Ivan Nagy

ROZHODOVÁNÍ A ŘÍZENÍ
v dopravě

Část III. Speciální modely

Obsah

Předmluva

Předkládaná zpráva je pokračováním seriálu zpráv, pojednávajících o využití výsledků bayesovské statistiky pro automatické řízení v dopravě. V této třetí části se budeme zabývat některými speciálními modely, vhodnými pro popis dopravních, ale jiných, dat. Jedná se o modely směsi distribucí, které umožňují pomocí množiny lineárních modelů popsat soustavy s několika stavy, i když model přepínání mezi stavy je sám o sobě nelineární. Dalším důležitým modelem je statický regresní model (případně jejich směs) s časově proměnnou konstantou, modelovanou pomocí polynomu nebo spline-funkce. Takový model je zvláště vhodný pro popis dat, u nichž je garantován určitý stupeň hladkosti. Oba dva modely jsou velmi vhodné pro popis dopravních dat.

Protože se tato práce zaměřuje především na ty, kteří se s danou problematikou teprve seznamují, je výklad zaměřen zejména na konkrétní příklady. Ty jsou ale voleny tak, aby postupy v nich použité byly snadno použitelné i pro obecnější případy.

Probírané úlohy předpokládají znalost předchozího dílu tohoto pojednání a matematiku na úrovni základního kurzu na některé z technických vysokých škol.

Další zprávy v této sérii se budou zabývat konkrétními dopravními problémy a návrhem jejich řešení, založeným na bayesovském přístupu k identifikaci a řízení.

Zpráva je určena především studentům, přicházejícím do styku s problematikou automatického řízení, aby jim pomohla překonat základní bariéru, kterou vytváří jednak poněkud rozšířený matematický aparát užívaný v této oblasti, ale zejména jeho použití, které se běžnému studentu může zdát neobvyklé.

Komentář k textu:

Poznámky jsou tištěny odlišným písmem a mají za úkol průběžně komentovat vykládaný text. Jsou odlišeny proto, aby bylo dobře možno je při dalším čtení přeskakovat a nerušily souvislost výkladu.

Odstavce označené znakem "piky" vyjadřují důležité poznatky. V zájmu snadnějšího zapamatování jsou často vyjádřeny "lidovou mluvou".

Cvičení mají za úkol upozornit na zajímavé problémy, nebo dávají čtenáři příležitost ověřit si porozumění vykládané problematice.

Kapitola 1

Identifikace multimodelů

Na rozdíl od předcházejících kapitol, kdy jsme pro popis sledované soustavy uvažovali jediný model, budeme nyní jako základ modelování uvažovat celou množinu různých modelů mezi kterými budeme přepínat. V každém okamžiku tedy bude modelovanou veličinu popisovat právě jeden z uvažovaných modelů. Tak lze kombinovat modely se stejnou strukturou, lišící se pouze v parametrech, ale také modely zcela odlišné. Zmíněný model nazveme multimodelem a jeho identifikací se budeme zabývat v této kapitole.

Význam multimodelů je v popisu soustav, jejichž dynamická struktura se během identifikace skokově mění, nebo jako přiblížení modelům s proměnnými parametry, kdy místo spojité změny parametrů uvažujeme skokově se měnící parametry s konečným počtem různých hodnot.

V této kapitole se budeme zabývat výhradně identifikací. Budeme proto uvažovat neřízený model, ve kterém vystupují data $d_t,\ t = 1, 2, ..., \partial t$ místo dříve uvažovaných signálů výstupu y_t a vstupu u_t.

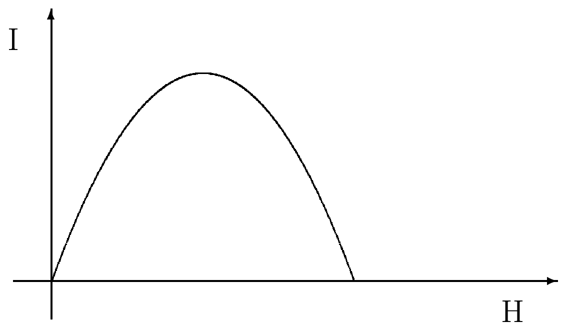
Symbol ∂x označuje celkový počet složek vektoru nebo posloupnosti x; např. ∂t označuje délku časového intervalu na kterém provádíme identifikaci.

1.1 Multimodely a jejich popis

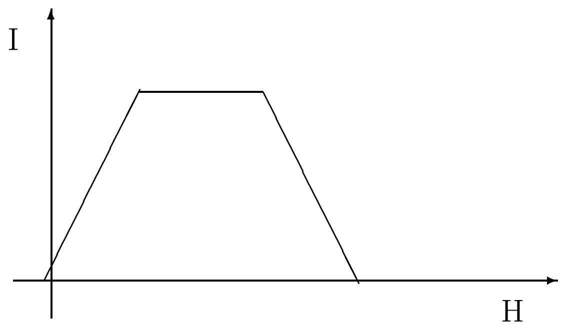
Motivace pro použití multimodelů

Významnou třídu tvoří nelineární reálné soustavy, které však lze přibližně popsat pomocí skupiny lineárních modelů tak, že podle polohy pracovního bodu je v každé chvíli aktivní právě jeden z uvažovaných modelů.

Jako příklad uveďme modelování závislosti hustota (H) – intenzita (I) dopravního toku na jednom místě dopravní komunikace. Tato závislost má tvar známé křivky podobné obrácené parabole.



Takový model je nelineární. Můžeme jej ale lokálně linearizovat tak, že jej nahradíme skupinou lineárních modelů (úseček) např. takto



kde

rostoucí úsečka představuje oblast volného provozu,
konstantní úsečka popisuje oblast nasycení a
klesající úsečka je obrazem kolabující dopravní situace.

V každém okamžiku leží ideální pracovní bod $[H, I]$ na některé úsečce, a tedy, stav je v každém okamžiku popsán některým lineárním modelem.

Model přepínače

Z uvedeného příkladu je patrné, že v multimodelu se kromě modelovaných dat d_t vyskytuje ještě další veličina. Její hodnota ukazuje na aktivní komponentu. Tuto veličinu označíme c_t a budeme ji nazývat **přepínač**. Její hodnoty v časech $t = 1, 2, ..., \partial t$ představují pořadové číslo aktivní komponenty

$$c_t \in \{1, 2, ..., \partial c\}$$

kde ∂c je počet komponent multimodelu.

Ve výjimečných případech, kdy víme, která komponenta je kdy aktivní, je tato veličina známá diskrétní funkce – posloupnost.

Ve většině případů však nevíme, která komponenta je aktivní, a c_t je posloupností náhodných veličin. Popisem těchto náhodných veličin je podmíněná hp

$$\mathcal{P}: \; f(c_t|d(t-1), \Theta) = f(c_t|\alpha_{c_t}) \tag{1.1}$$

kde

α_{c_t} je parametr, popisující pravděpodobnost, že přepínač nabyl hodnotu c_t.

Parametry $\alpha_c, \quad c = 1, 2, ..., \partial c$, tvořící vektor α s ∂c prvky, budeme dále interpretovat přímo jako vektor pravděpodobností toho, že v čase t je aktivní c-tá komponenta multimodelu (za podmínky, že známe pouze starší data $d(t-1)$ a nikoliv současnou hodnotu data d_t). Proto musí platit

$$\alpha_c \in [0,1], \quad \text{pro} \quad c = 1, 2, ..., \partial c \qquad \text{a} \qquad \sum_{c=1}^{\partial c} \alpha_c = 1.$$

V modelu přepínače je vyjádřena podmínka nezávislosti náhodné veličiny c_t na starších datech $d(t-1)$.

Abychom s uvedenou hp mohli reálně pracovat, je třeba ji vyjádřit v "rozumném " tvaru, tj. ve tvaru, kdy odhad parametrů je možno provádět na základě statistik, jejichž rozměry s časem nerostou. Víme, že toto platí pro hp z exponenciální třídy. V případě modelu přepínače budeme ještě konkrétnější a zvolíme *multinomické* rozdělení, kde za pravděpodobnosti hodnot náhodné veličiny (tj. aktivit jednotlivých komponent) zvolíme prvky vektoru parametrů α, tj.

$$\mathcal{P}: \; f(c_t|\alpha) = \alpha_{c_t}$$

nebo, jak bývá zvykem vyjadřovat multinomické rozdělení, v součinovém tvaru

$$\mathcal{P}: \; f(c_t|\alpha_{c_t}) = \prod_{c=1}^{\partial c} \alpha_c^{\delta(c,c_t)}$$

kde $\delta(c, c_t)$ je Kroneckerova funkce, definována takto

$$\delta(c, c_t) = \begin{cases} 1 & \text{pro} \quad c = c_t \\ 0 & \text{pro} \quad c \neq c_t \end{cases}, \quad c \in c^* \tag{1.2}$$

Tuto funkci budeme používat i v dalších modelech a nazveme ji **ukazatel**, protože index c, ve kterém nabývá jedničku, ukazuje na aktivní komponentu.

Modely komponent

Jednotlivé komponenty multimodelu vyjádříme jako běžné regresní modely ve formě podmíněných hustot pravděpodobnosti (hp) indexovaných pořadovým číslem komponenty c

$$\mathcal{K}_c: \; f(d_t|d(t-1), c, \Theta_c) = f_c(d_t|\varphi_t, \Theta_c), \quad c \in \{1, 2, ..., \partial c\} \tag{1.3}$$

kde

d_t je modelovaná veličina v čase t,

φ_t je regresní vektor, obsahující data v okamžiku t známá, na kterých závisí modelovaná veličina y_t a který pro jednoduchost uvažujeme společný pro všechny komponenty,

Θ je vektor parametrů, popisující komponenty,

Θ_c je subvektor vektoru Θ, příslušející c-té komponentě,

∂c označuje počet komponent.

V modelu komponenty je vyjádřena podmínka nezávislosti modelovaných dat d_t na datech starších než je řád modelu a skutečnost, že každá komponenta c závisí pouze na svých parametrech θ_c.

Pro další práci s modely komponent budeme předpokládat, že patří do *exponenciální třídy*, a to obecně, zatím bez další konkretizace. Je tedy

$$\mathcal{K}_c: \ f_c(d_t|\varphi_t, \Theta_c) = \exp\{q_c^T(\Theta) s_c(\phi_t)\}, \qquad c = 1, 2, ..., \partial c, \tag{1.4}$$

kde každá komponenta má svou vlastní parametrickou funkci q_c a obecně také svou vlastní datovou funkci s_c, závisející na tzv. rozšířeném regresním vektoru $\phi_t = [d_t, \varphi_t']'$. Parametrická funkce q a datová funkce s jsou n rozměrné sloupcové vektory. Datová funkce s bývá často společná pro všechny komponenty.

Podobně jako model přepínače, i když v tomto případě trochu neobvykle, vyjádříme model aktivní komponenty pomocí *ukazatele* v součinovém tvaru

$$\mathcal{K}_c: \ f_{c_t}(d_t|\varphi_t, \Theta_{c_t}) = \prod_{c=1}^{\partial c} f_c(d_t|\varphi_t, \Theta_c)^{\delta(c,c_t)} = \exp\left\{\sum_{c=1}^{\partial c} q_c^T(\Theta) s_c(\phi_t) \delta(c, c_t)\right\}.$$

Popis multimodelu

V běžném případě, kdy neznáme aktivity jednotlivých komponent, je multimodel popsán podmíněnou hp obou neznámých náhodných veličin d_t a c_t, kterou lze použitím řetězového pravidla dále rozložit

$$\begin{aligned} \mathcal{M}: \ f(d_t, c_t|d(t-1), \Theta_{c_t}, \alpha_{c_t}) &= f(d_t|d(t-1), c_t, \Theta_{c_t}) f(c_t|d(t-1), \alpha_{c_t}) = \\ &= f_{c_t}(d_t|\varphi_t, \Theta_{c_t}) \alpha_{c_t}. \end{aligned} \tag{1.5}$$

První hp v rozkladu je model komponenty a druhá model přepínače.

Multimodel, popisující aktivní komponentu v čase t, lze samozřejmě také vyjádřit v součinovém tvaru

$$\mathcal{M}: \ f(d_t, c_t|d(t-1), \Theta_{c_t}, \alpha_{c_t}) = \prod_{c=1}^{\partial c} f(d_t, c|d(t-1), \Theta_c, \alpha_c)^{\delta(c,c_t)} = \tag{1.6}$$

$$= \prod_{c=1}^{\partial c} \exp\left\{q_c^T(\Theta) s_c(\phi_t) \delta(c, c_t)\right\} \alpha_c^{\delta(c,c_t)}.$$

`Poznámka:` *Vyjádření modelu v součinovém tvaru je formální krok. Přesto má význam. Hp na pravé straně neobsahují neznámou funkci c_t. Obsahují jen indexovou proměnnou c za kterou postupně dosazujeme hodnoty $1, 2, ..., \partial c$. Funkce c_t je obsažena jen v ukazateli $\delta(c, c_t)$.*

`Poznámka:` *Ukazatel $\delta(c, c_t)$ můžeme chápat jako hp veličiny c_t jestliže její hodnotu známe, tj. víme, která komponenta je aktivní. Příslušná hodnota je jedna, ostatní jsou nuly.*

Popis predikce dat

Tak jako tomu bylo u obyčejných modelů, je úkolem multimodelu predikovat data d_t bez ohledu na to, která komponenta je zrovna popisuje. Jinými slovy, potřebujeme z předchozího vyjádření multimodelu eliminovat veličinu c_t. To lze provést výpočtem marginální hp podle vzorce

$$\begin{aligned} f(d_t|d(t-1),\Theta,\alpha) &= \sum_{c_t=1}^{\partial c} f(d_t,c_t|d(t-1),\Theta,\alpha) = \\ &= \sum_{c_t=1}^{\partial c} f_{c_t}(d_t|\varphi_t,\Theta_{c_t})\alpha_{c_t} \end{aligned} \tag{1.7}$$

Vidíme, že při volbě multinomického rozdělení pro model přepínače dostáváme prediktivní hp ve tvaru lineární kombinace komponent, kde vahami kombinace jsou pravděpodobnosti jednotlivých komponent. Takováto lineární kombinace se nazývá směs hustot pravděpodobnosti.

1.2 Odhad neznámých parametrů multimodelu

Bayesův vztah

Dosud jsme se zabývali matematickým popisem multimodelu. Ten je na obecné úrovni dán podmíněnou hp (1.5), závislou na neznámých parametrech Θ, α. Základem pro identifikaci těchto parametrů je, podobně jako u obyčejných modelů, Bayesův vztah, který za pomoci měřených dat d_t přepočítává apriorní hp $f(\Theta,\alpha|d(t-1))$ na aposteriorní hp $f(\Theta,\alpha|d(t))$. Bayesův vzorec má v tomto případě tvar

$$\mathcal{B}: \; f(\Theta,\alpha|d(t)) \propto f(d_t|d(t-1),\Theta,\alpha) f(\Theta,\alpha|d(t-1)), \tag{1.8}$$

který je rekurzivně počítán pro $t = 1, 2, ...\partial t$, a který startuje s apriorní hp $f(\Theta,\alpha|d(0))$ vybudované na základě apriorní znalosti.

Aproximace Bayesova vztahu

Princip aproximace

Z předchozího vyjádření Bayesova vztahu je patrné, že pro identifikaci je zapotřebí nikoliv multimodel, ale jeho marginální tvar (1.7), který jsme nazvali predikcí a který má vyjádření ve tvaru směsi hp. Po dosazení bychom dostali

$$\mathcal{B}: \; f(\Theta,\alpha|d(t)) \propto \left[\sum_{c=1}^{\partial c} f(d_t,c|d(t-1),\Theta,\alpha)\right] f(\Theta,\alpha|d(t-1)).$$

Protože tento vztah se počítá rekurzivně pro $t = 1, 2, .., \partial t$, dostáváme aposteriorní hp jako opakované součiny součtů v neuzavřeném tvaru (počet členů tohoto vyjádření neustále narůstá). Tento tvar je pro další počítání zcela nevhodný a je třeba jej aproximovat.

Navrženou aproximaci lze interpretovat v následujících dvou krocích

A. Vyjádření marginální hp (1.7) pomocí jediné podmíněné hp tak, jako kdybychom znali hodnotu náhodné veličiny $c = c_t$ a místo marginalizace bychom tuto hodnotu prostě dosadili.

B. Nahrazení ukazatele označujícího "známou" komponentu jeho optimálním bodovým odhadem, kterým je, pro kvadratické kriterium, střední hodnota, podmíněná dosud pozorovanými daty.

Schéma aproximace

Kroky, naznačené v předchozím odstavci, popíšeme podrobněji:

A. V Bayesově vzorci zaměníme směs hp jednou z nich, o které prohlásíme, že "to je ta pravá", která popisuje aktivní komponentu, a vyjádříme ji v součinovém tvaru

$$\sum_{c=1}^{\partial c} f(d_t, c|d(t-1), \Theta, \alpha) \to f(d_t, c_t|d(t-1), \Theta, \alpha) = \prod_{c=1}^{\partial c} f(d_t, c|d(t-1), \Theta, \alpha)^{\delta(c,c_t)}$$

B. Ukazatel, obsahující ve skutečnosti neznámou funkci c_t nahradíme kvadraticky optimálním bodovým odhadem, kterým je jeho střední hodnota

$$\delta(c, c_t) \to E[\delta(c, c_t)|d(t)] = \sum_{c=1}^{\partial c} \delta(c, c_t) f(c_t|d(t)) = f(c_t = c|d(t)), \quad c = 1, 2, ..., \partial c$$

podmíněná všemi dosud známými daty, včetně současných. Zde, ale i dále, používáme následující značení: $f(c_t = c|d(t)) = P(c_t = c|d(t))$, což je hp náhodné veličiny c_t v bodě c. ($P(.|.)$ označuje podmíněnou pravděpodobnost.)

Potřebujeme tedy znát příslušnou hp. Tu vyjádříme opět pomocí modelu komponenty takto

$$w_{c,t} = f(c_t = c|D(t)) = \int\int f(c_t = c, \Theta, \alpha|d_t, D(t-1)) d\Theta d\alpha \propto$$

$$\propto \int\int f(d_t|D(t-1), c, \Theta, \alpha) f(c_t = c|D(t-1)) f(\Theta, \alpha|D(t-1)) d\Theta d\alpha =$$

$$= \int\int f_c(d_t|\varphi_t, \Theta_c) \alpha_c f(\Theta|D(t-1)) f(\alpha|D(t-1)) d\Theta d\alpha$$

$$= \int f_c(d_t|\varphi_t, \Theta_c) f(\Theta|D(t-1)) d\Theta \int \alpha_c f(\alpha|D(t-1)) d\alpha \qquad (1.9)$$

pro $c = 1, 2, ..., \partial c$.

V předchozím odvození jsme v prvním řádku přidali a ihned vyintegrovali parametry Θ a α; při přechodu na druhý řádek jsme aplikovali Bayesův vzorec a ve třetím řádku již jen použili zavedené podmínky a značení pro jednotlivé hp. Výsledkem integrace pro jednu hodnotu c je číslo – pravděpodobnost, že $c_t = c$, tj., že v čase t je aktivní právě c-tá komponenta. Integrace provedeme pro všechny hodnoty veličiny $c = 1, 2, ..., \partial c$ a výsledkem bude vektor pravděpodobností aktivity jednotlivých komponent. Tento vektor "aktuálních pravděpodobností" jednotlivých komponent jsme označili $w_{c,t}$.

P o z n á m k a: *Všimněme si, že výsledný tvar výrazu pro výpočet hodnoty $w_{c,t}$ je dán součinem dvou nám již známých výrazů, které jsme uvedli v druhém díle tohoto pojednání.*

Jedná se o prediktivní hp pro regresní model s normálním šumem (2.23) a stejnou hp, ale pro model s diskrétním šumem (2.59).

Výsledek aproximace

V předchozím odstavci jsme vyjádřili model komponenty v součinovém tvaru a nahradili ukazatel $\delta(c, c_t)$ jeho aproximací $w_{c,t}$. Podobně, v součinovém tvaru (tzv. Dirichletův tvar) vyjádříme také hp parametrů

$$f(\Theta, \alpha|d(t)) = \prod_{c=1}^{\partial c} f(\Theta_c, \alpha_c|d(t)).$$

Označíme-li obecně symbolem $U_{c,t}$ (jako <u>u</u>kazatel) exponent, na který mocníme model komponenty, bude pro jednotlivé případy platit

$$\mathbf{A.}: \ U_{c,t} = \delta(c, c_t) \qquad \mathbf{B.}: \ U_{c,t} = w_{c,t}.$$

Bayesův vztah lze potom zapsat jednotně pro oba případy takto

$$\mathcal{B}: \ \prod_{c=1}^{\partial c} f(\Theta_c, \alpha_c|d(t)) = \prod_{c=1}^{\partial c} f_c(d_t, c|d(t-1), \Theta_c, \alpha_c)^{U_{c,t}} f(\Theta_c, \alpha_c|d(t-1))$$

Tento vzorec dále převedeme na rekurzi pro statistiky. Prozatím poznamenejme jen to, že aproximací jsme dosáhli svého cíle. Zbavili jsme se součtového vyjádření modelu komponenty, který způsoboval neuzavřenost Bayesova vztahu.

Uvedený tvar Bayesova vzorce rovněž ukazuje, že jeho výpočet je možno rozdělit a provádět pro každý činitel součinu (tedy pro každou komponentu $c = 1, 2, ..., \partial c$) zvlášť.

Bayesův vztah pro exponenciální třídu komponent

Bayesův vztah, tak jsme jej uvedli v předešlém odstavci, představuje funkcionální rekurzi. Je předpisem, jak funkce $f(\Theta_c, \alpha_c|d(t-1))$ přepočítat na funkce $f(\Theta_c, \alpha_c|d(t))$. Protože takové výpočty nelze na současné výpočetní technice provádět, potřebujeme tuto funkcionální rekurzi převést na číselnou. To uděláme podobně jako v případě obyčejných modelů tak, že vyjádříme modely v konkrétním tvaru – přepínač v multinomickém a komponenty obecně v exponenciálním.

Odpovídající (konjugovaný) tvar hp parametrů (viz minulý díl tohoto pojednání) je

$$f(\Theta_c, \alpha_c|d(t)) = \exp\{q_c^T(\Theta) S_c(t)\} \alpha_c^{\kappa_c(t)-1}, \quad c = 1, 2, ..., \partial c,$$

kde q_c je parametrická funkce z exponenciálního vyjádření modelu a $S_c(t)$ je statistika pro odhadování parametrů modelu c-té komponenty a $\kappa_c(t)$ je statistika pro odhad parametrů přepínače.

Po dosazení exponenciálního tvaru modelu komponent a jemu konjugovaného tvaru hp parametrů a po vypuštění symbolu pro součin má Bayesův vzorec následující tvar (pro jednotlivé komponenty platí)

$$\mathcal{B}: \ \exp\{q_c^T(\Theta) S_c(t) \alpha^{\kappa_c(t)} \propto \left[\exp\{q_c^T(\Theta) s_c(\varphi_t)\} \alpha_c\right]^{U_{c,t}} \exp\{q_c^T(\Theta) S_c(t-1)\} \alpha^{\kappa_c(t-1)} =$$

$$= \exp\{q_c^T(\Theta)[s_c(\varphi_t)U_{c,t} + S_c(t-1)]\}\alpha_c^{U_{c,t}+\kappa_c(t-1)}, \quad c = 1, 2, ..., \partial c \tag{1.10}$$

Porovnáním odpovídajících si statistik na obou stranách Bayesova vzorce dostáváme následující rekurze

$$S_c(t) = S_c(t-1) + s_c(\varphi_t)U_{c,t} \tag{1.11}$$
$$\kappa_c(t) = \kappa_c(t-1) + U_{c,t} \tag{1.12}$$

pro $c = 1, 2, ..., \partial c$.

Vlastní algoritmus pro odhadování parametrů Θ a α multimodelu můžeme shrnout takto:

Příprava identifikace:
sestav apriorní statistiky.

Vlastní identifikace:
for čas = 1 až N
změř data,
přepočti statistiky.

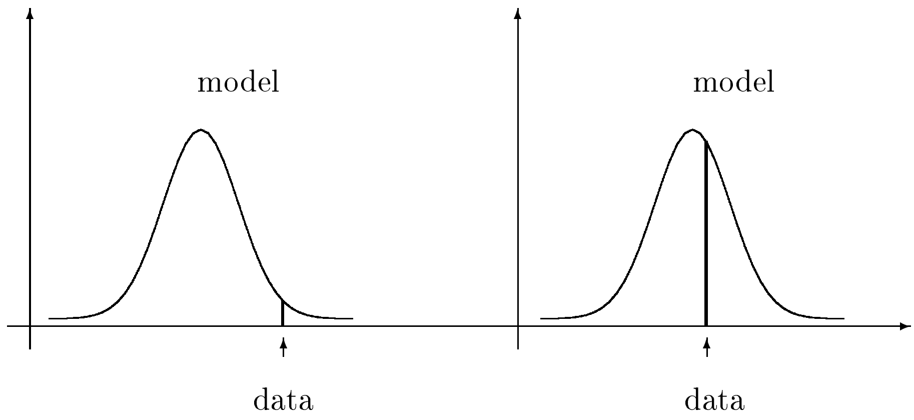
Závěr identifikace:
vypočti parametry ze statistik.

Struktura prostřední části uvedeného algoritmu počítaného v cyklu času je následující:

- změř data,
- udělej "pokusný" přepočet statistik pro $U_{c,t} = 1, \quad c = 1, 2, ..., \partial c$
- vypočti aktuální váhy komponent $w_{c,t}$ s právě přepočtenými statistikami,
- udělej "skutečný" přepočet statistik pro $U_{c,t} = w_{c,t}, \quad c = 1, 2, ..., \partial c$.

Komentář k algoritmu:

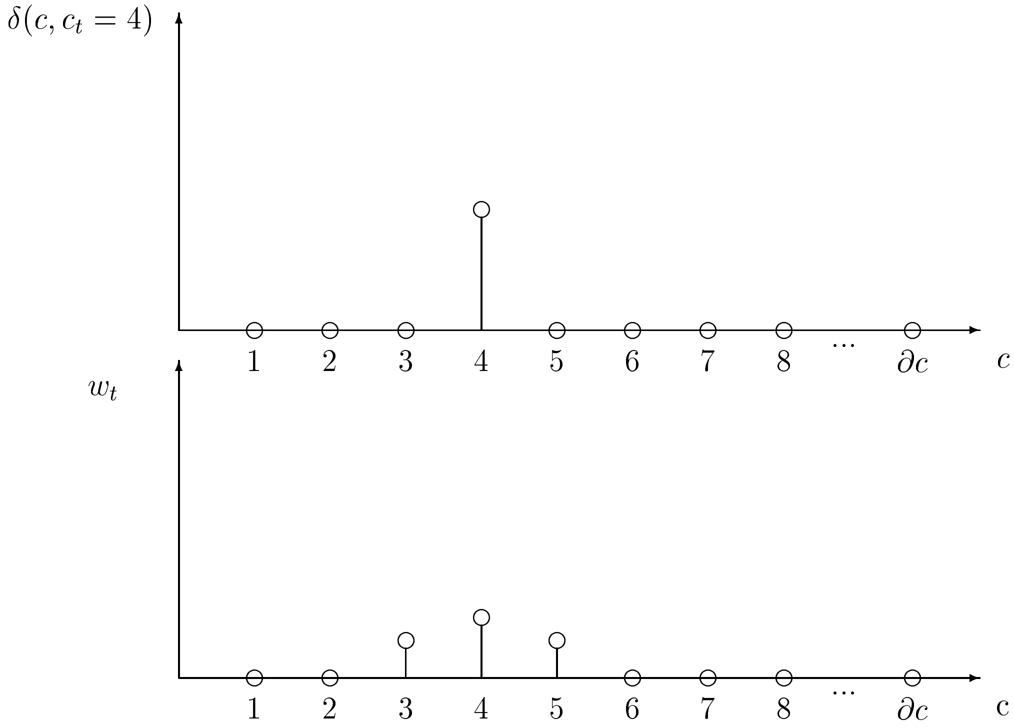
První přepočet statistik je třeba provést pro výpočet aktuálních vah (pravděpodobností aktivit) jednotlivých komponent. Při něm se stavíme do situace **A.**, kdy předpokládáme znalost aktivní komponenty. Přitom každou komponentu $c = 1, 2, ..., \partial c$ pokládáme vždy za známou a zkoušíme, jaká je pravděpodobnost, že právě naměřená data odpovídají této komponentě. To jsou právě hledané váhy $w_{c,t}$. Korespondenci dat a komponenty je možno dobře ilustrovat na statickém modelu, kde největší pravděpodobnost dat je přímo pod vrcholem hp



Na levém obrázku leží data daleko of středu, pořadnice pod hp (odrážející pravděpodobnost, že data pochází z tohoto modelu) je malá. Na pravém obrázku leží data blíže středu, pořadnice, a tedy i pravděpodobnost příslušnosti dat k modelu, je větší.

Pravděpodobnosti, které určíme po prvním přepočtu statistik, použijeme jako váhy $w_{c,t}$, se kterými přidáváme informaci ze současných dat do statistik jednotlivých komponent. Čím více data komponentě odpovídají, s tím větší vahou jsou zabudována do jejích statistik.

Bezprostřední důsledek aproximace je následující: jestliže víme přesně, která z komponent je aktivní, přepočteme její statistiky s jednotkovou váhou a ostatní ponecháme nedotčené. Pokud si ale nejsme aktivní komponentou jisti (což je důvodem k aproximaci), přidělujeme data do všech komponent, do každé s takovou váhou, která ji vzhledem k naměřeným datům náleží. Původní ukazatel na aktivní komponentu se tak "rozštěpí" a změní se na hustotu pravděpodobnosti pro jednotlivé komponenty. Tato situace je naznačena na následujícím obrázku



1.3 Příklady

K problematice multimodelů uvedeme dva příklady. Cílem prvního z nich je na jednoduché struktuře statických komponent ilustrovat teorii. V druhém příkladě ukážeme základní prvky programového systému MixTools, vyvíjeného v ÚTIA AV ČR a sloužícího, mimo jiné, k identifikaci obecně mnoharozměrných dynamických multimodelů. Pomocí tohoto toolboxu lze snadno definovat multimodel, na základě změřených dat určit počet komponent i jejich dynamickou strukturu, vytvořit apriorní statistiky a model identifikovat.

PŘÍKLAD 1.1 Statický multimodel s normálním šumem

Jako příklad ilustrující teorii uvedeme identifikaci multimodelu s dvěma statickými komponentami a normálním šumem. Obyčejným statickým modelem jsme se zabývali v předchozích částech tohoto pojednání. Nebudeme proto již opakovat odvození identifikace pro jednotlivé komponenty našeho multimodelu, ale využijeme již odvozených výsledků.

Modely komponent
Uvažujeme dvě statické komponenty

$$\mathcal{K}_c : \; f(d_t|\Theta_c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{-\frac{1}{2\sigma_c^2}(d_t^2 - 2\theta_c d_t + \theta_c^2)\right\}, \quad c = 1, 2.$$

s parametry θ_c a σ_c^2 označené indexem komponent c a s vektorem parametrů

$$\Theta' = [\Theta_1', \Theta_2'], \qquad \Theta_1' = [\theta_1, \sigma_1^2], \; \Theta_2' = [\theta_2, \sigma_2^2],$$

a společným regresním vektorem $\varphi_t = 1$ a rozšířeným regresním vektorem (nebo datovým vektorem) $\phi_t' = [d_t, 1]$.

Komponenty tohoto modelu vyjádříme ve tvaru exponenciální třídy

$$
\begin{aligned}
f(d_t|\Theta_c) &= \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{d_t^2}{2\sigma_c^2} + \frac{d_t\theta_c}{\sigma_c^2} - \frac{\theta_c^2}{2\sigma_c^2} - \log\sigma \right\} = \\
&= \frac{1}{\sqrt{2\pi}} \exp\left\{ \left[\frac{-1}{2\sigma_c^2}, \frac{\theta_c}{\sigma_c^2}, \frac{-\theta_c^2}{2\sigma_c^2} - \log\sigma \right] \left[d_t^2, d_t, 1 \right]' \right\} = \\
&= \frac{1}{\sqrt{2\pi}} \exp\left\{ q_c'(\Theta_c) s(\phi_t) \right\}
\end{aligned}
$$

kde

$$q_c' = [q_{1,c}, q_{2,c}, q_{3,c}], \quad s_c' = [s_{1,c}, s_{2,c}, s_{3,c}], \quad c = 1, 2$$

a jednotlivé prvky těchto vektorů jsou dány předchozím vztahem.

Struktura aposteriorní (a stejně apriorní) hp je

$$f(\Theta_c|d(t)) \propto \exp\left\{ -\frac{1}{2} \left(\nu_c(t)\theta_c^2 - 2S_{d1,c}(t)\theta_c + S_{d,c}^2(t) \right) \right\},$$

kde k_c, $S_{d1,c}$ a $S_{d,c}^2$ jsou statistiky pro odhad parametrů komponent.

Model přepínače

Hp přepínače má vždy multinomické rozdělení, tady

$$\mathcal{P}: \ f(c_t|\alpha) = \alpha_{c_t}, \quad c_t \in \{1, 2\},$$

kde

$$\alpha' = [\alpha_1, \alpha_2], \quad \alpha_1, \alpha_2 \geq 0, \ \alpha_1 + \alpha_2 = 1$$

Aposteriorní (jako i apriorní) hp má stejně jako model přepínače pevnou strukturu

$$f(\alpha_c) = \alpha_c^{\kappa_c(t)-1}$$

Přepočet statistik

Podle Bayesova vzorce (1.10) pro exponenciální třídu komponent, rozepsaného pro jednotlivé komponenty, dostáváme přepočet statistik podle (1.11) a (1.12) ve tvaru

$$
\begin{aligned}
S_{d,c}^2(t) &= S_{d,c}^2(t-1) + d_t^2 U_{c,t}, \\
S_{d1,c}(t) &= S_{d1,c}(t-1) + d_t U_{c,t}, \\
\kappa_c(t) &= \kappa_c(t-1) + U_{c,t}, \\
\nu_c(t) &= \nu_c(t-1) + U_{c,t},
\end{aligned}
$$

pro $c = 1, 2$ a $t = 1, 2, ..., N$

kde

$U_{c,t}$ je rovno 1 pro první statistik a rovná se $w_{c,t}$ pro druhý přepočet statistik,

$S_{d,c}(0)$, $S_{d1,c}(0)$, $\kappa_c(0)$ a $\nu_c(0)$ jsou apriorní statistiky,

N je počet kroku identifikace.

1.4 Programový systém MixTools

Tento systém umožňuje simulaci a identifikaci multimodelů. Základními prvky, se kterými pracuje, jsou: *faktor*, *komponenta* a *směs*. Vztah směsi a komponent je zřejmý teoretického pojednání. Hp, popisující multimodel, je lineární kombinace hp komponent, s vahami kombinace rovnými pravděpodobnostem jednotlivých komponent. Souvislost komponenty a jejích faktorů je nejlépe patrná opět na hp komponenty. Ukážeme ji pro dvourozměrný případ.

1.4.1 Komponenta a její faktory

Budeme uvažovat dvourozměrnou komponentu 1. řádu s konstantou. Ta popisuje dvourozměrnou náhodnou veličinu $y_t = [y_{1;t} y_{2;t}]'$ pomocí podmíněné hp s vektorem parametrů θ. Tuto dvourozměrnou hp lze rozepsat podle řetězového pravidla (index komponenty c pro přehlednost vynecháme)

$$f(y_{1;t}, y_{2;t}|y_{1;t-1}, y_{2;t-1}, \theta) = f(y_{2;t}|y_{1;t}, y_{1;t-1}, y_{2;t-1}, \theta) f(y_{1;t}|y_{1;t-1}, y_{2;t-1}, \theta).$$

Jednorozměrné hp v součinu jsou faktory. Odtud je zřejmé, že model komponenty jako hustotu pravděpodobnosti lze vyjádřit jako součin jednorozměrných hustot pravděpodobností modelů jednotlivých faktorů.

Strukturu faktorů a přepočet parametrů komponenty na parametry faktorů ukážeme na vyjádření modelu komponenty ve tvaru rovnice.

$$\begin{bmatrix} y_{1;t} \\ y_{2;t} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & k_1 \\ a_{2,1} & a_{2,2} & k_2 \end{bmatrix} \begin{bmatrix} y_{1;t-1} \\ y_{2;t-1} \\ 1 \end{bmatrix} + \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix} \begin{bmatrix} e_{1;t} \\ e_{2;t} \end{bmatrix},$$

kde náhodnou část modelu jsme vyjádřili ve tvaru "směrodatná odchylka" krát "normovaný šum" – kde směrodatná odchylka je Choleskyho odmocnina z kovarianční matice šumu a normovaný šum má rozptyl roven jedné.

Model komponenty dále upravíme tak, abychom dostali dva modely s oddělenými, tj. vzájemně nekorelovanými, šumy. Tyto modely budeme nazývat faktory.

Z matice r vytkneme diagonálu a zbytek invertujeme

$$\begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix} = \begin{bmatrix} r_{1,1} & 0 \\ 0 & r_{2,2} \end{bmatrix} \begin{bmatrix} 1 & r_{1,2}/r_{1,1} \\ 0 & 1 \end{bmatrix} \qquad \text{(vytknutí)}$$

$$\begin{bmatrix} 1 & r_{1,2}/r_{1,1} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -r_{1,2}/r_{1,1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & g_{1,2} \\ 0 & 1 \end{bmatrix} \qquad \text{(inverze)}$$

Inverzí g vynásobíme model komponenty a dostaneme

$$\begin{bmatrix} 1 & g_{1,2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{1;t} \\ y_{2;t} \end{bmatrix} = \begin{bmatrix} 1 & g_{1,2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & k_1 \\ a_{2,1} & a_{2,2} & k_2 \end{bmatrix} \begin{bmatrix} y_{1;t-1} \\ y_{2;t-1} \\ 1 \end{bmatrix} + \begin{bmatrix} r_{1,1} & 0 \\ 0 & r_{2,2} \end{bmatrix} \begin{bmatrix} e_{1;t} \\ e_{2;t} \end{bmatrix}.$$

Po úpravě je

$$\begin{bmatrix} y_{1;t} \\ y_{2;t} \end{bmatrix} = \begin{bmatrix} q_{1,0} & q_{1,1} & q_{1,2} & q_{1,3} \\ 0 & q_{2,1} & q_{2,2} & q_{2,3} \end{bmatrix} \cdot \begin{bmatrix} y_{2;t} \\ y_{1;t-1} \\ y_{2;t-1} \\ 1 \end{bmatrix} + \begin{bmatrix} r_{1,1} e_{1;t} \\ r_{2,2} e_{2;t} \end{bmatrix}.$$

kde koeficienty q dostaneme úpravou předchozího tvaru modelu.

Výsledný model lze rozepsat ma dva modely, každý se svým šumem. To jsou faktory výchozí dvourozměrná komponenty.

$$\begin{aligned} y_{1;t} &= q_{1,0}y_{2;t} + q_{1,1}y_{1;t-1} + q_{1,2}y_{2;t-1} + q_{3,1} + r_{1,1}e_{1;t} \\ y_{2;t} &= q_{2,1}y_{1;t-1} + q_{2,2}y_{2;t-1} + q_{3,2} + r_{2,2}e_{2;t} \end{aligned}$$

Každý faktor modeluje jednu složku výstupu y, tzv. datový kanál.

1.4.2 Základní prvky programu

Globální proměnné

V MixTools jsou definovány 3 globální proměnné, které je, po jejich deklaraci, možno kdekoliv použít. Jsou to proměnné

`TIME` – *dynamický čas* běhu programu (v teorii značený t)

`DATA` – *datová matice*, kde data jsou uspořádána po řádcích, tj. např. pro dvourozměrný výstup a TIME=1,2,3 je

$$\texttt{DATA} = \begin{bmatrix} y_{1;1} & y_{1;2} & y_{1;3} \\ y_{2;1} & y_{2;2} & y_{2;3} \end{bmatrix}$$

`DEBUG` – *úroveň dialogu* (DEBUG=1,2,3; vyšší číslo dá více informací při běhu programu)

Důležité proměnné

Dále jsou zavedena jména řady proměnných. Nejdůležitější jsou

`nchn` a `ndat` – *počet kanálů* (dimenze modelované veličiny) a *počet datových vzorků.*

Platí: `[nchn ndat]=size(DATA).`

`ychns` – *vektor modelovaných kanálů.* Např. `ychns=[1 2].`

`str` – *struktura faktoru*, která určuje veličiny a jejich pořadí v regresním vektoru faktoru. Je to dvouřádková matice, kde v prvním řádku je uvedeno číslo modelovaného kanálu a na odpovídajícím místě druhého řádku je jeho zpoždění. Např. pro dvě modelované veličiny a faktor

$$y_{1;t} = a_{1,1}y_{1;t-1} + a_{1,2}y_{2;t-1} + a_{1,3}y_{1;t-2} + a_{1,4}y_{2;t-2} + k + e_{1;t}$$

je odpovídající struktura

$$\texttt{str} = \begin{bmatrix} 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 2 & 2 & 1 \end{bmatrix}$$

tj., regresní vektor bude obsahovat první kanál jednou zpožděný, druhý kanál jednou zpožděný, první a druhý kanál dvakrát zpožděný a jedničku pro konstantu.

`dfcs` – *počet stupňů volnosti* komponent. Po normování na jednotkový součet dá pravděpodobnosti komponent.

`ncom` – *počet komponent* multimodelu.

`Coms` – *cell vektor ARX LS komponent*. Jeho složky jsou `Coms{i}`, i=1,2,...,`ncom` a obsahují informace o komponentách.

Komponenta `Coms{.}` obsahuje následující proměnné: (pro dvourozměrný model s třemi parametry)

proměnná	název	typ
`ychns`	*modelované kanály*	vektor 1x`nchn`
`str`	*struktura*	matice 2x3
`dfm`	*počet stupňů volnosti parametrů*	číslo
`type`	*typ komponenty*	číslo
`cove`	*kovarianční matice šumu*	matice 2x2
`Eth`	*regresní koeficienty*	matice 2x3
`Cth`	*kovariance regresních koeficientů*	matice 3x3

`Sim` – *směs pro simulaci – typ ARX LS*. Struktura je následující

proměnná	název	typ
`Facs`	*faktory*	cell vektor
`coms`	*faktory v komponentách*	matice (řád. komp., sl. faktory)
`dfcs`	*vektor stupňů volnosti*	vektor 1x`ncom`
`type`	*typ směsi*	číslo
`states`	*stavy směsi*	matlabská struktura

Struktura jednoho faktoru `Facs{.}` je

proměnná	název	typ
`ychn`	*číslo modelovaného kanálu*	číslo
`str`	*struktura faktoru*	matice
`dfm` *stupně volnosti*	číslo	
`type` *typ faktoru*	číslo	
`cove`	*rozptyl šumu*	číslo
`Eth`	*regresní koeficienty*	vektor
`Cth`	*kovariance regresních, koeficientů*	matice

`Mix` – *směs pro odhadování – typ ARX*. Její struktura je téměř stejná jako u `Sim`, místo parametrů `cove`, `Eth` a `Cth` však obsahuje statistiky v Choleskyho odmocnině z rozšířené informační matice.

Struktura jednoho faktoru `Facs{.}` je

proměnná	název	typ
`ychn`	*číslo modelovaného kanálu*	číslo
`str`	*struktura faktoru*	matice
`dfm`	*stupně volnosti*	číslo
`type`	*typ faktoru*	číslo
`LD`	*odmocnina z informační matice*	matice

`Poznámka:` *Symboly značí:*
ARX - autoregresní model s externím signálem – příslušná struktura obsahuje matici LD (rozklad informační matice);
ARX LS totéž, ale v "least square podobě", tj. struktura obsahuje prvky Eth, Cth, cove - což jsou parametry vypočtené z informační matice.

Důležité procedury

Podrobnější popis lze získat v Matlabu zadáním: `help` a jméno proceduty.

`Coms{i} =` **comarxls**`(ychns, str)`
konstrukce i té komponenty.

`Mix =` **mixconst**`(Coms, dfcs)`
konstrukce směsi (simulátoru nebo estimátoru).

`Mix1 =` **mix2mix**`(Mix, type)`
konverze směsi na jiný typ
(důležité typy: 21=ARX, 22=ARX LS, 23=matr ARX, 24=matr ARX LS).

mixsimul`(Sim, ndat)`
simulace směsi.

`Mix =` **mixest**`(Mix, frg, ndat, niter, opt)`
identifikace směsi.

`Mix =` **mixstats**`(Mix, ndat)`
přepočet statistik směsi (s finálními parametry).

1.4.3 Základní příklady (pomocné procedury)

Vytvoření směsi pro simulaci

```
function Mix = genmixs(ychns, str, dfcs, Pt)
% Mix = genmixs(ychns, str, dfcs, Pt)
% generates mixture for simulation (type 24)
%
% Mix    mixture for simulation
% ychns  modeled channels
% str    component structure
% dfcs   degrees of freedom for components
% Pt     regression par. and noise disp. of components
%        Par.Eth  and  Par.cove
%
% Design : I. Nagy, P. Nedoma
% Updated: June 2002
% Project: ProDaCTools
%
global DATA

ncom = length(dfcs);
nchn = length(ychns);
for i=1:ncom
  Com{i}  = comarxls(ychns, str);
  Com{i}.Eth = Pt{i}.Eth;
  if length(Pt{i}.cove)==1
    Com{i}.cove = eye(nchn)*Pt{i}.cove;
  else
    Com{i}.cove = Pt{i}.cove; end
end

Mix = mixconst(Com, dfcs);
```

Vytvoření směsi pro identifikaci

```
function Mix = genmixe(ncom, ychns, str, ndat, diagCth, diagcove, dfm, dfcs)
% generates mixture for identification (type 24)
% Mix = genmixe(ncom, ychns, str,  ndat, diagCth, diagcove, dfm)
% Mix = genmixe(ncom, ychns, str,  ndat, diagCth, diagcove) dfm=100
% Mix = genmixe(ncom, ychns, str,  ndat, diagCth)     diagcove=0.01
% Mix = genmixe(ncom, ychns, str,  ndat)              diagCth=1
% Mix = genmixe(ncom, ychns, str)                     ndat=size(DATA,1)
% Mix = genmixe(ncom, ychns)                          str=[0;1]
% Mix = genmixe(ncom)                                 ychns=1:size(DATA,1)
% Mix = genmixe;                                      ncom = 1;
%
% ncom   number of components
```

```
% ychns  modeled channels
% str    component structure
% nchn   number of channels
% ndat   size of data sample
% diagCth, diagcove  setting of Cth and cove
%
% Design : P. Nedoma
% Updated: January 2002
% Project: ProDaCTools
%
global DATA
if nargin<7, dfm      = 100;                      end
if nargin<6, diagcove = 0.01;                     end
if nargin<5, diagCth  = 100;                      end
if nargin<4, ndat     = length(DATA);             end
if nargin<1, ncom = 1;                            end
if nargin<8, dfcs = 0.1*ndat*ones(1,ncom);        end
if nargin<3, str      = [0; 1];                   end
if nargin<2, nchn = size(DATA,1); ychns = 1:nchn; end

Com  = comarxls(ychns, str);
Com.Cth  = diagCth*eye(max(size(Com.Cth)));
Com.cove = diagcove*eye(max(size(Com.cove)));
Com.dfm  = dfm;

Coms = cell(1, ncom);
for i=1:ncom
    Com.Eth = randn(size(Com.Eth));
    Coms{i} = Com;
end

Mix = mixconst(Coms, dfcs);
```

1.4.4 Jednoduchý příklad - statická verze

Simulace tří dvourozměrných statických komponent

```
prodini, echo off
 %
 % Komponenta
 % |y1(t)| =  |Eth1| + |cove1 cove12|.|e1(t)|
 % |y2(t)|    |Eth2|   |0     cove2 | |e2(t)|
 %
 ychns = [1 2];
  % modelované kanály
 str=[0;1];
  % struktura pro statické faktory
 dfcs = [10 15 30];
  % váhy komponent
 sd=1;
 Pt{1}.Eth = [0; 0];
 Pt{2}.Eth = [5; 0];
 Pt{3}.Eth = [3; 8];
 Pt{1}.cove = [.1 1; 1 .1]*sd;
 Pt{2}.cove = [1 0; 0 1]*sd;
 Pt{3}.cove = [.1 -1; -1 .1]*sd;
  % parametry komponent
 Sim = genmixs(ychns, str, dfcs, Pt);
  % vytvoření simulátoru
 ndat=1000;
  % počet kroků simulace
 DATA=zeros(2,ndat);

 % SIMULACE
 mixsimul(Sim, ndat)
 save simDat Sim DATA
   % uložení dat na disk

 % ZOBRAZENÍ
 plot(DATA(1,:),DATA(2,:),'.','MarkerSize',1)
 axis([-5,10,-5,15])
```

Identifikace tří dvourozměrných statických komponent

```
 prodini, echo off
 load simDat;

 % PŘÍPRAVA IDENTIFIKACE
 ncom = 3;
  % počet komponent
 ychns = [1 2];
```

```
 % modelované kanály
str=[0;1];
 % struktura pro statické faktory
ndat=1000;
 % počet kroků pro identifikaci
 dCt = 1e5; dco = 1e-3; dfm = 10;
 dfs = .1*ndat*ones(1,ncom);
  % počáteční statistiky
Mix0 = genmixe(ncom, ychns, str, ndat, dCt, dco, dfm, dfs);
%Mix0 = genmixe(ncom, ychns, str, ndat); !!! špatné předvolby
% vytvoření estimátoru

% IDENTIFIKACE
frg = .99;
Mix = mixest(Mix0, frg, ndat);
save simMix

% VÝSLEDKY
M24 = mix2mix(Mix, 24);
Eth1=M24.Coms{1}.Eth
Eth2=M24.Coms{2}.Eth
Eth3=M24.Coms{3}.Eth

set(figure(1),'Position',[50 100 900 600],'MenuBar','none')
subplot(121),mixplot(Sim)
title('Simulated mixture')
subplot(122),mixplot(Mix)
title('Estimated mixture')
```

Predikce tří dvourozměrných statických komponent

```
prodini, echo off
load simMix;

% PREDIKCE
nstep = 5;
 % počet kroku predikce
pchns = [1 2]; cchns = [];
 % predikované kanály a kanály v podmínce
pMix = mix2pro(Mix, pchns, cchns);
 % podmíněná směs
yp = []; time = []; ws = [];
for TIME = Mix.states.maxtd+1:ndat-nstep
  [Eths, coves, al, w] = profixn(pMix,[],[],nstep);
  th = GetTh(Eths);
  if 1, ypt = th*w';            % okamžité váhy
  else  ypt = th*al';           % průměrné váhy
  end                          % výpočet predikce
```

```
  yp=[yp ypt]; time=[time TIME]; ws=[ws w'];
end

% VÝSLEDKY
dd = DATA(pchns,time+1);        % predikovaná data
yp = yp(pchns,:);               % predikce
ep = dd-yp;                     % chyby predikce
SE = sqrt(var(ep')+mean(ep').^2)./std(dd')
 % chyby predikce vzhledem k směrodatné odchylce dat
al                              % váhy komponent

figure(1)
chn = 2;
plot(time, dd(chn,:),'ob','MarkerSize',3)
hold on
plot(time, yp(chn,:),'xr','MarkerSize',3)
hold off
 % predikce časových průběhů

if length(pchns)>1
  figure(2)
  subplot(121),plot(dd(1,:),dd(2,:),'.','MarkerSize',5)
  axis([-5 10 -5 10]), title('Data clusters');
  subplot(122),plot(yp(1,:),yp(2,:),'.','MarkerSize',5)
  axis([-5 10 -5 10]), title('Predicted clusters');
end
 % predikce shluků
```

1.4.5 Jednoduchý příklad - dynamická verze

Simulace tří dvourozměrných dynamických komponent

```
prodini, echo off
ychns = [1 2];
 % modelované kanály
str=[1 2 0;1 1 1];
 % struktura pro 1. řád modelu
 % váhy komponent
sd=.1;
Pt{1}.Eth = [.9 0 0; 0 .5 0];
Pt{2}.Eth = [.7 0 5; 0 .8 0];
Pt{3}.Eth = [.6 0 3; 0 .1 8];
Pt{1}.cove = sd;  % sám si určí dimenze !
Pt{2}.cove = sd;
Pt{3}.cove = sd;
 % parametry komponent
Sim = genmixs(ychns, str, dfcs, Pt);
 % vytvoření simulátoru
ndat=1000;
 % počet kroků simulace
DATA=zeros(2,ndat);

% SIMULACE
mixsimul(Sim, ndat)
save simDat3 Sim DATA
  % uložení dat na disk

% ZOBRAZENÍ
plot(DATA(1,:),DATA(2,:),'.','MarkerSize',1)
```

Identifikace tří dvourozměrných dynamických komponent

```
prodini, echo off load simDat3;

 % PŘÍPRAVA IDENTIFIKACE
 ncom = 3;
  % počet komponent
 ychns = [1 2];
  % modelované kanály
 str=[0 1 2;1 1 1];
  % struktura pro statické faktory
 ndat=1000;
  % počet kroků pro identifikaci
  dCt = 1e5; dco = 1e-3; dfm = 10;
  dfs = .1*ndat*ones(1,ncom);
   % počáteční statistiky
```

```
Mix0 = genmixe(ncom, ychns, str, ndat, dCt, dco, dfm, dfs);
%Mix0 = genmixe(ncom, ychns, str, ndat); !!! špatné předvolby
% vytvoření estimátoru

% IDENTIFIKACE
frg = .99;
Mix = mixest(Mix0, frg, ndat);
save simMix3

% VÝSLEDKY
M24 = mix2mix(Mix, 24);
Eth1=M24.Coms{1}.Eth
Eth2=M24.Coms{2}.Eth
Eth3=M24.Coms{3}.Eth

Data = DATA;
DATA = zeros(size(DATA));
mixsimul(Mix, ndat);
figure(1)
 % Porovnání simulace a identifikace přes datové shluky
subplot(121),plot(Data(1,:),Data(2,:),'bx')
subplot(122),plot(DATA(1,:),DATA(2,:),'rx')
```

Predikce tří dvourozměrných dynamických komponent

```
prodini, echo off
load simMix;

% PREDIKCE
nstep = 5;
 % počet kroku predikce
pchns = [1 2]; cchns = [];
 % predikované kanály a kanály v podmínce
pMix = mix2pro(Mix, pchns, cchns);
 % podmíněná směs
yp = []; time = []; ws = [];
for TIME = Mix.states.maxtd+1:ndat-nstep
  [Eths, coves, al, w] = profixn(pMix,[],[],nstep);
  th = GetTh(Eths);
  if 1, ypt = th*w';            % okamžité váhy
  else  ypt = th*al';           % průměrné váhy
  end                         % výpočet predikce
  yp=[yp ypt]; time=[time TIME]; ws=[ws w'];
end

% VÝSLEDKY
dd = DATA(pchns,time+1);        % predikovaná data
yp = yp(pchns,:);               % predikce
```

```
ep = dd-yp;                          % chyby predikce
SE = sqrt(var(ep')+mean(ep').^2)./std(dd')
 % chyby predikce vzhledem k směrodatné odchylce dat
al                                   % váhy komponent

figure(1)
chn = 2;
plot(time, dd(chn,:),'ob','MarkerSize',3)
hold on
plot(time, yp(chn,:),'xr','MarkerSize',3)
hold off
 % predikce časových průběhů

if length(pchns)>1
  figure(2)
  subplot(121),plot(dd(1,:),dd(2,:),'.','MarkerSize',5)
  axis([-5 10 -5 10]), title('Data clusters');
  subplot(122),plot(yp(1,:),yp(2,:),'.','MarkerSize',5)
  axis([-5 10 -5 10]), title('Predicted clusters');
end
 % predikce shluků
```

1.4.6 Obecný příklad

Simulace 'ncom', 'nchn'-rozměrných dynamických komponent

```
prodini, echo off
ncom = 5;
 % počet komponent
nchn = 3;
 % počet modelovaných kanálů
ychns = 1:nchn;
 % modelované kanály
str = [1:nchn 0; 1*ones(1,nchn) 1];
 % struktura pro první řád modelu
dfcs = 1+10*rand(1,ncom);
 % váhy komponent
sd=.1;
Pt{1}.Eth = [.9 0 0 0; 0 .5 0 0; 0 0 .6 0];
Pt{2}.Eth = [.7 0 0 5; 0 .8 0 2; 0 0 .4 1];
Pt{3}.Eth = [.6 0 0 3; 0 .1 0 8; 0 0 .7 6];
Pt{4}.Eth = [.3 0 0 2; 0 .9 0 6; 0 0 .7 3];
Pt{5}.Eth = [.4 0 0 8; 0 .7 0 2; 0 0 .5 1];
for i=1:ncom, Pt{i}.cove=sd; end
 % parametry komponent
Sim = genmixs(ychns, str, dfcs, Pt);
 % vytvoření simulátoru
ndat=1000;
 % počet kroků simulace
DATA=zeros(nchn,ndat);

% SIMULACE
mixsimul(Sim, ndat)
save simDat4 Sim DATA
  % uložení dat na disk

% ZOBRAZENÍ
ch1=1; ch2=2;
figure(1)
plot(DATA(ch1,:),DATA(ch2,:),'.','MarkerSize',1)
```

Identifikace 'ncom', 'nchn'-rozměrných dynamických komponent

```
prodini, echo off
load simDat4;

% PŘÍPRAVA IDENTIFIKACE
ncom = 5;
 % počet komponent
nchn = size(DATA,1);
```

```
 % počet modelovaných kanálů
ychns = 1:nchn;
 % modelované kanály
str = [1:nchn 0; 1*ones(1,nchn) 1];
 % struktura pro statické faktory
ndat = size(DATA,2);
 % počet kroků pro identifikaci
 dCt = 1e5; dco = 1e-3; dfm = 10;
 dfs = .1*ndat*ones(1,ncom);
  % počáteční statistiky
Mix0 = genmixe(ncom, ychns, str, ndat, dCt, dco, dfm, dfs);
%Mix0 = genmixe(ncom, ychns, str, ndat); !!! špatné předvolby
% vytvoření estimátoru

% IDENTIFIKACE
frg = .99;
Mix = mixest(Mix0, frg, ndat);
save simMix4

% VÝSLEDKY
M24 = mix2mix(Mix, 24);
for i=1:ncom, Eth1=M24.Coms{i}.Eth, end
Data = DATA;
DATA = zeros(size(DATA));
mixsimul(Mix, ndat);
 % Porovnání simulace a identifikace přes datové shluky
ch1=2; ch2=3;
figure(1)
d1=[Data(ch1,:) DATA(ch1,:)];
d2=[Data(ch2,:) DATA(ch2,:)];
mi1=min(d1); ma1=max(d1); mi2=min(d2); ma2=max(d2);
subplot(121),plot(Data(ch1,:),Data(ch2,:),'bx')
axis([mi1 ma1 mi2 ma2])
subplot(122),plot(DATA(ch1,:),DATA(ch2,:),'rx')
axis([mi1 ma1 mi2 ma2])
```

Predikce 'ncom', 'nchn'-rozměrných statických komponent

```
%
% ------------------------------------------------
prodini, echo off
load simMix4

% PREDIKCE
nstep = 1;
 % počet kroku predikce
pchns = [1 2]; cchns = [3];
 % predikované kanály a kanály v podmínce
```

```
pMix = mix2pro(Mix, pchns, cchns);
 % podmíněná směs
yp = []; time = []; ws = [];
for TIME = Mix.states.maxtd+2:ndat-nstep
  [Eths, coves, al, w] = profixn(pMix,[],[],nstep);
  th = GetTh(Eths);
  if 1, ypt = th*w';            % okamžité váhy
  else  ypt = th*al';           % průměrné váhy
  end                           % výpočet predikce
  yp=[yp ypt]; time=[time TIME]; ws=[ws w'];
end

% VÝSLEDKY
dd = DATA(pchns,time+1);        % predikovaná data
yp = yp(pchns,:);               % predikce
ep = dd-yp;                     % chyby predikce
SE = sqrt(var(ep')+mean(ep').^2)./std(dd')
 % chyby predikce vzhledem k směrodatné odchylce dat
al                              % váhy komponent

figure(1)
chn = 1;
plot(time, dd(chn,:),'ob','MarkerSize',3)
hold on
plot(time, yp(chn,:),'xr','MarkerSize',3)
hold off
 % predikce časových průběhů

if length(pchns)>1
  figure(2)
  subplot(121),plot(dd(1,:),dd(2,:),'.','MarkerSize',5)
  axis([-5 20 -5 10]), title('Data clusters');
  subplot(122),plot(yp(1,:),yp(2,:),'.','MarkerSize',5)
  axis([-5 20 -5 10]), title('Predicted clusters');
end
 % predikce shluků
```

Kapitola 2

Spline modely

Modelování dopravních veličin je důležitým prostředkem pro zjišťování okamžitého stavu dopravy v dané oblasti, jeho predikci a jeho případné řízení. Budeme se zabývat tzv. jednorozměrným modelováním dopravy, tj. modelováním jedné dopravní veličiny (např. intenzity provozu) na jednom místě. Měřením této veličiny ve velkém městě dostaneme funkci charakteristického tvaru, který je dán dopravní situací v městech. Přes noc a brzy ráno je je provoz téměř nulový. Okolo 7 hod. prudce narůstá a téměř během celého dopoledne je v saturaci. Na poledne poněkud zeslabí a během odpoledne opět vzroste. Na večer, kolem 19 hod., postupně slábne až po 21 hod. ustává. Sledovaná funkce má tendenci k periodicitě, avšak velmi zvláštní. Průběhy za všední dny jsou velmi podobné a tvoří jednu skupinu, víkendy jsou odlišné a tvoří druhou skupinu. Průběhy v jednotlivých skupinách se liší jen velmi málo, ale jejich rozdíly určitě netvoří bílý šum. Průběhy v obou skupinách se dále ještě liší podle ročního období a dalších více nebo méně pravidelných faktorů jako prázdniny, svátky apod. Nepravidelnými faktory, které mohou tvořit řadu dalších samostatných skupin, jsou různé události (návštěvy ze zahraničí, demonstrace, uzavírky komunikací atd.).

Jeden z možných a také dosud nejpoužívanějších přístupů k jednorozměrnému modelování dopravy spočívá v použití statického modelu, opírajícího se o pevný průběh intenzity dopravy v daném místě. Tento průběh se získá prostým měřením v jednotlivých dnech týdne a ověřením jejich "stejnosti" pomocí korelačních metod. Na základě tohoto průběhu se potom provádí predikce a řízení dopravy (viz např. metoda TRANSYT).

Předkládaný přístup vychází ze statického modelování a snaží se o jeho vylepšení ve dvou směrech. Za prvé: aproximací se bude průběžně určovat "ideální" průběh intenzity, za druhé: pomocí zapomínání bude možno sledovat pomalé změny ideálního průběhu intenzity během roku. Mezi jednotlivými skupinami funkcí je možno přepínat ručně (podle data), nebo je možno využít směsových modelů a přepínat nebo mixovat jednotlivé komponenty modelu automaticky.

2.1 Spline funkce

2.1.1 Definice spline funkce

Spline funkcí stupně n s defektem $n-k$ nazveme funkci $p(x)$ reálného argumentu x, definovanou na systému intervalů $I_i = [x_{i-1}, x_i]$, $\;i = 1, 2, ..., N$, která splňuje dvě podmínky

1. na každém intervalu I_i, $i = 1, 2, ..., N$ je tvořena polynomem $p_i(x)$ stupně n,
2. na svém definičním oboru je spojitá a má spojité derivace až do řádu $k < n$.

Body x_i, $i = 0, 1, ..., N$ se nazývají *uzly* spline-funkce.

Poznámka: *Defekt je počet nespojitých derivací, který spline funkci "odlišuje od polynomu". Nespojitost nejvyšších derivací dává spline-funkci proti čistému polynomu "větší pružnost".*

Poznámka: *Prostým názvem spline funkce se většinou označuje spline-funkce s defektem 1, tedy tvořená polynomy stupně n a mající $n-1$ spojitých derivací.*

Poznámka: *Nejčastěji používaná je spline funkce prvního nebo třetího stupně. Spline funkce sudého stupně se nepoužívají, pokud k tomu není zvláštní důvod.*

2.1.2 Vyjádření spline funkce pomocí báze

Spline funkce určitého stupně tvoří vektorový prostor. Lze v něm tedy zvolit bázi. Pro naše účely je výhodná báze tzv. *fundamentálních spline funkcí* $q_i(x)$, $i = 0, 1, ..., N$, která je definována požadavkem, aby i-tá bázová funkce měla hodnotu jedna v i-tém uzlu a ve všech ostatních uzlech byla nulová. Tedy platí

$$q_i(x_i) = 1,\ \forall i \quad \text{a} \quad q_i(t_j) = 0,\ \forall i \neq j$$

V takové bázi jsou váhové koeficienty přímo rovny hodnotám funkce v jednotlivých uzlech

$$f(x) = \sum_{i=0}^{N} a_i q_i(x)$$

kde

$f(x)$ je spline funkce, kterou chceme reprezentovat v bázi,
$a_i = f(x_i)$ jsou koeficienty reprezentace funkce f v bázi,
$q_i(x)$ jsou fundamentální bázové funkce.

Použití fundamentálních bázových funkcí je z praktického hlediska nejvýznamnější u spline funkcí prvního řádu. Bázové funkce jsou zde totiž nenulové jen na dvou intervalech sousedících s příslušným uzlem. Pro spline funkce vyšších řádů se tato vlastnost lokality bázových funkcí nezachová. Uvedeme si proto příklad pro spline funkci prvního řádu.

Příklad 2.1 [Báze fundamentálních spline funkcí]

V případě spline funkcí 1. řádu jsou rovnice bázových funkcí $q_i(x)$, $i = 0, 1, \ldots, N$ následující

$$q_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{pro } x \in I_{i-1} \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{pro } x \in I_i \\ 0 & \text{jinde} \end{cases}$$

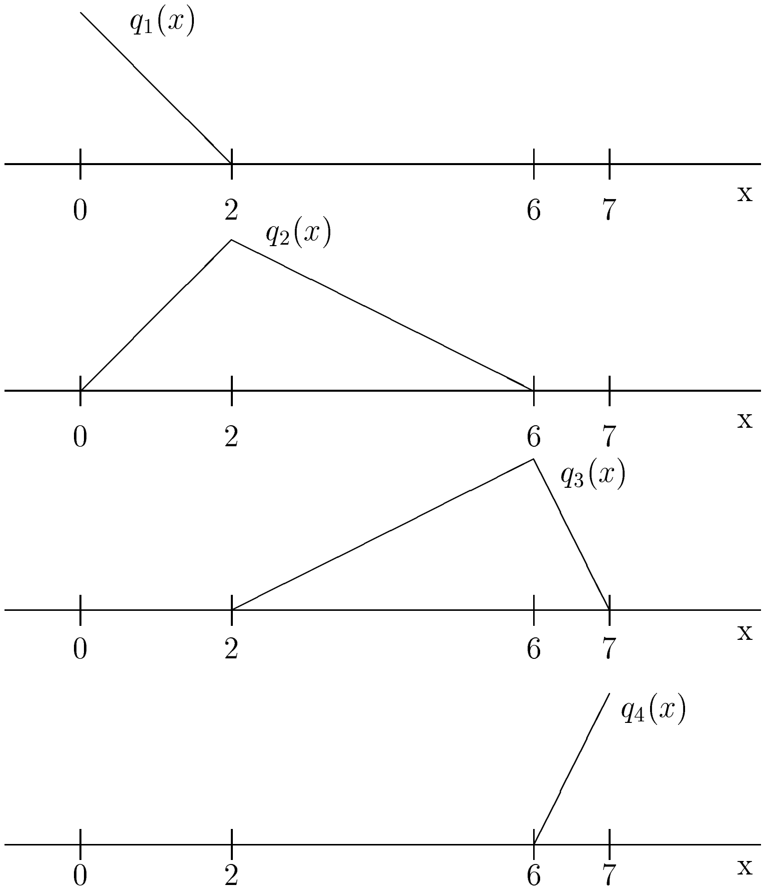
Uvažujme konkrétní situaci: lineární spline funkci f s uzly $x_i \in \{0, 2, 6, 7\}$ a odpovídajícími hodnotami v uzlech $f_i \in \{1, 5, 2, 3\}$.

Bázové funkce budou čtyři (dvě krajové budou mít jen jednu polovinu, dvě budou celé)

$$\begin{array}{rcl}
q_0(x) & = & 1 - 0.5x \quad \text{pro } I_1 \\
q_1(x) & = & \begin{cases} 0.5x & \text{pro } I_1 \\ 1.5 - 0.25x & \text{pro } I_2 \end{cases} \\
q_2(x) & = & \begin{cases} 0.25x - 0.5 & \text{pro } I_2 \\ 7 - x & \text{pro } I_3 \end{cases} \\
q_3(x) & = & x - 6 \quad \text{pro } I_3
\end{array}$$

Jinde jsou bázové funkce nulové.

Bázové funkce z tohoto příkladu jsou na následujícím obrázku.



Vyjádření spline funkce 1. řádu (tj. lineární lomené čáry s body zlomu v uzlech a hodnotami f_i v bodech zlomu) pomocí bázových funkcí je

$$f(x) = \sum_{i=0}^{3} f_i q_i(x) = 1.q_0(x) + 5.q_1(x) + 2.q_2(x) + 3.q_3(x)$$

`Poznámka:` *Správnost ověřte výpočtem.*

2.1.3 Interpolační spline funkce

O interpolační spline funkci hovoříme tehdy, jestliže požadujeme, aby spline funkce procházela předepsanými body. To lze samozřejmě jednoznačně splnit jen v případě, kdy počet předepsaných bodů přesně koresponduje s počtem uzlů, řádem a defektem spline funkce. Např. v případě defektu 1 musí být počet bodů roven počtu uzlů. Říkáme, že spline funkce interpoluje předepsané body.

Konstrukci interpolační spline funkce je nejlépe provést pomocí některé báze, např. pomocí již zmíněné báze fundamentálních spline funkcí. Pro účely modelování však tento druh spline funkcí není příliš zajímavý.

2.1.4 Aproximační spline funkce

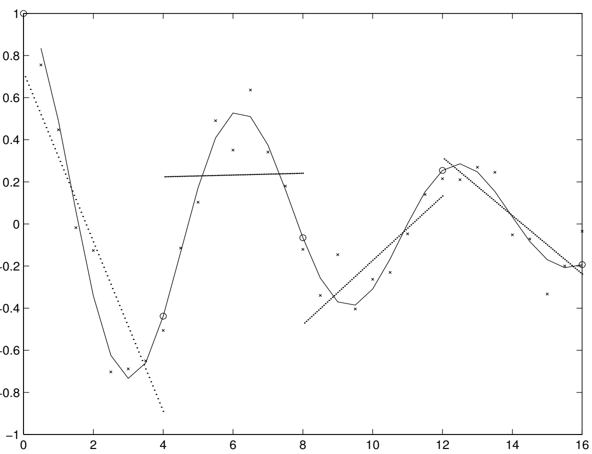
Aproximační spline funkci dostaneme tehdy, jestliže počet bodů, které má křivka sledovat, je větší a průchody křivky jednotlivými body nelze splnit. Potom se snažíme, aby tvar křivky byl takový, aby se křivka co nejvíce blížila jednotlivým bodům, např. ve smyslu nejmenších čtverců. Řekneme, že spline funkce aproximuje předepsané body.

Tento druh spline funkcí bude předmětem našeho zájmu při konstrukci spline modelů.

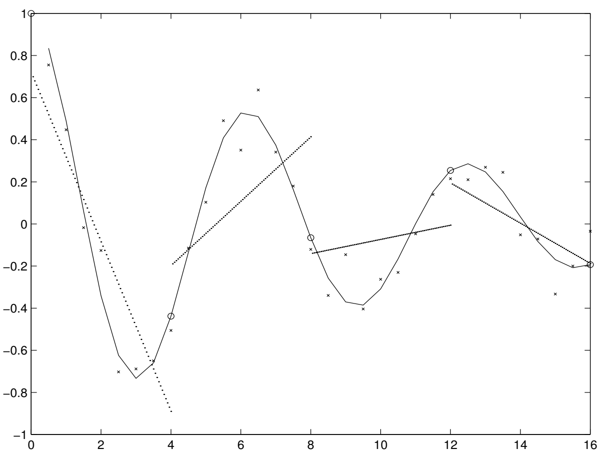
2.1.5 Zobecněné spline funkce

V obou předchozích případech jsme jako použitou křivku uvažovali spline funkci, tj. po částech polynomiální funkci s podmínkami hladkosti v uzlech. Podobně, jako jsme při přechodu od interpolačních spline funkcí k aproximačním ustoupili od požadavku přesného průchodu křivky předepsanými body, můžeme nyní uvažovat i o uvolnění přesných požadavků na hladkost křivky v uzlech. Splnění těchto požadavků budeme požadovat jen přibližně, a to s určitou váhou, např. opět metodou nejmenších čtverců. Přesnou spline funkci obdržíme s velkou váhou pro podmínky hladkosti, pro podmínky hladkosti s nulovou váhou dostáváme aproximaci nezávislými polynomy.

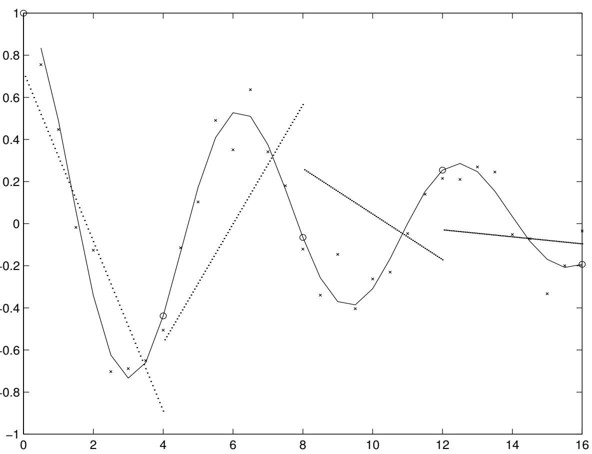
Vlastnosti zobecněných spline funkcí demonstruje seriál následujících obrázků (plná čára – původní funkce, křížky – měřené body, tj. funkce a šum, kroužky – uzly, tečkovaná čára – zobecněná spline funkce). Váhu podmínek hladkosti (zde spojitosti) označíme w. Kvalita aproximace je vyjádřena pomocí směrodatné odchylky chyby aproximace sde.



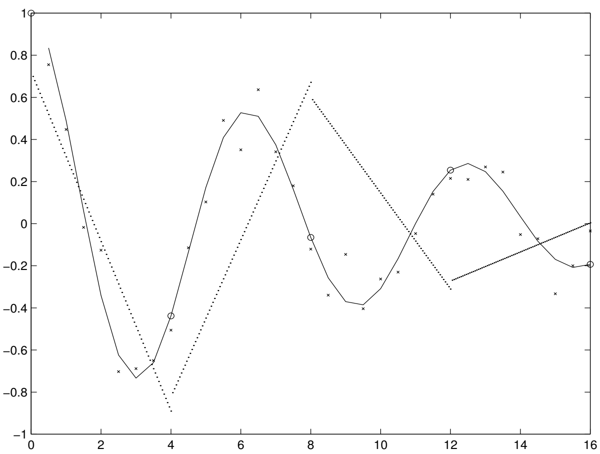
$w = 0,\ sde = 0.195$



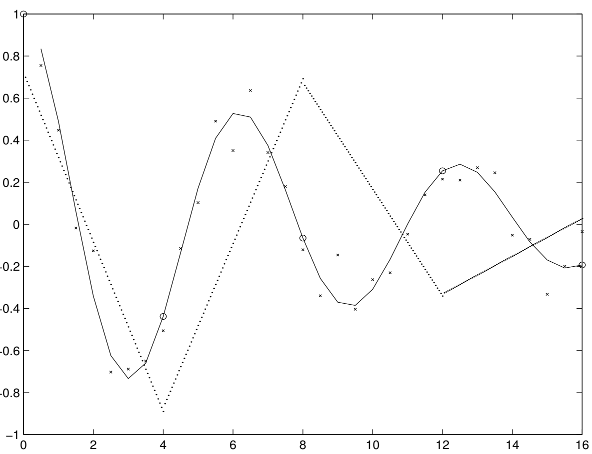
$w = 1,\ sde = 0.217$



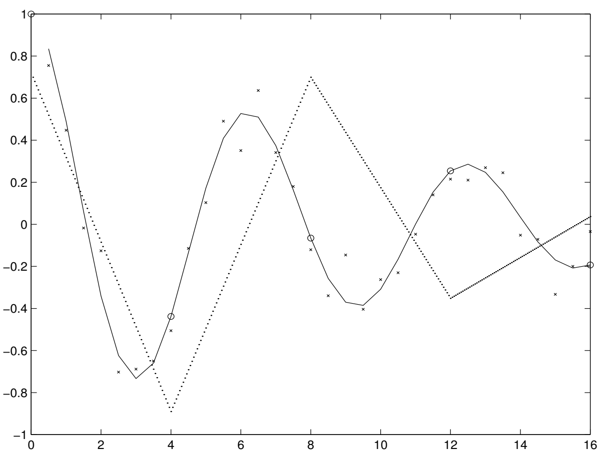
$w = 2,\ sde = 0.314$



$w = 5,\ sde = 0.417$



$w = 10,\ sde = 0.440$



$w = 100,\ sde = 0.449$

Z obrázků je zřejmé, že požadavek hladkosti jde na úkor kvality aproximace. Horší aproximace hladší funkcí však je většinou chápána jako filtrace zašuměné hladké křivky. Váha hladkosti pak určuje míru filtrace.

2.1.6 Práce se spline funkcemi

Pro vyjádření spline-funkce na intervalu I_i je výhodné zavést normovaný argument vztahem

$$\xi_i(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Tento normovaný argument na intervalu $[x_{i-1}, x_i]$ nabývá hodnot z intervalu $[0, 1]$. Jeho zavedením se vyhneme velkým hodnotám koeficientů použitých polynomů a dostaneme podmínky spojitosti v jednoduchém tvaru.

Celou spline funkci lze vyjádřit ve tvaru

$$f(x) = \sum_{i=1}^{N} p_i(\xi_i(x))\chi_i(x)$$

kde

$p_i(\xi_i(x))$ je i-tý polynom spline funkce,

$\chi_i(x)$ je indikátorová funkce, pro kterou platí

$$\chi_i(x) = \begin{cases} 1 & \text{pro } x \in I_i \\ 0 & \text{jinde.} \end{cases}$$

Příklad 2.2 [Normovaný argument]

Uvažujme polynom $p(x) = 1 - 2x + x^2$ na intervalu $x \in (0, 2)$. Posuneme-li tento interval o deset vpravo, bude

$$p(x - 10) = 1 - (x - 10)x + (x - 1)^2 = 121 + 22x + x^2.$$

Zavedeme-li normovaný argument $\xi(x) = \frac{x-10}{12-10} = 0.5x - 5$, bude mít posunutý polynom tvar

$$p(\xi(x)) = 1 + 4\xi + 4\xi^2,$$

s tím, že platí $\xi(10) = 0$ a $\xi(12) = 1$.

Tento tvar bude pro všechna posunutí stejný (samozřejmě vždy se svým normovaným argumentem) a normovaný argument bude vždy 0 na začátku intervalu a 1 na jeho konci.

Spline funkce 1. stupně

Na intervalu I_i je spline-funkce definována polynomem

$$p_i(\xi_i(x)) = a_i + b_i\xi_i(x). \tag{2.1}$$

Podmínky hladkosti se zde redukují na podmínky spojitosti, které pro dva navazující intervaly I_i a I_{i+1}, zahrnující uzly x_{i-1}, x_i a x_{i+1}, představují polynomiální rovnici

$$p_i(\xi_i(x)_i) = p_{i+1}(\xi_{i+1}(x)_i), \quad \text{tj.} \quad p_i(1) = p_{i+1}(0).$$

Dosadíme-li do předchozího vztahu vyjádření polynomů podle (2.1), dostaneme rovnici, vyjadřující spojitý přechod v uzlu x_i

$$a_i + b_i = a_{i+1}. \tag{2.2}$$

Odtud je vidět, že když stanovíme jednu počáteční podmínku vlevo (nejspíše hodnotu, kde spline funkce začíná), zbývá pro každý interval jen jediný volný koeficient. Druhý je určen podmínkami spojitosti.

Chceme-li tyto podmínky vyjádřit přibližně, použijeme tvar regresního modelu

$$a_i + b_i = a_{i+1} + \varepsilon, \tag{2.3}$$

kde

$a_i + b_i$ je známé číslo, (zde představuje změřenou hodnotu) a

ε je obecné označení chyby v přesném splnění podmínek (šum modelu).

Poznámka: *Podmínky lze použít buď bez šumu pro konstrukci "přesných" spline funkcí, nebo se šumem je přidat mezi podmínky aproximační. Výsledná spline funkce bude potom kompromisem mezi požadavky na aproximaci a spojitost.*

Spline funkce 3. stupně

Na intervalu I_i je spline-funkce definována polynomem

$$p_i(\xi_i(x)) = a_i + b_i\xi_i(x) + c_i\xi_i^2(x) + d_i\xi_i^3(x). \tag{2.4}$$

Podmínky hladkosti jsou zde vyjádřeny podmínkami na spojitost a návaznost 1. a 2. derivace, tedy pro uzel x_i

$$p_i(1) = p_{i+1}(0), \quad p_i'(1) = p_{i+1}'(0), \quad p_i''(1) = p_{i+1}''(0).$$

Po dosazení (2.4) dostáváme podmínky pro koeficienty polynomů

$$a_i + b_i + c_i + d_i = a_{i+1}, \tag{2.5}$$

$$b_i + 2c_i + 3d_i = b_{i+1}, \tag{2.6}$$

$$c_i + 3d_i = c_{i+1}. \tag{2.7}$$

Podmínky hladkosti lze samozřejmě použít stejně jako v případě spline funkce 1. stupně.

2.2 Spline model intenzity dopravního toku

Jak jsme již uvedli, budeme modelovat jednu dopravní veličinu (intenzitu dopravního toku) na jednom místě. Budeme uvažovat neřízený statický model ve tvaru

$$y_x = f_x + e_x \tag{2.8}$$

kde

x je časový okamžik měřený od určité referenční hodiny (např. 5.00 hod ráno).

y_x je modelovaná intenzita v časovém okamžiku x,

f_x je hodnota typického průběhu intenzity v čase x,

e_x je porucha, odchylka okamžité intenzity od její typické hodnoty.

Typickou hodnotu intenzity budeme modelovat pomocí zobecněné spline-funkce

$$f_x = \sum_{i=1}^{N} p_i(\xi_i(x))\chi_i(x) \tag{2.9}$$

kde p_i je polynom (2.1) nebo (2.4) na jehož koeficienty klademe "přibližné podmínky hladkosti".

Dosadíme-li předchozí vyjádření typické hodnoty průběhu intenzity do modelu (2.8), můžeme jej pro jeden den psát ve tvaru statického regresního modelu s proměnnou konstantou

$$y_x = \theta' \varphi_x + e_x \tag{2.10}$$

kde

θ je vektor parametrů (koeficientů všech polynomů tvořících spline funkci)
$\theta = [p_1, p_2, \ldots, p_N]'$, kde p_i je vektor koeficientů i-tého polynomu, řazený od absolutního členu.
Pro 1. resp. 3. stupeň spline funkce je

$$p_i = [a_i, b_i] \text{ resp. } p_i = [a_i, b_i, c_i, d_i], \quad i = 1, 2, \ldots, N,$$

φ_x je regresní vektor
$\varphi_x = [\phi_1\chi_1(x), \phi_2\chi_2(x), \ldots, \phi_N\chi_N(x)]$.
Opět pro 1. resp. 3. stupeň spline funkce a $i = 1, 2, \ldots, N$ platí

$$\phi_i = [1, \xi_i(x)], \quad \text{resp. } \phi_i = [1, \xi_i(x), \xi_i^2(x), \xi_i^3(x)].$$

Regresní vektor je tedy tvořen nulami, jen na místě, odpovídajícím právě aktuálnímu intervalu jsou umístěny normované argumenty příslušného polynomu. Takto konstruovaný regresní vektor představuje aproximační podmínky (tedy požadavek, aby spline funkce ležela blízko změřeným hodnotám).

Abychom zaručili požadovanou hladkost zobecněné spline-funkce, je třeba ještě zajistit "přibližné" splnění podmínek hladkosti, rovněž modelovaných ve tvaru regresního modelu (viz (2.3)), které pro 1. stupeň spline funkce jsou

$$a_i + b_i = a_{i+1} + \varepsilon \tag{2.11}$$

a pro 3. stupeň

$$a_i + b_i + c_i + d_i = a_{i+1} + \varepsilon \qquad \text{spojitost} \tag{2.12}$$

$$b_i + 2c_i + 3d_i = b_{i+1} + \varepsilon \qquad \text{1. derivace} \tag{2.13}$$

$$c_i + 3d_i = c_{i+1} + \varepsilon \qquad \text{2. derivace} \tag{2.14}$$

Vezmeme-li v úvahu strukturu vektoru parametrů, zavedenou modelem (2.10), odpovídají těmto podmínkám hodnoty výstupu rovny levým stranám předchozích rovnic (určeným z koeficientů předchozího polynomu) a regresní vektory, tvořené nulami a obsahující jedničku na příslušném místě odpovídajícím koeficientu z pravé strany rovnice. Tak např. pro 3. stupeň spline funkce a druhý interval I_2 bude tzv. virtuální regresní vektor φ roven:

$$\varphi = [0, 0, 0, 1, 0, \ldots] \qquad \text{pro spojitost}$$

$$\varphi = [0, 0, 0, 0, 1, 0, \ldots] \qquad \text{pro 1. derivaci}$$

$$\varphi = [0, 0, 0, 0, 0, 1, 0, \ldots] \qquad \text{pro 2. derivaci.}$$

Hodnoty výstupu budou postupně: $a_1 + b_1 + c_1 + d_1$; $b_1 + 2c_1 + 3d_1$ a $c_1 + 3d_1$.

Podobné vztahy platí i pro další intervaly, hodnoty na prvním intervalu jsou konstruovány na základě okrajových podmínek.

Identifikace parametrů probíhá např. metodou nejmenších čtverců, kam jako datový vektor dodáváme skutečný regresní vektor φ_x podle (2.10) a podmínky hladkosti respektujeme přidáním virtuálních regresních vektorů s příslušnými vahami. Velikost vah, s nimiž tyto regresní vektory použijeme určuje míru přesnosti splnění požadavků na hladkost aproximující zobecněné spline-funkce.

2.3 Program pro identifikaci spline-modelu

2.3.1 Hlavní program

```
cll, randn('seed',1234)
xx=0:4:16;                            % uzly (i neekvidistantní)
ff=func(xx);                          % konstrukce pro graf
h=.5;                                 % perioda vzorkování
ord=3;                                % řád spline funkce (1|3)
s=min(xx)+h:h:max(xx);                % body vzorkování
no=ord+2;

% Měření hodnot a identifikace
V=1e-5*eye(no);
yy=[]; y1=[]; aa=[];
for t=s                               % cyklus pro čas
  e=.1*randn;                         % šum
  ym=func(t);                         % výstup bez šumu
  y=ym+e;                             % výstup
  yy=[yy y]; y1=[y1 ym];              % body funkce (vektor y)
  [ti, ii]=xi(t, xx);                 % norm. argument a interval pro y
  in=n_int(t+h, xx);                  % interval pro příští y
  psi=datv(y,ti,ord);                 % datový vektor pro 1. řád
  V=V+psi'*psi;                       % update informační matice
  if in>ii                            % přechod na nový interval
    a=inv(V(2:no,2:no))*V(2:no,1);    % rozdělení inf. matice
    aa=[aa a];                        % koef. polynomů (po sloupcích)
    w=[1000 10 0];                    % váhy pro 0., 1. a 2. derivaci
    psi=datv0(w,a,ord);               % podmínky hladkosti
    V=psi'*psi;                       % inf. matice - podm. hladkosti
  end
end

% Výsledky
hs=.05;
```

```
ss=min(xx)+hs:hs:max(xx);            % body vzorkování
ys=[];
for t=ss                             % generování spline funkce
  [ti, ii]=xi(t, xx);                % norm. argument a č. intervalu
  psi=datv([],ti,ord);               % datový vektor
  yp=psi*aa(:,ii);                   % hodnota spline funkce
  ys=[ys yp];
end
ep=func(ss)-ys;                      % chyba aproximace
sde=std(ep)                          % směr. odch. chyby aproximace
sfg1
figure(1),plot(s,yy,'rx',s,y1,'r:',ss,ys,'c',xx,ff,'om','markersize',3)
```

2.3.2 Podpůrné funkce

Normovaný argument polynomu

```
function [xx, ii]=xi(xt, x)
% xx = xi(xt, x)
% výpočet normovaného argumentu pro spline-aproximaci
%
% xx   normovaný argument
% ii   pořadí aktuálního intervalu
% xt   aktuální argument
% x    vektor spline-uzlů
%
ii=n_int(xt,x);
xx=(xt-x(ii))/(x(ii+1)-x(ii));
```

Pořadí aktuálního intervalu spline funkce

```
function ii=n_int(xt, x)
% i = n_int(xt, x)
% pořadí aktuálního spline-intervalu
%
% ii   pořadí aktuálního intervalu
% xt   aktuální argument
% x    vector spline-uzlů
%
ii=sum(x<xt);
```

Konstrukce datového vektoru

```
function dv=datv(y,xi,ord)
% dv = datv(y,xi,ord)
% konstrukce datového vektoru
%
% dv  datový vektor = [data, regresní vektor]
%                   = [hodnota funkce, mocniny xi]
```

```
% y   změřená hodnota dat
% xi  normovaný argument
% ord řád spline-funkce
%
if isempty(y), dv=1;
else           dv=[y 1];
end
for i=1:ord
  dv=[dv xi^i];
end
```

Konstrukce datového vektoru pro podmínky hladkosti

```
function dv=datv0(w,a,ord)
% dv = datv0(w,a,ord)
%
% dv   datový vektor
% w    váhy pro podmínky hladkosti
% a    parametry polynomu z předchozího intervalu
% ord  řád spline-funkce
%
dv=w(1)*[sum(a) 1 zeros(1,ord)]; % pro 1. řád
if ord==3                        % pro 3. řád se přidá
  dv1=w(2)*[a(2)+2*a(3)+3*a(4) 0 1 0 0];
  dv2=w(3)*[2*a(3)+6*a(4) 0 0 2 0];
  dv=dv+dv1+dv2;
end
```