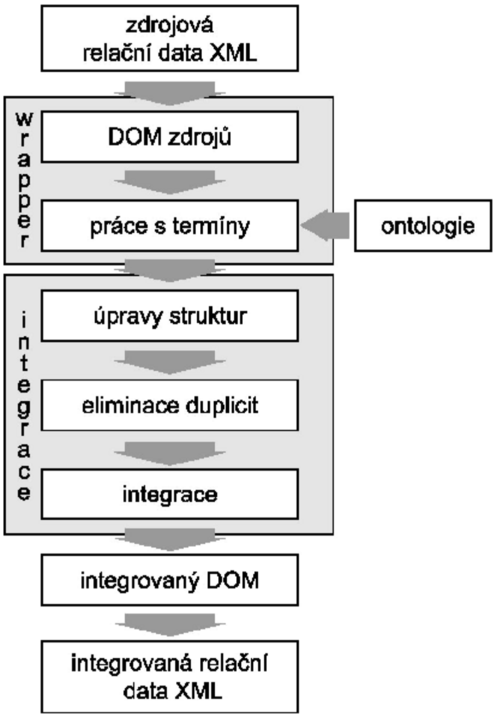
Schéma architektury navrhovaného systému a postupu úpravy a integrace datových zdrojů je patrné z obrázku na následující straně. Systém by měl mít dvě hlavní navazující části vykonávající jednak přípravu dat (moduly wrapperů [12] pro každý zdroj), jednak vlastní integraci. Příprava dat zahrnuje převod zdrojů na DOM reprezentaci a práci s použitými termíny.

Mnoho aplikací vychází z předpokladu, že rozdílná jména označují rozdílné věci. Na webu ovšem takový předpoklad učinit nelze. Je totiž například možné na jeden datový zdroj odkazovat několika různými způsoby. V některých případech je možné mezi rozdílnými pojmy použitými v datových zdrojích najít určitý vztah, dokonce i ekvivalenci. Používání ekvivalentních pojmů jako pojmů, které spolu nesouvisí, přitom vede v operaci integrace ke špatným výsledkům. Je proto výhodné před vlastní integrací s termíny pracovat.

Celý postup vlastní integrace vychází z integrace tzv. základní situace (viz dále). Její použití je z důvodů požadavků na vstupní data dosti omezeno. Proto jsou nejprve se zdroji provedeny některé úpravy: změna pořadí, odstranění či doplnění jednotlivých částí struktur. Aby výsledek integrace neobsahoval duplicitní informace, je nutné výskyt duplicit ošetřit.

Jednotlivé kroky následujícího "základního" algoritmu integrace budou podrobněji rozvedeny v dalších odstavcích 3.4 - 3.7.

Algoritmus integrace:



Obrázek 3: Postup při integraci dat

```
Uprav {Aplikace zvoleného postupu úpravy struktur}
Seřaď {Změna pořadí ve strukturách}
Odstraň_duplicity {Eliminace duplicit}
Integruj {Integrace základní situace}
```

3.4. Integrace základní situace

Situace, kdy jsou schémata integrovaných XML dokumentů naprosto stejná, je považována za základní. Při analogii k databázové tabulce budou mají zdroje identické databázové schéma, tj. stejné sloupce – stejný počet sloupců, jejich názvy a pořadí. Integrační operací je sjednocení obou zdrojů. Celý obsah elementu dokumentu z prvního i z druhého zdroje je sloučen do jediného elementu dokumentu výsledku. Toto sjednocení lze přirovnat k operaci sjednocení dvou databázových tabulek.

Algoritmus integrace základní situace:

```
Vytvoř kořen_výsledku
for i=1,2,...,počet_řádků1
  Zkopíruj zdroj1.řádek(i) do výsledku a Napoj na kořen
for j=1,2,...,počet_řádků2
  Zkopíruj zdroj2.řádek(j) do výsledku a Napoj na kořen
```

3.5. Eliminace duplicit

Jestliže by bylo možné nějakou informaci získat z obou zdrojů, ve výsledku by se objevila duplicitně. To by nebyl dobrý výsledek integrace, a proto je nutné potencionální výskyt duplicit řešit. V návrhu je výskyt násobných informací eliminován ještě před integrační operací.

Nejprve je nutné zjistit, které informace vedou na nežádoucí redundanci. Jeden ze zdrojů je označen jako referenční. Pro každou informaci z druhého zdroje je pak nutné ověřit, jestli se nedá získat již ze zdroje prvního – snahou vyhledat ji v referenčním zdroji. Skončí-li hledání úspěšně, násobná data budou odstraněna, a to z druhého (nereferenčního) zdroje.

V následujícím algoritmu je jako referenční zdroj *zdroj1*, duplicity jsou tedy odstraňovány ze *zdroj2*. Použitá metoda `Equals` je dvouhodnotová a porovnává dva stromy, zda jsou shodné.

Algoritmus odstraňování duplicit:

```
for i=1,2,...,počet_řádků1
    for j=1,2,...,počet_řádků2
        if zdroj1.řádek(i) Equals zdroj2.řádek(j)
             then Odstraň zdroj2.řádek(j)
```

3.6. Změna pořadí ve strukturách

Základní situaci integrace je možné aplikovat i na zdroje, jejichž struktury jsou shodné až na pořadí uzlů analogickým ke sloupcům tabulky. Příslušné pozice uzlů je pouze nutné změnit a vhodně seřadit. První zdroj je označen za referenční a dále se vychází z jeho struktury. Ve druhém zdroji je (podle referenčního) pořadí uzlů upraveno.

Algoritmus řazení sloupců:

```
for i=1,2,...,počet_sloupců1
  if zdroj1.řádek(1).sloupec(i).jméno
      $\neq$ zdroj2.řádek(1).sloupec(i).jméno
    then begin
      j:= Najdi_pozici zdroj1.řádek(1).sloupec(i) ve
                                               zdroj2.řádek(1)
      for k=1,2,...,počet_sloupců
           Odpoj zdroj2.řádek(k).sloupec(j)
           a Napoj ho na pozici i
    end
```

3.7. Úpravy struktur

Integraci základní situace není možné ihned použít v případě rozdílných struktur zdrojových dat. Jednoduchým případem neodpovídajících si schémat relačních XML zdrojů je, liší-li se struktury pouze v pořadí uzlů. Nabízí se pořadí uzlů vhodně upravit, to bylo ošetřeno v předchozí části. Spočívá-li ovšem rozdíl struktur zdrojů v odlišných sadách uzlů odpovídajících sloupcům tabulky, je třeba větších úprav. Volba vhodné operace závisí na okolnostech, na datech, která chceme integrovat, a na jejich významu.

Nejjednodušší možností je úprava struktur tak, že do výsledku budou zahrnuty pouze ty uzly sloupců, které se vyskytují ve všech zdrojích zároveň. Integrace proběhne přes průnik uzlů. Jestliže bude každý ze zdrojů nejprve upraven tak, že v něm budou nadbytečné sloupce odstraněny, všechny zdroje získají stejnou strukturu, případně bude třeba upravit pořadí. Pak bude moci být aplikován základní algoritmus integrace.

Algoritmus úpravy struktur na průnik:

```
A:= průnik sloupců
for i=1,2,...,počet_řádků1
     for j=1,2,...,počet_sloupců1
     if zdroj1.řádek(i).sloupec(j) not in A
          then Odstraň zdroj1.řádek(i).sloupec(j)
for i=1,2,...,počet_řádků2
     for j=1,2,...,počet_sloupců2
     if zdroj2.řádek(i).sloupec(j) not in A
          then Odstraň zdroj2.řádek(i).sloupec(j)
```

Obohacení struktury dat bez další kombinace datového obsahu je další možností, jak zpracovat zdroje rozdílných struktur. Výsledek bude mít strukturu sestávající ze všech uzlů sloupců jež do integračního procesu vstoupily, bez ohledu na původní zdroj. Každý zdroj bude obohacen o příslušný nový sloupec, jehož textová hodnota ale nebude z ostatních dat nijak odvozena. Nové textové hodnoty zůstanou buďto prázdné, nebo bude vložena speciální hodnota značící nezadání (např. hodnota NULL).

Algoritmus obohacení struktury:

```
B:=sjednocení sloupců
∀ prvek z B
     for i=1,2,..., počet_řádků1
          if prvek not in zdroj1.řádek(i)
               Vytvoř zdroj1.řádek(i).prvek
               zdroj1.řádek(i).prvek.text:='NULL'
     for i=1,2,...,počet_řádků2
          if prvek not in zdroj2.řádek(i)
               Vytvoř zdroj2.řádek(i).prvek
               zdroj2.řádek(i).prvek.text:='NULL'
```

Při obohacení struktury s kombinací dat je obohacena struktura zdrojů o nové sloupce stejným způsobem, jako v předchozím případě. S datovým obsahem se ovšem dále pracuje a je snaha data vhodně zkombinovat. K určení, jak data vzájemně souvisejí je možné využít průnik struktur zdrojů. Související data mají v takto určených uzlech stejné hodnoty. Analogii operace v takové situaci, kdy je nakombinována jak struktura, tak samotná data, lze ve světě databází spatřovat v operaci JOIN, tj. ve spojení dvou tabulek.

K vytvoření kombinace jsou využity kopie příslušných podstromů, vlastní původní podstrom zůstane beze změny – tak ho lze využít k dalším případným kombinacím. Nakonec jsou všechny původní podstromy, které vedly ke vzniku kombinace odstraněny, neboť jsou nová data obsažena v kopiích. Jak je ovšem naloženo s podstromy, které nebylo možné s ničím skombinovat, záleží na tom, co by měl výsledek obsahovat. Do výsledku je možné zahrnout pouze data, která vznikla kombinací obsahů zdrojů. Takovýto postup vede na výsledek integrace plně odpovídající databázové operaci spojení tabulek (inner join). Při takové situaci jsou ovšem ztracena data, která v druhém zdroji neměla odpovídající doplnění. Je-li požadováno všechna data zachovat, lze obohatit data, která obohatit lze, a ve zbylých případech doplnit strukturu o sloupce s nezadanými hodnotami. Tato operace a následná integrace je analogická databázovému vnějšímu spojení (outer join).

Algoritmus obohacení struktury a kombinace dat:

```
A := průnik sloupců
B := sjednocení sloupců
used1(1,2,...,počet_řádků1):=false
used2(1,2,...,počet_řádků2):=false
for i=1,2,...,počet_řádků1
   for j=1,2,...,počet_řádků2
      if ∀ prvek z A
           zdroj1.řádek(i).prvek.text =
                            zdroj2.řádek(j).prvek.text
                then souvisí:=true
                else souvisí:=false
      if souvisí = true then
           new:=Zkopíruj zdroj1.řádek(i) do zdroj1
           ∀ prvek z B
                if prvek not in new then
                         Zkopíruj zdroj2.řádek(j).prvek do new
```

```
       new:=Zkopíruj zdroj2.řádek(j) do zdroj2
       ∀ prvek z B
            if prvek not in new then
                 Zkopíruj zdroj1.řádek(i).prvek do new
for i=1,2,...,počet_řádků1
     if used1(i) then Odstraň zdroj1.řádek(i)
     else
            if inner_join
                 Odstraň zdroj1.řádek(i)
            if outer_join
                 ∀ prvek z B
                      if prvek not in zdroj1.řádek(i)
                        Vytvoř zdroj1.řádek(i).prvek
                        zdroj1.řádek(i).prvek.text:='NULL'
for i=1,2,...,počet_řádků2
     if used2(i) then Odstraň zdroj2.řádek(i)
     else
            if inner_join
                 Odstraň zdroj2.řádek(i)
            if outer_join
                 ∀ prvek z B
                      if prvek not in zdroj2.řádek(i)
                        Vytvoř zdroj2.řádek(i).prvek
                        zdroj2.řádek(i).prvek.text:='NULL'
```

4. Závěr

Integrace obecných dat pocházejících z webu je obtížná. Předpoklad XML formátu a relační struktury zdrojových dat však umožnil provést několik druhů integračních operací, z nich některé lze srovnávat s obdobnými operacemi prováděnými v oblasti relačních databází. Nicméně problematika je značně rozsáhlá - prezentovaný návrh pokrývá pouze část celého problému. Rozšiřování zpracovaného tématu je možné například v dalším využití existujících technik sémantického webu. Proto bych v tomto směru ve své práci ráda pokračovala.

Literatura

[1] T. Berners-Lee, J. Hendler and O. Lassila, “The Semantic Web”, *Scientific American*, vol. 284, 5, pp. 35–43, 2001.

[2] J. Euzenat, “Research Challenges and Perspectives of the Semantic Web”, *Report of the EU-NSF Strategic Research Workshop*, Sophia-Antipolis, Francie, říjen, 2001.

[3] W3C (WWW Consorcium). `http://www.w3.org`.

[4] W3C: Semantic Web. `http://www.w3.org/2001/sw/`.

[5] N. Bradley, “XML kompletní průvodce”, *Grada Publishing*, Praha, 2000, ISBN 80-7169-949-7.

[6] O. Lassila, R.R. Swick, “Resource Description Framework (RDF) Model and Syntax Specification”, *W3C Recommendation*, únor, 1999, `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.

[7] D. Beckett, “RDF/XML Syntax Specification (Revised)”, *W3C Working Draft*, leden 2003, `http://www.w3.org/TR/2003/WD-rdf-syntax-grammar-20030123`.

[8] Ch. Welty, N. Guarino, “Supporting Ontological Analysis of Taxonomic Relationships”, *Data & Knowledge Engineering*, vol. 39, pp. 51–74, 2001.

[9] J.H. Park, S.Ch. Park, "Agent-Based Merchandise Management in Business-to-Business Electronic Commerce", *Decision Support Systems*, vol. 35, pp. 311–333, 2003.

[10] Z. Linková, "Integrace dat a sémantický web", *Diplomová práce, FJFI ČVUT*, Praha, 2004.

[11] Document Object Model (DOM). `http://www.w3.org/DOM`.

[12] J.D. Ullman, "Information integration using logical views", *Theoretical Computer Science* vol. 239, pp. 189–210, 2000.

Relational Databases with Ordered Relations

Post-Graduate Student:
ING. RADIM NEDBAL

Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

radned@seznam.cz

Supervisor:
ING. JÚLIUS ŠTULLER, CSC.

Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

stuller@cs.cas.cz

Field of Study:
Mathematical engineering
Classification: 39-10-9

Abstract

This paper[1] describes an option to express our preferences in the framework of relational databases. Preferences have usually a form of a partial ordering. Therefore the question is how to deliver the semantics of ordering to a database system. The answer is quite straightforward.

1. Introduction

When retrieving data, it is difficult for a user of a classical relational database to express various levels of preferences.

Example 1 (Preferences represented by an ordering) *How could we express our intention to find an employee with a good command of English, or at least a good command of German, or at worst a good command of Russian? At the same time, we may want the employee to belong to the salesmen department or with higher preference to the management department. To sum up, we have the following preferences:*

A language:

1. *English,*
2. *German,*
3. *Russian,*

B department:

1. *management department,*
2. *salesman department,*

which can be formalized by an ordering, in general case by a partial ordering.

NAME	*LANGUAGE*	*DEPARTMENT*
Petr	*English*	*management*
Patrik	*German*	*management*
Pavel	*Russian*	*salesmen*
Dan	*Czech*	*clerk*
Robert	*English*	*president*
Martin	*German*	*management*
Marek	*Hungarian*	*clerk*

---

[1] The work was partially supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization"

We can see that Petr is preferred to Pavel for instance. However, what we can say about Robert for example? In this case, we need cartesian product operating on ordered relations.

The aim of this paper is to incorporate semantics of partial ordering into all the primitive operations, i.e. those that can not be expressed by means of others operations, of relational data model. The resulting data model should be capable of providing users with the most, according to their preferences, relevant data.

2. Relational data model

The relational data model is based on the term of relation. A table of a relational database corresponds to a relation and a row of that table is an element of the relation. However, the relational data model consists not only of the relations themselves, but it contains also operations on relations.

As a relation is a set, we have all the **set operations** plus **aggregation functions**, which are unary operations on sets returning a number, plus **arithmetic** for performing all the usual operations on numbers. As for the relational data model, Codd has introduced eight relational algebra operations:

1. Cartesian product $\times$,
2. Union $\cup$,
3. Intersection $\cap$,
4. Difference $\setminus$,
5. Restriction,
6. Projection,
7. Join,
8. Divide

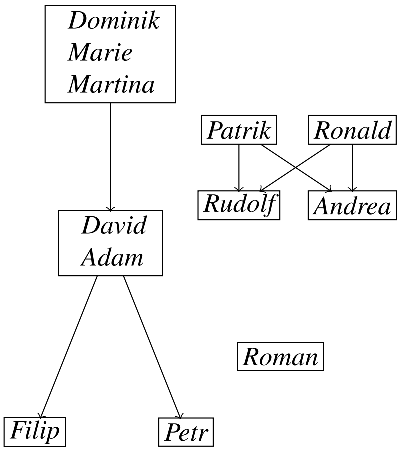
These operations are, however, not primitive [1] – they can be defined in terms of the others. In fact, of the set of eight, three (join, intersection and divide) can be defined in terms of the other five. Those other five operations (restriction, projection, cartesian product, union, and difference), by contrast, can be regarded as primitive, in the sense that none of them can be defined in terms of the other four. Thus, a minimal set of operations would be the set consisting of the five primitives – the *minimal set of relational algebra operations*.

3. Operations on Ordered Relations

Example 2 (Preferences of employees based on attribute values)

Relation scheme:
$R(\underline{NAME}, POSITION, LANGUAGE)$

Dominik	*President*	*English*
Marie	*Manager*	*English*
David	*Manager*	*German*
Petr	*Manager*	*Swedish*
Adam	*Manager*	*German*
Filip	*Programmer*	*Dutch*
Martina	*Programmer*	*English*
Patrik	*Programmer*	*French*
Rudolf	*Programmer*	*Italian*
Ronald	*Programmer*	*Spanish*
Andrea	*Programmer*	*Portuguese*
Roman	*Programmer*	*Russian*



We prefer employees speaking English to those speaking German, and at the same time we prefer German speaking employees to those who speak other germanic language. Similarly, we prefer Spanish and French to any other romanic language. We have no other preference. [2]

---

[2] The partial ordering is depicted using the standard Hasse diagram notation.

The ordering represents an extra information. To handle this information, we need appropriate operations. To maintain the same expressive power, we need operations corresponding to those that we have for the traditional relational model. In the following, we consider an ordered pair

$$[R, \leq^R],$$

of a relation R with its preference relation $\leq^R$.

3.1. Relational algebra operations

Restriction $R(\phi)$ returns a relation consisting of the set $\{r \in R | \phi(r)\}$ all tuples from a specified relation R that satisfy a special condition ϕ.

In the case of ordered relation $[R, \leq^R]$, we define:

$$[R; \leq^R](\phi) = [R(\phi); \leq^R_{R(\phi)}],$$

where

$$\leq^R_{R(\phi)} = R(\phi) \times R(\phi) \cap \leq^R$$

Projection $R[C]$ returns a relation consisting of all tuples that remain as (sub)tuples in a specified relation R after specified attributes have been eliminated.

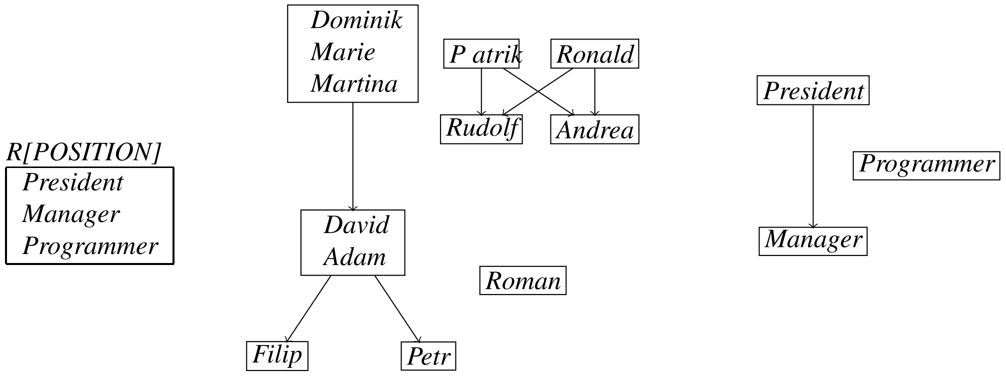
In the case of ordered relation $[R, \leq^R]$, we define:

$$[R; \leq^R][C] = [R[C]; \leq^{R[C]}],$$

where

$$\leq^{R[C]} = \{(p_i, p_j) |$$
$$\exists r_i, r_j \in R(r_i[C] = p_i \ \wedge \ r_j[C] = p_j) \ \wedge \ \forall r_i, r_j \in R(r_i[C] = p_i \ \wedge \ r_j[C] = p_j \Rightarrow r_i \leq^R r_j)\}$$

Example 3 (Ordering on a projection)



We prefer president to manager as all the presidents, which is in this case the only element, are preferred to all the managers in the input ordering. At the same time, we can say nothing about preferences of programmer and manager for instance as we can find incomparable couples of programmers and managers or those with contradictory preferences in the input relation.

Union $R_1 \cup R_2$ returns a relation consisting of all tuples appearing in either or both of two specified relations R_1, R_2.

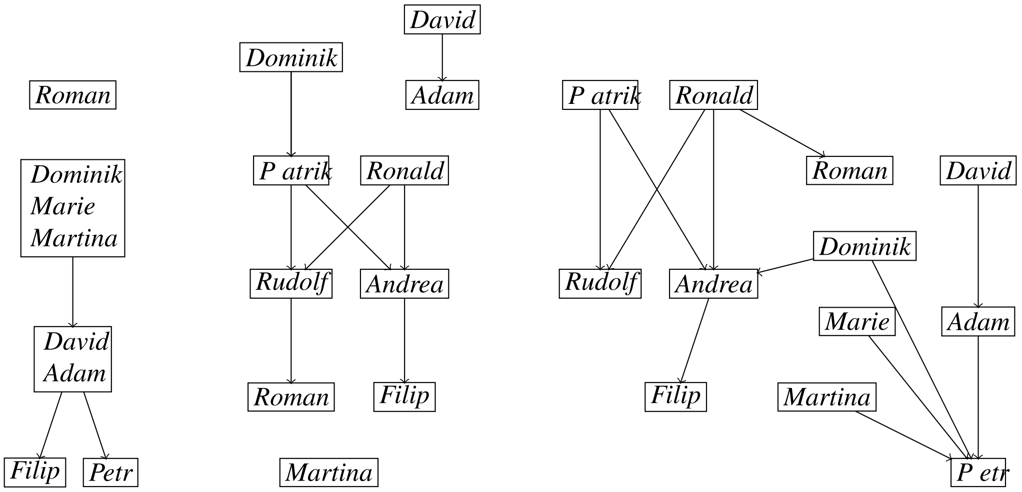
In the case of ordered relations $[R_1, \leq^{R_1}], [R_2, \leq^{R_2}]$, we define:

$$[R_1; \leq^{R_1}] \bigcup [R_2; \leq^{R_2}] = [R_1 \cup R_2; \leq^{R_1 \cup R_2}]$$

where

$$
\begin{aligned}
&\forall r_1, r_2 \in (R_1 \cup R_2)(r_1 \leq^{R_1 \cup R_2} r_2 \iff \\
&\qquad (r_1 \leq^{R_1} r_2 \quad \wedge \quad r_1 \leq^{R_2} r_2) \quad \vee \\
&\qquad (r_1 \leq^{R_1} r_2 \quad \wedge \quad r_1, r_2 \notin R_2) \quad \vee \\
&\qquad (r_1 \leq^{R_2} r_2 \quad \wedge \quad r_1, r_2 \notin R_1) \quad \vee \\
&\qquad (\exists r_3 \in (R_1 \cup R_2)(r_1 \leq^{R_1} r_3 \quad \wedge \quad r_3 \leq^{R_2} r_2)) \quad \vee \\
&(r_1 \in R_1 \cap R_2 \quad \wedge \quad r_2 \in R_1 \backslash R_2 \quad \wedge \quad r_1 \leq^{R_1} r_2 \quad \wedge \quad \forall r_3 \in R_1 \cap R_2 (r_2 \leq^{R_1} r_3 \Rightarrow r_1 \leq^{R_2} r_3)) \quad \vee \\
&(r_2 \in R_1 \cap R_2 \quad \wedge \quad r_1 \in R_1 \backslash R_2 \quad \wedge \quad r_1 \leq^{R_1} r_2 \quad \wedge \quad \forall r_3 \in R_1 \cap R_2 (r_3 \leq^{R_1} r_1 \Rightarrow r_3 \leq^{R_2} r_2)) \quad \vee \\
&(r_1 \in R_1 \cap R_2 \quad \wedge \quad r_2 \in R_2 \backslash R_1 \quad \wedge \quad r_1 \leq^{R_2} r_2 \quad \wedge \quad \forall r_3 \in R_1 \cap R_2 (r_2 \leq^{R_2} r_3 \Rightarrow r_1 \leq^{R_1} r_3)) \quad \vee \\
&(r_2 \in R_1 \cap R_2 \quad \wedge \quad r_1 \in R_2 \backslash R_1 \quad \wedge \quad r_1 \leq^{R_2} r_2 \quad \wedge \quad \forall r_3 \in R_1 \cap R_2 (r_3 \leq^{R_2} r_1 \Rightarrow r_3 \leq^{R_1} r_2)))
\end{aligned}
$$

Example 4 (Ordering on a union)



We can determine easily the ordering of the elements belonging to the intersection of the input relations and of the elements belonging to the symmetric difference of the input relations. Then we have to determine the ordering between elements from intersection and symmetric difference of the input relations. The possible contradictions following from the transitivity property of ordering have to be avoided.

Difference $R_1 \setminus R_2$ returns a relation consisting of all tuples appearing in the first R_1 and not the second R_2 of two specified relations.

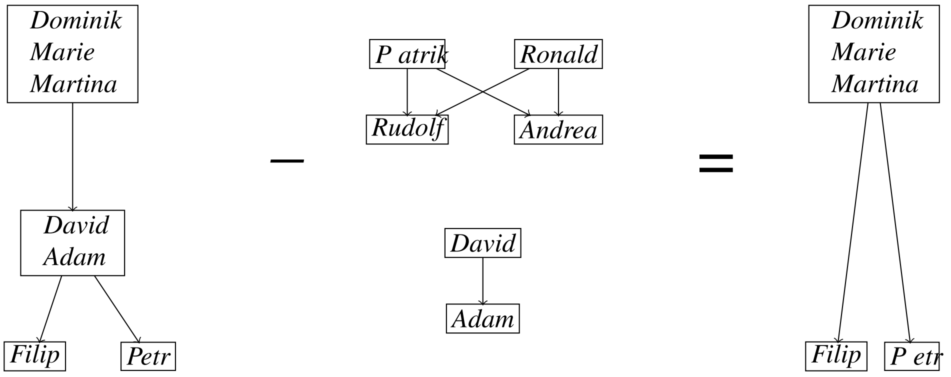
In the case of ordered relations $[R, \leq^{R_1}]$, $[R_2, \leq^{R_2}]$, we define:

$$[R_1; \leq^{R_1}] \setminus [R_2; \leq^{R_2}] = [R_1 \setminus R_2; \leq^{R_1 \setminus R_2}]$$

where

$$\leq^{R_1 \setminus R_2} = \leq^{R_1} \cap R_1 \setminus R_2 \times R_1 \setminus R_2$$

Example 5 (Ordering on a difference)



The difference ordering is the restriction of the input ordering on the the difference of the input relations.

Cartesian product $R_1 \times R_2$ returns a relation consisting of all possible tuples that are a combination of two tuples, one from each of two specified relations R_1, R_2.

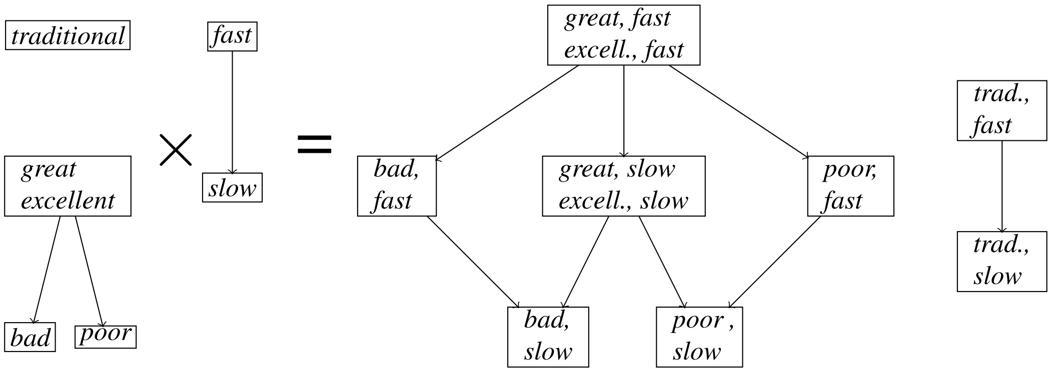
In the case of ordered relations $[R, \leq^{R_1}]$, $[R_2, \leq^{R_2}]$, we define:

$$[R_1; \leq^{R_1}] \times [R_2; \leq^{R_2}] = [R_1 \times R_2; \leq^{R_1 \times R_2}]$$

where

$$\leq^{R_1 \times R_2} = \{((r_1, r_2), (r_1', r_2')) | (r_1, r_1') \in \leq^{R_1} \wedge (r_2, r_2') \in \leq^{R_2}\}$$

Example 6 (Ordering on a cartesian product)



The output ordering is defined as a ordering of ordered pairs.

3.2. Aggregation Functions

In this subsection, we will extend the database relation R with a special element

$$\hat{r} = R(\phi), \quad \text{where} \quad \forall r \in R\ (\neg\phi(r))$$

In the following, the symbol R stands for this extended relation.

$$[R, \leq^R], \qquad \leq^R \text{ is a relation of a preference on R,}$$

$$\leq^R \text{ is generally no ordering on } R \text{ because } \leq^R \cap (\leq^R)^{-1} \not\subseteq I = (\leq^R)^0,$$

$$\leq^R \cap (\leq^R)^{-1} \subseteq R \times R,$$

$$[R; \equiv], \qquad \equiv \, = \, \leq^R \cap (\leq^R)^{-1}, \qquad \equiv \text{ is a relation of equivalence}$$

$$[R/\!\equiv; \leq^{R/\equiv}], \qquad \forall R_a, R_b \in R/\!\equiv (R_a \leq^{R/\equiv} R_b \iff a \leq^R b),$$

where

$$a, b \in R, \quad R_a = \{r \in R | r \equiv a\}, \quad R_b = \{r \in R | r \equiv b\}$$

$$\leq^{R/\equiv} \text{ is an ordering on } R/\!\equiv$$

$$[\mathscr{P}_{max}(R/\equiv); \leq^{\mathscr{P}_{max}(R/\equiv)}]$$

$$\begin{aligned}\mathscr{P}(R/\!\equiv) \supseteq \mathscr{P}_{max}(R/\equiv) = \{\tilde{R} \subseteq R/\equiv | \\ \forall R_a \in R/\!\equiv (\forall R_b \in R/\!\equiv (R_a \leq^{R/\equiv} R_b \Rightarrow R_a = R_b) \Rightarrow R_a \in \tilde{R}) \quad \wedge \\ \forall R_a \in \tilde{R}, \forall R_b \in R/\!\equiv (R_a \leq^{R/\equiv} R_b \Rightarrow R_b \in \tilde{R})\}\end{aligned}$$

$$\forall \tilde{R}_i, \tilde{R}_j \in \mathscr{P}_{max}(R/\equiv)(\tilde{R}_i \leq^{\mathscr{P}_{max}(R/\equiv)} \tilde{R}_j \iff \tilde{R}_i \supseteq \tilde{R}_j)$$

An aggregation function g in the classical relational data model is a function: $\mathscr{P}(R) \rightarrow \mathbb{R}$ operating on sets and returning numbers. In the case of relational databases with ordered relations, we define it as:

$$g: \quad \mathscr{P}([R; \leq^R]) \rightarrow [\mathbb{R}; \leq^{g(R)}],$$

where

$$\mathscr{P}([R; \leq^R]) = \{[R'; \leq^{R'}] | R' \subseteq R \wedge \leq^{R'} = \leq^R_{R'}\}$$

and

$$\begin{aligned}&\forall i, j \in \mathbb{R}(i \leq^{g(R)} j \iff \\ &\quad \exists \tilde{R}_j \in \mathscr{P}_{max}(R/\!\equiv)(g(\tilde{R}_j) = j \quad \wedge \quad \forall \tilde{R}_i \in \mathscr{P}_{max}(R/\!\equiv)(g(\tilde{R}_i) = i \Rightarrow \tilde{R}_i \leq^{\mathscr{P}_{max}(R/\equiv)} \tilde{R}_j))),\end{aligned}$$

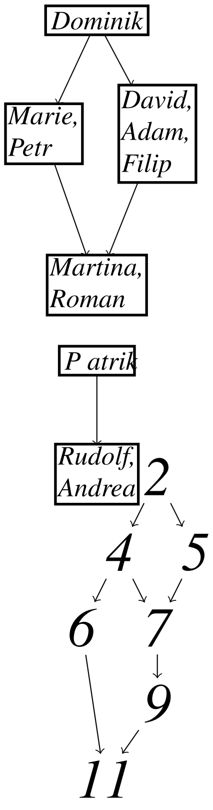
where g is defined with respect to the specific aggregation function as:

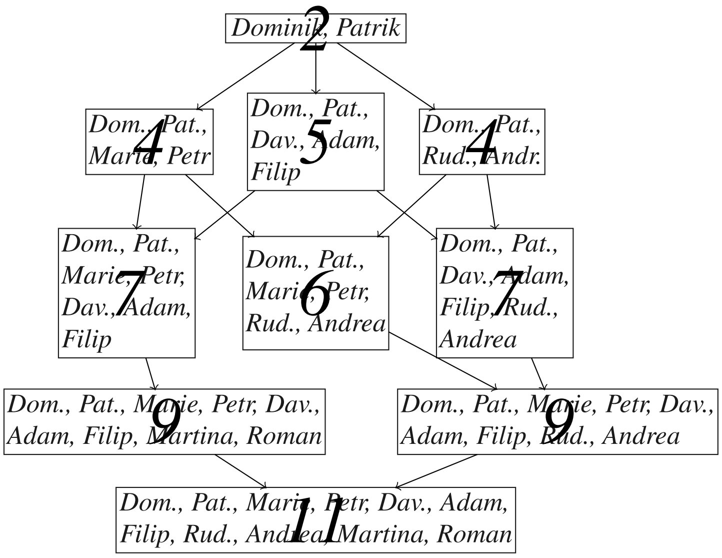
Count

$$[|R|; \leq^{|R|}]$$

$$g : \mathscr{P}_{max}(R/\!\equiv) \rightarrow |R|, \quad g(\tilde{R}) = \sum_{R_a \in \tilde{R}} |R_a|$$

Example 7 (Count on an ordered relation)





First we count the most preferred elements. Then the the less preferred elements are added. The rule is that we never add the elements that are in the hierarchy of the input ordering below the elements that we have not counted yet. The rationale behind this rule is that one always chooses the best elements possible. In this way, we get a lattice ordering of sets containing the maximal number of elements with the preference higher or equal to certain level. Then the classical count operation is applied and finally the resulting ordering determined.

The semantic of this final determination can be seen on the couple of 4 and 7 for instance: For any set of 7 elements having been chosen as the most preferred ones, there is its subset containing 4, more or equally preferred, elements. The elements with an equal preference are always taken into account together.

Max

$$[(-\infty; \max\{r.A|r \in R\}\rangle; \leq^{(-\infty;\max\{r.A|r\in R\}\rangle}]$$

$$g : \mathscr{P}_{max}(R/\!\!\equiv) \rightarrow (-\infty; \max\{r.A|r \in R\}\rangle, \quad g(\tilde{R}) = \max\{r.A|\exists R_a \in R/\!\!\equiv (r.A \in R_a \wedge R_a \in \tilde{R})\}$$

Min

$$[\langle \min\{r.A|r \in R\}; \infty); \leq^{\langle \min\{r.A|r\in R\};\infty)}]$$

$$g : \mathscr{P}_{max}(R/\!\!\equiv) \rightarrow \langle \min\{r.A|r \in R\}; \infty), \quad g(\tilde{R}) = \min\{r.A|\exists R_a \in R/\!\!\equiv (r.A \in R_a \wedge R_a \in \tilde{R})\}$$

Sum

$$[\mathbb{R}; \leq^{\text{Sum}(R)}],$$

$$g : \mathscr{P}_{max}(R/\!\!\equiv) \rightarrow \mathbb{R}, \quad g(\tilde{R}) = \sum_{\exists R_a \in R/\equiv (r.A \in R_a \wedge R_a \in \tilde{R})} r.A$$

Average

$$[\mathbb{R}; \leq^{\text{Avg}(R)}],$$

$$g : \mathscr{P}_{max}(R/\!\!\equiv) \rightarrow \mathbb{R}, \quad g(\tilde{R}) = \frac{\sum_{\exists R_a \in R/\equiv (r.A \in R_a \wedge R_a \in \tilde{R})} r.A}{\sum_{R_a \in \tilde{R}} |R_a|}$$

3.3. Arithmetic

We will consider a triplet of a relation R with a preference relation $\leq^R$ and basic arithmetic operations – denoted $\oplus$:

$$[R; \leq^R; \oplus]$$

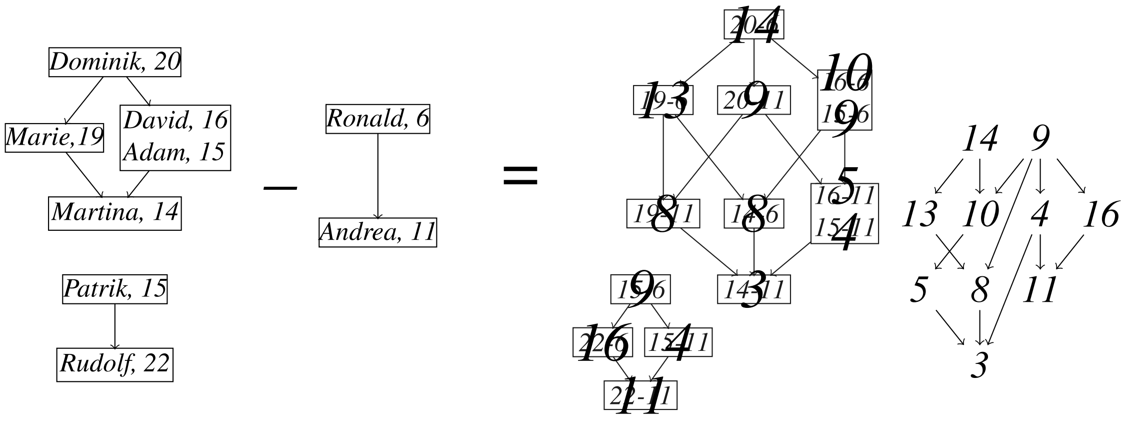
$$\oplus : [R_1; \leq^{R_1}][A] \times [R_2; \leq^{R_2}][B] \rightarrow [\mathbb{R}; \leq^{R_1[A] \oplus R_2[B]}]$$

where

$$\forall i, j \in \mathbb{R}\Bigg(i \leq^{R_1[A] \oplus R_2[B]} j \;\Leftrightarrow$$

$$\exists r_m \in R_1, r_n \in R_2 \Big(r_m.A \oplus r_n.B = j \wedge \forall r_k \in R_1, r_l \in R_2 \big(r_k.A \oplus r_l.B = i \;\Rightarrow\; (r_k, r_l) \leq^{R \times R} (r_m, r_n) \big) \Big) \Bigg)$$

Example 8 (Subtraction on an ordered relation) *Let's consider two input relations of programmers and managers respectively. We are interested in their names and years of practice only. The ordering reflects the preference based on their, say, proficiency. The question is: "What is the difference of years of practice between the most proficient programmers and managers?" We clearly need the arithmetic operation of subtraction.*



We have to consider all the possible couples of programmers and managers. The relation of these couples is ordered as ordered pairs of numbers. After performing the subtraction, the resulting ordering is determined.

The semantic of this final determination can be seen on the couple of 9 and 8 for instance: For any couple of a programmer and a manager having the difference of years of practise 8, there is another couple of a programmer and a manager that is above this couple in the hierarchy of preference and whose difference of years of practice is 9.

4. Conclusion

By means of redefinition of the *minimal set of relational algebra operations*, aggregation functions and arithmetic, we get operations corresponding to all the operations that we have in the relational database framework. Thus we maintain the expressive power of the classical relational model. As the new operations operate on and return ordered relations, we are able to handle an extra information of preference represented by an ordering. The result is the ability to retrieve more accurate data.

List of Symbols

$[a, b]$	an ordered pair of a and b
$R(\phi)$	a restriction of the relation R – the tuples satisfying a condition ϕ
$R[A]$	a projection of the relation R on the set of attributes A – subtuples of the relation R
$\leq^A$	an ordering relation with an index A (just a label)
$\leq_A$	a restriction of the ordering relation $\leq$ on the set A
$\leq^A_B$	a restriction of the ordering relation $\leq^A$ on the set B
$(\leq)^a$	a power a of the ordering relation $\leq$
$\equiv$	an equivalence relation
$R/\equiv$	$= \{R_a \vert R_a \subseteq R \wedge a \in R_a \wedge \forall r \in R(r \in R_a \Leftrightarrow r \equiv a)\}$
$\mathscr{P}(A)$	$= \{B \vert B \subseteq A\}$
$r.A$	the value that a tuple $r \in R$ acquires on an attribute A
$\oplus$	the general arithmetic operation $(+, -, \times, \div, \ldots)$

References

[1] C. J. Date, *An Introduction to Database Systems.* Pearson Education, 8th edition, 2004.

Precedence	Operation	Symbol
higher	projection	$R[A]$
$\uparrow$	restriction	$R(\phi)$
$\uparrow$	product	$\times$
$\uparrow$	difference	$\setminus$
lower	union, intersection	$\cup$, $\cap$

Table 1: Precedence of relation algebra operations

Digitální knihovny, biomedicínská data a znalosti

doktorand:
MGR. PETRA PŘEČKOVÁ
EuroMISE Centrum – Kardio
Ústav informatiky AV ČR
Pod Vodárenskou věží 2
182 07 Praha 8
preckova@euromise.cz

školitel:
PROF. RNDR. JANA ZVÁROVÁ, DRSC.
EuroMISE Centrum – Kardio
Ústav informatiky AV ČR
Pod Vodárenskou věží 2
182 07 Praha 8
zvarova@euromise.cz

obor studia:
Biomedicínská informatika
číselné označení: 3918V

Abstrakt

V tomto článku popisuji digitální knihovny, jejich vznik, definice, srovnání s klasickými knihovnami a seznamuji s principy vytváření digitálních knihoven. Jelikož se ve své disertační práci specializuji zejména na lékařské zprávy, soustřeďuji se v článku hlavně na digitální knihovny v medicíně. V druhé části se zabývám problematikou strukturalizace digitální informace a systémy umožňující sjednocování elektronických zdravotnických dokumentů jako je např. SNOMED, slovník MeSH nebo jazyk UMLS.

1. Úvod

Od poloviny devadesátých let minulého století je pojem digitální knihovny (digital libraries) velmi frekventovaně používán. Vyhledávací nástroj Google při zadání termínu "digital library" našel v červnu 2004 přibližně 7 760 000 záznamů.

Termín "digitální knihovna" je nejnovějším označením v dlouhé sérii názvů pro pojem, který byl popisován skoro od samého vzniku vývoje prvních počítačů.

Už v roce 1975 psal V. Bush o "memexu", počítačové aplikaci pro získávání informací. I když šlo o mechanické zařízení založené na technologii mikrofilmu, předjímal tento nástroj myšlenku hypertextů. Automatizace knihoven se začala rozvíjet začátkem 50. let 20. století pomocí děrovacích štítků. V roce 1965 J. C. R. Licklider poprvé použil spojení "knihovna budoucnosti", která popisovala jeho vizi plně počítačové knihovny. Skoro o 10 let později, v roce 1978 napsal F. W. Lancaster o blížící se "bezpapírové knihovně". Zhruba ve stejné době, v roce 1974, T. Nelson vynalezl a pojmenoval hypertext a hyperprostor. S postupem času se ještě začaly objevovat termíny jako "elektronická knihovna", "virtuální knihovna", "knihovna beze stěn", "bionická knihovna" a další.

Relativně nové použití termínu "digitální knihovna" vzniklo z Iniciativy digitálních knihoven (Digital Libraries Initiative), která byla založena spoluprací National Science Foundation, Advanced Research Projects

Agency a National Aeronautics and Space Administration v USA. Náhlý prudký nárůst internetu a rozvoj grafických webových prohlížečů donutil v roce 1994 tyto nadace, aby vydaly 24,4 milionů dolarů šesti americkým univerzitám na výzkum digitálních knihoven (Pool 1994). Tento termín se rychle uchytil mezi informatiky, knihovníky i v dalších profesích.

Význam pojmu digitální knihovna ale není tak jasný, jak by se na první pohled mohlo zdát. Nejprve si uveďme několik definic, vymezující tento pojem:

Digitální knihovna je spravovaná sbírka informací spolu s odpovídajícími službami, přičemž informace jsou uloženy v digitální podobě a jsou dostupné prostřednictvím sítě (Arms 2000). Tato definice zdůrazňuje, že to nejsou pouze digitální informace, které tvoří digitální knihovny, ale hlavními aspekty je jejich strukturalizace, kontext, správa, různé služby a to vše pomocí počítačové sítě.

I. H. Witten a D. Bainbridge popisují digitální knihovnu jako cílenou sbírku digitálních objektů, která zahrnuje objekty textové, vizuální a zvukové, spolu s metodami pro jejich zpřístupnění a získávání, stejně jako pro výběr, organizaci a uchovávání (Witten, Bainbridge 2003).

Podle D. J. Waterse jsou digitální knihovny organizace, které poskytují zdroje (včetně specializovaného materiálu) umožňující provádět výběr, strukturování a zpřístupnění sbírek digitálních prací, tyto práce dále distribuovat, udržovat jejich integritu a dlouhodobě uchovávat - a to vše s ohledem na snadné a ekonomické využití určitou komunitou nebo množinou komunit uživatelů (Waters 1998). I v této definici je zdůrazněna systematičnost uchovávání digitálních sbírek a také připomíná, že digitální knihovna je vytvářena a slouží pro potřeby určité konkrétní komunity uživatelů.

Digitální knihovny jsou tedy organizace, které využívají a zobrazují množství zdrojů, zejména intelektuálních zdrojů, které jsou zahrnuty ve specializovaných materiálech, ale nejsou organizovány stejným stylem jako tradiční knihovny. I přesto, že zdroje digitálních knihoven slouží podobným funkcím jako tradiční knihovny, jsou to knihovny v mnoha směrech odlišné. Například ukládání a vyhledávání je závislé pouze a jenom na počítačích a síťových systémech, na systémech vyžadujících spíše technické dovednosti než dovednosti pravého knihovníka či člověka pracujícího v kartotékách.

Jaké jsou tedy hlavní znaky digitálních knihoven:

- Cílem je, aby měl uživatel jednotný přístup k relevantním digitálním informacím a to bez ohledu na jejich formu, formát, způsob a místo uložení.
- Hlavním problémem není digitalizace fyzického materiálu, ale jeho organizace, strukturování a správa.
- Digitální knihovna není chápána jako uzavřený objekt. V literatuře se proto častěji užívá množné číslo - digitální knihovny.
- Digitální knihovny vyžadují síťové technologie, kterou umožňují propojení informačních zdrojů.
- Digitální knihovny nejsou vázány na dokumenty v tištěné formě.

Vlastnosti digitálních knihoven

Jak uvádí S. Makulová ve svém článku, je potřeba se zamyslet nad tím, do jaké míry by měly mít digitální knihovny vlastnosti tradiční knihovny. Je zřejmé, že digitální knihovny mají plno funkcí, které nenajdeme u tradičních knihoven. Na druhou stranu, digitálním knihovnám chybí mnoho funkcí tradiční knihovny a proto je potřeba si položit otázku, které funkce tradičních knihoven by měly zůstat zachované.

V následující tabulce jsou porovnávány vlastnosti digitálních knihoven v nejužším chápání, které je založené na tradičních knihovnách, až po nejvolnější chápání digitálních knihoven, které je založeno na současném internetu (Harter 1996).

ÚZKÉ CHÁPÁNÍ (založené na tradičních knihovnách)	ŠIRŠÍ CHÁPÁNÍ (střed mezi oběma krajními možnostmi)	NEJŠIRŠÍ CHÁPÁNÍ (volně založené na současném internetu)
informačními zdroji jsou objekty	informačními zdroji je většina objektů	informačními zdroji může být cokoli
objekty jsou vybírány podle kvality	některé objekty jsou vybírány na základě kvality	žádná kontrola kvality; žádné překážky při ukládání objektů
objekty jsou uloženy na fyzickém místě	objekty se nacházejí na logickém místě a mohou být distribuovány	objekty nejsou umístěny ani na fyzickém ani na logickém místě
objekty jsou organizované		objekty nejsou organizované
objekty procházejí kontrolou odborníků	je zde částečná kontrola odborníky	žádná kontrola
objekty jsou fixní (nemění se)	objekty se mění standardizovaným způsobem	objekty jsou nestabilní (kdykoli se mohou měnit)
objekty jsou trvalé (nezmizí)	zmizení objektů je kontrolováno	objekty jsou pomíjivé (mohou kdykoli zmizet)
důležitým pojmem je autorství	pojem autorství je oslaben	bez autora
přístup k objektům je limitován pouze specifickým skupinám uživatelů	přístup k některým objektům je limitován specifickým skupinám uživatelů	přístup k čemukoli je umožněn komukoli
k dispozici jsou nabídnuty služby, jako například referenční asistence		jedinými službami jsou ty, které umožňuje počítačový software (umělá inteligence)
existují lidští specialisté (knihovníci)		neexistují žádní knihovníci
existuje definovaná skupina uživatelů	některé skupiny objektů mají sdružené skupiny uživatelů	nejsou definované žádné skupiny uživatelů

Tabulka 1: Vlastnosti digitálních knihoven.

Principy vytváření digitálních knihoven

Vytvoření digitální knihovny je velmi nákladné. Než se začne s vytvářením takové knihovny,je potřeba mít na paměti několik základních principů, které představují základ návrhu, implementacea spravování jakékoli digitální knihovny. Podle A. T. McCray a M. E. Gallagher (2001) existuje 10 základních principů:

1. *Předvídat změny*. Změna technologií může přinést problémy. V současné době se technologie vyvíjejí tak rychle, že za nějakou dobu se může stát, že nemůžeme otevřít dokument uložen v nějakém starším formátu.

2. *Znát obsah*. Pro uživatele je obsah tím nejdůležitějším a nejhodnotnějším aspektem digitálních knihoven. Tvůrci digitálních knihoven proto musí rozhodnout o obsahu jejich digitální knihovny, což znamená, že musí vybrat objekty, které zde budou obsaženy, digitalizovat položky, které jsou pouze v analogové formě, označit položky standardizovaným jazykem jako je Standard Generalized Markup Language (SGML) a přiřadit metadata, která budou popisovat obsah a další atributy jednotlivých objektů.

3. *Zapojit správné osoby*. Ideální by bylo zahrnout osoby z různých prostředí, které nabízejí množství odborných znalostí z různých oborů, které přispějí k budování digitální knihovny. Nejdůležitější jsou samozřejmě dva obory: informatika a knihovnictví. Informatici si uvědomují možnosti i omezení technologie a jsou to vlastně oni, kdo vytváří systém. Knihovníci jsou ”strážci” informačních zdrojů, kteří nejenom že rozumí potřebám různých skupin, ale také otázkám vztahujícím se k uchovávání materiálů pro nepřetržitý přístup a použití.

4. *Navrhnout použitelný systém*. Většina digitálních knihoven je přístupná přes internet pomocí webových technologií, i když to není nezbytný znak digitálních knihoven. Jelikož jsou ale výhody webu tak veliké,

většina současných digitálních knihoven je tvořena tak, aby byla takto přístupná. Většina úspěšných tvůrců webových stránek počítá s mnoha faktory, jako jsou například technické odlišnosti mezi počítači a webovými prohlížeči, což zahrnuje i rychlost přístupu, a také rozdíly mezi uživateli. Prohlížeče se liší v tom, jak zobrazují informace, i když používají stejné základní komunikační protokoly (např. http nebo FTP) a standardní značkovací jazyky (jako jsou HTML nebo XML). Jelikož si uživatelé mohou změnit předem nastavené prostředí, čímž se rozumí velikost písma a další parametry, je lepší vždy vytvářet jednoduchý interface.

5. *Zajistit otevřený přístup*. Zajištění otevřeného přístupu je blízce spojeno s otázkami využitelnosti. Jedním způsobem, jak se dá tohoto docílit, je vyhnout se použití chráněného hardwaru a softwaru. Je rozumné vytvářet obsah za pomoci komerčně přístupných systémů a nástrojů a vyhnout se specializovanému softwaru a hardwaru, který by byl nutný k získání informací.

6. *Dávat si pozor na autorská práva*. Možnou hrozbou otevřenému přístupu k informacím se mohou stát problémy ohledně intelektuálního vlastnictví. Existující intelektuální vlastnictví a autorská práva poskytují ekonomickou a právní ochranu vydavatelům. V současné době nejsou jasné odpovědi na otázky ohledně uplatňování intelektuálního vlastnictví na informace, které jsou v digitální podobě. Internet a web vyvinuly skupiny, které věřily ve sdílení informací a ne v omezený přístup. To vedlo k dojmu, že cokoli je volně na webu přístupné, může být dále distribuováno. Lidé, vytvářející digitální knihovny, by měly mít povolení majitele autorských práv k digitalizaci materiálů. V ideálním případě by měl tento majitel označit citlivé informace a zanechat instrukce, jak s nimi má býti pracováno.

7. *Automatizovat kdykoli to půjde*. Jelikož budování digitální knihovny klade velké nároky na ty, kteří systém vytvářejí, čím automatizovanější nástroje se vybudují a budou používat, tím lépe se zužitkují vložené lidské zdroje. Tyto nástroje musí být jednoduché na používání.

8. *Osvojit si a přidržovat se standardů*. Používání standardů v systému má mnoho výhod. Aplikace jsou přístupnější a schopné pracovat společně.

9. *Zajistit kvalitu*. Všechny části vytváření knihovny (výběr, vkládání metadat, obrázků, využívání systému) by měly podléhat kvalitě. Nesprávná a nekompletní data ovlivňují kvalitu celé digitální knihovny. Tmavé, zkreslené a necelé obrázky nejsou v digitální knihovně vítané. Digitalizovaná videa a audia musí být pravidelně kontrolována, aby odpovídala současným audiovizuálním nástrojům. Některé kroky kontroly mohou být automatické, jiné vyžadují zásah člověka.

10. *Mějte na mysli přetrvávání*. V článku J. Rothenberga se dočteme o tom, že skupina 21 odborníků zjistila, že neexistuje žádný způsob, jak zaručit přetrvávání digitálních informací.

Při utváření digitálních knihoven, bychom měly přistupovat vážně k výše uvedeným bodům. S cenným obsahem by se mělo zacházet s péčí a měl by být poskytován v nejvyšší možné kvalitě. Tento cenný obsah by neměl zmizet. Měli bychom vědět, jak zacházet a chránit digitální materiál, aby se nestal zastaralým. Mělo bychom se snažit o otevřený přístup ke všem znalostem.

Digitální knihovny v medicíně

Ve zdravotnickém prostředí potřebují jak pacienti, tak i poskytovatelé zdravotnické péče rychlý a jednoduchý přístup k široké škále webových zdrojů. Pacienti a jejich rodiny potřebují informace, které jim vysvětlí jejich osobní situaci a lékaři potřebují informace, které se vztahují k jednotlivým pacientům. Tyto informace mohou lékaři pomoci při výběru pouze efektivních zákroků a diagnostických testů, mohou mu pomoci, aby nepřehlédnul nějakou diagnózu a aby minimalizoval možné komplikace. Nejnovější medicínské informace, pokud jsou popsány srozumitelným jazykem, mohou zmocnit pacienty, aby převzali kontrolu nad svým zdravím, aby se dozvěděli o prevenci a stali se informovanějšími při volbách týkajících se jejich léčby.

Na webu je možno najít několik takovýchto medicínských digitálních knihoven. Jedná se například MedlinePlus (http://medlineplus.gov/), což je největší medicínská knihovna vytvořená Národní lékařskou knihovnou v USA. Na informace v MedlinePlus se dá spolehnout, jsou ověřené a vždy aktualizované. V MedlinePlus můžete najít informace o více než 650 nemocech. Nachází se zde také seznam lékařů a nemocnic, lékařská encyklopedie, lékařský slovník, rozsáhlé informace o lécích na předpis i bez předpisu a odkazy na tisíce klinických experimentů.

ClinicalTrials.gov (http://clinicaltrials.gov/) poskytuje pravidelně aktualizované informace o klinických výzkumech na lidských dobrovolnících, poskytuje informace a cílech experimentů, kdo se jich může účastnit, místa, kde se experimenty provádí a další informace.

Virtual Children's Hospital (http://www.vh.org/pediatric/) je digitální knihovna shromažďující informace z pediatrie a příbuzných oborů. Cílem tohoto projektu je zpřístupňování těchto informací v co největší míře a hlavně v organizované podobě.

The Virtual Naval Hospital (http://www.dlib.org/dlib/may99/05dalessandro.html) je digitální knihovna lékařských věd pro Námořnictvo Spojených států amerických.

Další digitální knihovnou je PERSIVAL (PErsonalized Retrieval and Summarization of Image, Video and Language) (http://persival.cs.columbia.edu/), jehož primárním účelem je snadný přístup k lékařským informacím i literatuře v digitální knihovně, a to jak pro poskytovatele zdravotní péče, tak pro pacienty.

Problematika strukturalizace digitální informace

V této části bych se zaměřila na některé problémy, které souvisí s využitím textové informace uložené v digitální podobě z pohledu jejího využití v biomedicínských oborech a ve zdravotnictví.

S každým dnem se svět zdravotnictví stává o něco menším díky tomu, že medicínské znalosti a péče o pacienty jsou sdíleny bez ohledu na geografické hranice. Zároveň se ale navyšuje množství informací, které musí lékaři skladovat, sdílet a vyhledávat, aby mohli efektivně pokračovat ve své práci. Různé zdravotnické záznamy se musí sjednocovat a to nejenom během pacientova života, ale i mezi různými skupinami pacientů a mezi celými populacemi, aby se tak mohla zajistit ta nejlepší a správná léčba, aby se mohli sledovat trendy nemocí, atd.

Aby došlo ke sjednocení lékařské terminologie, vytvořili např. v Národní lékařské knihovně v USA jazyk UMLS (Unified Medical Language System). Cílem tohoto jazyka je umožnit rozvoj počítačových systémů, které by se chovaly jakoby "rozuměly" významu biomedicínského a zdravotnického jazyka. Národní lékařská knihovna vytváří a rozšiřuje znalostní zdroje UMLS (databáze) a přidružené softwarové nástroje (programy), které využívají vývojáři při budování a zlepšování elektronických informačních systémů, které vytvářejí, zpracovávají, vyhledávají, integrují a shromažďují biomedicínská a zdravotnická data a informace. Tyto UMLS databáze jsou také využívány v informatickém výzkumu. UMLS znalostní zdroje jsou univerzální. Nejsou optimalizované pro jednotlivé aplikace, ale mohou být používány v systémech, které vykonávají několik funkcí zahrnujících jeden nebo více druhů informací, např. v lékařských zprávách, vědecké literatuře, doporučeních, v datech veřejného zdraví. Softwarové nástroje napomáhají při přizpůsobování nebo využívání UMLS znalostních zdrojů pro určité účely. Lexikální nástroje pracují lépe v kombinaci s UMLS znalostními zdroji, ale mohou být používány i nezávisle.

Existují tři UMLS znalostní zdroje: Metathesaurus, Sémantická síť (Semantic Network) a Specializovaný slovník (SPECIALIST lexicon). Jsou rozšiřovány pomocí několika programů, které usnadňují jejich

Metathesaurus je velice rozsáhlá, víceúčelová a vícejazyčná lexikonová databáze, která zahrnuje informace o biomedicínských, zdravotnických a jim příbuzným pojmech. Dále obsahuje jejich různé názvy a vztahy mezi nimi. Metathesaurus vznikl z elektronických verzí mnoha různých thesaurů, klasifikací, souborů kódů a vyjmenovává regulované termíny, které se používají v péči o pacienta, při fakturaci zdravotnických služeb, ve veřejných zdravotnických statistikách, při indexování a katalogizaci biomedicínské literatury a při základním a klinickém výzkumu zdravotnických služeb. Hlavním cílem Metathesauru je spojit alternativní názvy stejných pojmů a identifikovat užitečné vztahy mezi různými pojmy.

Cílem sémantické sítě je poskytovat konzistentní kategorizaci všech pojmů zastoupených v UMLS Metathesauru.

Specializovaný slovník byl vytvořen za účelem poskytování lexikálních informací potřebných pro specializovaný systém zpracovávající přirozený jazyk (SPECIALIST Natural Language Processing System - NLP). Mělo by se jednat o veřejný anglický slovník zahrnující mnoho biomedicínských termínů. Každý pojem obsahuje syntaktické, morfologické a pravopisné informace, které jsou potřebné pro specializovaný NLP systém.

Další nástrojem pro zpracovávání lékařských zpráv je SNOMED CT (Systematized Nomenclature of Human and Veterinary Medicine). Jedná se o detailní klinickou referenční terminologii, založenou na kódování, která se skládá z 344 549 pojmů vztahujících se ke zdravotnictví. Tato terminologie umožňuje využívat zdravotnické informace kdykoli a kdekoli je to potřeba. SNOMED CT poskytuje "společný jazyk", který umožňuje konzistentní způsob získávání, sdílení a shromažďování zdravotnických dat od různých klinických skupin mezi které patří ošetřovatelství, medicína, laboratoře, lékárny i veterinární medicína. Pojmy poskytují společný jazyk pro komunikaci se zdravotnickými informacemi. SNOMED CT je dílem rozsáhlé spolupráce mezi světovými znalci klinické terminologie a je používán ve více než 40 státech.

Diagnóza pomocí terminologie SNOMED může obsahovat topografický kód, morfologický kód, kód živého organismu a funkční kód. Pokud existuje jasně definovaná diagnóza pro kombinaci těchto čtyř kódů, je definován specializovaný diagnostický kód. Například kód nemoci D-13510 (pneumokokový zánět plic) je ekvivalentem pro kombinaci těchto kódů:

T-2800 (topologický kód pro *plíce, nijak nespecifikované*),

M-40000 (morfologický kód pro *zánět, nijak nespecifikovaný*) a

L-251166 (pro *streptokokový zánět plic*) u živých organismů.

V souvislosti s medicínskou terminologií bych ještě ráda uvedla slovník MeSH (Medical Subjekt Headings). Jedná se o slovník kontrolovaný opět Národní lékařskou knihovnou v USA. Tvoří ho skupina pojmů, které hierarchicky pojmenovávají klíčová slova a tato hierarchie napomáhá při vyhledávání na různých úrovních specifičnosti. Klíčová slova v MeSH jsou uspořádána jak abecedně tak i hierarchicky. Na nejobecnější úrovni hierarchické struktury jsou široké pojmy jako např. "anatomie" nebo "mentální onemocnění". Hierarchie je jedenáctistupňová. Ve tomto tezauru se nachází 22 568 klíčových slov. Navíc je zde ale více než 139 00 tak zvaných doplňkových záznamů, které jsou uloženy v odděleném tezauru. Národní lékařská knihovna využívá MeSH k indexování článků ze 4600 světových předních biomedicínských časopisů pro MEDLINE/PubMED databázi. Využívá se také pro databázi katalogizující knihy, dokumenty a audiovizuální materiály, které Národní lékařská knihovna potřebuje. Každý bibliografický odkaz je spojován se skupinou termínů v MeSH a tyto termíny pak popisují obsah položky. Podobně i vyhledávací dotazy používají slovní zásobu z MeSH, aby našli články na požadované téma. Specialisté vytvářející MeSH slovník průběžně aktualizují a kontrolují. Sbírají nové pojmy, které se začínají objevovat ve vědecké literatuře nebo ve vznikajících oblastech výzkumu, definují tyto pojmy v rámci obsahu existujícího slovníku a doporučují jejich přidání do slovníku MeSH.

Závěr

Ve zdravotnictví je řada nástrojů pro strukturalizaci informací, které jsou založené na anglickém jazyce a pro český jazyk jsou tedy nepoužitelné. Z tohoto důvodu se snažíme o přístupy strukturalizace volného textu, které jsou nezávislé na použitém jazyce. Pro český jazyk je předpokladem vytvoření českého výkladového terminologického slovníku biomedicínských pojmů, který v současné době vytváříme v EuroMISE centru.

Poděkování

Tato práce vznikla částečně s podporou projektu LN00B107 Ministerstva školství, mládeže a tělovýchovy České republiky.

Literatura

[1] Arms, W. Y.: Digital Libraries. MIT Press, Cambridge, 2000. ISBN 0–262–01880–8

[2] Bush, V.: As We May Think. Atlantic Monthy 176 (1945), pp. 101–108.
http://www.isg.sfu.ca/ duchier/misc/vbush/

[3] Furnas, G. W., Rauch, S. J.: Considerations for Information Environments and the NaviQue Workspace. Proceeding of ACM DL ´98, pp. 79–88

[4] Lancaster, F. W.: Toward Paperless Information Systéme. New York. Academic Press, 1978

[5] Licklider, J. C. R.: Libraries of the Future. Cambridge, Mass. M. I. T. Press, 1965

[6] Makulová S.: Digitálne knižnice budúcnosti - nové smery výskumného programu.
http://www.aib.sk/infos/infos2003/15.html

[7] McCray A. T., Gallagher M. E.: Principles for Digital Library Development. Communications of the ACM, Vol. 44, No. 5, May 2001

[8] Nelson T. E.: Computer Lib. Chicago. Nelson, 1974

[9] Pool R.: Turbiny an Info-Glut into a Library. Science 266. 1994. pp. 20-22

[10] Přečková P.: Terminologický slovník 1.–12., Lékař a technika, Vol. 33–35, 2002–2004

[11] Rothenberg, J.: Avoiding Technological Quicksnad: Cindiny a Viable Technical Foundation for Digital Preservation. Rep. to Council on Library and Information Resources. 1999.
www.clir.org/pubs/reports/rothenberg/pub77.pdf

[12] Waters, D., J.: What are Digital Libraries? CLIR Issues, July/August 1998.
http://www.clir.org/pubs/issues/issues04.html

[13] Witten, I. H., Bainbridge, D.: How to Build a DigitalLibrary. Morgan Kaufmann, 2003. ISBN 1-55860-790-0

[14] http://medlineplus.gov/

[15] http://clinicaltrials.gov/

[16] http://www.vh.org/pediatric/

[17] http://www.dlib.org/dlib/may99/05dalessandro.html

[18] http://www.nlm.nih.gov/

[19] http://persival.cs.columbia.edu/

[20] http://www.nlm.nih.gov/

Modelling of Piezoelectric Materials

Post-Graduate Student:
ING. PETR RÁLEK
Katedra modelování procesů
Fakulta mechatroniky
Technická univerzita Liberec

Liberec, Czech Republic

petr.ralek@vslib.cz

Supervisor:
DOC. DR. ING. JIŘÍ MARYŠKA, CSC.
Katedra modelování procesů
Fakulta mechatroniky
Technická univerzita Liberec

Liberec, Czech Republic

jiri.maryska@vslib.cz

Field of Study:
Scientific engineering
Classification:

Abstract

Piezoelectric resonator is the thin stick or wafer made of the piezoelectric material, with two or more electrodes on its surface (see, e.g., [13]). In consequence of harmonic electric loading, the resonator oscillates. The most important parameters, describing the behavior of the resonator, are its *resonance frequencies* - frequencies of the oscillations with maximal amplitudes in some characteristic directions. Piezoelectric resonators are used, e.g., as stabilisators of frequencies of electric circuits, frequency filters, sensors of nonelectric quantities. For piezoelectric materials, resonance frequencies are typically determined by experimental or analytical methods. Analytical methods are, however, applicable only for some particular, simply posed problems and simply shaped resonators. The main disadvantage of the experimental testing is its high cost.

In this paper the finite element (FEM) model of the piezoelectric resonator based on the physical description of the piezoelectric material is described. Discretization of the problem then leads to a large sparse linear algebraic system, which defines the generalized eigenvalue problem. Resonance frequencies are subsequently found by solving this algebraic problem. Depending on the discretization parameters, this problem may become large, which may complicate application of standard techniques known from the literature. It should be pointed out, that typically we are not interested in all eigenvalues (resonance frequencies). For determining of several of them it seems therefore appropriate to consider iterative methods. Based on the finite element discretization of the mathematical model, we wish to propose, implement and test numerical algorithms for computing several resonance frequencies of piezoelectric resonators, and compare our results with experimental measurements.

1. Physical description

A crystal made of piezoelectric material represents a structure in which the deformation and electric field depend on each other. A deformation (impaction) of the crystal induces electric charge on the crystal's surface. On the other hand, subjecting a crystal to electric field causes its deformation. In linear theory of piezoelectrocity, derived by Tiersten in [11], this process is described by two constitutive equations - the **generalized Hook's law** (1) and the **equation of the direct piezoelectric effect** (2),

$$\mathrm{T_{ij}} = c_{ijkl}\,\mathrm{S_{kl}} - d_{kij}\,\mathrm{E_k}, \qquad i,j = 1,2,3, \tag{1}$$

$$\mathrm{D}_k = d_{kij}\,\mathrm{S_{ij}} + \varepsilon_{kj}\,\mathrm{E_j}, \qquad k = 1,2,3. \tag{2}$$

Here, as in other similar terms troughout the thesis, we use the convention known as the Einstein's additive rule ($a_{ij}b_j = \sum_{j=1}^{3} a_{ij}b_j$, see e.g. [12]). The Hook's law (1) describes dependence between the **stress tensor T**, the **strain tensor S** and the **vector of intensity of electric field E**,

$$\mathrm{S}_{ij} = \frac{1}{2}\left[\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right], \qquad i,j = 1,2,3, \qquad \mathrm{E_k} = -\frac{\partial \tilde{\varphi}}{\partial x_k}, \qquad k = 1,2,3,$$

where $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3)^{\mathrm{T}}$ is the **displacement vector** and $\tilde{\varphi}$ is the **electric potential**. The strain tensor $\mathbf{S}$ and the stress tensor $\mathbf{T}$ are symmetric [13]. The equation of the direct piezoelectric effect (2) describes the dependence between the **vector of electric displacement** $\mathbf{D}$, the strain and the intensity of electric field. Quantities c_{ijkl}, d_{kij} and ε_{ij} represent symmetric material tensors, playing role of the material constants. From the conditions of the thermodynamic stability ([10], part II), tensors c_{ijkl} and ε_{ij} have to be symmetric and positive definite.

Computing of oscillation of the pure *elastic continuum* is solved by analytical methods or by discretization of the continuum into lumped parameters, for which motion equations are solved. The finite element method (FEM) represents nowadays one of the most important discretization method. It divides the continuum into finite elements, where values of unknown functions in nodes of division are approximated with the help of special basis functions. As a result a system of ordinary differential equations is obtained. For description of widely used methods see e.g. in [3] or [4]. For piezoelectric continuum, oscillations of simply posed problems are usually solved by analytical methods (a survey of analytical methods is given in [13]). Experimental measurements are in many cases too expensive and therefore impractical. Mathematical modelling of more complicated settings require using of advanced numerical techniques. That is the motivation for using FEM. Its basic formulation was published by Allik back in 1970 [1], but the rapid progress in FEM modelling in piezoeletricity came in the last ten years.

1.1. Oscillation of the piezoelectric continuum

Consider resonator made of piezoelectric material with density ϱ, characterized by material tensors. We denote the volume of the resonator as Ω and its boundary as Γ. Behavior of the piezoelectric continuum is governed, in some time range $(0, \mathrm{T})$, by two differential equations: Newton's law of motion (3)and the quasistatic approximation of Maxwell's equation (4) (see, e.g., [6]),

$$\varrho \frac{\partial^2 \tilde{u}_i}{\partial t^2} = \frac{\partial \mathrm{T_{ij}}}{\partial x_j} \qquad i = 1,2,3, \qquad x \in \Omega, \quad \mathrm{t} \in (0, \mathrm{T}), \tag{3}$$

$$\nabla . \, \mathbf{D} = \frac{\partial \mathrm{D_j}}{\partial x_j} = 0. \tag{4}$$

Replacement of **T**, resp. **D** in (3) and (4) with the expressions (1), resp. (2), gives

$$\varrho \frac{\partial^2 \tilde{u}_i}{\partial t^2} = \frac{\partial}{\partial x_j}\left(c_{ijkl}\, \frac{1}{2}\left[\frac{\partial \tilde{u}_k}{\partial x_l} + \frac{\partial \tilde{u}_l}{\partial x_k}\right] + d_{kij}\, \frac{\partial \tilde{\varphi}}{\partial x_k}\right) \quad i = 1,2,3, \tag{5}$$

$$0 = \frac{\partial}{\partial x_k}\left(d_{kij}\, \frac{1}{2}\left[\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right] - \varepsilon_{kj}\, \frac{\partial \tilde{\varphi}}{\partial x_j}\right). \tag{6}$$

Initial conditions, Dirichlet boundary conditions and Neumann boundary conditions are added:

$$\begin{array}{rcl} \tilde{u}_i(.,0) & = & u_i, \quad x \in \Omega, \\ \tilde{u}_i & = & 0, \qquad i = 1,2,3, \quad x \in \Gamma_u, \\ \mathrm{T_{ij}} n_j & = & f_i, \qquad i = 1,2,3, \quad x \in \Gamma_f, \\ \tilde{\varphi}(.,0) & = & \varphi, \\ \tilde{\varphi} & = & \varphi_D, \quad x \in \Gamma_\varphi \\ \mathrm{D_k} n_k & = & q, \quad x \in \Gamma_q, \end{array} \tag{7}$$

where

$$\Gamma_u \cup \Gamma_f = \Gamma, \; \Gamma_u \cap \Gamma_f = \emptyset, \; \Gamma_\varphi \cup \Gamma_q = \Gamma, \; \Gamma_\varphi \cap \Gamma_q = \emptyset.$$

Right-hand side f_i represents mechanical excitation by external mechanical forces, q denotes electrical excitation by imposing surface charge (in the case of free oscillations, they are both zero). Equations (5)-(6) define the problem of harmonic oscillation of the piezoelectric continuum under given conditions (7). We will discretize the problem using FEM.

2. Weak formulation

Discretization of the problem (5)-(7) and the use of the finite element method is based on so called *weak formulation*. We briefly scetch the function spaces used in our weak formulation. We deal with the weak formulation derived in [9], chapters 28-35. For more details we recommend the reader to this book. We consider bounded domain Ω with Lipschitzian boundary Γ. Let $L_2(\Omega)$ be the Lebesgeue space of functions square integrable in Ω. *Sobolev space* $W_2^{(1)}(\Omega)$ is made of functions from $L_2(\Omega)$, which have generalized derivatives square integrable in Ω. To express values of function $u \in W_2^{(1)}(\Omega)$ on the boundary Γ, the *trace* of function u is established (see [9]; for function from $C^{(\infty)}(\overline{\Omega})$, its trace is determined by its values on the boundary).

Now, we establish

$$V(\Omega) = \{v | v \in W_2^{(1)}(\Omega), \quad v|_{\Gamma_1} = 0 \text{ in the sence of traces}\},$$

the subspace of $W_2^{(1)}(\Omega)$, made of functions, which traces fulfil the homogenous boundary conditions.

We derive the weak formulation in the standard way ([9], chapter 31). We multiply the equations (5) with testing functions $w_i \in V(\Omega)$, summarize and integrate them over Ω. As well, we multiply the equation (6) with testing function $\phi \in V(\Omega)$ and integrate it over Ω. Using Green formula, we obtain the integral equalities (boundary integrals are denoted with sharp brackets)

$$\begin{aligned}\left(\varrho \frac{\partial^2 \tilde{u}_i}{\partial t^2}, w_i\right)_\Omega + \left(c_{ijkl}\, \frac{1}{2}\left[\frac{\partial \tilde{u}_k}{\partial x_l} + \frac{\partial \tilde{u}_l}{\partial x_k}\right], \frac{\partial w_i}{\partial x_j}\right)_\Omega \\ + \left(d_{kij}\, \frac{\partial \tilde{\varphi}}{\partial x_k}, \frac{\partial w_i}{\partial x_j}\right)_\Omega = \left\langle f_i, w_i \right\rangle_{\Gamma_f},\end{aligned} \tag{8}$$

$$\left(d_{jik}\, \frac{1}{2}\left[\frac{\partial \tilde{u}_i}{\partial x_k} + \frac{\partial \tilde{u}_k}{\partial x_i}\right], \frac{\partial \phi}{\partial x_j}\right)_\Omega - \left(\varepsilon_{ji}\, \frac{\partial \tilde{\varphi}}{\partial x_i}, \frac{\partial \phi}{\partial x_j}\right)_\Omega = \left\langle q, \phi \right\rangle_{\Gamma_q}. \tag{9}$$

Let us denote

$$\mathrm{R}_{ij} = \frac{1}{2}\left[\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i}\right], \qquad i,j = 1,2,3.$$

Due to the symmetry of material tensors, equations (8) and (9) are equivalent to simplified forms of integral equalities,

$$\left(\varrho \frac{\partial^2 \tilde{u}_i}{\partial t^2}, w_i\right)_\Omega + \left(c_{ijkl}\, \mathrm{S}_{kl}, \mathrm{R}_{ij}\right)_\Omega + \left(d_{kij}\, \frac{\partial \tilde{\varphi}}{\partial x_k}, \mathrm{R}_{ij}\right)_\Omega = \left\langle f_i, w_i \right\rangle_{\Gamma_f}, \tag{10}$$

$$\left(d_{jik}\, \mathrm{S}_{ik}, \frac{\partial \phi}{\partial x_j}\right)_\Omega - \left(\varepsilon_{ji}\, \frac{\partial \tilde{\varphi}}{\partial x_i}, \frac{\partial \phi}{\partial x_j}\right)_\Omega = \left\langle q, \phi \right\rangle_{\Gamma_q}. \tag{11}$$

Weak solution: Let

$$\tilde{\mathbf{u}}_D \in ([W_2^{(1)}(\Omega)]^3, C^{(2)}(0,\mathrm{T})),\ \tilde{\varphi}_D \in (W_2^{(1)}(\Omega), AC(0,\mathrm{T}))$$

satisfy the Dirichlet boundary conditions (in the weak sence). Further, let

$$\tilde{\mathbf{u}}_0 \in ([W_2^{(1)}(\Omega)]^3, C^{(2)}(0,\mathrm{T})),\ \varphi_0 \in (W_2^{(1)}(\Omega), AC(0,\mathrm{T}))$$

be functions, for which equalities (10) and (11) are observed for all choices of testing functions

$$\mathbf{w} = (w_1, w_2, w_3) \in [V(\Omega)]^3,\ \phi \in V(\Omega).$$

Then we define the *weak solution* of the problem (5)-(7) as

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}_D + \tilde{\mathbf{u}}_0, \quad \tilde{\varphi} = \tilde{\varphi}_D + \tilde{\varphi}_0.$$

Weak solution, on the contrary to the classical solution, does not necesarilly have continuous spatial derivatives of the 2nd order. The weak solution has generalized spatial derivatives and statisfies the integral identities (10), (11).

3. Discretization of the problem

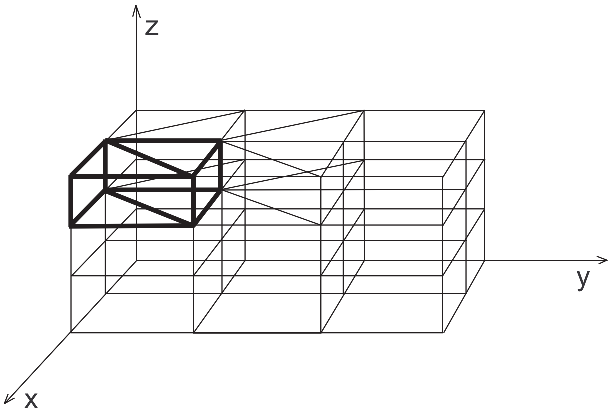


Figure 1: Division of a cubic crystal into layers and prizmatic elements

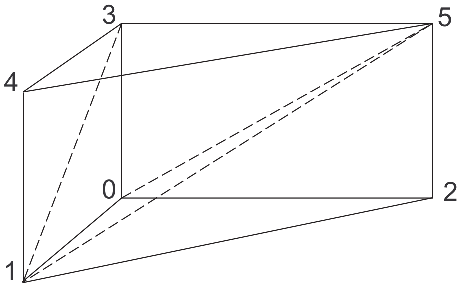


Figure 2: Division of a prizmatic element into three tetrahedrons 0125, 0153 a 1534

Finite element method constructs finite dimensional approximation of the weak solution. The domain Ω is decomposed into a set of *finite elements*, where special basis functions are established. Then, weak solution as the linear combination of these basis functions is looked for. The parts $\mathbf{u}_D, \varphi_D$ of the weak solution, satisfying the Dirichlet boundary conditions, then can be explicitly expressed in the linear system, resulting from discretization of the problem (10), (11), and are introduced in the paragraph 3.1.

In our case, we use the folowing FEM approximation. In two steps, we decompose the domain Ω (which is the volume of the resonator) into the finite set E^h of disjoint tetrahedral elements (the first step - shown in the figure 1 - means the division into the layers and prizmatic elements, the second part the division of the prizmatic elements into the tethrahedrons - figure 2). The domain Ω is approximated by the union of these tethrahedrons,

$$\Omega \sim \Omega^h = \bigcup_{e \in E^h} e, \qquad \bigcup_{e \in E^h} e = \overline{\Omega},$$

where ^{h} denotes the discretization parameter ($diam(e) < h \;\forall e \in E^h$). The boundary Γ is approximated as

$$\Gamma^h = \partial\Omega^h.$$

On the union Ω^h, we construct the finite dimensional approximation $V^h(\Omega)$ of the function space $V(\Omega)$. Functions from $V^h(\Omega)$ are piecewise linear and continuous on Ω^h and are zero on the boundary. For each tethrahedron $e \in E^h$, we define set $\Psi^h(e)$ of four linear multinomials,

$$\psi_i^e(x, y, z) = \alpha_{0i}^e + \alpha_{1i}^e x + \alpha_{2i}^e y + \alpha_{3i}^e z. \tag{12}$$

Consider an element $e = \{s^1, s^2, s^3, s^4\}$. Its *j*-th node s^j has coordinates (x_j, y_j, z_j). Basis functions can be uniquely defined by its values at the nodes s^j of the element and have to satisfy

$$\psi_i^e(s^j) = \delta_{ij}, \; \psi_i^e|_{\Omega^h - e} = 0 \qquad i, j = 1, 2, 3, 4.$$

The coefficients $\alpha_{.i}$ in (12) can be computed by inverting the matrix of node's coordinates (see, e.g. [8]). For each tethrahedron, the basis is made of four these linear multinomials. They generate the function space $V^h(e)$,

$$V^h(e) = \{\psi^h | supp(\psi^h) \subset e, \quad \psi^h \in W_2^1(e), \quad \psi^h|_{\Omega - e} = 0\}.$$

The union

$$\Psi^h(\Omega) = \bigcup_{e \in E^h} \Psi^h(e)$$

forms the basis of function space

$$V^h(\Omega) = \bigcup_{e \in E^h} V^h(e),$$

which is the finite dimensional approximation of the space $V(\Omega)^1$. The global approximations of the electric potential and displacement, lying in the space $V^h(\Omega)$, are:

$$\begin{aligned} \tilde{u}_i^h(\mathbf{x}) &= \sum_{\psi_j^h \in \Psi^h} u_i^j(t)\psi_j^h(\mathbf{x}), \quad u_i^j : (0, \mathrm{T}) \to \mathbf{R}, \quad \mathbf{x} \in \Omega, \quad i = 1, 2, 3, & (13) \\ \tilde{\varphi}^h(\mathbf{x}) &= \sum_{\psi_j^h \in \Psi^h} \varphi^j(t)\psi_j^h(\mathbf{x}), \quad \varphi^j : (0, \mathrm{T}) \to \mathbf{R}, \quad \mathbf{x} \in \Omega, \end{aligned}$$

and for its derivatives holds

$$\frac{\partial \tilde{u}_i^h}{\partial x_i}(\mathbf{x}) = \sum_{\psi_j^h \in \Psi^h} u_i^j(t) \frac{\partial \psi_j^h}{\partial x_i}(\mathbf{x}), \; \frac{\tilde{\varphi}^h}{\partial x_i}(\mathbf{x}) = \sum_{\psi_j^h \in \Psi^h} \varphi^j(t) \frac{\partial \psi_j^h}{\partial x_i}(\mathbf{x}). \tag{14}$$

Let the nodes of the division and the global basis functions be numbered, $(\psi_1^h, ..., \psi_r^h)$. We denote

$$\begin{aligned} \mathrm{U}^{\mathrm{T}} &= (u_1^1(t), u_2^1(t), u_3^1(t), u_1^2(t), u_2^2(t), u_3^2(t), ..., u_1^r(t), u_2^r(t), u_3^r(t)), & (15) \\ \Phi^{\mathrm{T}} &= (\varphi_1^1(t), \varphi_2^1(t), \varphi_3^1(t), \varphi_1^2(t), \varphi_2^2(t), \varphi_3^2(t), ..., \varphi_1^r(t), \varphi_2^r(t), \varphi_3^r(t)). & (16) \end{aligned}$$

U and Φ are values of displacement and electric potential at the nodes of division in time t. The approximations (13) are piecewise linear on Ω^h, approximations of derivatives are piecewise constant (in the spatial variable). We substitute approximations (13) and (14) into integral equalities (10) and (11). We require to them to be fulfiled for all basis functions ψ_s^h, $s \in \hat{r}$,

$$\left(c_{ijkl}\, \mathrm{S}^{\mathrm{h}}_{\mathrm{kl}}, \mathrm{R}^{\mathrm{h}}_{\mathrm{ij}}\right)_\Omega + \left(\varrho \frac{\partial^2 \tilde{u}_i^h}{\partial t^2}, \psi_s^h\right)_\Omega + \left(d_{kij}\, \frac{\partial \tilde{\varphi}^h}{\partial x_k}, \mathrm{R}^{\mathrm{h}}_{\mathrm{ij}}\right)_\Omega = \left\langle f_i, \psi_s^h \right\rangle_{\Gamma_f}, \tag{17}$$

$$\left(d_{jik}\, \mathrm{S}^h_{ik}, \frac{\partial \psi_s^h}{\partial x_j}\right)_\Omega - \left(\varepsilon_{ji}\, \frac{\partial \tilde{\varphi}^h}{\partial x_i}, \frac{\partial \psi_s^h}{\partial x_j}\right)_\Omega = \left\langle q, \psi_s^h \right\rangle_{\Gamma_q}. \tag{18}$$

The system of ordinary differential equations for values of displacement and potential in the nodes of division results, having block structure

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{K}\mathrm{U} + \mathbb{P}^{\mathrm{T}}\Phi = \mathrm{F}, \tag{19}$$

$$\mathbb{P}\mathrm{U} + \mathbb{E}\Phi = \mathrm{Q}. \tag{20}$$

The submatrix $\mathbb{K} \in \mathbf{R}^{3r,3r}$ is the elastic matrix, $\mathbb{M} \in \mathbf{R}^{3r,3r}$ is the massmatrix, $\mathbb{P} \in \mathbf{R}^{r,3r}$ is the piezoelectric matrix and $\mathbb{E} \in \mathbf{R}^{r,r}$ is the electric matrix. Matrices $\mathbb{K}, \mathbb{M}, \mathbb{E}$ are symmetric. Vectors F and Q represent

---

[1] The basis functions defined on nearby elements, which belong to the same node i of division, form together one global basis function. This function is normalized to have value one in the node i.

the mechanical and electrical excitation, respectively. F are nodal forces, resp. Q nodal charges. Each matrix has also the block structure (for definition, see [8]),

$$\mathbb{K} = \begin{pmatrix} \mathbb{K}_{11} & \mathbb{K}_{12} & ... & \mathbb{K}_{1r} \\ \mathbb{K}_{21} & \mathbb{K}_{22} & ... & \mathbb{K}_{2r} \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ \mathbb{K}_{r1} & \mathbb{K}_{r2} & ... & \mathbb{K}_{rr} \end{pmatrix}, \tag{21}$$

$$\mathbb{K}_{pq} \in \mathcal{R}^{3,3},$$

$$\mathbb{M} = \begin{pmatrix} \mathbb{M}_{11} & \mathbb{M}_{12} & ... & \mathbb{M}_{1r} \\ \mathbb{M}_{21} & \mathbb{M}_{22} & ... & \mathbb{M}_{2r} \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ \mathbb{M}_{r1} & \mathbb{M}_{r2} & ... & \mathbb{M}_{rr} \end{pmatrix}, \tag{22}$$

$$\mathbb{M}_{pq} \in \mathcal{R}^{3,3},$$

$$\mathbb{P} = \begin{pmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} & ... & \mathbb{P}_{1r} \\ \mathbb{P}_{21} & \mathbb{P}_{22} & ... & \mathbb{P}_{2r} \\ \vdots & \vdots & s & \vdots \\ \vdots & \vdots & s & \vdots \\ \mathbb{P}_{r1} & \mathbb{P}_{r2} & ... & \mathbb{P}_{rr} \end{pmatrix}, \tag{23}$$

$$\mathbb{P}_{pq} \in \mathcal{R}^{3,1},$$

$$\mathbb{E} = - \begin{pmatrix} \mathbb{E}_{11} & ... & \mathbb{E}_{1r} \\ \mathbb{E}_{21} & ... & \mathbb{E}_{2r} \\ \vdots & & \vdots \\ \mathbb{E}_{r1} & ... & \mathbb{E}_{rr} \end{pmatrix}, \tag{24}$$

where

$$\mathbb{E}_{pq} \in \mathcal{R}.$$

3.1. Boundary conditions

We deal with Dirichlet boundary conditions (7) for displacement and electric potential. The introduction of the boundary conditions is sketched on the fig. 3. First case is the homogenous boundary condition for displacement $\mathbf{u}^2$. Let there be in some nodes prescribed zero displacements (on the fig.3 marked with gray color). Then proper columns of the matrix (marked with gray color) are multiplied by zeros and can be eliminated. So can be eliminated the prescribed variables from the vector of unknowns. Now, the number of equation is bigger than the number of unknows, thus the rows (marked with gray color) belonging to the known variables can be eliminated. The resulting submatrices $\mathbb{K}$ and $\mathbb{M}$ are symmetric and positive definite (due to the positive definiteness of material tensors, see e.g. [5], chapter 20).

The similar situation occurs, when zero electric potential is prescribed[3]. Proper columns and rows can be eliminated and submatrix $\mathbb{E}$ becomes positive definite.

---

[2] It is possible to prescribe here the nonhomogenous displacement, but in practice, the zero displacement is established, e.g. due to resonator mounting.

[3] E.g. by the grounding of the resonator.

In the case of nonhomogenous Dirichlet boundary conditions for electric potential, there are some differences. The part of the vector with prescribed values is marked with the grid. The proper columns of the matrix are multiplied by prescribed values and the resulting vector can be set to the right-hand side of the linear system. The rows (marked with the grid) belonging to the known variables can be eliminated. Resulting matrix $\mathbb{E}$ is symmetric and positive definite. The linear system with different right-hand side results, with deflated matrix,

$$\hat{\mathbb{M}}\ddot{\mathrm{U}} + \hat{\mathbb{K}}\mathrm{U} + \hat{\mathbb{P}}^{\mathrm{T}}\Phi = \mathrm{F} + \mathrm{F}_{\varphi}, \tag{25}$$

$$\hat{\mathbb{P}}\mathrm{U} - \hat{\mathbb{E}}\Phi = \mathrm{Q} + \mathrm{Q}_{\varphi}. \tag{26}$$

F_{φ} represents generated electric force, Q_{φ} generated surface charge.

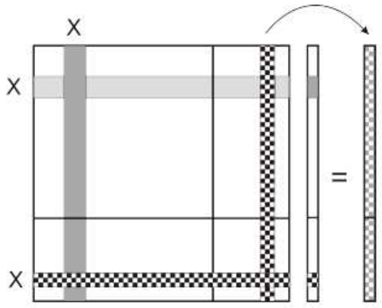


Figure 3: Introduction of boundary conditions into the linear system

3.2. Input errors of the model

In the process of derivation of the model, we have made some simplifications on the physical reality. Further, we must deal with other errors resulting from the used methods.

We use the linear approximation $\mathbf{u}^h \in V^h(\Omega)$ of the weak solution $\mathbf{u} \in W_2^{(1)}(\Omega^h)$. The theory of approximation error is introduced e.g. in [2], we only mention here, that for our problem the global approximation estimate is propotional to h,

$$||\mathbf{u} - \mathbf{u}^h||_{W_2^{(1)}} \sim \mathcal{O}(h).$$

The same holds for approximation error for weak solution of potential φ.

Using the numerical integration of constant, linear or quadratic functions on the tetrahedral elements, we don't generate other error.

First simplification was made in establishing the piezoelectric equations of state. In the Hook's law, resp. Maxwell's equation, we used the linear dependance of the strain on the deformation - in the reality, this dependance is nonlinear and material tensors of higher orders must be used (see e.g. [13]), multiplied by the higher derivatives of displacement and potential. By this simplification, the error of order $\mathcal{O}(h^2)$ is generated, which is less then the error made by linear approximation.

4. Dimension of the problem

The size of the matrices in (19) depends on the number of the nodes in division, say r. From (21)-(24) can be seen that the sizes of the submatrices are

$$\mathbb{K}, \mathbb{M} \in \mathcal{R}^{3r,3r}, \mathbb{E} \in \mathcal{R}^{r,r}, \mathbb{P} \in \mathcal{R}^{r,3r}.$$

The submatrices are sparse. The blocks $\mathbb{K}_{pq}, \mathbb{M}_{pq}, \mathbb{E}_{pq}, \mathbb{P}_{pq}$ (according to the terms (21)-(24)), $p, q \in \hat{r}$, are nonzero only if $p = q$ or if nodes s^p and s^q have common edge. For our scheme of discretization, the number of nonzero blocks in each submatrix is proportional to $12r$ (in the worst case).

When Dirichlet boundary conditions are prescribed, the dimension of the submatrices decreases,

$$\hat{\mathbb{K}}, \hat{\mathbb{M}} \in \mathcal{R}^{3r_1,3r_1}, \hat{\mathbb{E}} \in \mathcal{R}^{r_2,r_2}, \hat{\mathbb{P}} \in \mathcal{R}^{r_2,3r_1},$$

where r_1 is number of the nodes, where no Dirichlet BC for the displacement are precribed, r_2 is number of the nodes with no prescribed Dirichlet BC for the potential.

4.1. Points of interest

Let us write the system (25), (26) with introduced boundary conditions as

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{K}\mathrm{U} + \mathbb{P}^{\mathrm{T}}\Phi = \tilde{\mathrm{F}}, \tag{27}$$

$$\mathbb{P}\mathrm{U} - \mathbb{E}\Phi = \tilde{\mathrm{Q}}, \tag{28}$$

where on the right-hand side are sums of external and generated forces, resp. charges. The submatrices (here written without hats) have the properties described in paragraph 3.1. This system describes the general oscillation of piezoelectric element, with mechanical or electrical excitation. There are several ways to deal with this equation. Widely used method in is so called *static condensation*: substituting the potential from the second equation

$$\Phi = \mathbb{E}^{-1}(\mathbb{P}\mathrm{U} - \tilde{\mathrm{Q}})$$

into the first equation to get one equation for the displacement,

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{K}^{\star}\mathrm{U} = \tilde{\mathrm{F}} + \mathbb{P}^{\mathrm{T}}\mathbb{E}^{-1}\tilde{\mathrm{Q}},$$

where

$$\mathbb{K}^{\star} = \mathbb{K} - \mathbb{P}^{\mathrm{T}}\mathbb{E}^{-1}\mathbb{P}.$$

4.2. Free oscillation

The core of the behavior of oscillating the piezoelectric continuum lies in its free oscillation (when external excitaion is zero). Free oscillations (and computed eigenfrequencies) tells, when the system under external excitation can get to the resonance. Let us assume the harmonic oscillations, therefore

$$\varrho\ddot{\mathrm{U}} = -\omega^2\mathrm{U},$$

where ω is the frequency of oscillation. There are two kinds of free oscillations of a piezoelectric system.

In the first case, the electrodes are open, and the eigenfrequencies of the system can be found by solving eigenvalue problem,

$$\begin{pmatrix} \mathbb{K} - \omega^2\mathbb{M} & \mathbb{P}^{\mathrm{T}} \\ \mathbb{P} & \mathbb{E} \end{pmatrix} \begin{pmatrix} \mathrm{U} \\ \Phi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{29}$$

Static condensation gives

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{K}^{\star}\mathrm{U} = 0.$$

The equation is similar to pure elastic case, only elastic matrix $\mathbb{K}^{\star}$ contains the term representing the electromechanical coupling. Eigenfrequencies can be found by solving the eigenvalue problem

$$(\mathbb{K}^{\star} - \omega^2\mathbb{M})\mathrm{U} = 0.$$

These eigenfrequencies are called *antiresonance frequencies*. In antiresonance frequency, the system oscillates with maximal impedance.

In the second case, electrodes are short-circuited, and for thin piezoelectric layers electric potential is zero in the whole volume ($\Phi = 0$). The problem reduces to the standard elastic oscillation case,

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{K}\mathrm{U} = 0.$$

Eigenfrequencies of the system can be found by solving the generalized eigenvalue problem,

$$(\mathbb{K} - \omega^2\mathbb{M})\mathrm{U} = 0.$$

Matrix $\mathbb{M}$ is positive definite (say of order n), so the problem has n eigenvalues and eigenvectors as solution. Frequencies $\omega_1, ..., \omega_n$ are called *resonance frequencies*. In resonance frequency, the system oscillates with minimal impedance.

4.3. Damped oscillation

We only mention here, that if we deal with structural damping of the piezoelectric material, the first governing equation extends of damping term,

$$\mathbb{M}\ddot{\mathrm{U}} + \mathbb{H}\dot{\mathrm{U}} + \mathbb{K}\mathrm{U} + \mathbb{P}^{\mathrm{T}}\Phi = \mathrm{F},$$

where $\mathbb{H}$ is the structural damping matrix

$$\mathbb{H} = \alpha\mathbb{M} + \beta\mathbb{K},$$

where α and β are positive numbers, $\alpha + \beta = 1$.

4.4. Static problem

For the static case, the problem reduces to solving the system of linear equations,

$$\begin{aligned} \mathbb{K}\mathrm{U} + \mathbb{P}^{\mathrm{T}}\Phi &= \mathrm{F}, \\ \mathbb{P}\mathrm{U} - \mathbb{E}\Phi &= \mathrm{Q}. \end{aligned} \tag{30}$$

5. Numerical solution

For discretization and compilation of the global matrix, we have developed our own code. For solving the eigenvalue problem (29), we use the procedures from the Lapack++, resp. Arpack++ library, available on the internet. From Lapack++ (see [3]), we use algorithm based on generalized Schur decomposition. These algorithms solve the complete eigenvalue problem. From Arpack++ (see [4]), we use algorithm based on shift-invert method combined with LU factorization. This algorithm, in contrast to Lapack++ code, solve the partial eigenvalue problem and deal with the fact, that matrices are sparse.

6. Remarks

The mathematical model for computing the resonance frequencies of the piezoelectric resonator has been built. The results of the described model approximate well the measured results for tested simply shaped (rod or slide) resonators. It seems that our model can have real application, e.g. in desining shape of the resonators vibrating with required frequencies. The testing results were presented at the last seminar. In last year, the corrections of the physical formulation was made and typic problems were established. Nowadays, the computer modules for computing with dense matrices work well, but there are still problems with including the modules for the sparse matrices (arpack++) to the main module. We also wait for the measured results of problem of oscillation of planconvex resonator. We propose to give some results at the seminar.

References

[1] Allik H., Hughes T.J.R., “Finite element method for piezoelectric vibration”, *International journal for numerical methods in engineering*, vol. 2, pp. 151-157, 1970.

[2] Brenner S.C., Scott L.R., "The mathematical theory of finite element methods", *Springer-Verlag NY*, 1994.

[3] Brepta R., Pust L. and Turek F., "Mechanické kmitání", *Technical Guide vol. 71, Sobotáles*, Praha 1994.

[4] Nečas J., Hlaváček I., "Mathematical theory of elastic and elasto-plastic bodies: An introduction", *Elsevier*, Amsterdam 1981.

[5] Míka S., Kufner A., "Parciální diferenciální rovnice I", *SNTL*, Praha 1983.

[6] Milsom R.F., Elliot D.T., Terry Wood S. and Redwood M., "Analysis and Design of Couple Mode Miniature Bar Resonator and Monolothic Filters", *IEEE Trans Son. Ultrason.*, Vol. 30 (1983), pp. 140- 155.

[7] Piefort V., "Finite Element Modelling of Piezoelectric Active Structures", *PhD. thesis, FAS ULB*, 2001.

[8] Rálek P., "Modelování piezoelektrických jevů", *Diploma thesis, FJFI CVUT*, Prague 2001.

[9] Rektorys K., "Variační metody", *Academia Praha*, 1989.

[10] Semenčenko V.K., "Izbrannye glavy teoretičeskoj fiziki", *Voennogo izdatelstvo Ministerstva oborony SSSR*, Leningrad 1960.

[11] Tiersten H.F., "Hamilton's principle for linear piezoelectric media", *Proceedings of IEEE 1967*, pp. 1523-1524.

[12] Štoll I., Tolar J., "Teoretická fyzika", *Czech technical university*, Prague 1981.

[13] Zelenka J., "Piezoelektrické rezonátory a jejich použití v praxi", *Academia Praha*, 1981.

Estimations of Cardiovascular Disease Risk

A survey of our results from 2004

Post-Graduate Student:
RNDR. JINDRA REISSIGOVÁ

EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

reissigova@euromise.cz

Supervisor:
PROF. RNDR. JANA ZVAROVÁ, DRSC.

EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

zvarova@euromise.cz

Field of Study:
Biomedical informatics
Classification: 3918V

Abstract

The aim of this paper is to present main results from the year 2004 of my postgradual doctoral (PhD) thesis *Estimation of cardiovascular disease risk based on data from epidemiological studies*. Framingham risk functions published in 1991 and 1998 were validated in population of the Czech Republic, namely in middle-aged men from Prague taking part in longitudinal study of risk atherosclerosis factors (STULONG). In STULONG, the calibration ability of those risk functions was not good, the discrimination ability was acceptable.

1. Introduction

The name of my postgradual doctoral (PhD) thesis is *Estimation of cardiovascular disease risk based on data from epidemiological studies*. One of main goals of my thesis is to verify estimations of cardiovascular disease risk used in the Czech Republic.

As you know from my presentation of the results in 2003 in Paseky nad Jizerou [7], cardiovascular diseases (CVD) are the main cause of death in developed countries. Take an example the Czech Republic, trend in age-standardized mortality in men shows Figure 1.

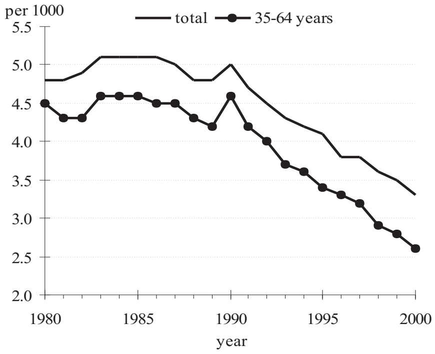


Figure 1: Directly age-standardized (world standard population, 1960) mortality from atherosclerosis CVD in men from the Czech Republic

The main and modifiable risk factors of CVD are cigarette smoking, hypertension, high blood fats, e.g. cholesterol, diabetes mellitus (DM), physical inactivity, obesity and overweight, see `http://www.who.int/cardiovascular_diseases/priorities/en/`. The mentioned risk factors can be controlled by changing lifestyle or pharmacologically treated. Therefore the aim is to identify high-risk CVD persons and to intervene their risk factors of CVD.

The statistical models are increasingly used to identify population at high risk. Based on epidemiological studies, statistical models estimating a individual's absolute risk for CHD event were derived [1], [2], [3], [6], [8], [9], [10]. The absolute CHD risk is the probability of developing a new CHD event within a given time period.

In the Czech Republic, one of the most used predictive models estimating the probability of developing CHD is the model of the Framingham Heart Study investigators. The aim of this paper was to verify the Framingham risk functions [1],[11] in 20-year lasting study of the risk factors of atherosclerosis (RFA) started in Prague in the year 1975.

2. Materials and methods

The longitudinal study of the risk factors of atherosclerosis (STULONG)

The design of STULONG has been already described in my report in 2003, see Figure 2 and read [7]. Briefly repeated, STULONG is the intervention prime preventive study with multiple risk factor intervention. The study was conducted by 2nd Dep. of Internal Medicine, 1st Faculty of Medicine and General Faculty Hospital, Charles University in Prague 2 in 1975–1999 (project leader prof. Boudík).

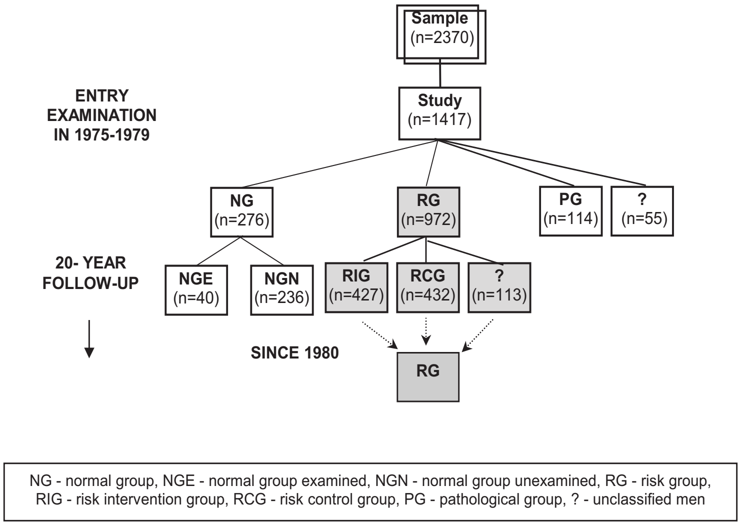


Figure 2: Design of the intervene prime preventive study of atherosclerosis (STULONG)

In 1975 total 2370 men aged 38–49 living in the 2nd district in the centre of Prague were randomly selected from list of electors. Of 2370 invited men, 1417 (59.8 %) men answered the invitation and underwent entry examination in 1975–1979. According to health status and occurrence of RFA (Table 1) at the entry into the study, each man was classified into one of three groups (normal, risk and pathological) differing in way of multiple risk factor intervention in the 20–year follow-up.

Normal Group (NG): men without any RFA, without CVD, without diabetes mellitus, without other serious disease not enabling long term follow-up and without pathological finding on ECG curve at the entry into the study.

Risk group (RG): men with at least one of RFA, without CVD, diabetes mellitus and other serious disease not enabling long term follow-up and without pathological finding on ECG curve at the entry into the study.

Pathological group (PG): men with CVD, diabetes mellitus or other serious disease not enabling long term follow-up or with pathological finding on ECG curve at the entry into the study.

Positive family history	death on the atherosclerotic diseases before the age of 65 years in the parents
Obesity	Brocca index (BI)$\geq$115 %, where BI=weight[kg]$/$(height[m] $-$ 100) $\cdot$ 100 %
Smoking	$\geq$15 cigarettes daily; or non-smoker less than one year and $\geq$15 cigarettes daily before
Hypertension	blood pressure $\geq$160 and/or 95 mmHg in two of three measurements; or hypertension in anamnesis
Hypercholesterolaemia	total cholesterol $\geq$260mg % (6.7 mmol/l)

Table 1: The risk factors of atherosclerosis at the entry into the study in1975–1979

The Framingham Heart study (FHS)

FHS is the prospective cohort study started in 1948 and continuing up to this day. The original objective of FHS was to identify the risk factors of CVD. For more information see `http://www.nhlbi.nih.gov/about/framingham/design.htm`.

Framingham risk function - 1991

In 1991, Weibull regression was used to develop the CHD risk function. The Framingham CHD risk function was derived from 2 590 men at the age of 30 to 74 years, who were free of cardiovascular disease (stroke, transient ischemia, CHD, congestive heart failure and intermittent claudication) at the time of examinations in 1971-1974 [1]. The Framingham function of

age [years],
systolic blood pressure (SBD – average of two office measurements) [mm Hg],
cholesterol (total serum cholesterol) [mg/dl],
high density lipoprotein cholesterol (HDL) [mg/dl],
smoking (1, cigarette smoking or quit within past year; 0, otherwise),
diabetes (1, diabetes; 0, otherwise) and
electrocardiography – left ventricular hypertrophy (ECG LVH) (1, definite; 0, otherwise)

was estimated to predict CHD includes angina pectoris, coronary insufficiency (unstable angina), myocardial infarction and sudden coronary death) developing within 4–12 years. There are some differences in the equation calculation of CHD risk for men and women. For men, the predicted probability (p) of CHD within t years is

$$p = 1 - \exp\left(-e^{u}\right), \tag{1}$$

where

$$u = \frac{\log(t) - \mu}{\sigma},$$
$$\sigma = \exp\left(-0.3155 - 0.2784 \cdot m\right),$$
$$\mu = 4.41818 + m,$$
$$m = a - 1.4792 \cdot \log(\text{age}) - 0.1759 \cdot \text{diabetes}\ ,$$
$$a = 11.1122 - 0.9119 \cdot \log(\text{SBP}) - 0.2767 \cdot \text{smoking} - 0.7181 \cdot \log(\text{cholesterol}/\text{HDL}) - 0.5864 \cdot \text{ECG LVH}.$$

Framingham risk function - 1998

In 1998, Cox proportional hazard regression model was used to derive the CHD risk function, see Table 2 [11], which estimates a 10-year individual's absolute risk of CHD (defined as in the Framingham risk function - 1991) for men. This model is based on 2489 men 30-74 years old at the time of their FHS examinations in 1971-1974. Similar model has been also published for women [11].

Variable	Relative Risk	95% CI
Age (years)	1.05	3.6
Blood pressure (mm Hg)[1]		
Normal including optimal (SBP$<$130, DBP$<$85)	1.00	Referent
High normal (SBP 130-139, DBP 85-89)	1.31	0.98-1.76
Hypertension stage I (SBP 140-159, DBP 90-99)	1.67	1.28-2.18
Hypertension stage II-IV (SBP$\geq$ 160, DBP$\geq$100)	1.84	1.37-2.49
Cigarette use	1.68	1.37-2.06
Diabetes (yes/no)	1.50	1.06-2.13
Total cholesterol (mg/dl)		
$<$200	1.00	Referent
200-239	1.31	1.01-1.68
$\geq$240	1.90	1.47-2.47
HDL-cholesterol (mg/dl)		
$<$35	1.47	1.16-1.86
35-59	1.00	Referent
$\geq$60	0.56	0.37-0.83

[1] When systolic (SBP) and diastolic (DBP) blood pressures fell into different categories, a man was classified into the higher category.

Table 2: Multivariate-adjusted relative risks for CHD, men

Statistical methods

For men from NG and RG the risk of coronary heart diseases were estimated according to the given Framingham risk functions from the year 1991 and 1998 (see above). The accuracy of the prediction of the Framingham risk function (1991, 1998) was measured by tests of calibration and discrimination.

Calibration of a model measured the degree of correspondence between the observed and predicted numbers of CHD events, and was tested by Hosmer-Lemeshow (H-L) chi-square goodness of the fit test.

Discrimination of a model measured the ability of model to distinguish observations with a positive or a negative outcome (CHD events). Discrimination was expressed by Receiver Operating Characteristics (ROC) curve. Area under ROC expresses how well the given model distinguishes between possible outcomes. Values vary between 0.0 and 1.0 with an area under ROC = 1.0 meaning that the model can perfectly distinguish between possible outcomes (fallen down with CHD vs. not fallen down with CHD).

When validating Framingham risk function 1991, the risk of CHD was estimated on the assumption that HDL is equal to 38.66 mg/dl (the level of HDL was not ascertained at the entry into the STULONG study). The accuracy of risk to predict CHD within 10-year period was evaluated by ROC curve and H-L chi-square goodness of the fit test. We have already presented Paseky nad Jizerou (25.-26. September 2003) and published the validation of the Framingham risk function - 1991 [7], in this work we precise our results.

When validating Framingham risk function 1998, the accuracy of risk to predict CHD within 10-year period was evaluated by the ROC curve, and H-L chi-square goodness of the fit test. Moreover, we compared the relative risks of CHD associated with the risk factors in FHS with those in STULONG. The relative risks in STULONG were estimated in the same way as in FHS, i.e. using the age-adjusted Cox proportional hazard regression model [11]. The equality of the relative risks between STULONG and FHS was tested by z-test [4]. Besides these methods, classification trees were used to analyze the association between the changes in the risk of CHD and the number of CHD events in time.

3. Results

First of all it is necessary to say, that we are going to publish the result of validation FHS risk function in international medical journals. According to their instructions the results were not allowed to be publish elsewhere. For this reason, in the lecture (in Paseky nad Jizerou, 29. September - 1. October 2004) we will present an overview of the results in detail, and here the results will be only briefly described.

Briefly summarized, the Framingham risk function 1991 was validated on base of entry examinations, the Framingham risk function 1998 on base of control examinations. The calibration ability of both Framingham risk functions was not good, the discrimination ability was around 70 %. The changes in the CHD risk (estimated according to Framingham risk function 1998) during the 5-year period influenced the number of CHD events in the next 5-year period.

4. Discussion

Coronary prediction estimate derived from a given population may not hold for another population. The Framigham risk function is based on subjects (almost all Caucasian, mostly Irish extraction) from the town of Framingham (a suburb west of Boston, USA). Although the authors recommend the model to use for individuals who resemble the study sample, the model is utilized for instance in European populations. However, if CHD incidence is much lower or higher in a given population than in that in Framingham, the Framingham risk model may be inappropriate [1]. Therefore, validation of CHD prediction models is the aim of epidemiological studies.

In this work we validated the Framingham risk functions published in 1991 and 1998. The difference between the numbers observed and predicted according to the Framingham risk functions was highly significant, the discrimination ability was acceptable. Our results may be influenced by the fact, that the risk functions used to estimate the risk of CHD assume the risk factors at the baseline remain constant over time. However, STULONG was a primary preventive study, and the risk profile of men taking part in our study may vary differently than the general population without the primary prevention. Despite these facts the number of CHD deaths in the 10-year period was significantly increasing with the risk score estimated at the baseline.

The Framingham risk functions published in 1991 overestimated the absolute risk of CHD in populations of Italian rural population, Western Europe, Israel and Germany. On the contrary the Framingham risk function 1998 underestimates the actual risk of CHD in Northern Ireland and France. The Framingham risk functions were recalibrated in some of those populations. After recalibrating for differing prevalence of risk factors and underlying rates of CHD events, the Framingham risk function estimates the number of CHD events close to those observed.

The study was supported by the project LN00B107 of the Ministry of Education of CR.

References

[1] K. M. Anderson, P. W. Wilson, P. M. Odell and W. B. Kannel, “An Updated Coronary Risk Profile, A Statement for Health Professionals”, *Circulation*, vol. 83, pp. 356–362, 1991.

[2] G. Assmann, P. Cullen and H. Schulte, “Simple Scoring Scheme for Calculating the Risk of Acute Coronary Events Based on the 10-year Follow-up of the Prospective Cardiovascular Munster (PROCAM) Study”, *Circulation*, vol. 105, pp. 310-315, 2002.

[3] R. B. D’Agostino, M. W. Russell, D. M. Huse, R. C. Ellison, H. Silbershatz, P. W. Wilson and S. C. Hartz, “Primary and Subsequent Coronary Risk Appraisal: New Results from the Framingham Study”, *American Heart Journal*, vol. 139, pp. 272–281, 2000.

[4] R. B. D'Agostino, S. Grundy S, L. M. Sullivan and P. Wilson, "Validation of the Framingham Coronary Heart Disease Prediction Scores: Results of a Multiple Ethnic Groups Investigation", *Journal of the American Medical Association*, vol. 11, pp. 180–187, 2001.

[5] S. M. Grundy , R. B. D'Agostino, L. Mosca, G. L. Burke, P. W. Wilson, D. J. Rader, J. I. Cleeman, E. J. Roccella, J. A. Cutler and L. M. Friedman, "Cardiovascular Risk Assessment Based on US Cohort Studies: Findings from a National Heart, Lung, and Blood Institute Workshop", *Circulation*, vol. 104, pp. 491-496, 2001.

[6] S. J. Pocock, V. McCormack, F. Gueyffier, F. Boutitie, R. H. Fagard and J. P. Boissel, "A Score for Predicting Risk of Death from Cardiovascular Disease in Adults with Raised Blood Pressure, Based on Individual Patient Data from Randomised Controlled Trials", *British Medical Journal*, vol. 323, pp. 75-81, 2001.

[7] J. Reissigová, "Validation of the Coronary Heart Disease Prediction", *Doktorandský den '03, Ústav informatiky, Akademie věd České republiky*, Matfyzpress, nakladatelství MFF UK v Praze, pp. 89–95, 2003.

[8] S. G. Thompson, S. D. Pyke and D. A. Wood, "Using a Coronary Risk Score for Screening and Intervention in General Practice. British Family Heart Study", *Journal of Cardiovascular*, vol. 3, pp. 301–306, 1996.

[9] T. F. Thomsen , M. Davidsen , H. I. T. Jorgensen , G. Jensen and K. Borch-Johnsen, "A New Method for CHD Prediction and Prevention Based on Regional Risk Scores and Randomized Clinical Trials; PRECARD and the Copenhagen Risk Score", *Journal of Cardiovascular*, vol. 8, pp. 291-297, 2001.

[10] H. Tunstall-Pedoe. "The Dundee Coronary Risk-Disk for Management of Change in Risk Factors", *British Medical Journal*, vol. 303, pp. 744-747, 1991.

[11] P. W. F. Wilson, R. B. D'Agostino, D. Levy, A. M. Belanger, H. Silbershatz and W. B. Kannel. "Prediction of Coronary Disease using Risk Factor Categories", *Circulation*, vol. 97, pp. 1834-1847, 1998.

Item Analysis of Educational Tests in System ExaME

Post-Graduate Student:
MGR. PATRÍCIA REXOVÁ
EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
rexova@euromise.cz

Supervisor:
DOC. RNDR. KAREL ZVÁRA, CSC.
Faculty of Mathematics and Physics
Charels University
Prague
zvara@karlin.mff.cuni.cz

Field of Study:
Probability and Mathematical Statistics
Classification: M4

Abstract

The paper contains an analysis of items of educational tests. The properties such as item difficulty, item discrimination or probability of guessing are studied within the framework of item response theory (IRT). Three methods for parameter estimation based on maximum likelihood (joint maximum likelihood (JML), marginal maximum likelihood (MML) and conditional maximum likelihood (CML)) are described. The asymptotic properties of these three estimators are mentioned. The possible connection with classical estimate of item difficulty is shown, using the Taylor approximation. Closeness of the classical estimates and the estimates based on logistic model is demonstrated on real data of the ExaME evaluation system, which is beeing developed by EuroMISE Center since 1998.

1. Introduction

Since 1998 the system ExaME for evaluation of a targeted knowledge is being developed [7]. The headstones of the ExaMe system are knowledge bases created for a specific target (mostly for knowledge covered by a special course). Each knowledge base consists of generalized multiple-choice questions (not limited number of answers, at least one answer is true, at least one is false).

The system is designed for two purposes:

1. for evaluation of a group of students in a computer classroom
2. for self-evaluation on remote places

For evaluation of a group of students, a teacher creates so-called fixed test by choosing appropriate questions and answers from the knowledge base. For student's evaluation on remote places, the system ExaME generates automatically so-called automated test.

The item analysis is important for both types of evaluation: When choosing the questions for the fixed test, the teacher is usually interested in item qualities, such as item difficulty, item discrimination power or probability of guessing the particular item. The teacher is also often interested in the reliability of the entire test (this was discussed in [5]). In remote-places evaluation, the system offers the student the possibility to specify the required difficulty of the test. That is why it is necessary that the system has estimated at least difficulties of all items.

1.1. Logistic model – Item Response Theory

In last decades an extensive theory for evaluation of item properties, called Item Response Theory (IRT) has been built. The theory is based on logistic regression and its fundamental component is the Rasch model, introduced by Danish statistician Georg Rasch [4]. In this model the probability of a correct response of person i on item j is given by

$$P(X_{ij} = x_{ij}; a_i, b_j) = \frac{\exp[x_{ij}(a_i - b_j)]}{1 + \exp(a_i - b_j)}, \tag{1}$$

where $x_{ij} = 1$ if the response of person i on item j is correct, and $x_{ij} = 0$ otherwise. In the model, a_i describes the level of ability of person i (either as an unknown parameter or as a random effect) and b_j is an unknown parameter describing the difficulty of item j.

A direct generalization of the Rasch model is the three parameter logistic model, where for each item two additional parameters are introduced: a discrimination parameter c_j and a guessing parameter d_j. The probability of a correct response of person i on item j is then given by

$$P(X_{ij} = x_{ij}; a_i, b_j, c_j, d_j) = d_j + (1 - d_j)\frac{\exp[x_{ij}c_j(a_i - b_j)]}{1 + \exp[c_j(a_i - b_j)]}. \tag{2}$$

Other extensions of the Rasch model are possible, too. Among these there are the extensions to polytomous models, such as the partial credit model, rating scale model, binomial trials and Poisson counts model. The majority of these models can be covered in generalized linear model. Other possible extensions are models for items in which response time or number of successful attempts are recorded. A well-arranged overview of extensions of the Rasch model can be found in [6].

An advantage of models containing more parameters is better description of the situation. A disadvantage is that with small sample sizes it may result in unstable estimates of item parameters.

1.2. Interpretation of item parameters

Another advantage of logistic models is a nice and clear graphical interpretation of parameters. Let's study the three parameter logistic model (2). If we define the item characteristic curve (ICC) of item j as $f_j(a) = P[X_{ij} = 1|a, b_j, c_j, d_j]$, by further analysis of this function we can easily see, that:

- If $c_j > 0$ then $f_j(a)$ is increasing (so that the better students are more likely to answer the item correctly), which is a reasonable assumption for an item.
- If $c_j > 0$, then $d_j = \lim_{a \to -\infty} f_j(a)$, therefore it describes the probability that person without any knowledge answers the item correctly.
- Difficulty parameter b_j can be understood as a value on the ability scale: If a person has ability $a = b_j$, then the probability that the person answers item j correctly is $\frac{1+d_j}{2}$, and so it is exactly in the middle between 1 and d_j.
- The first derivative of function f_j at point b_j is equal to $c_j\frac{1-d_j}{4}$, thus the discrimination parameter c_j is described by the slope of ICC in point b_j, more precisely it is equal to $f'(b_j)\frac{4}{1-d_j}$.

After estimating the parameters of an item, the item characteristic curve (see Figure 1) can be plotted out. Item characteristic curve describes the properties of an item very clearly: we can read easily its difficulty, the probability that persons with no knowledge answer it correctly. From ICC plotted on Figure 1 we can easily see that the described item can very well distinguish between the students with ability level between 0 and 2. On the other hand this item does not distinguish very well between students with lower ability level (nor between students with higher ability level).

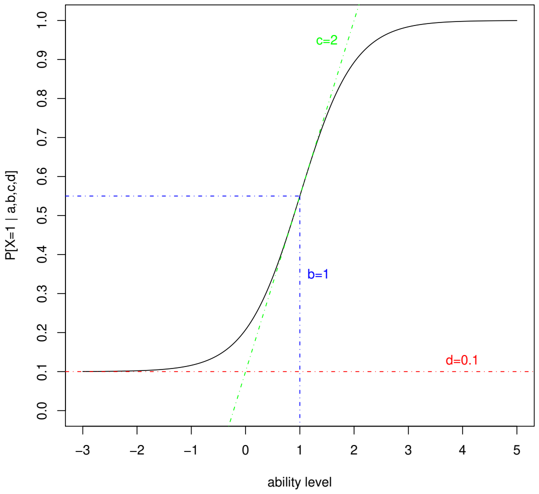


Figure 1: Item characteristic curve for an item with difficulty parameter $b_j = 1$, discrimination parameter $c_j = 2$ and guessing parameter $d_j = 0.1$

2. Estimation of item parameters

For the dichotomous Rasch model (1), there are available three likelihood based methods for item parameter estimation: joint maximum likelihood (JML), marginal maximum likelihood (MML) and conditional maximum likelihood (CML). The disadvantage of the logistic regression models is the fact that the estimation procedures for item parameters are hard to explain for non-statisticians. All the three algorithms based on maximum likelihood described in the next three subsections do use the iterative procedures. Connection of the classical estimator of item difficulty with the estimator based on logistic model is studied in the last, fourth subsection.

2.1. Joint maximum likelihood

Joint likelihood function for one-parameter Rasch model (1) is given by

$$p(\boldsymbol{x}; \boldsymbol{\omega}) = \prod_{i=1}^{n} \prod_{j=1}^{k} P(X_{ij} = x_{ij}; a_i, b_j), \tag{3}$$

with $\boldsymbol{\omega} = (\boldsymbol{b}^T, \boldsymbol{a}^T)$, $\boldsymbol{a} = (a_1, \ldots, a_n)$ representing the vector of abilities, $\boldsymbol{b} = (b_1, \ldots, b_k)$ representing the vector of difficulties of items and with $P(X_{ij} = x_{ij}; a_i, b_j)$ given by (1).

In the first method, the item parameters are estimated by maximizing (3) with respect to $\boldsymbol{\omega}$ given the data $\boldsymbol{x}$. As it was discussed already in [3], when keeping the number of item parameters and increasing the number of tested persons, this method leads to inconsistent estimators. This is caused by the fact that we have a problem in which a limited number of parameters of interest (item difficulties $\boldsymbol{b}$) are to be estimated in the presence of many nuisance parameters (abilities $\boldsymbol{a}$). Eliminating the nuisance parameters gives the solution for this problem. This elimination can be accomplished by the marginal or the conditional maximum likelihood method.

2.2. Marginal maximum likelihood

When estimating the item parameters using marginal maximum likelihood (MML) method, we usually assume that the abilities $\boldsymbol{A}$ constitute a random sample from an ability distribution with density $h(A; \boldsymbol{\xi})$, with $\boldsymbol{\xi}$ the parameters of the ability distribution. The joint probability can be then written as

$$p(\boldsymbol{x};\boldsymbol{b},\boldsymbol{\xi}) = \prod_{i=1}^{n} \int_{-\infty}^{\infty} \prod_{j=1}^{k} P(X_{ij} = x_{ij}|A_i; b_j) h(A_i;\boldsymbol{\xi}) dA_i, \tag{4}$$

with $P(X_{ij} = x_{ij}|A_i; b_j)$ again given by (1). Above mentioned so called marginal likelihood function is maximized with respect to $\boldsymbol{b}$ and $\boldsymbol{\xi}$. The nuisance parameters are eliminated by integrating them out. Often, the ability distribution is considered to be normal with unknown parameters μ_A and σ_A^2, which are estimated together with $\boldsymbol{b}$. Main problem of this method is the correct specification of ability distribution. If the distribution is not specified correctly, the method can lead to biased estimators of item parameters. The MML method can be used also without specifying a parametric form of the ability distribution. This nonparametric distribution is then estimated together with the item parameters. EM algorithm and MCMC method can be used for estimation.

2.3. Conditional maximum likelihood

The last approach to item parameter estimation is conditional maximum likelihood (CML) method. It results from the fact that if there exist sufficient statistics for the nuisance parameters, the model can be separated in a conditional part dependent only on the parameters of interest and a part which models the sufficient statistics. Since in the Rasch model (1) the total score $T_i = \sum_{j=1}^{k} X_{ij} = X_{i\bullet}$ is a sufficient statistics for $a_i,\ i = 1, \ldots, n$, the likelihood function (3) can be rewritten as:

$$p(\boldsymbol{x};\boldsymbol{\omega}) = \prod_{i=1}^{n} f(\boldsymbol{x_i}|t_i;\boldsymbol{b}) \prod_{i=1}^{n} g(t_i;\boldsymbol{b};a_i), \tag{5}$$

with $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ik})$ the response vector of person i. Maximization of the conditional likelihood

$$\prod_{i=1}^{n} f(\boldsymbol{x_i}|t_i;\boldsymbol{b}) \tag{6}$$

with respect to $\boldsymbol{b}$ leads under mild conditions to consistent and asymptotically normally distributed estimates (see [1]).

An interesting topic in CML estimates is their efficiency. The problem is that when estimating the item parameters, only the conditional likelihood (6) is used and the second part of the full likelihood (5), that is the marginal distribution of T, is neglected. Nevertheless, this second part could possibly contain some information on the item parameters. For evaluating the loss of information due to using the CML method, the F-information can be defined. This is a generalization of Fisher information matrix for the case when a part of the parameters is nuisance. The properties of F-information and the loss of information in CML estimation is in detail studied in [2].

2.4. Taylor approximation

When one is asked to estimate a difficulty of an item, probably the simplest thing he/she can think of is the proportion of correct responses to that item. In this section we would like to show that this classical estimator is justified and that it approximates the estimators mentioned above.

By maximizing the corresponding likelihood function, one can easily show, that the proportion of correct responses (more precisely $-\bar{x}_{\bullet j}$) is the maximum likelihood estimator (using any of the three above mentioned methods: JML, MML or CML) when considering the two-way ANOVA model

$$X_{ij} = a_i - b_j + e_{ij}, \tag{7}$$

with a_i ability level of person i, b_j difficulty of item j and $e_{ij} \sim \mathrm{N}(0, \sigma_e^2)$ a random error. The normality assumption is of course arguable, nevertheless, it is the item difficulty estimate that is of our interest, not the model itself.

Moreover, let's make a slight reparametrization of the Rasch model (1)

$$P[X=1|a_i,b_j] = \frac{e^{a_i-b_j}}{1+e^{a_i-b_j}} = \frac{e^{\mu+\alpha_i-\beta_j}}{1+e^{\mu+\alpha_i-\beta_j}} = f(\mu+\alpha_i-\beta_j), \tag{8}$$

with $\sum \alpha_i = \sum \beta_j = 0$. Let's consider the Taylor approximation

$$f(\mu+\alpha_i-\beta_j) \doteq f(\mu) + f'(\mu)(\alpha_i-\beta_j) = f(\mu) + f(\mu)(1-f(\mu))(\alpha_i-\beta_j) \tag{9}$$

Let's define $\eta = f(\mu) = \frac{e^{\mu}}{1+e^{\mu}}$ and

$$f(\mu+\alpha_i-\beta_j) \doteq \eta + \eta(1-\eta)(\alpha_i-\beta_j) \overset{def}{=} T_{ij}, \tag{10}$$

then the new joint likelihood function can be written as

$$L = \prod_{i=1}^{n}\prod_{j=1}^{k} T_{ij}^{x_{ij}}(1-T_{ij})^{(1-x_{ij})}. \tag{11}$$

Maximizing the new joint likelihood function with respect to η, α_i and β_j leads for β_j to the linear transformation of the classical estimator of item difficulty:

$$\hat{\beta}_j = -\frac{\bar{x}_{\bullet j} - \bar{x}_{\bullet\bullet}}{\bar{x}_{\bullet\bullet}(1-\bar{x}_{\bullet\bullet})}, \tag{12}$$

In this sense the classical estimator of item difficulty can be understand as a justified approximation of the estimator based on logistic regression.

3. Numerical example

The test on biomedical statistics of the ExaME system contains 12 items and it was given to 114 students, during the last three years. For each item we were interested whether the student answered fully correctly to that item ($x_{ij} = 1$) or not ($x_{ij} = 0$). When setting the item, the teacher had the possibility to evaluate subjectively its difficulty. The teacher assigned to each item a number from 1 (very easy item) to 3 (very difficult item). Besides the subjective estimates of difficulties, we estimated the difficulties using conditional maximum likelihood in the Rasch model and using the classical estimation (both of which were transformed using the linear transformation into the interval $\langle 1.00, 3.99\rangle$ so that the comparison with the subjective estimate was possible). For estimation, `clogit` and `lm` procedures of the software `R` were used. The 95% confidence intervals for estimates are given, too, the confidence intervals for item 25 are missing, because the estimates were considered fixed zero due to reparametrization conditions. For a better illustration, all the information was plotted in Figure 2.

item	25	26	27	28	29	30	31	32	33	34	35	36
subjective estimate	1.00	1.00	1.00	3.00	2.00	1.00	2.00	1.00	2.00	2.00	1.00	3.00
classical estimate	2.25	1.50	1.15	2.10	2.35	3.49	2.54	3.24	1.00	3.99	1.05	3.79
CI 95% lower		0.86	0.51	1.46	1.70	2.85	1.90	2.60	0.36	3.35	0.41	3.15
CI 95% upper		2.14	1.79	2.74	2.99	4.13	3.19	3.88	1.64	4.63	1.69	4.43
IRT estimate	2.19	1.50	1.16	2.05	2.28	3.38	2.46	3.12	1.00	3.99	1.05	3.72
CI 95% lower		0.88	0.52	1.45	1.67	2.73	1.85	2.49	0.36	3.29	0.42	3.06
CI 95% upper		2.12	1.80	2.66	2.89	4.02	3.07	3.74	1.64	4.69	1.69	4.39

Table 1: Estimates of item difficulties for 12 items of the test on biomedical statistics

As we can see, the subjective estimate often differs a lot from the other two estimates and thus it is worth estimating the item difficulty from the data. On the other hand, there is not a big difference between the classical estimate and the estimate based on logistic regression. The confidence intervals are very similar, too. Thus the classical estimate seems to be a good approximation of the estimate based on logistic regression.

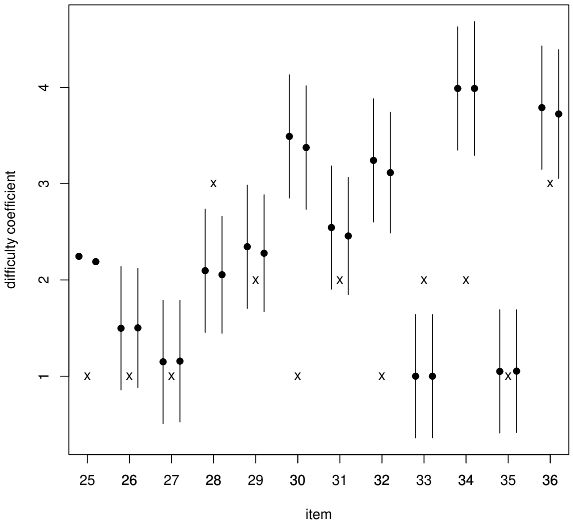


Figure 2: Estimates of item difficulties for 12 items of the test on biomedical statistics. Left: classical estimate, middle: subjective estimate, right: IRT estimate.

4. Discussion

The item response theory (IRT) based on logistic regression for item analysis of educational test was presented in this article. The three possible methods for item parameter estimation based on maximum likelihood were described: joint maximum likelihood, marginal maximum likelihood and conditional maximum likelihood. The asymptotic properties of these estimates were mentioned, which are, together with computational aspects of the three mentioned methods going to be a focus of a future author's research. The connection between the classical item difficulty estimator with the estimator based on logistic regression was given using the Taylor approximation, which gave a justification of the classical model.

Acknowledgement: The work was supported by the grant number LN00B107 of the Ministry of Education of the Czech Republic.

References

[1] Andersen E. B., ”Asymptotic Properties of Conditional Maximum-likelihood Estimators”, *Journal of the Royal Statistical Society*, Series B, vol. 32, No. 2, pp. 283–301, 1970.

[2] Eggen T. J. H. M., ”On the Loss of Information in Conditional Maximum Likelihood Estimation of the Item Parameters”, *Psychometrika*, vol. 65, pp. 337–362, 2000.

[3] Neyman J., Scott E. L., ”Consistent Estimators Based on Partially Consistent Observations”, *Econometrica*, vol. 16, No. 1, pp. 1–32, 1948.

[4] Rasch G., ”Probabilistic Models for Some Intelligence and Attainment Tests”, The Danish Institute of Educational Research, 1960.

[5] Rexová P., Zvára K., ”Reliability of Fixed Tests in the ExaME Evaluation System”, IJM Euromise 2004 Proceedings, Praha, 2004.

[6] van der Linden W. J., Hambleton R. K. , “Handbook of Modern Item Response Theory”, Springer Verlag, New York, 1997.

[7] Zvárová J., Zvára K., ”Evalulation of Knowledge using ExaMe Program on the Internet”, In: Iakovidis I, Maglavera S, Trakatellis A, eds. User Acceptance of Health Telematics Applications, Amsterdam: IOS Press 2000, pp.145-151 , 2000.

Rekonstrukce databázového modelu na základě dat (studie proveditelnosti)

doktorand:
ING. MARTIN ŘIMNÁČ

Katedra kybernetiky
Fakulta elektrotechnická

ČVUT Praha

rimnacm@cs.cas.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.

Ústav informatiky AV ČR
Pod Vodárenskou věží 2

182 07 Praha 8

stuller@cs.cas.cz

obor studia:
Databázové systémy
číselné označení: I

Abstrakt

Příspěvek[1] popisuje provedenou studii proveditelnosti databázově orientované části systému zajišťujícím automatickou extrakci dat z webových zdrojů (formáty XHTML, XML, CSV). Úkolem této části je transformace dat do automaticky vygenerovaného relačního modelu, který může být následně užit pro realizaci myšlenek sémantického webu.

V úvodní části je uvedena motivace pro implementaci takového nástroje. Součástí příspěvku je i částečné ohlédnutí za již implementovanými metodami, které autor v současné době zpracovává. V poslední části je nastíněna fuzzyfikace problematiky.

1. Motivace

Tak jako kola větrných mlýnů se nebudou točit bez větru, tak koncepce sémantického webu nebude přijata širokou veřejností bez relevantně využitelných informací v takovém rozsahu, jaký dnes nabízejí webové servery jak v podobě XHTML stránek, tak v podobě stažitelných dokumentů rozličných aplikací nebo různých webových služeb. Z toho důvodu je vhodné se zabývat nástrojem, který by pokud možno automaticky data z webových serverů získával a konvertoval je do strojově dále zpracovatelné podoby (např. relační databáze, XML, RDF). Součástí získávání dat může být i zahrnutí jejich dostupné sémantiky, zpravidla v rámci stránky vyjádřené pomocí formátování dokumentu.

Tato práce navazuje na diplomovou práci doktoranda [1], která se zabývala mapováním obecných webových prezentací. Základní mapovací jednotkou je webová stránka, výstupem algoritmu pak uspořádání stránek do stromové struktury. Jediným předpokladem tohoto algoritmu je strukturovanost webové prezentace. Současné úsilí hledá odpověď na otázku, zda-li lze efektivně mapovat i na nižší úrovni, než-li je webová stránka, tedy na úrovni strukturovaného obsahu stránky. Současné vyhledávací služby vracejí odkazy na stránky, které hledanou informaci obsahují. Je ale možné najít požadovanou informaci samu? Tím se dostáváme zpět k sémantickému webu. Lze tedy implementovat automatický nástroj, který by dokázal data najít, extrahovat a dále je prezentovat v kontextu jiných informací?

Praktickou motivací pro tuto úlohu je sledování časově proměnných veličin, např. cen různých počítačových komponent nebo vývoj devizových kurzů měn. Na základě takto získaných informací se můžeme ptát, který prodejce má nejvýhodnější služby, jaké jsou alternativy výrobků, jaké jsou trendy. A to bez striktní podmínky publikování informací jejich poskytovatelem ve formátu podporujícím paradigma sémantického webu. Nástroje využívající myšlenek sémantického webu navrhovanou metodikou mohou získat informace a dokáží prezentovat svoje přednosti. To může vést k všeobecnému přijetí této koncepce a budoucí moderní webové prezentace již budou "samozřejmě" zahrnovat i sémantiku prezentovaných dat.

---

[1] Práce byla částečně podpořena projektem 1ET100300419 programu Informační společnost (Tematického programu II Národního programu výzkumu v ČR): Inteligentní modely, algoritmy, metody a nástroje pro vytváření sémantického webu.

2. Současný stav problematiky

Tento příspěvek se zaměřuje na tu část problematiky, která se zabývá rekonstrukcí databázového modelu na základě vstupních dat. Tato úloha je v různých souvislostech řešena od zavedení relačních databází, první dílčí výsledky jsou publikovány od roku 1975 [2].

Poměrně velká pozornost byla v počátcích věnována sledování dotazů (příp. transakcí) [2, 3]. Na základě množin atributů, ke kterým bylo přistupováno v rámci jedné operace, byla statisticky vyhodnocována příbuznost atributů. Podle různých kritériích na hodnotu vzájemné příbuznosti atributů pak byly generovány množiny atributů, které byly sdružovány do relací. Tento způsob nezaručuje žádnou z normálních forem, spíše je využitelný pro fyzický návrh databáze a předzpracování (předpřipravení) dotazů.

Pro logický návrh jsou vhodnější metody analyzující závislosti mezi atributy. Jednotlivé závislosti mezi atributy mohou být znázorňovány pomocí hypergrafů [4]. Úkolem algoritmu je rozdělit relaci obsahující všechny atributy schématu do subrelací tak, aby tyto subrelace byly v požadované normální formě nebo splňovaly jiná kritéria. Metody lze rozdělit podle přístupu, buď přistupují shora dolů nebo zdola nahoru.

Přístup shora dolů spočívá v `dekompozici schématu`. V principu se algoritmus inicializuje jedinou relací obsahující všechny atributy schématu a tuto relaci testuje na podmínky specifikované normální formy [5] nebo na množiny různých druhů závislostí [6]. Pokud relace těmto podmínkám nevyhovuje, je rozdělena. Na dekomponované schéma jsou kladeny různé požadavky jako minimální redundance, reprezentativnost a separace [7].

Naopak přístup zdola nahoru vychází z funkčních závislostí a postupně `odstraňuje redundantní závislosti` vznikající díky jejich tranzitivitě (popsána dále). Odstraňování může být provedeno na základě analýz uzávěrů množiny atributů [8, 9] nebo při uvažování prvků těchto uzávěrů jako vzájemných podmnožin atributů [10].

3. Navrhovaná metodika

V současné době autor příspěvku analyzuje již navržené algoritmy v chronologickém pořadí a konfrontuje je s vlastní intuitivně navrženou metodikou, ke které byla provedena níže popsaná studie proveditelnosti. Cílem je najít algoritmus s přístupem zdola nahoru, který by bylo možné fuzzyfikovat a při rekonstrukci modelu uvažovat fuzzy-závislosti místo klasických závislostí.

Jako nevýhodu všech výše popsaných algoritmů můžeme označit fakt, že pracují se striktní definicí funkčních (příp. i jiných) závislostí, kterou některé závislosti v obecném případě na reálných datech nemusí splňovat. Předpokládejme tedy, že malé procento záznamů těchto dat danou funkční závislost nevykazuje. Výše uvedené algoritmy používají nefuzzy vstupy, tedy toto procento záznamů ignorují (čímž prakticky provedou defuzzyfikaci hned na svém vstupu) nebo docházejí k situaci neodpovídajícím schématům (uvažuje se pouze podmnožina skutečných závislostí). Alternativní přístup, kterým se autor hodlá zabývat, uvažuje fuzzy závislosti po celou dobu dekompozice a defuzzyfikace je provedena až na výsledném dekomponovaném schématu.

Podobně jako většina výše uvedených algoritmů omezme vstupní informace následovně:

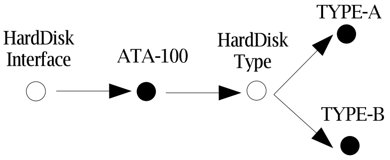
- Data budou interně uložena formou stromu atributů a jejich hodnot, která umožňuje uložit v databázi data, jejichž strukturu apriori neznáme. Tato reprezentace dat bude sloužit jako zdroj informací pro vygenerování relačního schématu.
- Data ve svém relačním schématu neobsahují cykly. Vylučujeme tak relace mezi stejnými entitními typy, např. relaci potomek. Tato podmínka vede na zjednodušení úlohy, některé úlohy vykazují pouze polynomiální složitost při acyklicitě [4].
- Žádné další informace nejsou k dispozici.

- Hodnoty atributů pro jednoduchost předpokládejme diskrétní.
- Jednoatributové primární klíče každé subrelace.

4. Integrace dat

Pro účely studie proveditelnosti byla použita podstatně zjednodušená verze grafového modelu sloužícího původně k integraci dat XML dokumentů [11].

- Uzly stromu jsou dvojího druhu, buď představují jméno atributu $attr_i$ nebo jeho hodnotu val_{ij}.
- Dvojice uzlů $(attr_i, val_{ij})$ je propojena orientovanou hranou.
- Všechny takové dvojice jednoho záznamu jsou hierarchicky propojeny tak, aby graf vykazoval stromovou strukturu.



Obrázek 1: Příklad struktury integrovaných dat

Kvalita integrace je dána počtem hran grafu vztaženou na počet uložených záznamů. Poměrně snadno lze ukázat, že počet hran je minimální, pokud posloupnost atributů $\{A_k\}$ je hierarchicky uspořádána tak, že

$$|D(A_i)| < |D(A_j)| \Rightarrow i < j \,. \tag{1}$$

Symbol $|D(A_k)|$ označuje počet prvků (diskrétních hodnot) domény k–tého atributu.

Takto provedené uspořádání atributů však nic neříká o vztazích mezi atributy, příp. o dekompozici atributů do databázového schématu a je tudíž pro získávání sémantických informací na základě dat nepoužitelné.

5. Závislosti atributů

Pro dekompozici relací mezi atributy použijeme definici funkční závislosti a využijeme některé vlastnosti těchto závislostí. Je užito značení podle [12]. Pro stanovení funkční závislosti používáme intenzivního přístupu.

5.1. Výklad základních pojmů

Definujme `funkční závislost` dvou atributů X a Y téhož entitního typu E s instancemi $R = \{r_k\}$. Říkáme, že atribut Y je závislý na atributu X (značíme $X \to Y$) právě tehdy, když

$$\forall r_i, r_j \in R : y(r_i) \neq y(r_j) \Rightarrow x(r_i) \neq x(r_j) \,. \tag{2}$$

V zobecněném případě pak můžeme hovořit o závislostech množin atributů.

$$\forall r_i, r_j \in R : \overline{y}(r_i) \neq \overline{y}(r_j) \Rightarrow \overline{x}(r_i) \neq \overline{x}(r_j) \,, \tag{3}$$

což v atomickém zápisu znamená

$$\forall r_i, r_j \in R : y_0(r_i) \neq y_0(r_j) \wedge \cdots \wedge y_m(r_i) \neq y_m(r_j) \Rightarrow x_0(r_i) \neq x_0(r_j) \vee \cdots \vee x_n(r_i) \neq x_n(r_j) \,. \tag{4}$$

Naopak atributy označíme za nezávislé (značíme $X \nrightarrow Y$), pokud

$$\exists r_i, r_j \in R, i \neq j : y(r_i) \neq y(r_j) \wedge x(r_i) = x(r_j)\,. \tag{5}$$

Atributy X a Y označíme jako vzájemně závislé (značíme $X \leftrightarrow Y$), pokud

$$X \leftrightarrow Y \Leftrightarrow X \rightarrow Y \wedge Y \rightarrow X\,. \tag{6}$$

Pro naše účely doplňme k těmto definicím ještě následující 2 tvrzení:

Transivita Nechť X, Y, Z jsou atributy entitního typu E. Pak

$$X \rightarrow Z \wedge Z \rightarrow Y \Rightarrow X \rightarrow Y\,. \tag{7}$$

Hierarchie Nechť $\overline{X}, \overline{Y}, \overline{Z}$ jsou neprázdné množiny atributů. Pak

$$\overline{Z} \subset \overline{X} : \overline{Z} \rightarrow \overline{Y} \Rightarrow \overline{X} \rightarrow \overline{Y}\,. \tag{8}$$

5.2. Testování funkčních závislostí

Předpokládejme, že ve výše popsané stromové struktuře máme uloženo celkem (C) záznamů a testujeme funkční závislost $X \rightarrow Y$. Z této struktury extrahujme všechny hodnoty atributu Y a po větvích stromu k nim nalezneme odpovídající hodnoty atributu X. Není–li odpovídající hodnota atributu X nalezena, uvažujeme, že atribut nabývá hodnoty NULL. Seřaďme nyní extrahované atributy podle hodnot atributu X a sekundárně pomocí atributu Y.

Označme e_i extrahované dvojice atributů. Pak případ, kdy je porušena funkční závislost, lze detekovat pomocí

$$x(e_i) = x(e_{i-1}) \wedge y(e_i) \neq y(e_{i-1})\,. \tag{9}$$

Počet takových záznamů označíme jako c. Abychom získali nefuzzy funkční závislost, defuzzyfikujeme takto otestovanou závislost, např. pomocí prahování počtu c záznamů (např. ve významu maximální přípustné chyby f):

$$\begin{array}{l} \frac{c}{C} < f \Rightarrow X \rightarrow Y \\ \frac{c}{C} > f \Rightarrow X \nrightarrow Y \end{array} \quad f \in <0, 1>\;. \tag{10}$$

Diskutujme výpočetní složitost testu. Mějme N atributů. Přijmeme zjednodušující předpoklad, že všechny záznamy popisují jednu relaci (mají shodné atributy). Pak extrakce hodnot atributů je složitosti $o(NC)$. Efektivní složitost je nižší díky stromovému uspořádání.

Druhou složkou je seřazení hodnot, uvažujme $o(C\log(C))$.

Poslední složkou je samotný test spočívající v průchodu všech záznamů a porovnání se záznamem předcházejícím, tj. složitost $o(C)$. Efektivní složitost může být podstatně nižší díky možnosti agregace shodných záznamů.

Celková složitost je dána součtem dílčích složitostí:

$$o(NC) + o(C\log(C)) + o(C)\,. \tag{11}$$

Jak je patrno, nejsložitější operací je extrakce hodnot. Proto je vhodné provést extrakci pouze jednou ale pro všechny dvojice atributů. Složitost testu všech dvojic:

$$o(NC) + N(N-1)(o(C\log(C)) + o(C)) = o(NC) + o(N^2C\log(C)) + o(N^2C) = o(N^2C\log(C))\,. \tag{12}$$

5.3. Matice závislostí

Při provedené studii se ukázalo vhodné zavést pojem `matice závislostí`. Nechť model obsahuje N atributů. Pak matice závislostí prvního řádu

$$M^1 = \{m_{ij}\}, m_{ij} = \begin{cases} -1 & A_i \rightarrow A_j \\ 1 & A_j \rightarrow A_i \quad i,j = 1..N\,. \\ 0 & jinak \end{cases} \tag{13}$$

Matice závislostí prvního řádu umožňuje dekompozici níže uvedených modelů závislostí.

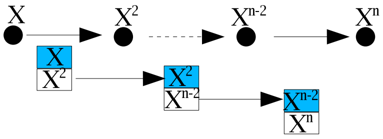
5.3.1 Model hierarchické závislosti: Tato závislost říká, že primární klíč je cizím klíčem předchozí relace, přičemž relaci tvoří dvojice (primární, cizí) klíč. Pro model formálně platí:

$$\forall i = 2..N : A_i \rightarrow A_{i-1}\,. \tag{14}$$

Pak díky transitivitě (7) platí, že

$$\sum_k m_{ik} > \sum_k m_{jk} \Leftrightarrow i < j\,. \tag{15}$$

Pokud vstupní data (s libovolným uspořádáním atributů) lze popsat pomocí modelu (14), pak tento model lze jednoznačně ze vstupních dat rekonstruovat na základě uspořádání atributů podle kritéria (15).



Obrázek 2: Příklad hierarchické závislosti

5.3.2 Model hierarchické závislosti se závislými atributy: Tento model vychází z předchozího modelu (14), avšak každý primární klíč je vzájemně závislý s jiným jedním atributem, který není závislý na žádném ze svých následníků. Model formálně popíšeme:

$$\forall i = 1..N/3 \quad \forall k > 3i-2 : A_{3i} \rightarrow A_{3i-1} \leftrightarrow A_{3i-2} \wedge A_k \nrightarrow A_{3i-1}\,. \tag{16}$$

Opět na základě transitivity (7) dokážeme, že

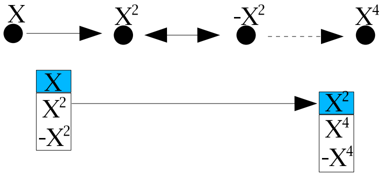
$$\sum_k m_{ik} > \sum_k m_{jk} \Rightarrow i < j\,. \tag{17}$$

Díky faktu, že

$$\forall s = 1..N/3 : \sum_k m_{ik} = \sum_k m_{jk} \Leftrightarrow i = 3s-1 \wedge j = 3s-2 \tag{18}$$

platí implikace (17) pouze jedním směrem.

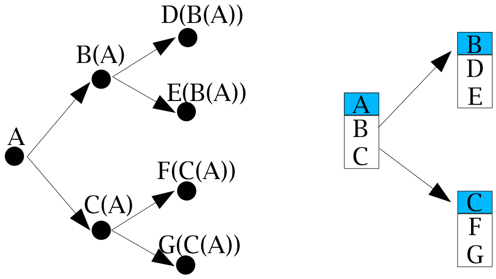
Určení primárního klíče není jednoznačné, protože k cizích klíčů nadřazené subrelace je navzájem závislých. Všechny cizí klíče nadřazené subrelace jsou rovnocennými kandidáty na primární klíč subrelace. Rekonstruovaný model je tedy jednoznačný až na určení primárního klíče podřízené subrelace.



Obrázek 3: Příklad hierarchické závislosti se závislými atributy

5.4. Modely s vícearitními závislostmi

Na základě předchozího odstavce se můžeme domnívat, že použitím podobných kritérií bude možné postupně rozšiřovat množinu modelů závislostí. Uvažme, že libovolná z relací má navzájem nezávislé cizí klíče. Pak sice selhává postup z minulého odstavce, kritérium však můžeme rozšířit o matice závislostí vyšších řádů (řád odpovídá aritě závislosti) a toto kritérium bude využívat vlastnosti hierarchických závislostí podle (8).



Obrázek 4: Příklad vícearitní závislosti

Selhání modelování se v tomto případě projevuje porušením podmínky

$$A_i \rightarrow A_j \Leftrightarrow i > j\,. \tag{19}$$

Tato podmínka platila ve všech předchozích modelech, avšak neplatí v případě, že relace obsahuje více nežli jeden cizí klíč.

5.5. Studie proveditelnosti

Na základě intuitivní myšlenky autora byla provedena studie proveditelnosti. Pro úplnost byla nastíněna i zcela původní myšlenka (čistě grafový přístup, odvození na základě počtu prvků domén jednotlivých atributů, odvozený z (1)), která však byla slepá, avšak poznatky z tohoto řešení lze parciálně využít pro snížení výpočetní složitosti a paměťových nároků na uložení ”surových” dat. Studie proveditelnosti poukázala na směr dalšího řešení problematiky.

Metodice můžeme vytýkat kombinatorickou explozi při testování vícearitních funkčních závislostí, avšak v tomto kontextu lze argumentovat podstatnou redukcí prohledávaného stavového prostoru. Navíc při použití fuzzy–závislostí je možné vícearitní funkční závislosti pouze odhadovat a testovat teprve v případě použití takové závislosti ve výsledném databázovém schématu.

Z uvedených dílčích výsledků můžeme usuzovat, že řešení této úlohy pomocí nastíněné metodiky má smysl. Při následné rešerži literatury se jeví perspektivní použití některých myšlenek jiných algoritmů, např. [10]. Zajímavým v tomto kontextu může být i nasazení genetických algoritmů na matici závislostí (13) při vhodně definovaném kritériu tak, aby nebylo nutné procházet NP–úplné testy na normální formu subrelace [5].

Během studie byly některé části implementovány na databázovém serveru PostGres, což přináší ve-dle ověření teoretických odvození i možnost testování na reálných datech. Tyto výsledky je možné zpřístupnit v rámci osobních stránek autora [13].

6. Budoucí práce

Během studie proveditelnosti bylo rovněž i experimentováno přímo s fuzzy závislostmi atributů, tedy bez provedení defuzzyfikace (10). Ty lépe vystihují závislosti mezi vstupními daty, která mohou být zatížena chybami nebo mohou být víceznačná.

Fuzzy míru příslušnosti můžeme formálně zavést jako procentuální vyjádření počtu záznamů (podle (9)), které splňují testovanou funkční závislost, tedy

$$\mu_{ij} = \frac{1-c}{M}, \text{ kde } M \text{ je počet všech záznamů, z nichž } 1-c \text{ splňuje } A_i \rightarrow A_j\,. \tag{20}$$

Matici závislostí pak modifikuje

$$\widetilde{M}^1 = \{\mu_{ij} - \mu_{ji}\} \quad \forall i,j = 1..N\,. \tag{21}$$

Tato modifikace umožňuje pracovat po celou dobu dekompozice z fuzzy–závislostmi. To vede k přesnějšímu popisu ze vstupních dat extrahované sémantiky, obzvláště vzhledem k intenzivnímu, daty orientovanému, přístupu ke generovanému schématu.

7. Závěr

Doktorand si klade za cíl provést detailní rozbor podobných metod a některou z metod fuzzyfikovat, případně použít metodiku novou (vyplývající ze studie proveditelnosti). Zajímavá bude konfrontace výsledků těchto metod právě s ohledem na extrahovanou sémantiku dat.

Tato metoda by měla korespondovat s rámcem sémantického webu. Předpokládá se, že bude implementován celý nástroj na získávání informací z webových stránek, příp. z jiných, veřejně přístupných, internetových zdrojů. Teoretické aspekty práce pak budou součástí disertační práce doktoranda.

Míra rozpracovanosti tématu odpovídá době necelých 3 měsíců, po kterou se autor danou problematikou detailně zabývá. Autor se snaží zohledňovat především praktickou část problematiky, o čemž svědčí i částečná implementace nástroje na základě dílčích teoretických výsledků.

Literatura

[1] M. Řimnáč, “Mapa webové sítě”, *Diplomová práce* Katedra řídicí techniky, FEL, ČVUT. 2004.

[2] J.A. Hoffer, D.G. Severance, “The Use of Cluster Analysis in Physical Data Base Design”, in *First International Conference on Very Large Data Bases*, pp. 69–86, 1975.

[3] B.N. Shamkant, R. Minyoung, “Vertical Partitioning for Database Design – A Graphical Algorithm”, in *SigMod*, pp. 440–450, 1989.

[4] G.Ausiello, A. D’Atri and M. Moscarini, “Chordality Properties on Graphs and Minimal Conceptual Connections in Semantic Data Models”, in *Symposium on Principles of Database Systems*, pp. 164–170. 1985.

[5] G.Grahme, K. Räihä, “Database Decomposition into Fourth Normal Form”, in *Conference on Very Large Databases*, pp. 186–196, 1983.

[6] M.A. Melkanov, C. Zaniolo, “Decomposition of Relations and Synthesis of Enitity–Relationship Diagram”, in *Entity-Relationship Approach Conceptual Modelling*, pp. 277–294, 1979.

[7] J.A. Hoffer, D.G. Severance, "A Sophisticate's Introduction to Database Normalisation Theory", in *Fourth International Conference on Very Large Data Bases*, pp. 113–124, 1978.

[8] J. Biskup, U. Dayal and P.A. Bernstein, "Synthesizing Independend Database Schemas", in *SigMod*, pp. 143–150, 1979.

[9] P.A. Bernstein, J.R. Swenson and D.C. Tsichristzis, "A Unified Approach to Functional Dependecies and Relations", in *SigMod*, pp. 237–245, 1975.

[10] D. Maier, D.S. Warren, "Specifying Connections for Universal Relation Scheme Database", in *SigMod*, pp. 1–7, 1979.

[11] D. Rosaci, G. Terracina and D. Ursino, "A Framework for Abstracting Data Sources Having Heterogennous Representation Formats", in *Data & Knowledge Engineering*, vol. 48, pp. 1–38. 2004.

[12] J. Ullman, "Principle of Database Systems", *Computer Sience Press*. 1980.

[13] M. Řimnáč, "Osobní stránka", *http://www.cs.cas.cz/∼rimnacm/* . [online].

Alternative Target Functions for Multilayer Neural Networks

Post-Graduate Student:
RNDR. MILAN RYDVAN
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
rydvan@cs.cas.cz

Supervisor:
RNDR. LADISLAV ANDREJ, CSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
andre@cs.cas.cz

Field of Study:
Theoretical Computer Science
Classification: I1

Abstract

This paper offers an overview of alternative target functions for multilayer neural networks, in particular biquadratic functions, relief error networks, genetically trained task-tailored functions and entropy-based functions. The alternative functions are described, suitable training algorithms are derived and the alternative functions are compared mutually and with the frequently used least square error function, using the problem of stock price prediction as the testing problem. This comparison shows that the proposed functions show better results and generalization abilities than the least square error function.

1. Motivation

This work deals with the model of *multilayer neural networks*. The model is widely known; the definition can be found for example in [5]. Multilayer neural networks employ supervised training, using a finite training set $T = \{(\vec{x}_i, \vec{d}_i)\}$ of pairs of input vectors and desired output vectors. The aim of training is to find such parameters of the network (weights, thresholds) that minimize a *target function* $E(d_{ij}, y_{ij})$, summed over all the output neurons and all the training patterns, where y_{ij} stands for the actual output of the network's j-th output neuron for the i-th training pattern. Because we will work with networks with a single output neuron, we will, for the sake of simplicity, omit the indices in the following text and use d and y when speaking about the desired and actual output of the output neuron for the particular training pattern. The theory can however be easily and intuitively extended to the case with more output neurons.

Rummelhart [5] proposed least square error function

$$E(d, y) = (y - d)^2 \tag{1}$$

as the target function for multilayer neural networks and it has been widely used till today. Its advantages include the fact that it is simple and natural. The fact that it penalizes the distance between the desired and the actual output makes it applicable, with a better or worse success, on all kinds of problems without requiring a specific knowledge about the character of the problem. Its graphs is shown in Figure 1.

This strong point of the least square error function is however also its weakness. The price for being so generally usable is that it cannot represent special knowledge we might have about the problem we are to solve.

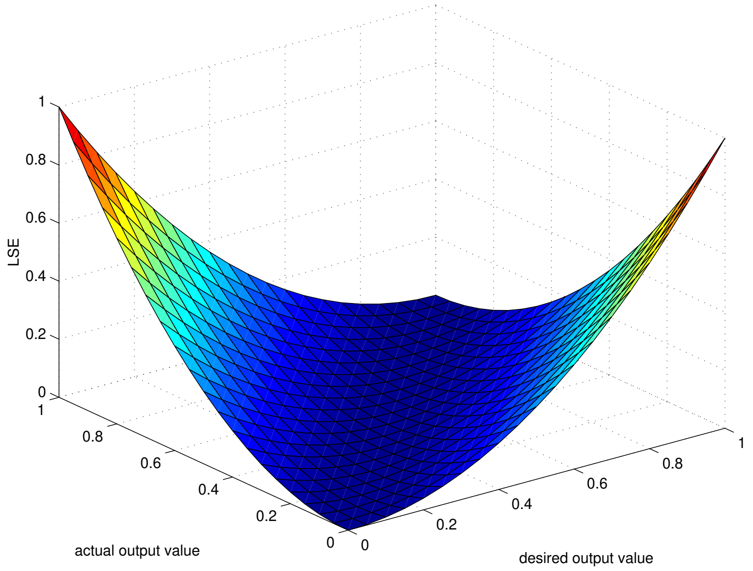


Figure 1: The least-square error function

The motivation for the first three alternative target functions therefore was to allow use of as non-restricted target function as possible, so that a function expressing the known specific features of the problem could be used. Three gradual steps towards this goal are presented in the following sections.

The motivation for the fourth proposed target function is different. The aim was to find and analyze an alternative to the least square error function sharing its advantage of general usability. Such an alternative was found, based on the cross-entropy function.

2. Biquadratic target functions

As we have said in the previous section, our aim is to use a target function that describes specific knowledge we have about the problem we are trying to solve. In the mentioned problem of stock price prediction, a suitable target function might be based on the profit a broker abiding by our prediction would achieve on the market, or rather on a model of this profit. A simple, yet rather realistic model follows:

$P = d - c$ iff $y > c$, (price rise prediction, recommendation to buy)

$P = -d - c$ iff $y < -c$, (price fall prediction, recommendation to sell)

$P = 0$ otherwise, (stagnation or small change prediction, no action recommended),

where y is the price growth/fall predicted by our system, d is the real growth/fall achieved on the stock exchange and c represents the transaction costs. A suitable target function would then be

$$E(d, y) = -P. \tag{2}$$

It is also natural as well, being based on the principle ”the higher is the profit, the smaller is the target (error) function”. Its graph is shown by the circled crosses in Figure 3. A problem of such a function is that it is non-continuous and therefore non-differentiable, while the most widely used training algorithms for multilayer neural networks (for example, the Back-Propagation) require the target function to be differentiable in y.

The first approach to solve this problem is to approximate the desired target function by a function that has the required property. We have chosen *biquadratic function* of the form

$$E(d, y) = A_2 d^2 + A_1 d + B_2 y^2 + B_1 y + C d y + D, \tag{3}$$

where A_2,..., D are constants. It is a natural generalization of the least-square error function (1). It is also easily differentiable, being a polynomial in both of its variables (an y in particular). It is therefore easy to derive and use the BP-training algorithm.

The values of the constants A_2,..., D can be determined by any interpolation method that fits a biquadratic function through the points of P for a selected grid of pairs (d, y). The graph of the resulting biquadratic target function for our testing problem of stock price prediction is shown in Figure 2.

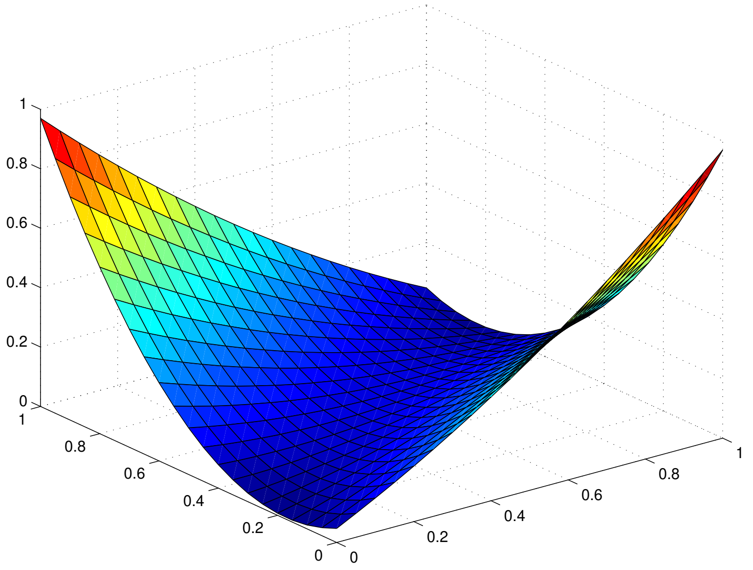


Figure 2: The biquadratic target function

3. Relief error networks

This section describes another approach to the solution of the problem mentioned in the previous section. It was developed in cooperation with Iveta Mrázová, for details see [4].

In order to avoid restraining ourselves to a single type of the approximating function, we will use another neural network, called *relief error network* (REN), to approximate the desired target function, for example (2). The relief error network will treat the actual and desired outputs of the main network as its inputs and the corresponding error values as its desired output. It can be then trained in a standard way, using e.g. the BP-training algorithm. It is necessary to pay proper care to the approximation and generalization abilities of the relief error network, because it will be then used for training of the main neural network. Figure 3 shows the graph of the target function produced by the relief error network trained to approximate the profit-model function (2).

The trained REN is then added modularly to the main neural network. For each training pattern of the main network, the REN uses the main network's actual output and desired output for this pattern as its inputs, and computes its own output, i.e. the main network's target function value for this pattern.

A problem that remains to be solved is training of the main neural network. It is however simplified by the fact that the target function produced by the REN is analytical - continuous and differentiable - under

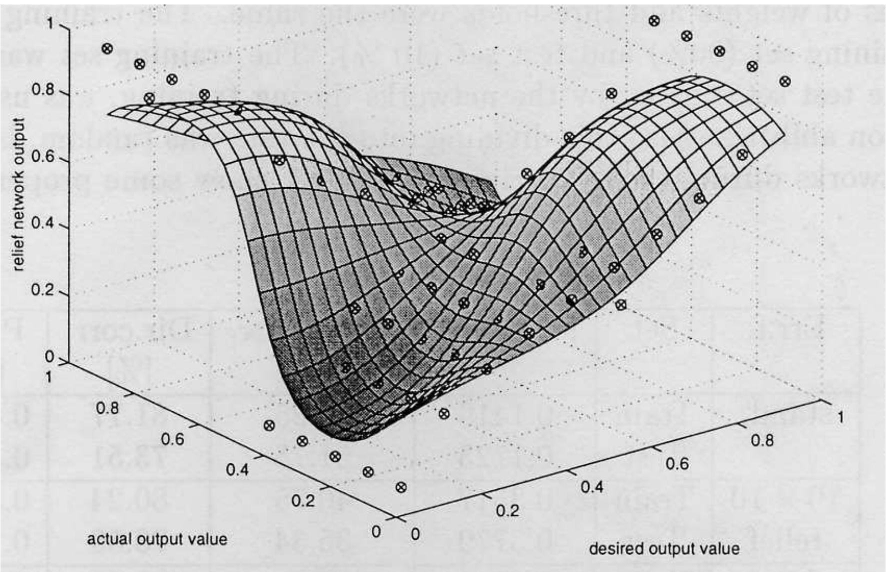


Figure 3: The target function, computed by the trained relief error network. The crosses represent the grid of the training patterns, their height then the desired error value

the condition that such transfer functions are used in the REN. That is fulfilled for example the standard sigmoid transfer function (6). Moreover, if the standard sigmoid is used also for the REN's output neuron, the REN's values (target function values for the main network) are bounded in $(0, 1)$ and have their minima close to 0, which can be useful during the training of the main network.

In order to train the main network using the REN, we can apply the idea of the Back-Propagation, derived in [5]. The computing of the error terms begins in the highest layer of the relief error network, and continues downwards, through both the REN and the main network, in accordance with the Back-Propagation principle. The weights and thresholds are adapted only in the main network; the REN is already trained and its parameters remain unchanged.

4. Arbitrary target functions

The logical final step in the effort to make the target function as unrestricted as possible is to use the desired function itself (in our testing problem of stock price prediction for example the profit-modelling target function (2)) rather than any of its approximation. The problem is that such a function may generally be non-differentiable. The gradient training methods, such as the Back-Propagation, therefore cannot be used.

A solution to this problem is to use training methods that do not pose such (and preferably any) requirements on the target function. One of these methods is the use of genetic algorithms. Genetic algorithms (see for example [3] for more detailed information) perform distributed cooperating search in the solution space. Each prospective solution is coded in the form of a chromosome, a string of one-bit, two-bit or real values. Each chromosome is assigned a *fitness*, reflecting how suitable the corresponding solution is. The GA maximizes the fitness using genetic operators on a population of chromosomes. *Selection* ensures the overall improvement of the fitness, *crossover* combines schemes in existing individuals in order to create new patterns in new individuals and *mutation* makes random modifications, helping the system to produce new schemes and avoid local minima.

When training neural networks using GAs, the chromosome can consist of real-valued genes, each representing a single parameter of the network — a weight or a threshold. The fitness of such chromosome-network is the negative value (because GAs maximize fitness) of the target function applied on the network and the training set, divided by the size of the training set, i.e. the average target function value per training pattern.

Genetic training of neural networks usually has several drawbacks compared to gradient methods — it tends to be slower and its results are worse. On the other hand, it does not suffer from the local minima problem so much. However, the main benefit genetic training has for our task is that it allows usage of unrestricted target functions, which with a suitable use of the target function can balance or even outperform the drawbacks of genetic training.

5. Entropy-based target function

The last proposed alternative function represents a different way of research. It is an alternative to the least square error function (1) that is also usable generally, without a prior problem-specific knowledge. To propose such a function, we have used the notion of entropy.

Entropy is a quantity originating in thermodynamics, describing the measure of disorder in a system. In other words, it therefore means that it describes also the measure of information contained within a system. We will use this fact when applying an entropy-based function as a target function for neural networks.

The target function we propose is based on the *cross-entropy function*:

$$E_c = \left(d \ln \frac{d}{y} + (1-d) \ln \frac{1-d}{1-y} \right), \qquad d, y \in (0,1). \tag{4}$$

In order to be able to use this function as a target function for neural networks trained by the Back-Propagation training algorithm, we need to compute its derivative according to y:

$$\frac{\partial E_c}{\partial y} = d \cdot \frac{y}{d} \cdot \frac{-d}{y^2} + (1-d) \cdot \frac{1-y}{1-d} \cdot \frac{1-d}{(1-d)^2} = \frac{1-d}{1-y} - \frac{d}{y}. \tag{5}$$

We can see that $E_c = 0$ if and only if $d = y$; the minimum of E is located in these points. For the graph of E_c, see Figure 4.

6. Stock-price prediction

We have compared the performance of the proposed target functions with the least-square error function on the problem of stock price prediction. The aim was to predict the stock price change on the following trading day, knowing a history of (five) previous price changes, plus additional information about the previous day's trading, such as the volume of trade, the position of the latest known price in the long-term history, the supply/demand ratio etc. The raw data from the stock exchange were subject to extensive pre-processing. It was for example necessary to transform the outputs of the network, which represent the expected price change, i.e. generally a real number, into the interval $(0,1)$, because it is required by the target function (4). This was achieved by transformation using the sigmoidal function

$$\bar{y} = f(y) = \frac{1}{1+e^{-y}}. \tag{6}$$

Among other pre-processing methods there was e.g. application of the Principal Component Analysis (see [2]) on the data, which normalizes them and therefore increases the performance of training.

We used Matlab as the platform for programming the experiments. Matlab's Neural Network toolbox was used, together with the author's implementation of the alternative target functions. In order to implement

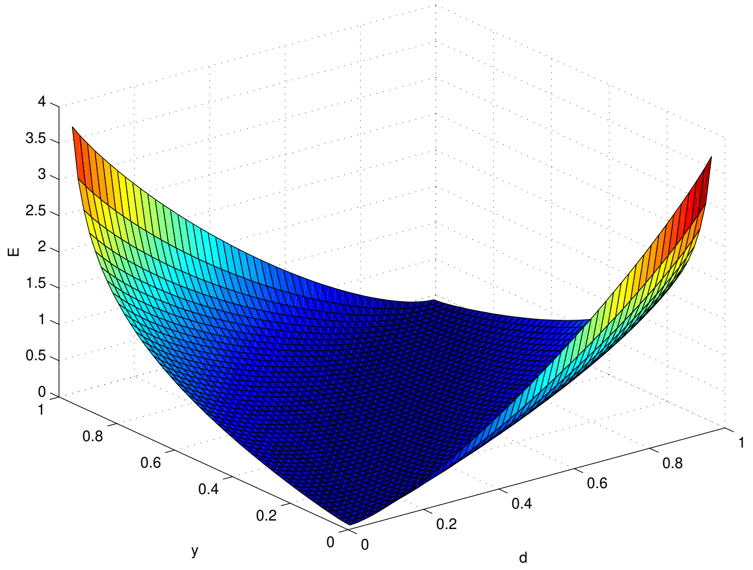


Figure 4: The cross-entropy target function

genetic training, we have interconnected this toolbox with a GA toolbox developed by Houck, Joines and Kay ([1]). Series of experiments were carried out, in order to determine and tune the parameters of the tested methods. We used the same architecture (9-15-1) for the standard least-square error function and for each of the alternative target functions, in order to keep the conditions of the compared models as similar as possible.

The networks using the least square error function (1) and the biquadratic target function (3) were trained using the Back-Propagation training algorithm, using the learning rate 0.01. The number of the training cycles was limited to 1,000 unless a chosen target function value was reached earlier. The maximum number of training cycles was however rarely needed; the target function limit was usually reached sooner.

A series of examples was used also to determine the best architecture, training set and training parameters of the relief error network. A set of 121 training patters forming a 10x10 grid in the desired/actual output space, architecture 2-5-1 and a learning rate of 0.1 have produced the best results. The learning rate for the main network was again 0.01.

For the genetic training of the profit-based target function (2) we achieved the best results with a population of 1000 chromosomes, using the normalized geometric ranking, simple one-point crossover and uniform mutation as the genetic operators. Using normalized geometric ranking, the probability of selecting the i-th individual from the population equals

$$P_i = \frac{q}{1-(1-q)^P}(1-q)^{r-1},$$

where r is the rank of the i-th individual according to the fitness, q is the probability of selecting the best individual and P is the population size. The parameter q was set to 0.08. Simple crossover just randomly selects a point in the chromosomes of the parents and creates the offspring by exchanging the parents' genes located rightwards of the position. Uniform mutation randomly selects one gene and assigns it a uniform random number from the permitted space of values (interval $(-10, 10)$ was used as the permitted space for the gene values).[1] The probabilities of crossover and mutation were 0.5 and 0.2, respectively. The evolution continued until the best individual reached the fitness of 0.43 or for 200 generations.

---

[1] For detailed definition of the mentioned genetic operators, see [1].

Finally, let us deal with the cross-entropy target function (4), trained using the Back-Propagation training algorithm. This target function has shown to be very sensitive on the learning rate. The reason is that $\lim_{y \to 0} \frac{\partial E_C}{\partial y} = -\infty$ and $\lim_{y \to 1} \frac{\partial E_C}{\partial y} = \infty$ (see (5)), which causes extreme and possibly diverging changes of the network's parameters in these cases. Values $y = 0$ and $y = 1$ after pre-processing of the data represent infinite slump and growth of the stock price, respectively, (see (6)) and similar extreme values therefore should not appear in a trained network. They may however appear in a "newborn“ network that has been created randomly and has not undergone much training yet. This problem may be solved by applying a very low learning rate α. This however makes the training process very slow and increases the risk of getting stuck in a (very) local minimum. Therefore, we have chosen a method of variable learning rate. At the beginning of the training, when the chance of extreme values of y is larger, α is low ($2.5 \cdot 10^{-4}$). It is then doubled twice, after 100th and 200th iteration of the Back-Propagation, when it thus reaches $1 \cdot 10^{-3}$.

In order to estimate and compare generalization abilities of the methods, we divided the known data into a *training set*, which was used during the training period, and the *test set*, unseen by the networks during training and used for measuring their performance on unknown data. The training set contained 75% of the data; the test set contained the remaining 25%.

We compared the proposed methods by carrying out 100 experiments. During each of them, five networks were trained - one using the standard least-square error function and one using each of the proposed alternative target functions. Table 6 describes the averaged results both on the training set and on the test set. The division into the training/test set was carried out randomly for each of the 100 experiments; in each experiment it was however the same for all five target functions tested.

Target function	Set	Square error	Dir. corr.	Profit
LSE	Train	0.033	81.8%	0.472%
	Test	0.050	73.1%	0.164%
BIQ	Train	0.069	78.3%	0.377%
	Test	0.087	73.7%	0.211%
REN	Train	0.159	79.1%	0.331%
	Test	0.178	72.7%	0.177%
PROFIT	Train	0.167	79.9%	0.353%
	Test	0.190	72.6%	0.175%
ENTR	Train	0.041	76.4%	0.322%
	Test	0.049	72.7%	0.184%

Table 1: Comparison of performance of the least-square error function and of the proposed alternative target functions on the problem of stock price prediction, separately for the training set and the test set. Several measures of success are presented - the summed-square error, the direction correctness (the percentage of correct prediction of price rise/decrease) and the modelled daily profit.

The test set results suggest that the proposed alternative target functions have outperformed the standard least-square error function in terms of the most decisive criterion - the model of the achieved profit. The direction correctness (i.e. the success rate showing how often they predict the trend correctly) is roughly the same for all the target functions used. Finally, measured by the summed square error, the standard error function is better than the profit-based target functions, which is however not surprising - minimizing the square error was not their task. What is interesting is that the cross-entropy target function outperformed the standard least-square error function even in this criterion, even though it was not its task, either.

The comparison between the training and test set results says that their difference is smaller in the case of the alternative target functions, which suggest that their generalization ability might be better and their tendencies to get overtrained lower. This is most visible in the case of the cross-entropy and biquadratic target functions, which outperformed the standard error function in two out of the three used measures of success on the test set, even though their results on the training set were worse.

Let us say a few words also with the speed of the training process. The cross-entropy target function and the biquadratic error function were fastest — they needed a low number of training cycles (144 and 170 in average, respectively) and each cycle was rather quick, thanks to the simplicity of the target functions, resulting in quick and simple computation of the weight/threshold changes. The standard error function was placed third; its training cycles were quick, too, but the training needed a higher number of the cycles (570). The use of the relief error network was slower, because computation of the target function during presentation of each training pattern requires to run a neural network. This caused that despite the not so high number of required training cycles (246 in average), the training time was longer. The training of the non-approximated profit function was the slowest, because of the character of genetic training — the target function summed over the whole training set must be computed for each network/individual in each generation, and the number of the individuals and of the populations was rather high.

7. Conclusion

Out of four proposed alternative target functions for the multilayer neural networks, two have achieved better results than the standard least-square error function in a shorter training time and the other two have outperformed the standard function, too, even though their training was slower.

The profit-based target functions suggested that it is possible to incorporate problem-specific knowledge into the training process using the target function. On the other hand, the cross-entropy target function proposes an alternative to the least-square error function that does not require such knowledge and yet speeds up and improves the training process. The proposed alternative target functions also seem to have better generalization abilities.

The results presented in this article show that studying target functions of neural networks and proposing alternatives can improve the results of their training. Two paths were suggested. The first one leads towards problem-tailored target functions, which may often have non-analytic forms and will require approximation or special training algorithms, but which are capable of expressing the specific knowledge we may have about the problem. The second path leads towards generally usable target functions that will possibly have better properties that the standard error function and yet are applicable on most of the problems that are being solved using multilayer neural networks. Both paths seem to be passable.

References

[1] Chris Houck, Jeff Joines and Mike Kay, *"A Genetic Algorithm for Function Optimization: A Matlab Implementation"*, NCSU-IE TR 95-09, 1995, see `http://www.ie.ncsu.edu/mirage/#GAOT`.

[2] T. Masters, *"Advanced Algorithms for Neural Networks"*, John Wiley & Sons, 1995.

[3] M. Mitchell, *"An Introduction to Genetic Algorithms"*, MIT Press, 1997.

[4] I. Mrázová, M. Rydvan *"Relief Error Networks for Economic Predictions"*, Proceedings of conference Nostradamus 99 (323), (1999), pp. 135-140.

[5] D. E. Rummelhart, G. E. Hilton and R. J. Williams, *"Learning representations by backpropagating errors"*, Nature (323), (1986)

Short Survey on Bioinformatics with Fuzzy Logic

Post-Graduate Student:
MGR. MARTIN SATURKA
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
martin.saturka@cs.cas.cz

Supervisor:
PROF. RNDR. PETR HÁJEK, DRSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
hajek@cs.cas.cz

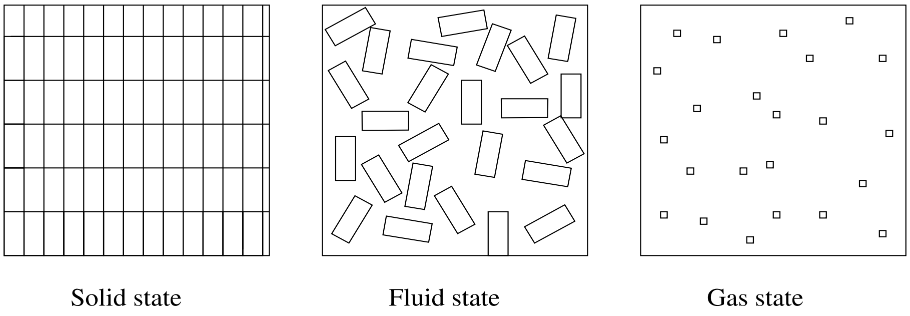
Field of Study:
Theoretical Computer Science
Classification: I1

Abstract

We survey parts of bioinformatics theory with respect to DNA chip microarray data analysis. First, we outline information structures and bioinformatics itself. Next to it, we describe so called fuzziness and we show generalized logical connectives which are usable for data preprocessing and structuring. Finally, we describe several classes of aggregative operators.

1. Introduction

There are three main structural states of physical matter with respect to its organization: solid, liquid and gas phases. Their classical representants are ideal crystal, ideal fluid and ideal gas respectively. They are depicted at Figure 1. While crystal has fixed and regular structure, gas has random and dynamical structure. Organization of fluids lies between the too extremes. In case of solid state, we deduce all the structural properties of the matter from just one point of it. In case of liquid state, our deduction is limited to a bounded region. We can not deduce anything on distinct parts of matter in case of gas state.



Solid state Fluid state Gas state

Figure 1: Organizational states of matter

Sometimes, live matter (i.e. organisms) is put in line with fluids. It seems to be rational, since both structures are partially regular. However, there are some controversies. First, organisms are not just spread fluid matter. Second, there are several patterns for the "middle" setting. We sketch three possible structures at Figure 2. The case A is for partially sublimated matter - if we are lucky, we can deduce investigated properties to large part of the matter. However, in adverse situation, we can not deduce at all. The case B is for fluids and they were mentioned above. The case C is for so called organismal matter. We can deduce just small amount of matter properties from one point knowledge. However, as we investigate more points in the matter, we can deduce much more - not just on bounded surroundings of the investigated points. It is usually the case

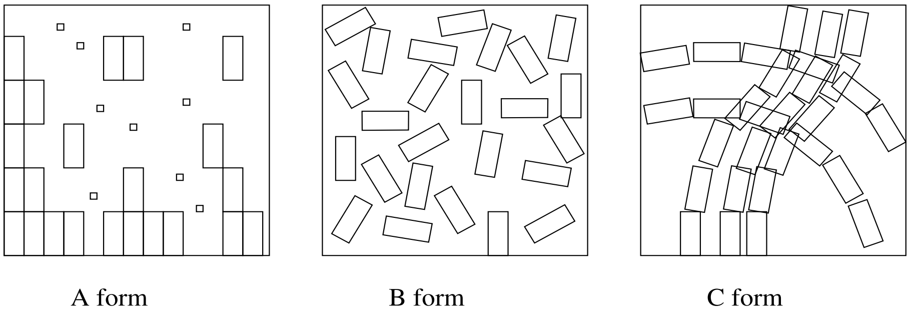


Figure 2: Fluid-like forms of organization

we assume to be the interesting one. And we believe, it is the case for organisms. Nevertheless, we do not say that it is specific property for living organisms.

Features being dashed at Figure 2, case C, are covered inside investigated matter. They characterize particular objects, but we do not know the features a priori. The task is to unravel the features. Since the features are too complex and diversed to be covered by a few formulas, we try to spring them by data mining methods. Usually, our work is separated into three parts. First, theoretical algorithms have to be invented. Second, we have to implement the algorithms into software. Third, programs are used on biological data. We focus to the first part in this survey. Especially, we concentrate on use of Hájek's observational calculus and fuzzy logic.

2. Fuzzy logic and bioinformatics

Fuzzy logic [2] is fruitful of structures which can be used for data mining. Unfortunately, the word of "fuzzy" is used for many different ideas [6]. First, we use the notion of fuzzy as is formalized in mathematical fuzzy logic: i.e. logic of comparable truth values. Second, bioinformatical data [1] we focus on, have their values in real intervals. It means that value e.g. 0.5 is for actual half-large variable. For example, one variable can be age: people can range from very young (value $\approx$ 0.1), somewhat young (value $\approx$ 0.3) to very old (value $\approx$ 1.0) ones, see example at Figure 3.

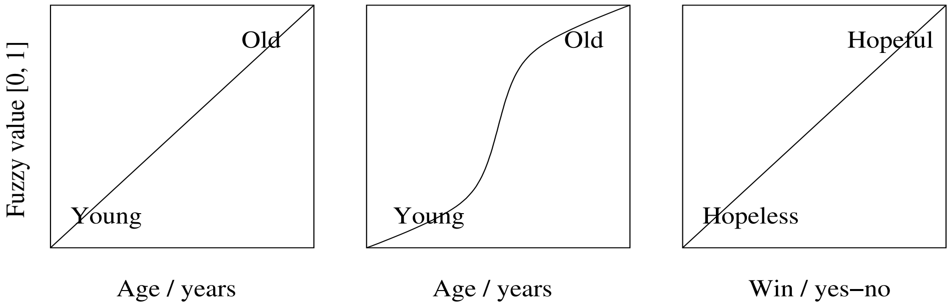


Figure 3: Different meanings of fuzziness

It is not necessary to have linear dependence of a fuzzy value on the real quantity. In case of bioinformatical data, the dependence frequently contains logarithmical transformation. One reason for it is gaining distribution of data which is more symmetrical and normal like.

Contrary to the above case of real continuous data, there are situations with crisp (i.e. two valued - yes/no) data when the meaning of fuzziness is used too. For example, the crisp variable can be a win in a future with its fuzzy value expressing the chance or our hope to win, see at Figure 3. Fuzzy variables which are used for description of such situations, are just measures of probability or believe that investigated crisp data occur. It is notable to say that we do not use fuzziness for such two valued data since one just expresses value of uncertainty there.

We develop methods for biological data that can usually have their values greater or lesser than a middle value. This is motivated by gene expressions. Values of expression are by default viewed as either being in a middle region or altered ones. In case of alteration, the values can be greater (i.e. activated expression) or lesser (i.e. inhibited expression).

Some common examples can be temperature or favor of cup of tea. In case of cup temperature, the tea can have middle temperature - it is neither warm nor cold, it can be cold, it can be warm. Likewise, the tea favor can be as negative (dislikes), neutral or positive (likes). It is shown at Figure 4.

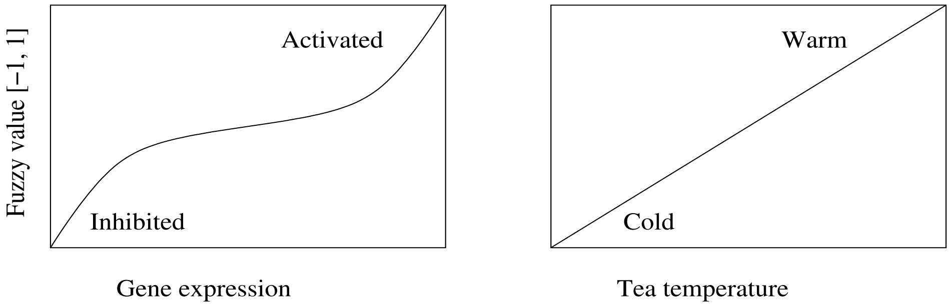


Figure 4: Twofold value alteration

It is natural to use interval of $[-1, 1]$ to express such values. In fact, we use pairs of values for it. It can be gained by usage of generalized logical connectives. The new connectives, say plications, are extension to implication and coimplication as uninorms are extension to t-norms and conorms. It means that in case of a plication, say $P(x, y)$, it generally holds neither $P(x, y) = 1$ for $x \leq y$ nor $P(x, y) = 0$ for $x \geq y$. The new connectives can be used not only for pairs of values on single properties, but they can be reused for general pairs. In such a case, they can express time changes. It is useful tool for time series data and it plays role of time differentials.

Together with it, we reuse principles invented as monadic observational predicate calculus [3, 4]. It has two subsequent parts. Particular measured properties are used as logical formulas and they are combined by logical connectives. Next to it, generalized forms of quantifiers are evaluated on pairs of formulas to check their connections. It can be viewed as counting on a relational table:

	var 1	var 2	...	var M
obj 1	0.3	0.8		0.2
obj 2	0.5	0.7		0.9
...				
obj N	0.1	0.5		0.4

The exemplary table above shows starting point for observational calculus (on fuzzy data) computing. Separate columns are for particular variables, for example genes or cups of teas. Separate rows are for particular objects, we measure the variables on. They can be patients or drinkers. Filled values (set into interval $[0, 1]$) can express amount of gene activation / inhibition or tea positive / negative favor, respectively. We look for rules that say e.g. ”who likes tea of kind 1, dislikes tea of kind 2”, ”when both genes 1 and 2 are activated then gene 3 is activated too”.

Combination of variables is done by connectives of fuzzy logic. Since amount of variables in bioinformatics (i.e. genes) is rather big, it is necessary to cluster them during computations. It is not disadvantage. It is known that groups of genes behave similarly and to find the groups is one of tasks of bioinformatics. Evaluations are done by so called generalized quantifiers. They combine ideas of classical quantifiers and ideas of statistical estimators and tests [5]. They can be, for example, estimates of quantiles (of holding a formula) or tests for e.g. 0.9 value of them on a value of significance.

3. Feature aggregation

When we have found and enumerated relevant rules we may want them to combine to express a final value which describe investigated system. The value of the object in the interest can be similarity to another (complex) object, inclination of a relevant gene to be activated or inhibited, or favor of the prepared tea.

We generally have pieces of evidence for both greater final values and lesser final values. Their combination should behave as uninorms. It means that combination of two positive values should tend to be greater, combination of two negative values should be lesser, and combination of one positive and one negative value should lie between them. We can describe such behavior as acting of individual rules on the final value that is glued onto one end of a spring, the second end of the spring is glued to zero value. We call such an operator a dinorm, an example is at Figure 5.

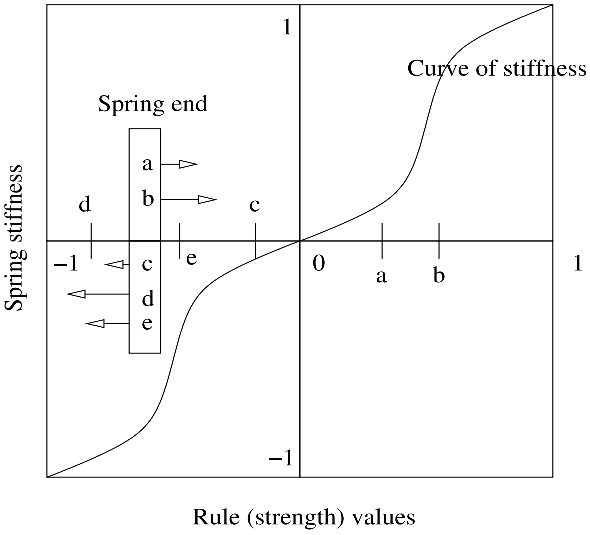


Figure 5: Dinorm example

We need continuity, rather uniform one, to have stable aggregative operators. However, it is impossible for uninorms. In fact, uninorms have unnatural behavior on combination of two opposite extreme values: it must be an extreme too. It can be overwhelmed by abandoning associativity, either weak or strong. It is not so bad since e.g. (arithmetical) mean is not associative too. We just can not separate the final operator into recursive action of one (associative) binary operator.

Still, we can state less conditions (than recursiveness) on reducibility of the operator. The operator may be, for example, separable into two (several) associative operators. In such a case, we say that the operator obey weak non-associativity. This imitates double values in preprocessing and formula combination steps: first, we combine separately positive and negative values by conorms, and second, we combine the two result values by coimplication (of the lesser one to the greater one). Generally, we do not suffer from lack of associativity since it is not required for aggregation operators - we do not use them as logical connectives.

We usually want to have evaluated the power of our result from statistical point of view. Since we have an amount of both objects and rules, we can use some multidimensional methods, e.g. bootstrapping. It yields strength and plausibility of localization of the final value on whole [-1, 1] interval. It means that we can state e.g. that the final result value is greater than or equal to 0.5 with a value of significance, and it is greater than or equal to 0.3 with a greater value of significance.

References

[1] P. Baldi, G. W. Hatfield, “DNA Microarrays and Gene expression, From Experiments to Data Analysis and Modeling”, *Cambridge University Press*, 2002.

[2] P. Hájek, “Metamathematics of fuzzy logic”, *Kluwer*, 1998.

[3] P. Hájek, T. Havránek, ”Mechanizing Hypothesis Formation, Mathematical foundations for a general theory”, *Springer-Verlag Berlin-Heidelberg-New York*, 1978.

[4] P. Hájek, T. Havránek and M. K. Chytil, “Metoda GUHA, Automatická tvorba hypotéz”, *Academia Praha*, 1983.

[5] T. Havránek, “Statistika pro biologické a lékařské vědy”, *Academia Praha*, 1993.

[6] G. J. Klir, T. A. Fogler, “Fuzzy Sets, Uncertainty, and Information”, *Englewood Cliffs, Prentice-Hall*, 1988.

Kernel Based Regularization and Neural Networks

Post-Graduate Student:
MGR. TEREZIE ŠIDLOFOVÁ
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

terka@cs.cas.cz

Supervisor:
RNDR. VĚRA KŮRKOVÁ, DRSC.
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

vera@cs.cas.cz

Field of Study:
Theoretical Informatics
Classification: I1

Abstract

We study approximation problems formulated as regularized minimization with kernel-based stabilizers. These approximation schemas exhibit easy derivation of solution to the problem in the shape of linear combination of kernel functions (one-hidden layer feed-forward neural network). We prove uniqueness of such a solution if one exists and discuss existence in special cases. We exploit the article by N. Aronszajn [1] on reproducing kernels and use his formulation of product of kernels and resulting kernel spaces to show possible use of such a construction in practical applications.

This work was supported by GA ČR grant 201/02/0428.

1. Preliminaries

A *normed linear space* W is any vector space over $\mathbb{R}$ or $\mathbb{C}$ with a *norm* $\|.\|$, where for all $x, y \in W$, $\lambda \in \mathbb{R}$ (or $\mathbb{C}$).

1. $\|x\| \geq 0$ and $\|x\| = 0$ only if $x = 0$
2. $\|\lambda x\| = \lambda \|x\|$, and
3. $\|x + y\| \leq \|x\| + \|y\|$.

Banach space $(B, \|.\|)$ is any normed linear space that is complete in its norm. A *Hilbert space* is a Banach space in which the norm is given by an inner product $\langle ., . \rangle$, that is $\|x\| = \langle x, x \rangle^{1/2}$.

Let d, k be positive integers, $\Omega \subseteq \mathbb{R}^d$. We let $(C(\Omega), \|.\|_C)$ denote the space of continuous functions on Ω with maximum norm. Next, C_k will denote all functions with continuous Fréchet derivative up to order k and C_∞ all infinitely differentiable functions. We say that $f \in C_\infty$ belongs to the *Schwartz space* $\mathcal{S}(\mathbb{R}^d)$ if $p \cdot D^\alpha f$ is a bounded function for any multiindex $\alpha = (\alpha_1, \ldots, \alpha_d)$ and any polynomial $p = \sum_i c_{\beta_i} x_1^{\beta_{i_1}} \ldots x_d^{\beta_{i_d}}$ on $\mathbb{R}^d$ (where $D^\alpha(f) = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1} \ldots \left(\frac{\partial}{\partial x_d}\right)^{\alpha_d}$). For convenience let us define (following [9]) the *normalized Lebesgue measure* m_d on $\mathbb{R}^d$ as $\mathrm{d}m_d(x) = (2\pi)^{-d/2}\mathrm{d}x$.

The Lebesgue space $(\mathcal{L}_p(\Omega), \|.\|_p)$ of functions on Ω with integrable p-th power will be renormed: $\|f\|_p = \left\{\int_\Omega |f|^p \mathrm{d}m_d\right\}^{1/p}$. This will simplify the use of *Fourier transform* $\hat{f}$ of the function $f \in \mathcal{L}^1(\mathbb{R}^d)$: $\hat{f}(t) = \int_{\mathbb{R}^d} f(x) e^{-it \cdot x} \mathrm{d}m_d$, where $t \in \mathbb{R}^d$ and $t \cdot x = t_1 x_1 + \cdots + t_d x_d$.

Let B be a Banach space, $\Omega \subseteq B$ and let $f : \Omega \times \Omega \to \mathbb{R}$ be a symmetric function (that is $f(x, y) = f(y, x)$). Then f is *positive definite* if for any $a_1, \ldots, a_n \in \mathbb{C}$ and $t_1, \ldots, t_n \in \Omega$

$$\sum_{i,j=1}^{n} \overline{a}_i a_j f(t_i, t_j) \geq 0,$$

where $\overline{a}$ is complex adjoint of a. We call the function *strictly positive definite* if the inequality is strict.

Let V and W be vector spaces over the same body. Then $\mathcal{L} : W \to V$ is a *linear mapping* if and only if $\mathcal{L}(\lambda x + \mu y) = \lambda \mathcal{L} x + \mu \mathcal{L} y$ for all $x, y \in W$ and $\lambda, \mu \in F$ (where $F = \mathbb{R}$ or $\mathbb{C}$). If $V = W$, we call $\mathcal{L}$ an *operator*, if $V = F$ we call it a *linear form* or a *functional* on W.

For a functional $\mathcal{F} : X \to (-\infty, +\infty]$ we write $\operatorname{dom} \mathcal{F} = \{f \in X : \mathcal{F}(f) < +\infty\}$ and call this set the *domain* of $\mathcal{F}$. *Continuity* of $\mathcal{F}$ in $f \in \operatorname{dom} \mathcal{F}$ is defined as usual. A functional is *sequentially lower semicontinuous* if and only if the convergence of $\{f_n\}$ to f implies $\mathcal{F}(f) \leq \liminf_{n\to\infty} \mathcal{F}(f_n)$. Functional $\mathcal{F}$ is *weakly sequentially lower semicontinuous* if and only if $f_n \rightharpoonup f$ implies $\mathcal{F}(f) \leq \liminf_{n\to\infty} \mathcal{F}(f_n)$.

A functional $\mathcal{F}$ is *convex* on a convex set $E \subseteq \operatorname{dom} \mathcal{F}$ if for all $f, g \in E$ and all $\lambda \in [0, 1]$, $\mathcal{F}(\lambda f + (1 - \lambda) g) \leq \lambda \mathcal{F}(f) + (1 - \lambda) \mathcal{F}(g)$. Functional $\mathcal{F}$ is *(strongly) quasi-convex* if for all $f, g \in E, f \neq g$ it holds: $\mathcal{F}\left(\frac{1}{2} f + \frac{1}{2} g\right)\ (<) \leq \max\{\mathcal{F}(f), \mathcal{F}(g)\}$.

2. Reproducing Kernel Hilbert Spaces

Reproducing Kernel Hilbert Space (shortly RKHS) was defined by Aronszajn, 1950 ([1]) as Hilbert space $\mathcal{H}$ of functions (real or complex) defined over $\Omega \subseteq \mathbb{R}^d$ with the property, that for each $x \in \Omega$ the evaluation functional on $\mathcal{H}$ given by $\mathcal{F}_x : f \mapsto f(x)$ is bounded. This implies existence of a positive definite symmetric function $k : \Omega \times \Omega \to \mathbb{R}$ (the so called *reproducing kernel*) corresponding to $\mathcal{H}$ such that

1. for any $f \in \mathcal{H}$ and $y \in \Omega$ the following reproducing property holds

   $$f(y) = \langle f(x), k(x, y) \rangle ,$$

   where $\langle ., . \rangle$ is scalar product in $\mathcal{H}$ and
2. for every $y \in \Omega$, the function $k_y(x) = k(x, y)$ is an element of $\mathcal{H}$.

Note that the reproducing kernel is unique for a given $\mathcal{H}$. On the other hand, every positive definite symmetric function is a reproducing kernel for exactly one Hilbert space, that can be described as $\operatorname{comp}\{\sum_{i=1}^{n} a_i k_{x_i}; x_i \in \Omega, a_i \in \mathbb{R}\}$, where comp means completion of the set. See paragraph 2.1 for sketch of proofs.

Next we will consider product of Reproducing Kernel Hilbert Spaces, for potential applications of this construction see the discussion below Theorem 2.1. For $i = 1, 2$ let F_i be an RKHS of functions on Ω_i, let k_i be the corresponding kernel. Consider the following set of functions on $\Omega = \Omega_1 \times \Omega_2$

$$F' = \{\sum_{i=1}^{n} f_{1,i}(x_1) f_{2,i}(x_2) \mid n \in \mathbb{N}, f_{1,i} \in F_1, f_{2,i} \in F_2\} .$$

Clearly, F' is a vector space, it is not complete though. For its completion, we first define a scalar product on F'. Let f, g be elements of F' expressed as $f(x_1, x_2) = \sum_{i=1}^{n} f_{1,i}(x_1) f_{2,i}(x_2)$, $g(x_1, x_2) = \sum_{j=1}^{m} g_{1,j}(x_1) g_{2,j}(x_2)$. We define

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \langle f_{1,i}, g_{1,j} \rangle_1 \langle f_{2,i}, g_{2,j} \rangle_2 ,$$

where $\langle \cdot, \cdot \rangle_i$ denotes the scalar product in F_i. It is a routine to check that this definition does not depend on the particular form in which f and g are expressed and that the properties of scalar product are satisfied. We define norm on F' by $\|f\| = \sqrt{\langle f, f \rangle}$. Finally, let F be the completion of F'. It can be shown ([1]) that the completion exists not only as an abstract Hilbert space but that F is in fact a space of functions on Ω. We call F the product of F_1 and F_2 and write $F = F_1 \otimes F_2$.

Theorem 2.1 *([1]) For $i = 1, 2$ let F_i be an RKHS on Ω_i with kernel k_i. Then the product space $F = F_1 \otimes F_2$ on $\Omega_1 \times \Omega_2$ is an RKHS with kernel given by*

$$k((x_1, x_2), (y_1, y_2)) = k_1(x_1, y_1) k_2(x_2, y_2)\,,$$

where $x_1, y_1 \in \Omega_1$, $x_2, y_2 \in \Omega_2$.

2.1. Proofs

All the proofs presented here have been sketched in [1].

Lemma 2.2 *Let $\mathcal{K}(\Omega)$ be a real valued RKHS with k as kernel. Then $\mathcal{K}_C := \{f_1 + if_2; f_1, f_2 \in \mathcal{K}\}$ with $\|f_1 + if_2\|^2 = \|f_1\|^2 + \|f_2\|^2$ is a complex RKHS with the same k as kernel.*

Proof: $\mathcal{K}_C$ is clearly a Hilbert space. Evaluation functionals remain linear and bounded, i.e. $\mathcal{K}_C$ is RKHS. And for any $f \in \mathcal{K}$ it holds: $if(y) = \langle if(x), k(x, y) \rangle$. ■

We see that it is sufficient to consider only complex RKHS.

Lemma 2.3 *Let $\mathcal{K}(\Omega)$ be a Hilbert space with a reproducing kernel k. Then k is unique.*

Proof: Suppose we have two reproducing kernels k, k' and $k \neq k'$. Then for some x, y we have $0 < \|k(x, y) - k'(x, y)\|^2 = \langle (k - k')(x, y), (k - k')(x, y) \rangle = \langle (k - k')(x, y), k(x, y) \rangle - \langle (k - k')(x, y), k'(x, y) \rangle = (k - k')(y, y) - (k - k')(y, y) = 0$, which is a contradiction. ■

Lemma 2.4 *Let $\mathcal{K}(\Omega)$ be a Hilbert space with the property that all evaluation functionals $\mathcal{F}_x$ are linear and bounded. Then there exists a reproducing kernel k satisfying properties (i) and (ii) that is positive definite. On the other hand from (i) and (ii) we obtain linear bounded (continuous) evaluation functionals.*

Proof: $\mathcal{F}_y$ is a linear bounded (i.e. continuous) functional on Hilbert space $\mathcal{K}(\Omega)$. Thus by Fréchet-Riesz Theorem [6, p. 19] we have $a_y \in \mathcal{K}$ such that $\mathcal{F}_y(f) = \langle f(x), a_y(x) \rangle$. We put $a_y(x) = k(x, y)$ obtaining the reproducing kernel.

To check the desired properties (symmetry and positive definiteness) we use the reproducing property: $\sum_{i,j=1}^n \overline{a}_i a_j k(x_i, x_j) = \langle \sum_{j=1}^n a_j k(y, x_j), \sum_{i=1}^n a_i k(y, x_i) \rangle = \| \sum_{j=1}^n a_j k(y, x_j) \|^2 \geq 0$ and $k(x, y) = \langle k(z, y), k(z, x) \rangle = \overline{\langle k(z, x), k(z, y) \rangle} = \overline{k(y, x)}$.

To prove the last statement it is sufficient to observe, that: $|f(y)| = |\langle f(x), k(x, y) \rangle| \leq \|f\| \langle k(x, y), k(x, y) \rangle^{1/2} = \|f\| k(y, y)^{1/2}$. ■

Lemma 2.5 *To every $k(x, y)$ satisfying the properties (i) and (ii) there corresponds one and only one Hilbert space $\mathcal{H}$ admitting k as a reproducing kernel.*

Proof: Let us take the class of all functions of the form $\sum \alpha_k k(x, y_k)$ with the norm $\| \sum_{k=1}^{n} \alpha_k k(x, y_k)\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{\alpha}_i \alpha_j k(x_i, x_j)$. To complete the space we add limits of all Cauchy sequences (relative to the above norm - gives point-wise convergence). ∎

Theorem 2.6 *Let F be a linear class of functions with scalar product defined on Ω satisfying all the properties of a Hilbert space with the exception of completeness (an incomplete Hilbert space). The class can be completed if and only if*

1. *for every fixed $y \in \Omega$ the linear functional $\mathcal{F}_y(f)$ is bounded in F*
2. *for a Cauchy sequence $\{f_m\} \subset F$ the condition $f_m(y) \to 0$ for every y implies $\|f_m\| \to 0$.*

If the completion is possible, it is unique.

Proof: See [1, p. 347]. ∎

3. Learning from data as minimization of functionals

Learning from data usually means to fit a function to a set of data $z = \{(u_i, v_i); i = 1, \ldots, N\} \subseteq \mathbb{R}^d \times \mathbb{R}$. The problem is what type of functions will we use for the fitting, because there are infinitely many ways to go through the given points. And even if we have a reasonable set of functions (admissible set) to pick from, there is no guarantee that the problem will have a solution and that the solution will be unique.

Typically it is not necessary that the function fits the data exactly, we approximate. Thus nice functions (smooth, continuous) come into question and the solution generalizes better (see [5]). Some of these properties are easily expressed by the set of admissible functions, but we might have more complicated (global) external information (a-priori knowledge) about the problem and want to add it, too.

Mathematical expression of these ideas lies in formulating a functional that would among admissible functions pick the one, that is reasonably close to the data and also agrees with global property assumptions ([2], [4], [8], [10], [12]). Existence and uniqueness of such a solution can be secured by minimizing a functional over a corresponding set of functions.

The task to find an optimal solution to the setting of approximating a data set $z = \{(u_i, v_i)\}_{i=1}^{N} \subseteq \mathbb{R}^d \times \mathbb{R}$ by a function from a general function space X (minimizing the error) is ill-posed. Thus we impose additional (regularization) conditions on the solution ([5]). These are typically things like a-priori knowledge, or some smoothness constraints. The solution f_0 has to minimize a functional $\mathcal{F} : \Omega \to \mathbb{R}$ that is composed of the error part and the “smoothness” part:

$$\mathcal{F}(f) = \mathcal{E}_z(f) + \gamma \Phi(f),$$

where $\mathcal{E}_z$ is the error functional depending on the data $z = \{(u_i, v_i)\}_{i=1}^{N} \subseteq \mathbb{R}^d \times \mathbb{R}$ and penalizing distance from the data, Φ is the regularization part — the so called stabilizer — penalizing ”distance from the global property” and γ is the regularization parameter giving the trade-off between the two terms of the functional to be minimized.

To prove existence and uniqueness of solution to such a problem we will use some results from mathematical analysis. Uniqueness of solution to the minimization problem can be secured by strong quasi-convexity of the minimized functional (see Lemma 3.2). The error functionals are naturally convex and to have quasi-convexity we need the other part of the minimization functional to do the job. In fact if the second part was quasi-convex, we would succeed. So we are searching for regularization parts that are quasi-convex. A wide range of such functionals are second powers of norms of Hilbert spaces (for example RKHS).

Here we start using Reproducing Kernel Hilbert Spaces to obtain existence and uniqueness of solution and derive the form of the solution easily. We build an RKHS that fits to our problem and obtain a unique well defined solution to the problem. The idea is to minimize our functional over this RKHS (using advantages of Hilbert spaces) and having the regularization part in the form of a norm on this RKHS. Then we obtain existence and uniqueness easily and by reproducing property of the kernel also the form of the solution.

Let $\mathcal{H}$ be an RKHS over $\Omega \subseteq \mathbb{R}^d$ with kernel k and norm $\|.\|_k$. We construct the minimization functional composing of error part $\mathcal{E}_z(f)$ based on data $z = \{(u_i, v_i); i = 1, \ldots, N\} \subseteq \mathbb{R}^d \times \mathbb{R}$ and let the regularization part be $\Phi(f) = \|f\|_k^2$ forming $\mathcal{F}(f) = \mathcal{E}_z(f) + \gamma\Phi(f)$ with $\gamma \in \mathbb{R}^+$. Now uniqueness of solution to such a problem comes clearly from strong quasi-convexity of the functional $\mathcal{F}$ (see Lemma 3.2, 3.3).

To show existence of solution many authors consider sufficient to derive the shape of a solution (without explicitly showing that it is a solution). We don't regard this approach convincing (there may be no solution at all); however we are able to prove existence in special cases only, see for example [11].

Derivation of the shape of the solution to the regularized minimization problem has been shown already in [5] but without taking advantage of RKHS, in [4], [8] and others known as Representer theorem, for a concrete case see [11]. All the proofs are based on a theorem from mathematical analysis.

Theorem 3.1 *Let the functional $\mathcal{F}$ defined on a set E in a Banach space X be minimized at a point $f_0 \in E$, with f_0 an interior point in the norm topology. If $\mathcal{F}$ has a derivative $\mathrm{D}\mathcal{F}_{f_0}$ at f_0, then $\mathrm{D}\mathcal{F}_{f_0} = 0$.*

Employing this theorem we obtain solution to the kernel-based minimization problem in the form of

$$f_0(x) = \sum_{i=1}^{N} c_i k(x, u_i),$$

where x_i are the data points and $k(\cdot, \cdot)$ the corresponding kernel.

3.1. Examples of minimization functionals and RKHS

An error functional is usually of the form $\mathcal{E}_z(f) = \sum_{i=1}^{N} V(f(u_i), v_i)$. A typical example of the empirical error functional is the classical mean square error:

$$\mathcal{E}_z(f) = \frac{1}{N}\sum_{i=1}^{N}(f(u_i) - v_i)^2.$$

In [5] a special stabilizer based on the Fourier Transform was proposed:

$$\Phi_G(f) = \int_{\mathbb{R}^d} \frac{|\hat{f}(s)|^2}{\hat{G}(s)} \mathrm{d}m_d(s),$$

where $\hat{G} : \mathbb{R}^d \to \mathbb{R}_+$ is symmetric ($\hat{G}(s) = \hat{G}(-s)$) function tending to zero as $\|s\| \to \infty$ (the last holds for any $G \in \mathcal{L}_1$). That means $1/\hat{G}$ is a low-pass filter.

Thus the functional $\mathcal{F}_G$ to be minimized is of the form:

$$\mathcal{F}_G(f) = \mathcal{E}_z(f) + \Phi_G(f) = \frac{1}{N}\sum_{i=1}^{N}(f(u_i) - v_i)^2 + \gamma \int_{\mathbb{R}^d} \frac{|\hat{f}(s)|^2}{\hat{G}(s)} \mathrm{d}m_d(s),$$

where $\gamma \in \mathbb{R}^+$. Now we show how to build an RKHS corresponding to the regularization part of our functional:

Let us define

$$g(x,y) = G(x-y) = \int_{\mathbb{R}^d} \hat{G}(t) e^{it.x} e^{-it.y} \mathrm{d}m_d(t).$$

For $g \in \mathcal{S}(\mathbb{R}^{2d})$ symmetric positive definite we obtain an RKHS $\mathcal{H}$ (using the classical construction, see [4], [10],[12]). We put $\langle f, g \rangle_{\mathcal{H}} = \int_{\mathbb{R}^d} \frac{\hat{f}(s)\hat{g}^*(s)}{\hat{G}(s)} \mathrm{d}m_d(s)$ and obtain the norm $\|f\|_{\mathcal{H}}^2 = \int_{\mathbb{R}^d} \frac{|\hat{f}(s)|^2}{\hat{G}(s)} \mathrm{d}m_d(s)$, for $\mathcal{H} = \mathrm{comp}\,\mathrm{span}\{G^{\dagger}(x,.), x \in \mathbb{R}^d\}$, where $\mathrm{comp}\{\dots\}$ denotes completion of the set $\{\dots\}$ and a^* means complex conjugate of a. It is easy to check the reproducing property of G on $\mathcal{H}$, that is $\langle f(x), G(x-y) \rangle_{\mathcal{H}} = f(y)$.

Special types of reproducing kernels and following RKHS are the well known Gaussian kernel $k(x,y) = e^{-\|x-y\|^2}$ with Fourier transform $\hat{k}(s) = e^{-\frac{\|w\|^2}{2}}$ or in one dimension, the kernel given by $k(x,y) = e^{-|x-y|}$ with Fourier transform $\hat{k}(s) = (1+s^2)^{-1}$. The norm for this RKHS is of the form $\|f\|_k = \int \frac{|\hat{f}|^2}{(1+s^2)^{-1}} = \|f\|_{\mathcal{L}_2}^2 + \|f'\|_{\mathcal{L}_2}^2$. So we see we obtain a Sobolev space W_2^1.

As a more general example we will consider the product of kernels introduced in Section 2. Suppose that apriori knowledge of our data suggests to look for the solution as a member of product of two functional spaces. In one dimension the data may be clustered thus being suitable for approximation via Gaussian kernels. In the other dimension we have only information on smoothness of the data, hence we will use kernel resulting in Sobolev norm. Employing Theorem 2.1 we obtain a kernel for the product space of the form:

$$k((x_1,x_2),(y_1,y_2)) = e^{-\|x_1-y_1\|^2} \cdot e^{-|x_2-y_2|},$$

where $x_1, y_1 \in \Omega_1$, $x_2, y_2 \in \Omega_2$. Taking advantage of this being an RKHS we have the form of the solution to such a type of minimization:

$$f_0(x_1,x_2) = \sum_{i=1}^{N} c_i e^{-\|x_1-u_{i,1}\|^2} \cdot e^{-|x_2-u_{i,2}|}.$$

We expect this approximation scheme to exhibit nicer approximation properties since it can be better fitted to special types of data.

3.2. Proofs

Lemma 3.2 (Da71) *A strongly quasi-convex functional $\mathcal{G}$ can achieve its minimum over a convex set C at no more than one point.*

Proof: Let $\mathcal{G}$ attain its minimum at f_1 and f_2 (i.e., $\mathcal{G}(f_1) = \mathcal{G}(f_2) = \inf_{f \in C} f(x)$) and $f_1 \neq f_2$. Then $\frac{1}{2}f_1 + \frac{1}{2}f_2 \in C$, but $\mathcal{G}(\frac{1}{2}f_1 + \frac{1}{2}f_2) < \max\{\mathcal{G}(f_1), \mathcal{G}(f_2)\} = \inf_{f \in C} \mathcal{G}(f)$, which is a contradiction. ∎

Lemma 3.3 *Functional $\mathcal{E}_z$ is convex and functional Φ_G is strongly quasi-convex on RKHS $\mathcal{H}$. Hence, also $\mathcal{F}$ is strongly quasi-convex on K.*

Proof: For the first part, $\mathcal{E}_z(f)$ as an error functional is convex. (See for example 3.1. The sum of N elements, each of which is a convex functional, as (real) function $w \mapsto \frac{1}{N}(w - v_i)^2$ is convex.)

To deal with the other functional, we will prove that in any Hilbert space the norm $\|.\|$ is strongly quasi-convex, that is $\|\frac{1}{2}x + \frac{1}{2}y\| < \max\{\|x\|, \|y\|\}$ for any distinct x, y in the space. We will use the parallelogram law to show the fact. In any Hilbert space it holds that $\|x+y\|^2 + \|x-y\|^2 = 2(\|x\|^2 + \|y\|^2)$ and so we get:

$$\frac{1}{4}\|x+y\|^2 = \frac{2}{4}(\|x\|^2 + \|y\|^2) - \frac{1}{4}\|x-y\|^2.$$

Hence $\|\frac{1}{2}x + \frac{1}{2}y\|^2 \leq \frac{1}{2}(2\max\{\|x\|^2, \|y\|^2\}) - \frac{1}{4}\|x-y\|^2$. As for $x \neq y$ we have $\|x-y\|^2 > 0$, we get the desired claim. (Observe that $\Phi_G(f) = \|f\|_k^2$ in Section 3.1.)

So we have $\mathcal{F}_G$ a sum of a convex and a strongly quasi-convex functional and so clearly $\mathcal{F}_G$ is strongly quasi-convex as claimed. ■

4. Conclusion

We have shown how to employ RKHS in approximation theory and stressed advantages of this approach. Inspired by the article [1] we introduce kernel-product based approximation and try to show possible practical usage.

Further work shall be concentrated on the product issue comparing it to standard approximation methods. We also want to see to the question of existence of the solution of minimization problem in a more general scope.

References

[1] Aronszajn N. (1950) *Theory of Reproducing Kernels*. Transactions of the AMS, **68**, 3, p. 337-404.

[2] Cucker F., Smale S. (2001). *On the Mathematical Foundations of Learning*. Bulletin of the American Mathematical Society **3**9, 1-49.

[3] Daniel J. W. (1971). *The Approximate Minimization of Functionals*. Prentice-Hall, Inc.

[4] Girossi F. (1998). *An Equivalence between Sparse Approximation and Support Vector Machines*. Neural Computation **10**, 1455-1480, MIT. (A.I. Memo No. 1606, MIT, 1997)

[5] Girossi F., Jones M., Poggio T. (1995). *Regularization Theory and Neural Networks Architectures*. Neural Computation, **7**, 219-269.

[6] Lukeš J. (2002). *Zápisky z funkcionální analýzy*. Karolinum, UK Praha.

[7] Lukeš J., Malý J. (1995). *Measure and Integral*. Matfyzpress, Praha, 1995.

[8] Poggio T., Smale S. (2003). *The Mathematics of Learning: Delaing with Data*. Notices of the AMS **5**0, 5, 536-544.

[9] Rudin W. (1991). *Functional Analysis*,. 2nd Edition, McGraw-Hill, NY.

[10] Schölkopf B., Smola A. J. (2002). *Learning with Kernels*. MIT Press, Cambridge, Massachusetts.

[11] Šidlofová T. (2004). *Existence and Uniqueness of Minimization Problems with Fourier Based Stabilizers*. Compstat 2004, Prague.

[12] Wahba G. (1990). *Spline Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia.

Automatická tvorba analytického popisu systému

doktorand:
ING. MILAN ŠIMŮNEK
Katedra informačních technologií
Vysoká škola ekonomická
Praha
simunek@cs.cas.cz

školitel:
DOC. RNDR. PAVEL DRBAL, CSC.
Katedra informačních technologií
Vysoká škola ekonomická
Praha
drbal@vse.cz

obor studia:
Informatika
číselné označení: I

Abstrakt

Současné metodiky pro analýzu a návrh programových systémů kladou značné nároky na znalosti a zkušenosti analytiků. Kvalitně provedená analýza a návrh vyžaduje značné mentální úsilí podpořené navíc dostatkem zkušeností. To se odráží ve vysokých nákladech na vyškolení a udržení kvalitních analytiků, ale i na jejich nedostatku na pracovním trhu.

Disertační práce se snaží poukázat na možnost využití některých metod umělé inteligence pro automatizaci procesu analýzy programového systému popsaného pomocí zadání ve volném textu. Zautomatizování této činnosti nebo i pouhé přiblížení analyzované oblasti by přineslo nemalé finanční úspory a ušetřilo i čas nutný pro vyškolení kvalitních analytiků.

Řešení prezentované v disertační práci je založeno na použití evolučních algoritmů a mobilních agentů. Evoluční algoritmy umožňují zjednodušit celou úlohu nahrazením obtížné transformace text na analytický popis transformací k ní opačnou – analytický popis na text, která je výrazně jednodušeji algoritmizovatelná. Mobilní agenti se používají pro rychlé a přesné vyhledání pojmů v rozsáhlé bázi znalostí.

Tato práce je myšlena jako jeden z prvních pokusů o automatizaci procesů softwarového inženýrství s využitím přístupů a prvků z oboru umělé inteligence.

Klíčová slova:

analýza a návrh IS, evoluční algoritmy, reprezentace znalostí, vyhledávání znalostí

1. Charakteristika současného stavu

Současné metodiky pro analýzu a návrh programových systémů kladou značné nároky na znalosti a zkušenosti analytiků. Kvalitně provedená analýza a návrh vyžaduje značné mentální úsilí podpořené navíc dostatkem zkušeností. To se odráží ve vysokých nákladech na vyškolení a udržení kvalitních analytiků, ale i na jejich nedostatku na pracovním trhu.

V současné době lze pozorovat velmi nízkou míru spolupráce mezi obory softwarového a znalostního inženýrství. Přestože se jedná o velmi blízké obory, nedochází k podstatné výměně informací, postupů a metod, které se ukázaly v praxi jako přínosné.

Nabízí se otázka, zda by nešlo využít současných poznatků v oblasti umělé inteligence a automatického zpracování znalostí k řešení nedostatku volných analytiků na trhu. Konečným cílem by mohla být snaha o vytvoření systému, který může do jisté míry zastupovat zkušeného analytika při analýze a návrhu programového systému. Začínající analytik se na systém bude moci obrátit jako na kolegu-experta s žádostí o radu. I pro zkušené analytiky by byl systém přínosem, protože by jej mohli používat v případech, kdy by si s vlastním řešením nebyli jisti. Systém doporučí řešení, které bude moci být iteračně doplňováno pomocí změn v zadání. Nakonec může být buď zcela přijato jako výsledné řešení nebo bráno pouze jako inspirace pro vlastní řešení vytvořené člověkem-analytikem. Bude tak docházet k výrazným úsporám v čase potřebným na řešení zcela bez pomoci a zároveň i ke snižování nákladů, které by jinak byly třeba pro zaplacení rozsáhlejšího týmu analytiků.

2. Vymezení cíle práce

Základním cílem práce je v prvé řadě poukázat na možnost interdisciplinárního přístupu k analýze a návrhu a zejména na možnost využití metod a algoritmů z oblasti umělé inteligence.

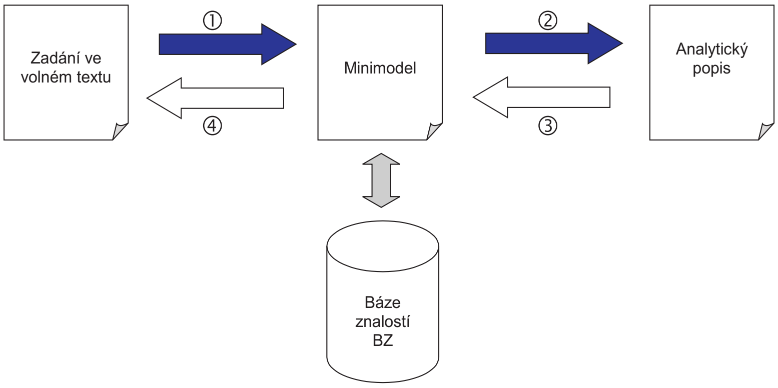
Vlastní obsah práce má potom za hlavní cíl navrhnout základní postup tvorby analytického popisu a specifikovat jeho jednotlivé etapy a datové struktury, které jsou v nich používány. Přitom je kladen důraz na netradiční přístupy, které byly prozatím opomíjeny a které nabízejí oproti klasickým metodám potenciálně velmi dobré výsledky. Pro vybrané etapy je potom cílem navrhnout konkrétní algoritmy zpracování.

K dosažení hlavního cíle bylo nejprve navrženo vyřešit tři dílčí cíle, které jsou v práci detailně specifikovány. Jedná se o tyto kroky:

- vytvoření minimodelu z textového zadání;
- porovnání minimodelu s analytickým popisem;
- vytvoření analytického popisu IS podle zadání.

Kde textové zadání je vstupem popisujícím programový systém v textové podobě; minimodel je speciální datová struktura sloužící pro krátkodobé uchování znalostí o aktuálně zpracovávané předmětné oblasti; a analytický popis je výstup tvořený množinou analytický modelů v grafické podobě a k nim doprovodných textů s doplňujícími informacemi a omezeními.

Dílčí kroky ve formě jednotlivých transformací vidíme i na následujícím obrázku.



Základní typy dat a transformace

Prvním krokem je zpracování textového zadání a vytvoření interní reprezentace v zadání obsažených znalostí – minimodelu (MM). Druhým krokem je potom vytvoření analytického popisu z minimodelu. Samostatnou úlohu, která bude využita v druhém kroku je porovnání minimodelu a existujícího analytického popisu.

Na vstupu zpracování se objevuje zadání ve formě volného textu. Tato forma reprezentace znalostí je pro další strojové zpracování nevhodná a je třeba transformovat znalosti obsažené v textu (pokud možno beze ztráty věcného významu) do jiné, strojově lépe zpracovatelné formy.

Při snaze porozumět volnému textu budeme využívat znalosti uložené v bázi znalostí. Minimodel je jakousi krátkodobou pamětí obsahující podmnožinu znalostí z BZ vybranou s ohledem na věcnou oblast, které se týká zpracovávaný text. Reprezentace znalostí v minimodelu vychází z formátu, ve kterém jsou uloženy znalosti v bázi znalostí. Vzhledem k předpokládanému menšímu rozsahu minimodelu oproti BZ si však můžeme dovolit ukládat do minimodelu i některé duplicitní informace. Tím zjednodušíme a zrychlíme zpracování textu a umožníme odhalení i dalších souvislostí.

Zjištění shody mezi zadáním a již existujícím analytickým popisem je dílčí úlohou, která nám (jak si ukážeme) pomůže při samotné tvorbě analytického popisu. Při porovnávání předpokládáme, že analytický popis byl již vytvořen (ať již ručně nebo automaticky) a naším úkolem je zjistit, zda si navzájem zadání a analytický popis odpovídají.

Přestože porovnání AP je evidentně jednodušší úloha než vytvoření AP, setkáváme se i při porovnávání s celou řadou problémů. Jako největší z nich se jeví již pouhá definice shody nebo podobnosti. Musíme přesně definovat, kdy jsou dva popisy (textový v zadání a analytický) shodné, kdy jsou „velmi podobné“, kdy „méně podobné“ a kdy zcela odlišné. Dále musíme nalézt společný popis (jazyk), do kterého převedeme jak textové zadání, tak i informace z analytického popisu, abychom mohli zavést nějakou míru shody a definovaným způsobem ji zjišťovat.

3. Obsah a struktura práce

Tato práce popisuje jeden z prvních pokusů o syntézu postupů a metod znalostního inženýrství (umělé inteligence a automatického získávání znalostí) s postupy a metodami používanými v softwarovém inženýrství při objektově orientované analýze a návrhu (OOAD) informačních systémů.

4. Navržený způsob řešení

V zásadě existují dva možné různé postupy získání analytického popisu z textového zadání. První z nich, který můžeme označit jako přímý, sleduje myšlenkový proces v mozku člověka–analytika, který má za cíl vytvořit analytický popis k systému definovanému textových zadáním. Po přečtení a pochopení textu je postupně v iteračních krocích tvořen a zpřesňován analytický popis a jeho jednotlivé modely. Druhý, v této práci preferovaný, postup obrací obvyklou posloupnost kroků a s využitím evolučních algoritmů se snaží postupovat opačně – od analytického popisu směrem k textovému zadání.

4.1. Klasický přímý postup

V přímém postupu procházíme očekávanou sekvencí kroků od syntaktické a sémantické analýzy přes vytvoření jednotlivých analytických modelů až konečně k porovnání původního zadání s dosaženým výsledkem. Během jednotlivých fází se upravuje aktivní kontext v bázi znalostí a zvyšuje se tak pravděpodobnost nalezení správných pojmů k použitým termínům, a tím i upřesnění celého popisu. Celý postup se opakuje tak dlouho, až je dosaženo požadované míry shody mezi zadáním a výsledkem (podrobněji viz [41]). Základní postup vypadá takto:

1. Vytvoření minimodelu z textového zadání
    - + úprava aktivních kontextů v BZ
2. Vytvoření analytického popisu (analytických modelů a doprovodných textů)
    - **tento bod je asi hlavní kámen úrazu, protože je obtížný i pro člověka–analytika a velmi špatně se algoritmizuje!**
3. Porovnání zadání s analytickým popisem
4. Je-li dosaženo požadované míry shody mezi zadáním a analytickým popisem, tak **KONEC**

5. Zpět k bodu 1)
    - v následujících krocích může být dosaženo odlišných výsledků v důsledku změny aktivních kontextů v BZ a tím pádem i nalezení jiných pojmů k hledaným termínům

Už práce [41] ukázala, že proces porozumění volnému textu je iterační, tj. opakující se tak dlouho, až je dosaženo požadované kvality/přesnosti výsledku. Kvalitu výsledků zjišťujeme porovnáním textového zadání s analytickým popisem.

Přímý postup se hodí v případech, kdy je cesta od zadání k řešení relativně přímá a nehrozí „uvíznutí“ v některé slepé uličce. V opačném případě může být velmi těžké navrhnout algoritmy, které uvíznutí rozpoznají a vrátí postup zpět na správnou cestu.

4.2. Hrubá síla s účelovou funkcí

Druhá možná cesta k vytvoření analytického popisu je do jisté míry opakem předchozí a může se zdát na první pohled poněkud krkolomná. Její výhody však spočívají v překonání některých potíží, které omezují konvenční postup a jeho použitelnost. V tomto přístupu je kladen důraz na dobře zvládnutou úlohu porovnání zadání s analytickým popisem a evoluční algoritmy, které umožňují v přijatelném čase řešit s dostatečně dobrými výsledky i exponenciálně složité úlohy.

Velkou výhodou obráceného postupu je fakt, že přímá cesta od textového zadání k analytickému popisu je opravdu velmi obtížná. Mnoho metodik (viz např. [39] se snaží tuto cestu popsat a nabízí různé pomůcky a mezikroky, jak vytvořit správný analytický popis. Naproti tomu cesta od analytického popisu k textovému zadání se zdá být jednodušší. Máme-li dostatečně výkonný počítač, můžeme si dovolit vytvářet velké množství analytických popisů a doufat, že se nám podaří „náhodou“ vytvořit takový, který bude odpovídat zadání. Je zřejmé, že tento postup je možné použít pouze při úplné automatizaci celého procesu, protože bude nutné prověřit statisíce a možná milióny analytických popisů. V žádném případě nesmí být kdekoliv vyžadovaná reakce člověka. Zároveň musí být „náhodné“ vytváření usměrňováno tak, aby se zvyšovala pravděpodobnost nalezení vhodného řešení.

Pro tento účel se jeví jako vhodné použít evoluční algoritmy. Pro evoluční algoritmy je klíčová vhodná datová reprezentace a dále účelová funkce (*fitness-function*), podle které vybíráme nejúspěšnější jedince. V našem případě bude účelová funkce vyjadřovat shodu mezi analytickým popisem na jedné straně a textovým zadáním na straně druhé. Jak analytický popis tak i textové zadání budeme muset převést na minimodel. Čím podobnější oba minimodely budou, tím lépe. Jedinci populace budou reprezentovat možné analytické popisy a evolucí se budou blížit hledanému popisu odpovídajícímu co nejvíce textovému zadání. Celý postup šlechtění populace analytických popisů vypadá takto:

1. Vytvoření minimodelu z textového zadání
    - + úprava aktivních kontextů v BZ
2. Vytvoření prvotní populace analytických popisů
    - s využitím slov v textu zadání a pojmů v MM
3. Ohodnocení každého analytického popisu z populace na základě účelové funkce
    - převod na minimodel a porovnání s minimodelem textového zadání
4. Je-li dosaženo požadované míry shody mezi zadáním a analytickým popisem, tak **KONEC**
5. Vytvoření nové populace analytických popisů
    - odstranění nejhorších jedinců
    - vytvoření nových jedinců

  - zejména jako potomků úspěšných jedinců aktuální populace
  - mutace a křížení

6. Zpět k bodu 3)

Z uvedeného postupu je patrné, že místo velmi obtížné transformace *Minimodel → Analytický popis* stačí zvládnout ne tak obtížnou transformaci *Analytický popis → Minimodel* a následné porovnání dvou minimodelů.

Je však třeba zdůraznit zásadní omezení evolučních algoritmů a to je získání pouze sub-optimálního řešení. Nikdy nemáme jistotu, že získané řešení je jediné správné a dokonce ani, že neexistuje ještě lepší řešení (algoritmus neuvízl v lokálním optimu). Přesto považuji tento postup za prakticky použitelný. Vždyť i pouhý nástin možného řešení velmi pomůže člověku–analytikovi, který jej může dále upravovat a rozšiřovat.

5. Metody dosažení cílů

Pro dosažení cílů byly použity zejména metody a algoritmy používané v oblasti umělé inteligence (AI) a dále vlastní zkušenosti s analýzou a návrhem programových systémů pomocí objektových metodik. Metody a algoritmy AI byly potom v souladu se zkušenostmi aplikovány na tvorbu analytického popisu. Kromě možných různých způsobů reprezentací znalostí jsou v této práci použity zejména evoluční algoritmy, principy distribuované umělé inteligence a *fuzzy* logiky.

6. Naplnění cílů práce

Cíle stanovené pro disertační práci se podařilo splnit. K základnímu cíli (poukázání na možnost interdisciplinárního přístupu) byl splněn i hlavní cíl – navržen základní postup tvorby analytického popisu. Zde navržený původní postup obrací s využitím evolučních algoritmů klasickou přímou cestu na „zpáteční“ ve směru od analytického popisu k textovému zadání.

Dílčí cíle byly splněny navržením podrobných postupů pro jednotlivé fáze transformace od textového zadání přes minimodel až k analytickému popisu, včetně schopnosti porovnat existující analytický popis s minimodelem (resp. textových zadáním) a poskytnout tak metriku pro selekční funkci evolučních algoritmů. Pro navrhovaný postup je klíčovou datovou strukturou báze znalostí, ve které jsou uloženy všechny dostupné znalosti o okolním světě. Pro jejich ukládání byl vytvořen vhodný způsob reprezentace znalostí umožňující rychlé vyhledání správného pojmu k termínu v textu a také odvozování znalostí.

Zároveň byly navrženy i některé detailní metody a algoritmy:

- Pro vyhledávání nejlepšího odpovídajícího pojmu v BZ podle termínu v textu byl navržen speciální algoritmus využívající mobilních agentů („neuronových vzruchů“) volně se pohybujících po bázi znalostí. Podobně jako v lidském mozku jsou v aktuálních centrech aktivity vzruchy nejčetnější a vzrůstá tak pravděpodobnost rychlého nalezení významově správného pojmu k termínu.
- S odvozováním znalostí úzce souvisí i neméně důležitá schopnost *fuzzy* porovnávání dvou znalostí pomocí míry v intervalu $<0;1>$. To umožňuje rozpoznat určitou míru podobnosti i mezi vzdáleně podobnými modely a výrazným způsobem tak urychlit evoluční výběr.

7. Přínosy k řešení zvolené oblasti

Z hlediska vědeckého přístupu se tato práce zabývá aplikací obecných algoritmů na vybranou oblast. Na základě problémů, které se objevují při snaze o algoritmizaci přímého postupu obvyklého při analýze a návrhu systémů, se tato práce pokouší ukázat možnost obráceného postupu pokus-omyl a jeho implementaci pomocí evolučních algoritmů, které se jeví jako velmi perspektivní. Neméně důležitá je i schopnost přesného uchovávání znalostí o okolním světě a rychlé nalezení požadované informace.

Za hlavní přínos této práce považuji právě interdisciplinární přístup, který používá evoluční algoritmy pro překlenutí potíží, které se obvykle vyskytují při snaze o přímý postup tvorby analytických modelů IS, a to jak při strojovém zpracování, tak i při návrhu samotným člověkem-analytikem. Zároveň jsou využívány a dále rozvíjené poznatky z diplomové práce.

Další podstatný přínos vidím v obecném chápání znalostí, definici rolí pojmů vystupujících ve vztahu pomocí dalších pojmů z BZ a připuštění nejednoznačnosti ve vyjadřovací schopnosti BZ. Víceznačné možnosti vyjadřování zásadním způsobem rozšiřují možnosti uchovávání znalostí v BZ. Na druhou stranu neovlivňují kvalitu výsledku, protože v navrženém postupu nepotřebujeme znalosti interpretovat, ale naopak jak textové zadání, tak i analytický popis do struktury podobné BZ převádíme. Nejblíže jsou tomuto chápání znalostí konceptuální grafy – [45] nebo http://www.jfsowa.com/cg/. Ty jsou navíc rozšířené o kvantifikátory a další prvky, které umožňují popsat i komplikovanější výroky typu existence, „věřit v něco“ apod. Možnost využití konceptuálních grafů pro dokonalejší anotaci informačních zdrojů popisuje článek [34].

Kvalitními zpracování textu zadání i analytického popisu napomáhá nedeterministické vyhledávání správného pojmu k termínu v textu pomocí agentů, kteří se volně pohybují po BZ. Inspirací pro ně byly neuronové vzruchy, které probíhají v lidském mozku. Agenti nepomáhají pouze v rychlém a přesném vyhledávání správných pojmů, ale i k udržování aktivního kontextu a v neposlední řadě i při úklidu báze znalostí.

Kvalitu porovnávání znalostí zvyšuje i *fuzzy* přístup, protože nerozlišujeme pouze dva případy – shodné × odlišné, ale celou spojitou škálu podobnosti vyjádřenou jako číslo v intervalu <0;1>. Pro porovnávání znalostí byla navržena schopnost substituce mezi příbuznými pojmy, která umožňuje rozpoznání i vzdáleně podobných struktur.

V neposlední řadě otevírá navržený postup relativně snadnou cestu k paralelizaci, která je u tohoto typu úloh velmi důležitá.

Protože se jedná o interdisciplinární problém, je práce určena jak odborníkům na softwarové, tak i na znalostní inženýrství. Nabízí netradiční pohled, který oběma skupinám může poskytnou inspiraci pro jejich práci.

Zde navržený postup může sloužit jako základ pro další rozvoj uvedené problematiky. Není totiž v silách jednoho jedince navrhnout a implementovat celý systém.

Literatura

[1] C. Altrock, “Fuzzy logic & NeuroFuzzy applications in business & finance”, *Prentice Hall PTR*, ISBN: 0-13-368465-2, 1995.

[2] S. W. Ambler, “Architecture and Architecture modeling Techniques”, http://www.agiledata.org/essays/enterpriseArchitectureTechniques.html, *first published 2002*, [22. 5. 2004].

[3] P. Berka a kol. “Expertní systémy”, *skripta VŠE*, 1998.

[4] P. Berka, “Dobývání znalostí z databází”, *Academia*, ISBN 80-200-1062-9, 2003.

[5] M. Beran, J. Wiederman, “Kogitoid – operační systém myslícího počítače”, *Vesmír*, vol. 10, ročník 81, pp. 576–578, ISSN: 0042-4544, 2002.

[6] J. Bůcha, “Strojové učení”, *Vesmír*, vol. 9, ročník 80, pp. 503–505, ISSN: 0042-4544, 2001.

[7] J. Černý, “Dějiny lingvistiky”, *Votobia*, ISBN: 80-85885-96-4, 1996.

[8] A. Cawsey, “The essence of Artificial inteligence”, *Prentice Hall Europe*, ISBN: 0-13-571779-5, 1998.

[9] P. Coveney, R. Highfield, “Mezi chaosem a řádem”, *Mladá Fronta*, ISBN: 80-204-0989-0, 2003.

[10] F. Daneš, Z. Hlavsa, a kol. “Větné vzorce v češtině”, *Academia*, 1987.

[11] E. W. Dijsktra, “A discipline of programming”, *Prentice–Hall*, Englewood Cliffs, 1976.

[12] F. Daneš, M. Grepl, Z. Hlavsa, “Mluvnice češtiny 3 – Skladba”, *Academia*, 1987.

[13] P. Drbal, a spolupracovníci, “Objektově orientované metodiky a metodologie”, *skripta VŠE*, ISBN: 80-7079-740-1, 1997.

[14] P. Drbal, “Pojmový model češtiny – aplikace objektového přístupu (The concept Model of Czech Language – Object approach)”, *OBJEKTY'2001 Praha*, ISBN 80-231-0829-X, 2001.

[15] P. Drbal, “Jak vytvořit a zkontrolovat vlastní metodiku (The Creating and Testing of The Special Method)”, *OBJEKTY'2001 Praha*, ISBN 80-231-0829-X, 2001.

[16] P. Drbal, “Metodiky, extrémy a praxe (Methods, Extremes and Praxis)”, *OBJEKTY'2002 Praha*, ISBN 80-231-0947-4, 2002.

[17] J. Fiala, “Čím hloupější jsou lidé, tím inteligentněji vypadají počítače”, *Vesmír*, vol. 10, ročník 81, pp. 554–556, ISSN: 0042-4544, 2002.

[18] E. Gamma, R. Helm, R. Johnson, J. Vlissides, “Design patterns: elements of reusable object-oriented software”, *Addison-Wesley*, Boston, ISBN: 0201633612, 2001.

[19] M. Grepl, P. Karlík, “Skladba češtiny”, *Votobia*, ISBN: 80-7198-281-4, 1998.

[20] I. Havel, “Člověk k obrazu počítače”, *Vesmír*, vol. 10, ročník 81, p. 543, ISSN: 0042-4544, 2002.

[21] I. Jacobson, G. Booch, J. Rumbaugh, “The Unified Software Development Process”, *Addison-Wesley*, ISBN: 0201571692, 1999.

[22] R. Jiroušek, “Metody reprezentace a zpracování znalostí v umělé inteligenci”, *skripta VŠE*, ISBN: 80-7079-701-0, 1995.

[23] G. J. Klir, B. Yuan, “Fuzzy sets and fuzzy logic”, *Prentice-Hall New Jersey*, ISBN 0-13-101171-5, 1995.

[24] J. Kosek, “XML pro každého”, *Grada Publishing*, ISBN: 80-7169-860-1, 2000.

[25] J. Kosek, M. Šimůnek, “Systém TOPIC – Příručka uživatele”, *skripta VŠE*, ISBN: 80-7079-907-5, 1996.

[26] M. Labský, “Projekt Slovní druhy”, *pracovní text*, 1999.

[27] J. Lampinen, “A bibliography of Differential Evolution Algorithm”, *Lappeenranta University of Technology, technical report*, http://www.lut.fi/∼jlampine/debiblio.htm .

[28] A. Lukasová, “Reprezentace znalostí v asociativní sítích”, *Znalosti 2001*, ISBN: 80-245-0190-2, 2001.

[29] V. Mařík, O. Štěpánková, J. Lažanský, a kolektiv “Umělá inteligence”, *Academia*, vol. (1) a (2), ISBN: 80-200-0502-1, 1997.

[30] D. Mitra, W. P. Bond, “Component-Oriented Programming as an AI-Planning Problem”, *In Proceedings of 15th International Conference on Industrial and Engineering. Applications of Artificial Intelligence and Expert Systems, Editors: T. Hendtlass, M. Ali (Eds.), Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, ISSN: 0302-9743.

[31] T. Mitchell, “Machine Learning”, *McGraw-Hill*, Boston, ISBN 0-07-042807-7, 1997.

[32] V. Novák, “Fuzzy množiny a jejich aplikace”, *SNTL Praha*, ISBN 99-00-01390-X, 1988.

[33] V. Novák, “Fuzzy logika v reprezentaci znalostí”, *tutoriál Znalosti 2003 Ostrava*, 2003.

[34] H. Palovská, “Kontextové vyhledávání pomocí konceptuálních grafů”, *Znalosti 2003 Ostrava*, ISBN: 80-248-0229-5, 2003.

[35] J. Polák, V. Merunka, A. Carda, "Umění systémového návrhu, Objektově orientovaná tvorba informačních systémů pomocí původní metody BORM", *Grada Praha*, ISBN: 80-247-0424-2, 2003.

[36] L. Popelínský, "Strojové učení a přirozený jazyk", *tutoriál Znalosti 2003 Ostrava*, http://www.fi.muni.cz/∼popel, ISBN: 80-248-0229-5, 2003.

[37] L. Popelínský, T. Pavelek, "Mining lemma disambiguation rules from Czech corpora", *Proc. of 3rd Eur. Conf. PKDD'99, Prague Czech Republic*, vol. LNCS 1704, pp. 498–503, 1999.

[38] I. Polášek, P. Pavlák, "Využitie znalostného inžinierstva pri aplikácii vzorov", *Systémová integrace*, vol. 12, pp. 13–30, ISSN: 1210-9479, 2003.

[39] J. Rumbaugh, M. Blaha, W. Premerlani, F. Hedí, W. Lorensen, "Object-oriented Analysis and Design, OMT Methodology", *Prentice-Hall*, 1996.

[40] M. Ševčenko, "Online Presentation of an Upper Ontology", *Znalosti 2003 Ostrava*, pp. 153–162, ISBN: 80-248-0229-5.

[41] M. Šimůnek, "Přínos objektových přístupů pro porozumění nestrukturovanému textu", *diplomová práce VŠE*, 2000.

[42] F. Slanina, M. Kotrla, "Sítě „malého světa"", *Vesmír*, vol.11, ročník 80, pp. 611–614, ISSN: 0042-4544, 2001.

[43] J. Šlechta, "Použití jazyka UML při vývoji systémů pracujících v reálném čase", *Tvorba software Ostrava*, ISBN:80-85988-83-6, 2003.

[44] M. Šlouf, "Projekt Větné vzorce", *pracovní text*, 1999.

[45] J. F. Sowa, "Knowledge Representation, logicla, philosophical and computation foundations", *Brooks/Cole*, 2001.

[46] R. Storn, K. Price, "Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", *Technical Report*, vol. TR-95-012, http://www.icsi.berkeley.edu/techreports/1995.abstracts/tr-95-012.html, ICSI, March 1995,

[47] V. Svátek, M. Labský, "Objektové modely a znalostní ontologie – podobnosti a rozdíly", *OBJEKTY*, 2003.

[48] V. Svátek, "Ontologie a WWW, Katedra informačního a znalostního inženýrství", *VŠE Praha*, http://nb.vse.cz/∼svatek/onto-www.doc, úplná verze článku pro konferenci DATAKON 2002.

[49] Z. Telnarová, "Modelování znalostí", *Tvorba software Ostrava*, ISBN: 80-85988-83-6, 2003.

[50] J. Voříšek, "Strategické řízení informačních systémů a systémová integrace", *Management Press Praha*, ISBN: 80-85943-40-9, 1997.

[51] P. Vydržal, "Je perspektivní použití strojového učení pro desambiguaci významu slov v češtině?", *Znalosti 2001*, ISBN: 80-245-0190-2, 2001.

[52] L. A. Zadeh, "Fuzzy sets", *Information and Control*, vol. 8, 1965.

[53] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. Systems Man Cybernet*, vol. 3, Num. 1, 1973.

[54] I. Zelinka, J. Lampinen, L. Nolle, "SOMA – Self-Organizing Migration Algorithm", *Folia Facultatis Scientiarum Naturalium Universitatis Masarykiana Brunensis*, vol. 11, 2002.

Security in Mobile Environment

Post-Graduate Student:
ING. ROMAN ŠPÁNEK

Department of Software Engineering
Faculty of Mechatronics
Technical University of Liberec

Liberec, Czech Republic

roman.spanek@vslib.cz

Supervisor:
ING. JÚLIUS ŠTULLER, CSC.

Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

stuller@cs.cas.cz

Field of Study:
Electroengineering and informatics
Classification: 2612v045

Abstract

Advances in cellular mobile technology have engendered a new paradigm of computing, called mobile computing. New challenges have arisen and solutions are proposed based on various approaches. One of the most important challenges is security which has been found nowadays ubiquitous in computing as a whole. The paper[1] presents a quick survey emphasizing security paradigm and also ad hoc networks are kept in mind and briefly discussed.

1. Introduction

Several challenges exist in the mobile environment which is generally divided into a collection of cells operated by base stations (BS) located in the center of each cell. A mobile database system is depicted on Figure 1 on the next page. One or more BSs is connected with a Base Station Controller (BSC), which coordinates BSs using locally stored software and commanded by the Mobile Switching Center (MSC). A fixed host is a set of general purpose computers connected with BSs through a high-speed wired network. Database Servers (DBS) realize data processing without affecting the mobile network. DBS communicate with Mobile Units (MU) only through a BS. Every MSC contains Home Location Register (HLR) which keeps user profiles and the real-time client location. MSC, in addition, contains also Visitor Location Register (VLR) with information about users who are actually within the MSC cells. When a MU moves out from current cell to another which is operated by different MSC, a new tuple is added into the VLR registry and the HLR is also updated accordingly. This is called *two-tier* architecture and makes user's location transparent to MSCs and therefore MUs. Through the MSCs mobile units can communicate to the Public Switched Telephone Network (PSTN).

Rest of the paper is organized as follows: section 2 summarizes main issues in the mobile computing and some possible solutions are also briefly sketched. Section 3 is dedicated to the proposed security algorithm and personalization is also mentioned as our next research direction. The ad hoc network and related problems are mentioned in section 4. Section 5 concludes the paper and a brief overview on our future research is presented.

---

[1] The work was partially supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realisation".

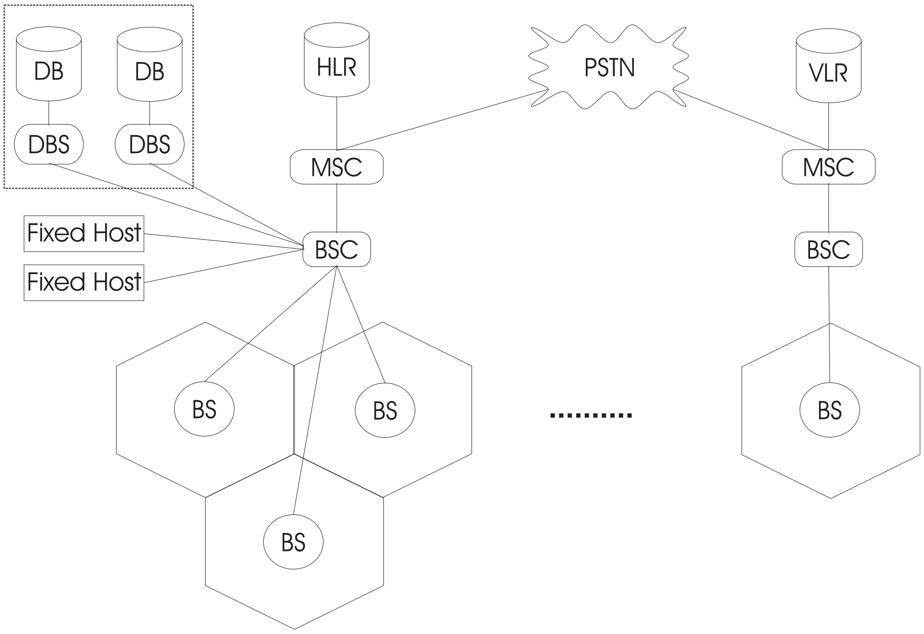


Figure 1: Mobile database system architecture

2. Main issues in mobile computing

2.1. Handoff

For a MU freely moving through the cellular network when crossing a cell boundary the corresponding signal level decline under minimum threshold and network disconnection consequently occurs. Therefore the MU has to switch the BS (this is called *Handoff*).

Three handoff strategies have been proposed:

- *Mobile-controlled handoff* (the MU continuously monitors the signal level and when it decreases under predefined threshold the handoff procedure is initiated);
- *Network-controlled handoff* (BSs measure the signal level and issue handoff process);
- *Mobile-assisted handoff* (the MU is responsible for measuring signal level but MSs are responsible to issue handoff procedure).

When the MU's signal level decreases under minimum acceptable level, the BS disconnects the MU from the network and sends messages to BSs in way to found one which will be able to serve the MU in movement. The selected BS establishes a new communication channel and the MU continues in the new cell with the new BS serving its requests. This approach is called *HARD HANDOFF* because the MU is disconnected from the mobile network for a while. In spite of the precedent disadvantage this kind of handoff is broadly used in the cellular networks all over the world.
The different approach, called *SOFT HANDOFF*, uses the different schema how a new link between a MU and a BS can be established. When handoff occurs the MU is in short time connected to the both BSs, to one to which it has been connected and to one to which it is being connected. In this approach the MU is all the time connected with a BS and is able to continuously broadcast.

2.2. Throughput

The limited wireless line throughput is a very strict constraint which is usually mentioned by all proposals. Proposals have to take into account also MUs' restricted battery power, quite unstable wireless lines with unpredictable handoffs and network disconnections so that most load and network conduct ought to be served by wired lines and powered BSs. Wired lines between BSs can be treated as sufficiently efficient and via those lines ought be sent messages without causing any obstacles to the mobile network.

2.3. Channel reusing

Amount of channels is obviously limited to a quite small number. Because of this limitation a channel reusing is used. The adjacent BSs use different channels so that no interference can occur. Those channels are reused by BSs located within the sufficient radius so the interference is under acceptable threshold. This schema suffers with inefficient channel utilization because the BSs under heavy traffic require more channels then the idle BSs. To cope with this limitation the Dynamic Channel Assignment (DCA) has been proposed [1], [2]. No channels are initially assigned to the cells and the channels are allocated on a BS's demand when necessary. Some additional schemas, based on the DCA, have been proposed like i.e. the Scheduled Channel Assignment (SCA). The SCA estimates traffic's and movement's peaks and the channels are allocated with respect to these peaks.

Quite different approach has been proposed in [3]-[6]. In this approach each BS has assigned finite number of channels. When BS is becoming *hot* (has only few free available channels) the channel borrowing algorithm is triggered. This algorithm takes into account information about the adjacent *hot* BSs and transfers free channels from *cold* (having plenty of available channels) BS.

2.4. Data management and location dependent data

The mobile data management as whole presents many challenges. Some of them will be addressed in the next lines.

When a MU issues a request for a data stored on a wired server, the BS sends it to the wired network and also receives the reply. But the MU location may be changed so that handoff would occur. Furthermore MU would have been disconnected from the mobile network (e.g. battery failure, line failure, i.e.). Therefore the requested data have to be sent to appropriate BS if MU has changed location or will have to be processed in different way if the MU has been disconnected from the network.

The location dependent data are frequently addressed in the mobile computing.
Common query like: "City of bird", "Mother's maiden name", etc. usually fetches the same data, independent on the location where it has been issued.
On the other hand query issued by user through its phone: "Where is the nearest hospital?" fetches different data with respect to the MU's location. This type is referred as "*Location Dependent Data (LDD)*" and the previous one as "*Location Free Data (LFD)*". The LDD gives rise to *Location Dependent Query (LDQ)* and *Location Aware Query (LAQ).* MU location is therefore required to be transparent to data source handling the requested (hospital in our example) information.

This is usually addressed as *Location management*. The different approaches can be used to locate MU in the mobile network and message consuming has to be considered again. The first one is called the *Deterministic approach* and the MU location is periodically updated by sending the location message. Choosing the interval and the condition for the location message issuing can be found as the main differences between the approaches. The *Probabilistic approach* on the other hand uses MU's movement patterns and likelihood's algorithms to manage MU's location. The location management belongs to one of the most important paradigms in the mobile databases.

2.5. Transaction management

Transaction management in the mobile computing is quite similar with distributed database systems. Each transaction is divided into *Fragments* usually executed on different places and also being location dependent. *Location Mapping* is consequently used to choose the geographic location where the requested data are stored. A mobile transaction definition follows.

A Mobile Transaction is a triple $\langle F_i, L_i, FLM_i \rangle$ *where* F_i *is a set of execution fragments,* L_i *a set of location, and* FLM_i *is a set of fragment location mappings.*

Due to handoff and low wireless line throughput in the mobile environment it is very difficult to support transactions with the traditional two or three phase commit protocols broadly in use in stationary (database) systems. Thereby new transaction methods have been found. One solution has been proposed by V. Kumar in [7], solving problem of unstable wireless line with unpredictable handoffs and limited throughput by a time stamp. The time stamp is used, the transaction's participants wait until the time stamp exceeds and only if all transaction's participants have replied commit message the transaction is committed (otherwise is aborted). So the time stamp has to be set very carefully. Too large may cause an unnecessary delay, but too short may cause aborting transaction despite of its correctness.

2.6. Ad hoc network

A network without BSs and stationary units is referred to as an *Ad hoc network*. The network structure without fixed infrastructure is built on freely moving MUs, which communicate with their neighbors (MUs in transmission range), and act also as routers for packages which are not addressed for them. In such an environment security problems are greater due to absence of any authority (like a BS in the cellular network) responsible for management of packets and authentication procedures.

2.7. Security

Security problems can be found in almost all environments and are common for mobile and traditional computing. The mobile environment face us with new obstacles and questions. Sharing information inside a selected group of users in a simple form with respect to the bandwidth utilization is one of the most important. A security scheme based on the grouping algorithm has been proposed in [8]. Grouping and personalization are more precisely described in the next section.

3. Grouping and personalization

Some proposals have addressed the security problem and some of them employed the grouping algorithms. Sharing secure information is very difficult to achieve in such unstable environment (as the mobile one) with a permanent threat of a tap. Tapping can be partially solved by cryptography algorithms with public key encryption.

3.1. Grouping algorithm

A grouping algorithm has been proposed in [9] but with a limited number of members. A different solution has been mentioned in [8]. Author started with a group as a base unit for the whole human society and employs the Hyper Graph theory. The Hyper graphs are used because of semantics which can provide solution for groups with huge amount of users, where it is inefficient to store complete data about each user by each user. In this approach user stores *secure cookie* (SC) [10] for holding information. The SCs are used instead of the traditional cookies because of enhanced security. In contrast to the traditional cookies which stored in a simple text format and can be easily stolen by malicious users, the SC includes necessary user's information: *Group Name, User Name, Cookie Time Stamp, User Trustiness Value, User Group ID, User Password, User IP Address and Seal Cookie* which is made as digital signature of all precedent values using the public key encryption and prevent malicious users into change the SC. The SC is subsequently used for user authentication to the group of which he is a member. The semantics given by the hyper graphs has power to store and manage a large groups in a simple way.

Figure 2 shows a hyper graph structure. Mu1 is the vertex representing particular mobile unit; r_1 is its role in the group and a_1 is the association which is responsible for interconnecting MUs from the same group. Vertex representing MU's *rank (Mu1)* is linked through the *meta-incidence* (i_1), the *meta-edge* (m_1) and

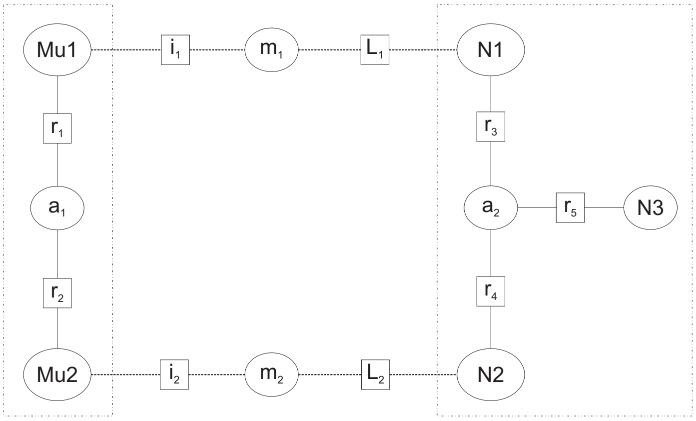


Figure 2: Hyper graph representing the group structure, MU's role and rank in the group

the *lift-incidence* (L_1) with connected component built from the vertices N1, N2, N3; the roles r_3, r_4, r_5 and the association a_2 which is again used as interconnection for common vertices. Assume that N1 has the "Sale Manager" value, N2 has the "IT Manager" and their common type N3 has therefore the "Manager" value. Roles for the mentioned vertices can be left blank or can be connected to a connected component. The MU's roles r_1, r_2 have to be linked in a similar way with connected components so that appropriate semantic is given. The semantics can be easily revealed and reads as follows: The Mobile unit 1 has the nexus "Sale Manager" and its role in the group is "Trusted Member" (note that the value "Trusted member" has been assumed for the role r_1 and has been achieved through link with a connected component). In a similar way the role and the nexus of the Mu2 can be achieved. Users simply ask the system if the user demanding data is either trusted or not. This information is derived from the hyper graph structure. So the user issues only yes/no query and system replies with minimum wireless messages optimally including yes/no value. Relations between users are therefore transparent to each.

Meta-incidences and meta-edges are used for interconnecting different connected components. Lift-incidences are also used for interconnection as well but have act as a direction manager, so that a server managing a group can easy distinguish which is either vertex represents a MU or vertex represents a nexus.

When a MU is about to built a group, the basic connected component represents basic roles (e.g. "Administrator") and nexuses (e.g. "Group Creator") have to be built. After the precedent step users are allowed to join the group by connecting its roles and nexuses. User can join a group on invitation issued by the trusted user or after completing the group's prerequisites (e.g. publishing on valuable conference). Note that the similar process can be found in the human society.

Each group user has its own trustiness value which is used for the user behavior validation. The trustiness value can be under evolution. When a user behavior is very valued its trustiness value is increased and with respect to this value the user's group nexus and rank can be enhanced. When a new user join the group its trustiness value is set to default (it is usually quite small) value. With respect to the user's behavior and its group assets it is either increased or decreased. Users with sufficient authority (derivate from its nexuses and roles) can make connected components for their own purposes, manage roles and nexuses of other participants and they can invite a new user.

Important aspect of proposed approach is that the whole structure is built on the hyper graph theory and no additional tools are required.

3.2. Personalization

Personalization is the very important research stream. It is also based on the human society behavior; humans need their privacy, their living space; personalization brings those prospects to computing. This will be our next research direction and security problems can be narrowly addressed with the personalization.

The first step was made by grouping algorithm and the next step will be creating both the living space and the privacy for each user in a group so that its good wouldn't be broken.

On the other hand personalization brings more obstacles because it is straightforwardly opposite to data sharing in the simplest form as possible.

4. Ad hoc network

The ad hoc network [11] absents any stationary and trusty structure like a BS in the cellular network. MUs are responsible for packets forwarding, routing and service discovery. From this environment new challenges are raised and have to be solved because of permanently growing amount of such networks and customer demands.

The ad hoc network is permanently changing, because of MU movement and can be imagined as a cellular network where MUs acts instead of BSs. From this specification have risen two different kind of attack:

- active; where a misbehavior node consume some energy to perform harmful operation; nodes acts in this kind of misbehavior are called *malicious*
- passive; consists of lack of cooperation and consequent harmful operations; nodes performing the passive attack to save energy are considered as *selfish*

Malicious nodes can brake down packets forwarding by *modifying routing information*, by *fabricating false routing information* and by *impersonating other nodes* in the network. Recent studies have revealed the new attack known as the *wormhole* attack. In the wormhole attack case the malicious node sends packets via tunnel to another network through a private network and shared them with other malicious nodes. Dangerousness of wormhole attack and its difficult revelation is gained by the routing protocols which try finding shortest path from a packet's source and its destination. From this viewpoint the wormhole nodes act as the shortest path.

Another kind of harmful behavior is *spoofing* when a malicious node impersonates legitimate nodes. *Integrity attack* should be also kept in mind. In this kind of attack malicious nodes alter protocols fields in order to deny communication with the legitimate nodes (it is also known as denial of service).

Several proposals have been aimed to solve precedent security problems [12] and most of them solved the active attacks with successfulness but the passive attack remains only half solved.

5. Conclusions

The mobile computing and mobile databases are quickly growing and evolving area with quickly increasing number of users. This part of computing brings both new possibilities and obstacles indeed.

The entire paper is dedicated to the mobile computing and brings an overview on the obstacles and their solutions as have been proposed in the recent years. The security part is emphasized and solution based on the paper proposed by the authors is described more precisely. The ad hoc networks are also briefly taken in account and the problems raised from the specific environment are sketch with the possible solutions.

Next research will be dedicated to enhance security schema based on the grouping algorithm and also the implementation task will be considered. For that purpose various mathematical theories will be taken in account to propose the one with the simplest and most efficient implementation. The ad hoc network and its security tasks will be kept in consideration as well.

Next turn will be personalization task. It is a very important question for the human society and the computing as whole.

References

[1] S. Nanda, D.J. Goodman, “Dynamic Resource Acquisition in Distributed Carrier Allocation for TDMA Cellular Systems”, *Proceedings GLOBECOM*, pp. 883–888, 1991.

[2] E. Re, R. Fantacci and G. Giambene, “Handover and Dynamic Channel Allocation Techniques in Mobile Cellular Networks”, *Transactions on Vehicular Technology*, vol. 44, 1995.

[3] S.K. Das, S.K. Sen and R. Jayaram, “A Novel Load Balancing Scheme for the Tele-Traffic Hot Spot Problem in Cellular Networks”, *ACM/Baltzer Journal on Wireless Networks*, vol. 4, 4, pp. 325–340, 1998.

[4] S. Mitra, S. DasBit, “Load Balancing Strategy Using Dynamic Channel Assignment and Channel Borrowing in Cellular Mobile Environment”, *Proceedings, International Conference, ICPWC*, pp. 278–282, 2000 (December). Architechture and Protocols, 1991 October.

[5] S.K. Sen, P. Agrawal, S.K. Das and R. Jayaram, “An Efficient Distributed Channel Management Algorithm for Cellular Mobile Network”, *IEEE International Conference ICUPC*, 646–650, 1997 October.

[6] I.I. Jiang, S.S. Rappaport, “Cbwl: a New Channel Assignment and Sharing Method for Cellular Communication Systems”, *IEEE Transactions on Vehicular Technology*, vol. 43, 1994 May.

[7] V. Kumar, N. Prabhu, M. Dunham and Y.A. Seydim, “TCOT - A Timeout-Based Mobile Transaction Commitment Protocol”, *Special Issue of IEEE Transaction on Computers*, vol. 51, No. 10, pp. 1212–1218, 2002.

[8] R. Spanek, “Security in Mobile Environment base on Grouping Algorithm”, in preparation.

[9] P.K. Behera, P.K. Meher, “Prospects of Group-Based Communication in Mobile Ad hoc Networks”, *Springer-Verlag Berlin Heidelberg*, 2002.

[10] J. Park, R. Sandhu and S. Ghanta, “RBAC on the Web by Secure Cookies”, *Database Security XIII: Status and Prospects*, Kluwer 2000.

[11] Basagni, “Remarks on Ad Hoc Networking”, *Springer-Verlag Berlin Heidelberg*, 2002.

[12] Molva R., Michiardi P., “Security in Ad Hoc Networks”, *Springer-Verlag Berlin Heidelberg*, 2003.

MUDRLite - Health Record Tailored to Your Needs

Post-Graduate Student:
MGR. JOSEF ŠPIDLEN
EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

spidlen@euromise.cz

Supervisor:
RNDR. ANTONÍN ŘÍHA, CSC.
EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2

182 07 Prague 8

riha@euromise.cz

Field of Study:
Biomedical Informatics
Classification: 3918V

Abstract

Nowadays most hospitals use an electronic form of health records included into their hospital information systems, but these systems are more focused on the hospital management part than the clinical one. Their usage is more suitable for the hospital management than for physicians. The health record in the information system is not as much structured as necessary, it includes a lot of free-text information, e.g. discharge letters, and the set of structured collected attributes is fixed and practically impossible to be extended. Physicians gathering information for the purpose of medical studies often use various proprietary solutions based on MS Access databases or MS Excel Sheets.

The EuroMISE Centre - Cardio is developing an electronic health record (EHR) application called MUDRLite, which could easily fill the gap among existing EHRs. MUDRLite is being created within the applied research in the field of EHR design, which is based on experience gathered during cooperation in the TripleC project. MUDRLite development is an extra branch in the MUDR (MUltimedia Distributed Record) development within my postgraduate study.

MUDRLite itself is a kind of interpreter, which has to be filled in with a configuration XML file. The XML file completely describes the visual aspects and the behavior of the EHR application. It includes simple 4GL-like constructions written in the MUDRLite Language. This enables - using the event-oriented programming principles - to program various handling of range of actions, e.g. filling a form with a result of an SQL statement after clicking on a button. MUDRLite can be tailored to particular needs of a health care provider. That makes the MUDRLite application easy to use in a specific environment. In the first instance, we are testing it in the Neurovascular Department of the Central Military Hospital in Prague.

1. Introduction

The European Centre for Medical Informatics, Statistics and Epidemiology - Cardio (EuroMISE Centre - Cardio) focuses on new approaches to the electronic health record (EHR) design, including electronic medical guidelines and intelligent systems for data mining and decision support [1]. Cooperating in those research tasks within my postgraduate study I concentrate mainly on the EHR architecture and data storing principles. The participation of EuroMISE in the project I4C-TripleC [2, 3, 4] of the 4th Framework Program of the European Commission as well as the CEN TC 251 standards and the cooperation with physicians produced much experience, which resulted into a list of 15 requirements on EHR systems [5].

To realize an EHR system, which would fulfill these requirements, EuroMISE Centre is developing an EHR application called MUDR (MUltimedia Distributed Record) [6, 7, 8, 9, 10]. MUDR have it's origin in my diploma thesis [11] which is now being extended, reevaluated, new features are being added etc. Following the requirements stated in [5], the modular structure of the system was defined. It is based on a 3-tier architecture, using a database layer, an application layer and a user interface layer, which enables the separation of physical data storage, application intelligence and the client applications.

The set of collected attributes varies in different departments, organizations and also during time. MUDR uses a dynamically extensible and modifiable structure of items based on a so-called *knowledge base* and *data file* principles as mainly described in [6, 10]. This approach allows the reorganization without change of database structure. It makes the system absolutely universal, but it brings also complications. It is quite difficult to develop universal user interfaces, which would be friendly and comfortable enough. Deploying the MUDR health record into a particular environment demands some effort; the knowledge base must be modeled and built, all the MUDR components must be installed and configured.

Currently most hospitals use an electronic form of health records included into their hospital or clinical information systems, but these systems are often more concentrated on the hospital management part than the clinical part. The usage of such systems is more suitable for the hospital management than for physicians. The health record is not structured as much as necessary, it includes a lot of free-text information, and the set of collected attributes is fixed and practically impossible to be extended. Physicians gathering information for the purpose of medical studies often use various proprietary solutions based on MS Access databases or MS Excel Sheets. MUDR usage in such cases is possible, but this solution may be too complicated and unavailing. Furthermore, the result may not be as user-friendly as a special application dedicated to particular user needs. Those were main reasons why to start another research branch called MUDRLite.

2. MUDRLite

The usage of MUDRLite health record would be an easier solution. MUDRLite is also created within the applied research in the field of EHR design; MUDRLite development is an extra branch in the MUDR development and a part of my postgraduate study; it simplifies both the MUDR architecture and the MUDR data-storing principles.

2.1. MUDRLite Architecture

MUDRLite architecture is based on 2 layers. The first one is a relational database. Currently, MS SQL server versions 7 and 2000 are supported. The second layer is a MUDRLite User Interface running on a Windows based operating system.

The database schema corresponds to the particular needs and varies therefore in different environments, unlike fixed database schema in the MUDR data layer. MUDRLite universality is based on a different approach. The database schema can be designed using standard data modeling techniques, e.g. E-R Modeling. MUDRLite User Interface is able to handle various database schemas. This feature often simplifies the way of importing old data stored using different databases or files.

2.2. MUDRLite User Interface

All the visual aspects and the behavior of the MUDRLite User Interface are completely described by an XML file. The end-user sees a set of forms. A form can be defined by a form element as follows:

```
<form name="new_hosp_form" label="New Hospitalization" author="JS"
 date="27.1.2004" language="en" sizeX="420" sizeY="410"> ... </form>
```

The attributes describe the internal name of the form, the label, which will be presented to the user, who and when has created the form, which language is used in the form and the visual size of the form. The controls on the form are described using various sub-elements like `<button>`, `<combobox>`, `<groupbox>`, `<textbox>`, `<datagrid>`, `<checkbox>` etc.

A control is placed on a form using following syntax:

```
<label name="lbl_patient" label="Patient:" posX="200" posY="315"
 sizeX="80" sizeY="23" tabIndex="2" color="indigo">
```

The attributes `name`, `label`, `posX`, `posY`, `sizeX`, `sizeY`, `tabIndex` and `color` are coincidently used in all elements describing various controls on a form. They describe the internal name of the control, the label presented to the user, position and size of the control, the tab index and the color of the control. Translating just the labels in the configuration file a new localized MUDRLite User Interface can be created very fast. A choice of over 160 color names is at your disposal. The names are defined in the GDI+ library of the .NET Framework [12]. They cover as well the usual colors as the system defined colors, e.g. the application workspace color, the window color, the window frame color etc. There are some simple elements, e.g. `<groupbox>` or `<label>`, which use just these attributes. Another attribute called `readonly` is very commonly used to determine the possibility of editing a value. More attributes are used by the other elements that will be described in more detail.

The most typical control is a text box defined by the `<textbox>` element. The additional attributes `acceptsReturn` and `acceptsTab` determine, whether the user can enter line breaks and tabulators. By the `multiline` attribute the form designer enables viewing more than one line of text in the text box. Scrollbars are placed using the `scrollbars` attribute, which can take on `none`, `horizontal`, `vertical` and `both` values.

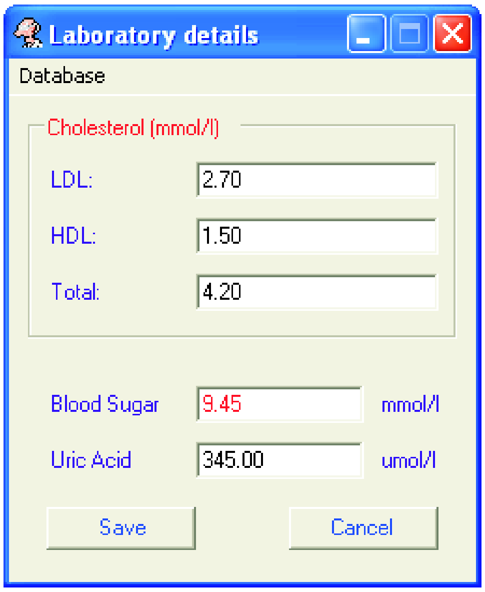
A data grid is also a common control element in database applications. In MUDRLite the `<datagrid>` element uses the `colwidth` attribute to set the preferred column width. Other visual aspects like the columns's titles are being set using the same techniques as for setting the data grid content. They are described later.

Enumerative variables are often bound using a combo or a list box controls. A `<combobox>` element uses the `maxDropDownItems` numerical attribute to specify, how many items can be shown together. The `sorted` attribute determines whether the items should be alphabetically sorted. There are two different styles of the combo box control, a list style, where the user can just select a value from a fixed list, and an editable style, where adding new values to the list is possible. The style is being set by the attribute named `dropDownStyle`. There are two ways how to fill in the combo box with values. The first one is using the `<item>` sub-elements, which define each item by the `value` and `display` attributes. It means that the value stored in database doesn't have to be the same the user can see in the form. This helps to maintain the 3^{rd} normal form (3NF) of the database while keeping the form user friendly. A second way to fill in the combo box is to use an SQL statement. This statement will be processed while loading the form. It should return two columns: a value member and a display member; or just one column in case the value and display members are identical. Technically this is realized using the `<items>` sub-element with the `command` attribute.

There are some more controls being prepared like tree views, tab panels, image boxes, scroll bars, progress bars, status bars, tool bars and tool tips, but they have not yet been completely implemented. Their full description will be included into the MUDRLite Designer's Manual after they have been finished. A small example of a user defined form can be seen in the Figure 1. It is defined as follows:

```
<form name="lab_details" label="Laboratory details" author="JS"
      language="en" date="5.7.2004" sizeX="250" sizeY="260">

 <label    label="LDL:"   posX="20" posY="38" sizeX="80" sizeY="25"/>
 <label    label="HDL:"   posX="20" posY="68" sizeX="80" sizeY="25"/>
 <label    label="Total:" posX="20" posY="98" sizeX="80" sizeY="25"/>

 <textbox  name="t_ldl" posX="100" posY="35" sizeX="130" sizeY="15"/>
 <textbox  name="t_hdl" posX="100" posY="65" sizeX="130" sizeY="15"/>
 <textbox  name="t_tot" posX="100" posY="95" sizeX="130" sizeY="15"/>

 <groupbox label="Cholesterol (mmol/l)" posX="10" posY="10"
          sizeX="230" sizeY="120" color="red"/>
```

```
 <label label="Blood Sugar" posX="20" posY="158"
        sizeX="80" sizeY="25"/>
 <label label="mmol/l"    posX="197" posY="158" sizeX="40" sizeY="25"/>
 <label label="Uric Acid" posX="20"  posY="188" sizeX="80" sizeY="25"/>
 <label label="umol/l"    posX="197" posY="188" sizeX="40" sizeY="25"/>

 <textbox posX="100" posY="185" sizeX="90" sizeY="15"/>

 <button label="Save"   name="but_save" posX="20" posY="220"
         sizeX="80" sizeY="23" color="blue"/>
 <button label="Cancel" name="but_exit" posX="150" posY="220"
         sizeX="80" sizeY="23" color="blue"/>
</form>
```



LDL: 2.70
HDL: 1.50
Total: 4.20
Blood Sugar 9.45 mmol/l
Uric Acid 345.00 umol/l

Figure 1: Example of a simple user-defined form.

This definition creates a form as seen in the Figure 1, but there are two problems left. There is no action connected with pressing the two buttons. The other problem is that even though the form should load the details of selected patient after having started, it always opens empty. Both the problems can be solved using the MUDRLite Language.

2.3. MUDRLite Language

MUDRLite Language (MLL) is a simple event based language used to describe the behavior of the MUDRLite User Interface. It contains 4GL-like constructions, which allow processing of database operations.

The MLL constructions are included in the XML configuration file using the `<action>` element. This element can be placed as a sub-element under a form or under a control element. This determines whether the action should be bound to the entire form or just to a control element in the form. Each action starts on an event. There can be various event types. On controls, the most typical events are a click and a double click. On forms, a typical event is loading a form, closing a form, etc. Events are bound to actions by the `invoke` attribute.

There are various action types. The action content is always described in the content of the `<action>` element. Simple actions work with controls on a form. You can for example clear, hide or show a control using the `<clear>`, `<hide>` and `<show>` sub-elements. The target controls are specified by the `result` attribute referring by their names. Typically, more target controls are specified simply separating their names by commas. Sometimes a reference to a control placed on another form is needed. In this case a dot separated full name is used. It is constructed by the form name connected by a dot with a control name. A dedicated name `parent` is used to reference the parent form; that is the form, which had been active before the current form opened. Therefore no form can be named "parent". The dot separated full name can be more complicated in case you need to address a special part of a control, not the whole control. A typical example is addressing a column of a data grid. In this case a dot is used to connect the control name with the control part name, e.g. the column name of a data grid. In this case the full name can look like this: `parent.patients_grid.patient_id`.

Other action types are working with whole forms. A form can be closed by the `<exit_form>` action element. A new form can be opened by the `<new_form>` element with the `name` attribute, which specifies the form to be opened.

Most powerful actions are using the MLL Language to communicate with the database and to set/get values into/from controls. An extended form of SQL is used in the content of these actions in the `command` attribute. The control names in these commands are bounded with a pair of colon marks. The result controls are specified by their names in the result attribute. Again, more result controls can be addressed separating their names or dot separated full names with commas. A select action can be executed by this code:

```
<action invoke="load">
  <select
    command = "select t.tot, t.ldl, t.hdl, t.bs, t.uacd
               from   laboratories t
               where  t.patient = ':parent.patient_id:' and
                      t.id = ':parent.exam_grid.lab_id:'"
    result  = "t_tot, t_ldl, t_hdl, t_bs, t_ua" />
</action>
```

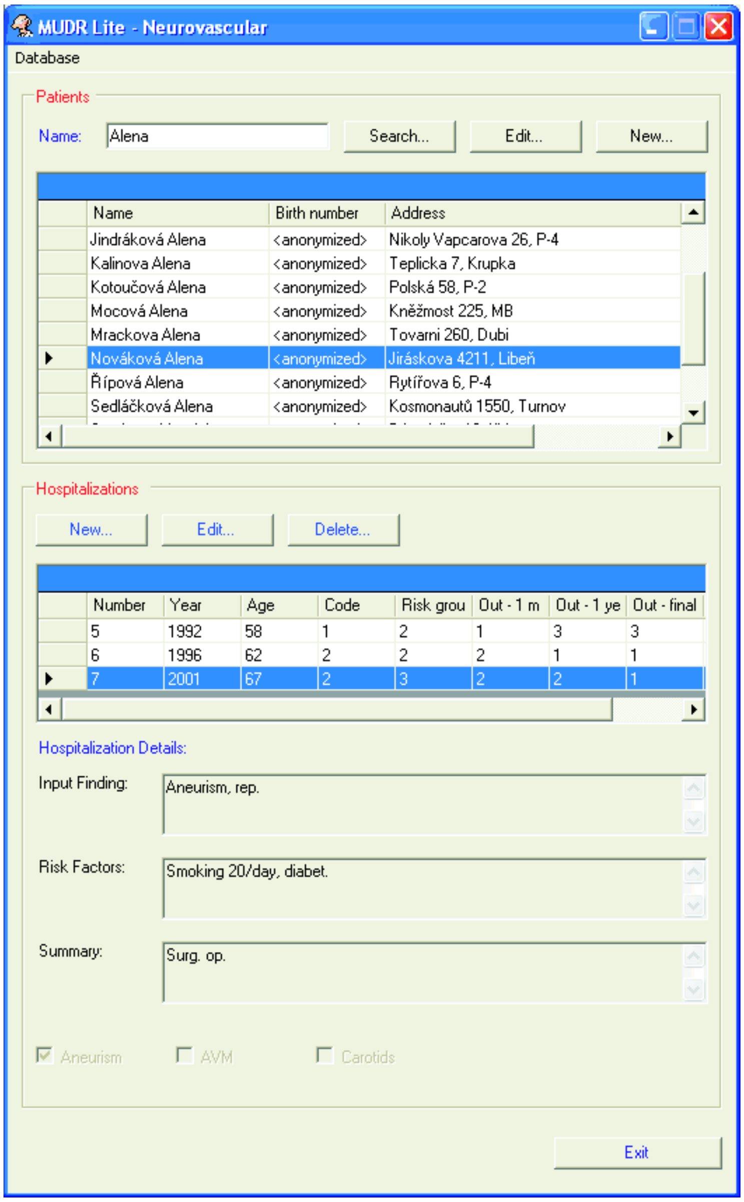
This example shows how the "Laboratory details" form is filled with details of the laboratory examination currently selected in a grid on the parent form. It also demonstrates a simple trick. The patient's identifier is stored in an invisible label, which has the role of an internal temporary variable. When the user presses the "Save" button, this variable is used to specify the patient who should be updated with an MLL update action.

Typically the count of values returning by the select command should correspond to the count of controls specifying in the `result` attribute. An exception is using a data grid as a result control. In this case the column names of the data grid are being specified by the "as" SQL expression.

Insert, update and delete operations are performing by the `<insert>`, `<update>` and `<delete>` elements using the same principles as the `<select>` element.

There is an additional action included in the first MUDRLite version. A text can be read by the computer using the `<speak>` action specifying the spoken text by the `text` attribute. So far just the English pronunciation can be used.

Of course, many different actions can be put one after another in a sequence. Figure 2 shows a more complex example – a form designed for the Neurovascular Department of the Central Military Hospital in Prague (English translation).



Name: Alena

	Name	Birth number	Address
	Jindráková Alena	<anonymized>	Nikoly Vapcarova 26, P-4
	Kalinova Alena	<anonymized>	Teplicka 7, Krupka
	Kotoučová Alena	<anonymized>	Polská 58, P-2
	Mocová Alena	<anonymized>	Kněžmost 225, MB
	Mrackova Alena	<anonymized>	Tovarni 260, Dubi
▶	Nováková Alena	<anonymized>	Jiráskova 4211, Libeň
	Řípová Alena	<anonymized>	Rytířova 6, P-4
	Sedláčková Alena	<anonymized>	Kosmonautů 1550, Turnov

	Number	Year	Age	Code	Risk grou	Out - 1 m	Out - 1 ye	Out - final
	5	1992	58	1	2	1	3	3
	6	1996	62	2	2	2	1	1
▶	7	2001	67	2	3	2	2	1

Input Finding: Aneurism, rep.

Risk Factors: Smoking 20/day, diabet.

Summary: Surg. op.

☑ Aneurism ☐ AVM ☐ Carotids

Figure 2: MUDRLite form designed for the Neurovascular Department of the Central Military Hospital in Prague.

3. MUDRLite Deployment

MUDRLite deployment to a particular environment needs preparation phases. First of all, the physicians must specify what information should be stored. This must be done precisely. A model including all collected attributes must be built. It must include the attribute types, the relationships among them, units of measurement, specification of numerical attributes' precisions etc. Together with data engineers an entity-relationship model (E-R Model) is built.

Two more phases are following in parallel. One of them is the data migration from an existing system. MUDRLite is seldom being deployed to an environment, where no data has been collected yet. This is being done using various techniques and SQL Server Data Transformation Services. The result of the second phase is the definition of MUDRLite user-defined forms and MUDRLite application behavior using the MLL Language. Various XML editors can serve to help creating the XML configuration file. A special application called ”MUDRLite Forms Designer” is being prepared to simplify this phase. After the XML configuration file is finalized, MUDRLite should be tested. Then the application is being fine-tuned according the physicians’ comments.

4. Results and Future Work

MUDRLite testing confirmed that this health record is flexible enough to allow dynamical changes of the database structure demanding no or just small changes in the configuration XML file. The XML file can be constructed using various XML editors, but we are preparing a MUDRLite Forms Designer to make this process easier. The 2-tier architecture separates the user interface from the data storing part. This enables a remote access to the health record. To make the remote access more flexible we plan to develop a Pocket PC version of the MUDRLite User Interface, which should run on various portable devices.

We have also verified the functionality and simplicity of the MLL Language. It is useful and sufficient for many applications, mostly thanks to the power of the SQL. But we still would like to increase the power of the MLL Language. We plan to do this by including arithmetical expressions and elements of logical conditions into the language. We are aware of the fact that we also have to keep it as simple as possible.

On the other hand we found out that the bottleneck that eliminates usage of the system as a clinical information system lies in the absence of interfaces to other EHR systems. That is why we have started the research how to support such interfaces. Because of the varying database layer we can not implement such support directly into the MUDRLite application but we are trying to suggest special communication modules which could be connected to MUDRLite. Each module should support one EHR data standard, e.g. the Czech data standard ”DASTA” or in Western Europe commonly used ”HL7”. We suppose there will always be another additional configuration file needed to define mapping between values stored in the database and attributes transferred via the selected standard.

5. Conclusion

Our interest is to increase the quality of EHR systems, to simplify data sharing and data migration among various EHR systems and to help in overcoming the classical free-text based health record. This is the way, which would increase the quality of healthcare, which brings benefit for the patient first of all. Most healthcare providers use a kind of an EHR system. But often the health record is not structured as much as necessary. Physicians gathering information for the purpose of medical studies often use varied proprietary methods.

We present the MUDRLite universal solution. It is an easy way to build electronic health record tailored exactly to your needs. We recognize just one problem that can emerge in case the user needs more security and finer rights policy than the database server can provide. Better policy can be hard to gain because we are mainly using the database server authentication and authorization (A/A) rules that seem to be sufficient in most environments. In the first instance, we are deploying MUDRLite to the Neurovascular Department of the Central Military Hospital in Prague.

Acknowledgment

The work was partially supported by the grant number LN00B107 of the Ministry of Education of the Czech Republic.

References

[1] J. Rauch, “Mining for Statistical Association Rules”, In: *J. Fong, M.K. Ng (eds.): The Fifth Pacific/Asia Conference on Knowledge Discovery and Data Mining*, University of Hong Kong, pp. 149–158, 2001.

[2] I. Iakovidis, "Towards Personal Health Record: Current Situation, Obstacles and Trends in Implementation of Electronic Healthcare Record in Europe", In: *Int J Med Inform*, Vol. 52, pp. 105–115, 1998.

[3] F.H. Pierik, A.M. Ginneken, T. Timmers, H. Stam and R.F. Weber, "Restructuring Routinely Collected Patient Data: ORCA Applied to Andrology", In: *J.H. Bemmel, A.T. McCray (eds.): Yearbook of Medical Informatics 98, Health Informatics and the Internet*, Schattauer, Stuttgart, pp. 257–263, 1998.

[4] A.M. Ginneken, "The Computerized Patient Record: Balancing Efort and Benefit", In: *Int J Med Inform*, Vol. 65, pp. 97–119, 2002.

[5] P. Hanzlíček, "Development of Universal Electronic Health Record in Cardiology", In: *G. Surján, R. Engelbrecht, P. McNair (eds.): Health Data in the Information Society*, IOS Press, Amsterdam, pp. 356–360, 2002.

[6] J. Špidlen, A. Říha, "Electronics Health Record and Telemedicine", In: *F. Hakl (ed.): Institute of Computer Science AS CR - PhD Conference Proceedings*, Matfyzpress, Paseky nad Jizerou, pp. 133–143, 2003 (in Czech).

[7] J. Zvárová, P. Hanzlíček and J. Špidlen, "Electronic Health Record in Cardiology: Pilot Application in the Czech Republic", In: *MIST2002 Proceedings*, John Wiley & Sons Pte Ltd., Taiwan, pp. 10–13, 2002.

[8] J. Špidlen, J. Adášková, H. Heroutová, J. Zvárová and I. Mazura, "Statistical Processing of Genetic Information in the MUDR Health Record", In: *J. Vignerová, J. Riedlová, P. Bláha (eds.): Anthropology and Society*, Charles University, Prague, p. 190, 2003.

[9] P. Hanzlíček, J. Špidlen and J. Zvárová, "The Internet in Connecting Electronic Health Record Mobile Clients", In: *Technology and Health Care*, Vol. 10, Num 6, IOS Press, pp. 502–503, 2002.

[10] J. Špidlen, P. Hanzlíček, "The Implementation of Formalized Medical Guidelines in the MUDR Electronic Health Record", In: *V. Svátek (ed.): Znalosti 2003 Proceedings of the 2nd Annual Conference*, VŠB-TU Ostrava, Ostrava, pp. 386–391, 2003 (in Czech).

[11] J. Špidlen, "Database Representation of Medical Information and Guidelines", *Diploma Thesis*, Charles University in Prague, Faculty of Mathematics and Physics, Prague, 2002 (in Czech).

[12] Microsoft Corporation, "Microsoft .NET Framework 1.1 Class Library Reference Volume 7: System.Windows.Forms, System.Drawing, and System.ComponentModel", Microsoft Press, ISBN: 0-7356-1818-6, 2003.

Probabilistic Clinical Decision Support Systems

Post-Graduate Student:
ING. FILIP ZÁMEK
EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
zamek@euromise.cz

Supervisor:
RNDR. ANTONÍN ŘÍHA, CSC.
EuroMISE Centrum – Cardio
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8
riha@euromise.cz

Field of Study:
Biomedical Informatics
Classification: 3918V

Abstract

The paper provides a brief overview of the state of the art in probabilistic medical decision support systems. An example of applied clinical decision support system is given and its deficiencies are discussed. The aims of proposed dissertation project are formulated in conclusions.

1. Introduction

Clinical decision plays a pivotal role in defining the quality and cost of medical care. Numerous computer systems have been developed to support these decisions. Generally, these systems can be regarded as a kind of expert systems. The problems that expert systems can deal with can be classified into two types: mainly *deterministic* and mainly *stochastic* problems. Consequently, expert systems can be classified into two types according to the nature of problems they are designed to solve: *deterministic* and *stochastic expert systems*. Deterministic problems can be formulated using set of rules that relates several well-defined objects. Expert systems that deal with deterministic problems are known as rule based expert systems, because they drew their conclusions based on a set of rules using a logical reasoning mechanism. In the medical field, the relationship among symptoms and diseases are known only with a certain degree of uncertainty. These kinds of problems may also include some deterministic elements, but they are largely stochastic problems because:

- The facts may not be accurate. For example, a patient may not be sure whether or not he had a fever last night. Thus, there is degree of uncertainty in the information associated with each patient (subjectivity, imprecision, lack of information, errors, missing data etc.)
- The abstract knowledge is not deterministic. The relationship among diseases and symptoms are not deterministic because the same group of symptoms may be associated with different diseases. The presence of a set of symptoms does not always imply the presence of disease.

Thus a need of an expert system that deals with uncertain situations is clear.

2. Probabilistic Expert Systems

In stochastic or uncertain situations it is necessary to introduce some means for dealing with uncertainty. For example, some expert systems use the same structure as rule-based expert systems, but introducing some measure associated with the uncertainty of rules and their premises. Some propagation formulas can be used to calculate the uncertainty associated with conclusions. Several uncertainty measures have been proposed during the last decades. Example of these measures *certainty factors*, *fuzzy logic*, Dempster and Shaffer *theory of evidence*. Another intuitive measure of uncertainty is *probability*, where *joint probability distribution* (JPD) of a set of variables is used to describe the relationship among the variables, and conclusions are drawn using certain well known probability formulas. Expert systems that use probability as measure of uncertainty are known as *probabilistic expert systems* and the reasoning strategy they use is known as probabilistic reasoning or probabilistic inference. This article is devoted to probabilistic expert systems based on Bayes theorem, used in field of medicine. To illustrate the use of Bayes theorem, suppose that a patient can be either healthy (has no disease) or have one of $m-1$ diseases $\{D_1, \dots D_{m-1}\}$. For simplicity of notation, let D be a random variable that can take one of m possible values $\{d_1, \dots, d_m\}$ where $D = d_i$ means that patient has no disease. Suppose also we have n associated symptoms $\{S_1, \dots S_n\}$. Now, given that patient has a set of symptoms $\{s_1, \dots s_k\}$ we wish to compute the probability that the patient has the disease D_i, that is $D = d_i$. Then using Bayes' theorem, we obtain

$$p\left(d_i|s_1, \dots s_k\right) = \frac{p\left(d_i\right)p\left(s_1, \dots s_k|d_i\right)}{\sum\limits_{d_i} p\left(d_i\right)p\left(s_1, \dots s_k|d_i\right)} \tag{1}$$

- The probability $p\left(d_i\right)$ is called the *marginal*, *prior* or *initial* probability of disease $D = d_i$ because it can be obtained before knowing any symptoms.
- The probability $p\left(d_i|s_1, \dots s_k\right)$ is the *posterior* or *conditional* probability of disease $D = d_i$ because it is computed after knowing the symptoms.
- The probability $p\left(s_1, \dots s_k|d_i\right)$ is referred to as the *likelihood* that patient with the disease $D = d_i$ will show the symptoms$S_1 = s_1, \dots, S_k = s_k$.

Thus we can use 1 to update the posterior probability distribution using both the prior and the likelihood.

3. Decision Support

Symptoms are observable, but diseases are not. But since diseases and symptoms are related, medical doctors usually use the symptoms as a basis for diagnosis. A difficulty of this approach, however, is that the relationship among symptoms and diseases are not perfect. By studying relationships among symptoms and diseases medical doctors gain more knowledge and experience, and hence they become more able to diagnose diseases with a high degree of certainty. However, should be recognized, that when making decision in uncertain situations, the decision can be incorrect. There are two possible types of errors in uncertain situations:

- A *false positive* decision, also known as *type I* error (A patient does not have the disease but the doctor concludes that patient has the disease)
- A *false negative* decision, also known as *type II* error (A patient has the disease but doctor concludes that the patient does not have the disease)

Thus, when making a decision, the doctor is subject to making one of the above two errors depending on the true state of nature. In some situations, however, the consequences of one error can be far more damaging than the consequences of other. Ideally doctor would keep the probabilities of making the above errors to a minimum, but the relative risks associated with the two types of errors have to be taken into consideration when making a diagnosis. To illustrate, suppose that a new patient with an unknown disease comes to the medical center. After an examination by a doctor, it is determined, that a patient has k symptoms $s_1, \dots s_k$. The question that booth the patient and the doctor need to know is, given these symptoms, which disease is the patient most likely to have? The answer to this question can be obtained by computing the posterior probabilities of $D = d$for each disease $d = d_i$ given the symptoms $s_1, \dots s_k$, that is $p\left(d_i|s_1,\dots s_k\right)$. This probability can be computed using Bayes formula. Thus given the patient has the symptoms $s_1,\dots s_k$, the doctor may conclude that the disease the patient is most likely to have is the one with $\max_i\left\{p\left(d_i|s_1,\dots s_k\right)\right\}$. If $\max_i\left\{p\left(d_i|s_1,\dots s_k\right)\right\}$ is close enough to 1, the doctor may decide that patient has corresponding disease. Otherwise, additional examination and identification of new symptoms (evidence) may be necessary. Bayes formula can still be used to compute the new conditional probabilities for each disease given all the accumulated symptoms (information). This process has to be repeated, adding more evidence, until $\max_i\left\{p\left(d_i|s_1,\dots s_k\right)\right\}$ gets close enough to 1. When this occurs, the medical doctor can make a decision and terminate the process of diagnosis. How *close* is *close enough* to 1 is usually decided by doctor, depending on risks associated with erroneous decision.

4. Knowledge Base

Main components of expert systems are shown schematically in the figure 1, where arrows represent flow of information. Expert system is usually the product of collaborative work of the *subject matter human experts* and the *knowledge engineers*, with the *user* in mind. The subject matter specialists are responsible for providing the knowledge engineers with an ordered and structured knowledge base and well defined and well explained set of relationships. The knowledge base of a probabilistic expert system consists of a set of variables $\{X_1,\dots X_n\}$ and a JPD over all the variables $p\left(x_1,\dots,x_k\right)$. Thus, to construct the knowledge base of a probabilistic expert system, wee need to specify the JPD of the variables. The most general model is based on direct specification of the JPD. That is, numerical value (parameter) is associated with every possible combination of values. Unfortunately, the direct specification of JPD involves a huge number of parameters. For example, with n binary variables, the most general JPD has 2^n parameters (the probabilities $p\left(x_1,\dots,x_k\right)$ of every possible realization $x_1,\dots,x_k$ of the variables) a number so large that no existing computer can handle it even for n as small as tens. In most practical situations, however, many subsets of the variables can be independent or conditionally independent. In such cases, simplification of the most general model can be obtained by exploiting the independency structure among variables. This may result in significant reductions of model parameters. Examples of such simplifications are so called *The Dependent Symptom Model (DSM)*, *The Independent Symptoms Model (ISM)*, *The Independent Relevant Symptoms Model (IRSM)* and *The Dependent Relevant Symptoms Model (DRSM)* explained in [1]. These four models,

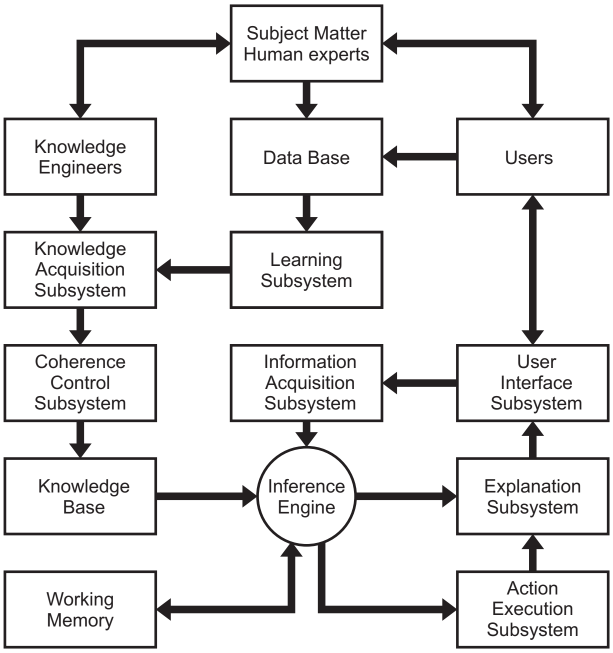


Figure 1: Typical components of an expert system

however, are ad hoc models that apply mainly in the medical field. More general and powerful probabilistic models are for example *Markov networks model*, *Bayesian networks model*, models specified by input lists and *conditionally specified models*. Whatever model is adopted the knowledge base must contain the set of variables of interest and the minimum set of parameters (probabilities) needed to specify the JPD of the variables. The *knowledge acquisition subsystem* controls the flow of new knowledge from the human experts to the knowledge base. It determines what new knowledge is indeed or whether the received knowledge is indeed new, i. e. whether or not it is included in knowledge base and if necessary, transmits them to the knowledge base. The *coherence control subsystem* is an essential component of an expert system. It controls its consistency and prevents any incoherent knowledge from reaching the knowledge base.

5. Inference Engine

The *inference engine* is the heart of every expert system. The main purpose of this component is to draw conclusions by applying the *abstract knowledge* to the *concrete knowledge*. In medical diagnosis the symptoms of a given patient (concrete knowledge) are analyzed in the light of the symptoms of all diseases (abstract knowledge). The abstract knowledge is represented by set of variables and the associated of probabilities needed to construct the JPD of the variables. This type of knowledge is stored in knowledge base. The concrete knowledge consists of a set of variables known to the user. This information is referred as *evidence set*. It is stored in a *working memory*. The inference engine uses both the abstract and concrete knowledge to answer certain questions or queries posed by the user. Examples of such queries are:

- *Before a patient is examined by a doctor, which disease is the patient most likely to have?* Here, no evidence is available. The patient is yet to be examined and the set of symptoms that the patient exhibits is yet to be examined. The problem is to compute the marginal (initial) probability distribution of D, $p\left(D=d_i\right),\ i=1,\ldots,m.$
- *Given that patient has a subset of symptoms $S_1 = s_1, \ldots, S_k = s_k$, which disease is the patient likely to have?* The evidence set in this case consists of the of the values $s_1, \ldots, s_k$. The problem at hand is to compute conditional probability distribution of D given the evidence set, $p\left(D=d_i|s_1,\ldots,s_k\right),\ i=1,\ldots,m$

The marginal probability distribution of D, $p\left(D=d_i\right)$, is also known as *prior distribution*, because this is computed before the evidence. The conditional probability distribution of d_i given a realization of the set of symptoms $p\left(d_i|s_1,\ldots,s_k\right)$ is known as *posterior distribution* because it is computed after knowing the evidence. The marginal (prior) probability distribution can be thought of as a special case of the conditional (posterior) probability distribution, where set of observed symptoms is the empty set, $\emptyset$.

One task of the inference engine in probabilistic expert systems is to compute the conditional probabilities of different diseases when new symptoms or data become known. The inference engine is responsible for updating the conditional probabilities

$$p\left(d_i|s_1,\ldots s_k\right)=\frac{p\left(d_i,s_1,\ldots s_k\right)}{p\left(s_1,\ldots s_k\right)};\quad i=1,\ldots,m \tag{2}$$

for all possible subset of the symptoms, and to decide which ones have high conditional probabilities. Normally a small nuber is selected and these are listed for the user to observe and make appropriate actions. In equation above, the role of the term $p\left(s_1,\ldots s_k\right)$ is to act as normalizing constant. Therefore, a decision based on the maximum of $p\left(d_i|s_1,\ldots s_k\right)$ coincides with that based on the maximum of $p\left(d_i,s_1,\ldots s_k\right)$. Thus, the ratios

$$R_i=\frac{p\left(d_i,s_1,\ldots s_k\right)}{\max\limits_i p\left(d_i,s_1,\ldots s_k\right)};\quad i=1,\ldots,m \tag{3}$$

provide information about the relative significance of any of the diseases. Bayes theorem is used to compute the posterior probabilities with ease when we have only a few diseases and symptoms. But when the number of variables (diseases and/or symptoms) is large, which is usually the case in practice, more efficient models and methods are needed to compute both the initial and the posterior probabilities. These methods are known as *evidence or uncertainty propagation methods*.

6. Other components

If the initial knowledge is very limited and conclusions cannot be reached, the inference engine utilizes the *information acquisition subsystem* in order to obtain the required knowledge and resume the inference process until concussions can be reached. In some cases, the user can provide the required information. A *user interface* subsystem is needed for this and other purposes. The user interface subsystem is the liaison between the expert system and the user. Thus, in order for an expert system to be an effective tool, it must incorporate efficient mechanism to display and retrieve information in easy way. The *action execution subsystem* is the component that enables the expert system to take actions. These actions are based on conclusions drawn by the inference engine. Explanation of these actions can be provided to the user by the *explanation subsystem*.

One of the main features of an expert system is the ability to learn. We shall differentiate between structural and parametric learning. By *structural learning* we refer to some aspects related to the structure of knowledge. Thus, discovery of a new relevant symptom for a given disease is an example of structural learning. By *parametric learning* we refer to estimating the parameters needed to construct knowledge base. So, estimation of probabilities associated with symptoms or diseases is an example of parametric learning. Another feature of expert system is their ability to gain experience based on available data. Collected data can be used by the *knowledge acquisition system* and by the *learning subsystem*.

7. An example - the Iliad

Comprehensive source of information about world-wide available medical expert and knowledge-based systems is [5]. This web site provides information on over 73 state-of-the-art medical expert and knowledge based systems. These systems cover several medical fields. As typical example of probabilistic expert system based on Bayesian reasoning is the Iliad system. It was developed at the University of Utah, Department of Medical Informatics. It is an expert system designed to cover all of internal medicine as well as additional specialties. It has evolved over the years from a teaching tool (internal medicine residents) to a more robust commercial product. Iliad is an expert system which can be used as a diagnostic consultant, a reference tool, and an educational system. The Iliad knowledge base covers more than 1200 diseases in the fields of internal medicine, pediatrics, dermatology, psychiatry, obstetrics and gynecology, peripheral vascular diseases and sleep disorders and provides treatment protocols for each. The systems data dictionary contains 1500 syndromes and treatment protocols and 11900 findings, which describe the diagnostic elements used in the system. More than 1400 digitized images are included for reference. Iliad offers three primary operating modes: consultation, simulation and reference. The consultation mode allows the user to enter findings, such as symptoms, physical signs and laboratory test results for actual patients. Then, Iliad generates a different list of diagnoses ranked by likelihood, also providing cost effective suggestions for confirming or ruling out any diagnosis on the differential list. Once a diagnosis is made, Iliad provides treatment suggestions for every disease in its knowledge base. As an educational tool, Iliad can generate a variety of simulated patient cases that can help students to practice and test their diagnostic skills by providing them with an opportunity to study many cases that they might only seldom encounter in practice. In this mode, Iliad evaluates the students differential diagnosis and work-up strategy. The reference mode provides the user with the ability to browse Iliads extensive medical reference library. For a diagnostic expert system, each disease or diagnostic decision may be represented as a list or table called a frame. Decision frame incorporates findings (symptoms of the disease) and associated logic. Iliad frames are represented by probabilistic (Bayesian) or Boolean reasoning (true, false — yes, no) together with heuristics (risk flags, display logic). Combination of the foundations of Boolean and Bayesian logic with heuristics result is a better model with more accurate findings and an improved user experience.

Positives

- Generated useful differential diagnoses
- Helped in multiple problem patients
- Defined what tests to order for non-routine disorders (teaching, recall)
- Reminded of less common diagnosis for patient with vague signs (e.g. vomiting, abdominal pain)

Negatives

- Difficult to use at the bedside
- Requires too much data entry - too much time required
- Thought more educational than practical
- Hasn't changed patient management decisions, but appreciated understanding of likelihood ratios.
- Usually confirms what 'I already know' or is as confused as 'I already am'.
- Knowledge base is incomplete
- Lack of integration with patient information system
- The result of some differential diagnoses may be considered irrelevant or inappropriate

8. Conclusion

Recent years have seen an enormous development in medical expert systems. Nevertheless, the systems now available still show some deficiencies (cf. those mentioned above) and have failed to gain widespread acceptation by physicians. The aim of the present project is the detailed study of methods how to remove the deficiencies, the creation of a new (prototype) implementation of a probabilistic expert system and its testing in the setting as close to the real one as possible. It will be especially necessary to interconnect (integrate) the system with an *Electronic Health Record* (EHR) which will eliminate multiple input of the same patient data and make usage of the system more comfortable for a physician.

9. Acknowledgement

This work was partially supported by the project number LN00B107 of the Ministry of Education of the Czech Republic.

References

[1] E. Castilio, J.M. Gutiérrez and A.S. Hadi, "Expert Systems and Probabilistic Network Models", Springer-Verlag, Monograph in Computer Science, 1997.

[2] H.R. Warner, D.K. Sorenson and O. Bouhaddou, "Knowledge Engineering in Health Informatics", Springer-Verlag, Computer and Medicine Series, 1997.

[3] J.H. van Bemmel, M.A. Musen, "Handbook of Medical Informatics", Springer-Verlag, 1997.

[4] E.H. Shortliffe, L.E. Perreault, "Medical Informatics: Computer Applications in Health Care and Biomedicine", Springer-Verlag, Computer and Medicine Series, 2001.

[5] `http://medexpert.imc.akh-wien.ac.at`

Ústav Informatiky AV ČR
DOKTORANDSKÝ DEN ’04

Vydal
MATFYZPRESS
vydavatelství
Matematicko-fyzikální fakulty
University Karlovy
Sokolovská 83, 186 75 Praha 8
jako svou 132. publikaci

Obálku navrhl František Hakl

Z předloh připravených v systému LaTeX
vytisklo Reprostředisko MFF UK
Sokolovská 83, 186 75 Praha 8

Vydání první

Praha 2004

ISBN 80-86732-30-4