Recognition of Properties by Probabilistic Neural Networks

Jiří Grim[1] and Jan Hora[2]

[1] Institute of Information Theory and Automation
P.O. BOX 18, 18208 PRAGUE 8, Czech Republic
[2] Faculty of Nuclear Science and Physical Engineering
Trojanova 13, CZ-120 00 Prague 2, Czech Republic
grim@utia.cas.cz, hora@utia.cas.cz

Abstract. The statistical pattern recognition based on Bayes formula implies the concept of mutually exclusive classes. This assumption is not applicable when we have to identify some non-exclusive properties and therefore it is unnatural in biological neural networks. Considering the framework of probabilistic neural networks we propose statistical identification of non-exclusive properties by using one-class classifiers.

Keywords: Probabilistic neural networks, Non-exclusive classes, One-class classifiers, Biological compatibility.

1 Introduction

The statistical approach is known to enable general and theoretically well justified decision making in pattern recognition. Given the probabilistic description of the problem in terms of class-conditional probability distributions, we can classify objects described by discrete or continuous variables. The Bayes formula provides full classification information in terms of *a posteriori* probabilities of a finite number of classes. A unique decision, if desirable, can be obtained by means of Bayes decision function which minimizes the probability of error. We recall that the classification information contained in the *a posteriori* probabilities is partly lost if only a unique decision is available [5].

On the other hand, introducing Bayes formula, we assume that the unconditional distribution of the recognized data vectors can be expressed as a weighted sum of class-conditional distributions, according to the formula of complete probability. In this way we implicitly assume the classes to be mutually exclusive. Nevertheless, the probabilistic classes may overlap in the sample space, they are mutually exclusive just in the sense of the complete probability formula.

The abstract statistical concept of mutually exclusive classes is rather unnatural in biological systems since most real life categories are non-exclusive. In this respect the multiclass Bayes decision scheme is unsuitable as a theoretical background of neural network models. In the following we use the term *property* to emphasize the fact that the recognized object may have several different properties simultaneously. In order to avoid the strict assumption of mutually

C. Alippi et al. (Eds.): ICANN 2009, Part II, LNCS 5769, pp. 165–174, 2009.

exclusive classes we propose recognition of properties by probabilistic neural networks based on one-class classifiers. We assume that for each property there is a single training data set. To identify a property we evaluate the log-likelihood ratio of the related conditional probability distribution and of the product of unconditional univariate marginals. Hence the only information about alternative properties is assumed to be given in the form of global marginal distributions of all involved variables. The proposed recognition of properties has two qualitative advantages from the point of view of biological compatibility: a) it is applicable both to the non-exclusive and exclusive properties (cf. Sec. 5) and b) provides a unified approach to recognition of properties and feature extraction (cf. [4]).

The concept of probabilistic neural networks (PNNs) relates to the early work of Specht [10] who proposed a neural network model closely related to the non-parametric Parzen estimates of probability density functions. In comparison with other neural network models the PNN of Specht may save training time essentially but, according to Parzen formula, one neuron is required for each training pattern. Moreover, there is a crucial problem of the optimal smoothing of Parzen estimates in multidimensional spaces. The PNN approach of Specht has been modified by other authors and, in some cases, simplified by introducing finite mixtures [9]. In this paper we refer mainly to our results on PNNs published in the last years (cf. [3] - [6]). Unlike previous authors we approximate the class-conditional probability distributions by finite mixtures of product components. The product-mixture-based PNNs do not provide a new technique of pattern recognition but they may contribute to better understanding of the functional principles of biological neural networks [3], [4].

In the following we first discuss the theoretical differences between multiclass classifiers (Sec. 2) and one-class identification of properties (Sec. 3). In Sec. 4 we summarize basic features of probabilistic neural networks and their application to identification of properties. In Sec. 5 we compare both schemes in application to recognition of handwritten numerals.

2 Multiclass Bayes Decision Scheme

Considering the statistical pattern recognition we assume that some multivariate observations have to be classified with respect to a finite set of mutually exclusive classes $\Omega = \{\omega_1, \ldots, \omega_K\}$. The observation vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}$ from the N-dimensional space $\mathcal{X}$ (which may be real, discrete or binary) are supposed to occur randomly according to some class-conditional distributions $P(\boldsymbol{x}|\omega)$ with *a priori* probabilities $p(\omega), \omega \in \Omega$. Recall that, given an observation $\boldsymbol{x} \in \mathcal{X}$, all statistical information about the set of classes Ω is expressed by the Bayes formula for a posteriori probabilities

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})}, \qquad P(\boldsymbol{x}) = \sum_{\omega \in \Omega} P(\boldsymbol{x}|\omega)p(\omega), \qquad \boldsymbol{x} \in \mathcal{X} \tag{1}$$

where $P(\boldsymbol{x})$ is the joint unconditional probability distribution of $\boldsymbol{x}$.

The posterior distribution $p(\omega|\boldsymbol{x})$ may be used to define a unique final decision by means of the Bayes decision function [1]

$$d : \mathcal{X} \to \Omega, \qquad d(\boldsymbol{x}) = \arg\max_{\omega \in \Omega}\{p(\omega|\boldsymbol{x})\}, \quad \boldsymbol{x} \in \mathcal{X} \tag{2}$$

which is known to minimize the probability of classification error.

Remark 2.1. The Bayes decision function is a typical multiclass classifier directly identifying one of a finite number of classes on output. A weak point of Bayes classification is the unknown probabilistic description to be estimated from training data. In multidimensional spaces the number of training samples is usually insufficient to estimate the underlying distributions reliably and, in case of large data sets, even the computational complexity may become prohibitive.

Alternatively there are numerous non-statistical methods like support vector machines, AdaBoost, back-propagation perceptron and others having proved to yield excellent results in different practical problems. Unlike Bayes formula they are typically based on complex separating surfaces suitable to distinguish between two classes. In such a case the multiclass problems have to be reduced to multiple binary problems. It is possible to construct individual binary classifier for each class (one-against-all approach), to distinguish each pair of classes (all-pairs method) [8], or to use a more general method of error-correcting output codes which can utilize binary classifiers for all possible partitions of the set of classes [2]. We recall that, by nature of the underlying separating planes, multiclass solutions based on binary classifiers are discrete and therefore the *a posteriori* probabilities, if desirable, have to be approximated by heuristic means. There is no exact relation to the probability of classification error, usually the learning algorithm minimizes some heuristic criterion (e.g. a margin-based loss function [1]). As it can be seen, from the point of view of "binary" approximating multiclass decision functions, the concept of properties is basically irrelevant. □

3 Identification of Properties

In case of non-exclusive classes we assume that the multivariate observations may have some properties from a finite set $\Theta = \{\theta_1, \ldots, \theta_K\}$. Considering a single property $\theta \in \Theta$ we are faced with a two-class (binary) decision problem. For any given sample $\boldsymbol{x} \in \mathcal{X}$ we have to decide if the property is present or not. In other words, the decision is *positive*, (θ) if the property has been identified and *negative*, $(\bar{\theta})$ if it has not been identified. Since both alternatives are mutually exclusive, we can solve the binary classification problem in a standard statistical way. In full generality we denote $p(\theta)$ the *a priori* probability that the property θ occurs and $p(\bar{\theta}) = 1 - p(\theta)$ denotes the complementary *a priori* probability that the property is missing. Analogously, we denote $P(\boldsymbol{x}|\theta)$ and $P(\boldsymbol{x}|\bar{\theta})$ the conditional probability distributions of $\boldsymbol{x} \in \mathcal{X}$ given the property θ and $\bar{\theta}$ respectively. Thus, given the probabilistic description of a binary problem $\{\theta, \bar{\theta}\}$ we can write

$$P(\boldsymbol{x}) = P(\boldsymbol{x}|\theta)p(\theta) + P(\boldsymbol{x}|\bar{\theta})p(\bar{\theta}), \quad \boldsymbol{x} \in \mathcal{X}, \quad (p(\bar{\theta}) = 1 - p(\theta)) \tag{3}$$

[1] Here and in the following we assume that possible ties are uniquely decided.

and by using Bayes formula

$$p(\theta|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\theta)p(\theta)}{P(\boldsymbol{x})}, \quad p(\bar{\theta}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\bar{\theta})p(\bar{\theta})}{P(\boldsymbol{x})}. \tag{4}$$

we obtain the related decision function in the form

$$\Delta : \mathcal{X} \to \{\theta, \bar{\theta}\}, \quad \Delta(\boldsymbol{x}) = \begin{cases} \theta, & p(\theta|\boldsymbol{x}) \geq p(\bar{\theta}|\boldsymbol{x}), \\ \bar{\theta}, & p(\theta|\boldsymbol{x}) < p(\bar{\theta}|\boldsymbol{x}), \end{cases} \quad \boldsymbol{x} \in \mathcal{X}. \tag{5}$$

Remark 3.1. As mentioned earlier (cf. Remark 2.1), the problem of mutually exclusive classes can be formally decomposed into a set of "one-against-all" binary classification problems. If we define for each class $\omega_k \in \Omega$ the property $\theta = \{\omega_k\}$ and the opposite property, $\bar{\theta} = \Omega \setminus \{\omega_k\}$, then we can construct the two corresponding components $P(\boldsymbol{x}|\theta)p(\theta)$ and $P(\boldsymbol{x}|\bar{\theta})p(\bar{\theta})$ in terms of the class conditional distributions $P(\boldsymbol{x}|\omega)$. In particular, we can write

$$P(\boldsymbol{x}|\theta)p(\theta) = P(\boldsymbol{x}|\omega_k)p(\omega_k), \quad p(\theta) = p(\omega_k), \tag{6}$$

$$P(\boldsymbol{x}|\bar{\theta})p(\bar{\theta}) = \sum_{\omega \in \Omega_k} P(\boldsymbol{x}|\omega)p(\omega), \quad p(\bar{\theta}) = \sum_{\omega \in \Omega_k} p(\omega), \quad \Omega_k = \Omega \setminus \{\omega_k\}. \tag{7}$$

Expectedly, the classification accuracy of the multi-class decision function (2) may be different from that of the binary decision functions (5) based on the distributions (6) and (7). We discuss the problem in Sec. 4 in detail. □

Let us recall that by introducing binary classifiers we assume the training data to be available both for the property θ and for its opposite $\bar{\theta}$. Unfortunately, in real life situations it is often difficult to characterize the negative property $\bar{\theta}$ and to get the corresponding representative training data. In such cases the binary classifier (5) cannot be used since the probabilistic description of the negative property is missing. In this sense the identification of properties is more naturally related to one-class classifiers [11], [12] when only a single training data set for the "target" class is available.

Given some training data set $\mathcal{S}_\theta$ for a property $\theta \in \Theta$, we can estimate the conditional distribution $P(\boldsymbol{x}|\theta)$ and the property θ could be identified by simple thresholding. Nevertheless, the choice of a suitable threshold value is difficult if any information about the opposite property $\bar{\theta}$ is missing [12]. In the following we assume a general "background" information in the form of unconditional marginal distributions $P_n(x_n)$ of the variables x_n which is applicable to all properties $\theta \in \Theta$. This approach is motivated by the underlying PNN framework since the information about the unconditional marginal probabilities $P_n(x_n)$ may always be assumed to be available at the level of a single neuron.

In order to identify a property $\theta \in \Theta$ we propose to use a one-class-classifier condition based on the log-likelihood ratio

$$\pi_\theta(\boldsymbol{x}) = \log \frac{P(\boldsymbol{x}|\theta)}{\prod_{n \in \mathcal{N}} P_n(x_n)} \geq \epsilon. \tag{8}$$

Note that asymptotically the mean value of the criterion $\pi_\theta(\boldsymbol{x})$ converges to the Kullback-Leibler discrimination information between the two distributions $P^*(\boldsymbol{x}|\theta)$ and $\prod_{n\in\mathcal{N}} P_n(x_n)$

$$\bar{\pi}_\theta = \frac{1}{|\mathcal{S}_\theta|}\sum_{x\in\mathcal{S}_\theta}\log\frac{P(\boldsymbol{x}|\theta)}{\prod_{n\in\mathcal{N}}P_n(x_n)} \quad\rightarrow\quad \sum_{x\in\mathcal{X}} P^*(\boldsymbol{x}|\theta)\log\frac{P^*(\boldsymbol{x}|\theta)}{\prod_{n\in\mathcal{N}}P_n(x_n)}. \tag{9}$$

The last expression is nonnegative and for independent variables it is zero. In this sense it can be interpreted as a measure of dependence of the involved variables distributed by $P^*(\boldsymbol{x}|\theta)$ (cf. [3]).

We recall that the criterion $\pi_\theta(\boldsymbol{x})$ does not depend on the a priori probability $p(\theta)$. The property $\theta \in \Theta$ is identified if the probability $P(\boldsymbol{x}|\theta)$ is significantly higher than the corresponding product probability $\prod_{n\in\mathcal{N}} P_n(x_n)$. The threshold value ϵ in (8) can be related to the log-likelihood function of the estimated distribution $P(\boldsymbol{x}|\theta)$ (cf. (21)). Thus the only information about the negative properties $\bar{\theta}$ is contained in the unconditional product distribution $\prod_{n\in\mathcal{N}} P_n(x_n)$ which implies the assumption of independence of the variables x_n.

4 Probabilistic Neural Networks

Considering PNNs we approximate the class-conditional distributions $P(\boldsymbol{x}|\omega)$ by finite mixtures of product components

$$P(\boldsymbol{x}|\omega) = \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|m)f(m), \quad F(\boldsymbol{x}|m) = \prod_{n\in\mathcal{N}} f_n(x_n|m). \tag{10}$$

Here $\mathcal{M}_\omega$ are the component index sets of different classes, $\mathcal{N} = \{1,\ldots,N\}$ is the index set of variables, $f(m)$ are probabilistic weights and $F(\boldsymbol{x}|m)$ are the products of component specific univariate distributions $f_n(x_n|m)$.

In order to avoid the biologically unnatural complete interconnection of neurons we have introduced the structural mixture model [5], [6]. In particular, considering binary variables $x_n \in \{0,1\}$, we define

$$F(\boldsymbol{x}|m) = F(\boldsymbol{x}|0)G(\boldsymbol{x}|m,\phi_m)f(m),\ m\in\mathcal{M}_\omega \tag{11}$$

where $F(\boldsymbol{x}|0)$ is a "background" probability distribution defined as a fixed product of global marginals

$$F(\boldsymbol{x}|0) = \prod_{n\in\mathcal{N}} f_n(x_n|0) = \prod_{n\in\mathcal{N}} \vartheta_{0n}^{x_n}(1-\vartheta_{0n})^{1-x_n}, \quad (\vartheta_{0n} = \mathcal{P}\{x_n = 1\}) \tag{12}$$

and the component functions $G(\boldsymbol{x}|m,\phi_m)$ include additional binary structural parameters $\phi_{mn} \in \{0,1\}$

$$G(\boldsymbol{x}|m,\phi_m) = \prod_{n\in\mathcal{N}}\left[\frac{f_n(x_n|m)}{f_n(x_n|0)}\right]^{\phi_{mn}} = \prod_{n\in\mathcal{N}}\left[\left(\frac{\vartheta_{mn}}{\vartheta_{0n}}\right)^{x_n}\left(\frac{1-\vartheta_{mn}}{1-\vartheta_{0n}}\right)^{1-x_n}\right]^{\phi_{mn}}. \tag{13}$$

The main advantage of the structural mixture model is the possibility to confine the decision making only to "relevant" variables. Making substitution (11) in (6), we can express the probability distributions $P(\boldsymbol{x}|\omega), P(\boldsymbol{x})$ in the form

$$P(\boldsymbol{x}|\omega) = \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|m)f(m) = F(\boldsymbol{x}|0) \sum_{m\in\mathcal{M}_\omega} G(\boldsymbol{x}|m,\phi_m)f(m), \tag{14}$$

$$P(\boldsymbol{x}) = \sum_{\omega\in\Omega} p(\omega)P(\boldsymbol{x}|\omega) = F(\boldsymbol{x}|0) \sum_{m\in\mathcal{M}} G(\boldsymbol{x}|m,\phi_m)w_m, \quad w_m = p(\omega)f(m).$$

As the background distribution $F(\boldsymbol{x}|0)$ cancels in the Bayes formula we obtain

$$p(\omega|\boldsymbol{x}) = \frac{\sum_{m\in\mathcal{M}_\omega} G(\boldsymbol{x}|m,\phi_m)w_m}{\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j} = \sum_{m\in\mathcal{M}_\omega} q(m|\boldsymbol{x}), \quad \omega\in\Omega. \tag{15}$$

$$q(m|\boldsymbol{x}) = \frac{w_m G(\boldsymbol{x}|m,\phi_m)}{\sum_{j\in\mathcal{M}} w_j G(\boldsymbol{x}|j,\phi_j)}, \quad \boldsymbol{x}\in\mathcal{X}. \tag{16}$$

Thus the posterior probability $p(\omega|\boldsymbol{x})$ becomes proportional to a weighted sum of the component functions $G(\boldsymbol{x}|m,\phi_m)$ each of which can be defined on a different subspace. In other words the input connections of a neuron can be confined to an arbitrary subset of input nodes. The structural mixtures (14) can be optimized by means of EM algorithm in full generality (cf. [3] -[6]).

In view of Eq. (15) the structural mixture model provides a statistically correct subspace approach to Bayesian decision-making. In particular, considering Eq. (15), we can write the decision function (2) equivalently in the form

$$d(\boldsymbol{x}) = \omega_k : \quad \sum_{m\in\mathcal{M}_{\omega_k}} q(m|\boldsymbol{x}) \geq \sum_{m\in\mathcal{M}_\omega} q(m|\boldsymbol{x}), \quad \forall\omega\in\Omega_k,\ \boldsymbol{x}\in\mathcal{X}. \tag{17}$$

Remark 4.1. Applying binary classifier (5) to the multiclass problem of Remark 3.1, we can write (cf. (4), (6), (7))

$$\Delta(\boldsymbol{x}) = \{\omega_k\} : \quad p(\omega_k|\boldsymbol{x}) \geq p(\Omega_k|\boldsymbol{x}) = 1 - p(\omega_k|\boldsymbol{x}), \quad \boldsymbol{x}\in\mathcal{X} \tag{18}$$

and, after substitution (15), we obtain the following equivalent form of Eq. (18)

$$\Delta(\boldsymbol{x}) = \{\omega_k\} : \quad \sum_{m\in\mathcal{M}_{\omega_k}} q(m|\boldsymbol{x}) \geq \frac{1}{2}, \quad \boldsymbol{x}\in\mathcal{X}. \tag{19}$$

Condition (19) is stronger than (17) and, unlike the multiclass decision function (17), it may happen that no class will be identified by the binary classifiers (18) for a given $\boldsymbol{x}\in\mathcal{X}$. However, the two different decision functions (17) and (19) will perform comparably in multidimensional problems. In high dimensional spaces the mixture components $F(\boldsymbol{x}|m)$ in (14) are almost non-overlapping and therefore the conditional weights $q(m|\boldsymbol{x})$ have nearly binary properties by taking values near zero or one. It can be seen that if for some $m_0\in\mathcal{M}_{\omega_k}$ the value

$q(m_0|\boldsymbol{x})$ is near to one then both the multiclass and binary classifiers (17) and (19) will decide equally $d(\boldsymbol{x}) = \Delta(\boldsymbol{x}) = \omega_k$. In numerical experiments we have obtained false positive and false negative frequencies differing in both schemes in several units only. □

The structural mixture model (14) is particularly useful to identify the properties by means of the one-class-classifier condition (8). In view of definition of the background distribution $F(\boldsymbol{x}|0)$ and considering the properties $\theta = \{\omega\}, \omega \in \Omega$ defined by numerals, we can write (cf. (8), (12), (14)):

$$\pi_\omega(\boldsymbol{x}) = \log\left[\frac{P(\boldsymbol{x}|\omega)}{\prod_{n\in\mathcal{N}} P_n(x_n)}\right] = \log \sum_{m\in\mathcal{M}_\omega} G(\boldsymbol{x}|m,\phi_m) f(m) \geq \epsilon_\omega. \tag{20}$$

The mean value of the criterion $\pi_\omega(\boldsymbol{x})$ is actually maximized by EM algorithm since the background distribution $F(\boldsymbol{x}|0)$ is fixed and *a priori* chosen. Thus, having estimated the conditional distributions $P(\boldsymbol{x}|\omega)$ we can derive the threshold values ϵ_ω for each property $\omega \in \Omega$ from the related log-likelihood function:

$$\epsilon_\omega = \frac{c_0}{|\mathcal{S}_\omega|} \sum_{x\in\mathcal{S}_\omega} \pi_\omega(\boldsymbol{x}) = \frac{c_0}{|\mathcal{S}_\omega|} \sum_{x\in\mathcal{S}_\omega} \log \sum_{m\in\mathcal{M}_\omega} G(\boldsymbol{x}|m,\phi_m) f(m). \tag{21}$$

Here the coefficient c_0 can be used to control the general trade-off between the false positive and false negative decisions.

The proposed statistical recognition of properties based on the threshold condition (20) is closely related to the mutually exclusive Bayesian decision making. Note that by choosing $\omega \in \Omega$ which maximizes $\pi_\omega(\boldsymbol{x})$ we obtain Bayes decision function very similar to (17). The only difference is the missing *a priori* probability $p(\omega)$ which implies the latent assumption of equiprobable classes.

5 Numerical Example

To illustrate the problem of recognition of properties we have applied PNNs to the widely used benchmark NIST Special Database 19 (SD19) containing about 400000 handwritten numerals. It is one of the few sufficiently large databases to test statistical classifiers in multidimensional spaces. The SD19 digit database consists of 7 different parts - each of about 60000 digits. They were written by Census Bureau field personnel stationed throughout the United States, except for one part (denoted as hsf_4) written by high school students in Bethesda, Maryland. Thus different parts of the SD19 database differ in origin and also in the quality. In particular, the digits written by students are known to be more difficult to recognize. Unfortunately, in the report [7] there is no unique recommendation concerning the choice of the training and test set respectively, except that the hsf_4 data should not be used as a test set. A frequent choice of the test set is to use numerals written by "independent" persons - not involved in preparation of training data. However, the use of "writer independent" test data is incorrect from the statistical point of view. The purpose of any benchmark data

Table 1. Recognition of numerals from the NIST SD19 database by differently complex structural mixtures. The classification error is given in the last row.

Experiment No.	I	II	III	IV	V	VI	VII	VIII
Number of Components	10	40	80	100	299	858	1288	1571
Number of Parameters	10240	38758	77677	89973	290442	696537	1131246	1462373
Classification error in %	**11.93**	**6.23**	**4.81**	**4.28**	**2.93**	**2.31**	**1.95**	**1.84**

Table 2. Classification error matrix obtained in the Experiment VIII. Each row contains frequencies of decisions for test data from a given class. The last column contains percentage of false negative decisions. The last row contains the total frequencies of false positive decisions in percent of all test patterns.

CLASS	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	false n.
0	19950	8	43	19	39	32	36	0	38	17	1.1 %
1	2	22162	30	4	35	7	18	56	32	6	0.9 %
2	32	37	19742	43	30	9	8	29	90	16	1.5 %
3	20	17	62	20021	4	137	2	28	210	55	2.6 %
4	11	6	19	1	19170	11	31	51	30	247	2.1 %
5	25	11	9	154	4	17925	39	6	96	34	2.1 %
6	63	10	17	6	23	140	19652	1	54	3	1.6 %
7	7	12	73	10	73	4	0	20497	22	249	2.1 %
8	22	25	53	97	30	100	11	11	19369	72	2.1 %
9	15	13	25	62	114	22	3	146	93	19274	2.5 %
false p.	0.09%	0.07%	0.27%	0.20%	0.17%	0.23%	0.07%	0.16%	0.33%	0.35%	**1.84%**

is to test the statistical performance of classifiers regardless of any "practically useful" aspects. For this reason the statistical properties of the training- and test data should be identical since otherwise we test how the classifier "overcomes" the particular differences between both sets. In order to guarantee the same statistical properties of both training and test data sets we have used the odd samples of each class for training and the even samples for testing. We have normalized all digit patterns (about 40000 for each numeral) to a 32x32 binary raster. In order to increase the natural variability of data we have extended the training data sets four times by making three differently rotated variants of each pattern (by -10,-5,+5 degrees) - with the resulting sets of about 80000 patterns for each numeral. The same procedure has been applied to the test data, too.

First, considering the problem of mutually exclusive classes, we have used the training data to estimate the class-conditional mixtures for all numerals separately by means of EM algorithm. Each digit pattern has been classified by the *a posteriori* probabilities computed for the most probable variant of the four rotated test patterns, i.e. for the variant with the maximum probability $P(\boldsymbol{x})$.

Table 3. Identification of properties (numerals) by means of one-class classifier. Each row corresponds to one test class, the columns contain frequencies of the identified numerals respectively.

CLASS	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	false n.
0	18815	2	954	23	30	292	76	0	406	103	6.8 %
1	6	21857	55	46	2756	111	52	4436	5039	410	2.2 %
2	4	9	18660	105	5	2	6	6	207	3	6.9 %
3	6	2	43	18971	1	1733	0	12	3177	373	7.7 %
4	1	0	6	1	18494	5	5	83	265	3229	5.5 %
5	7	2	4	918	0	17211	35	0	1246	282	6.0 %
6	50	10	30	0	60	888	18833	0	360	1	5.7 %
7	1	5	601	324	209	4	0	19817	242	6735	5.4 %
8	9	13	22	620	19	289	6	5	18201	154	8.0 %
9	3	4	6	70	1722	90	2	1060	1266	18667	5.6 %
false p.	0.0%	0.0%	0.9%	1.0%	2.4%	1.7%	0.1%	2.8%	6.1%	5.6%	**6.0%**

This approach simulates a biological analyzer choosing the best position of view. Table 1 shows the classification accuracy of differently complex mixture models, as estimated in eight independent experiments. The total numbers of mixture components and of the component specific parameters ($\sum_m \sum_n \phi_{mn}$) are given in the second and third row of Tab. 1 respectively. The last row contains the classification error in percent. It can be seen that the underlying mixture model is rather resistant against overfitting.

Table 2 comprises detailed classification results of the decision function from the Experiment VIII (cf. Tab. 1, last column) in terms of error frequency matrix. Each row contains frequencies of different decisions for the respective class with the correct classifications on diagonal. The last "false negative" column contains the error frequencies in percent. Similarly, the last "false positive" row of the table contains frequencies of incorrectly classified numerals in percent.

Table 3 shows how the properties (numerals) can be identified by means of one-class classifier (20). The threshold values have been specified according to the Eq. (21) by setting $c_0 = 0.75$ after some experiments[2]. Each column contains frequencies of decisions obtained by the one-class classifier (20) for the respective numeral (first row). Hence, the numbers on the diagonal correspond to correctly identified numerals. The last column contains percentage of false negative decisions defined as complement of the correctly identified patterns. The last row contains percentage of the false positive decisions which correspond to incorrectly identified numerals. Note that the only difference between Tables 2 and 3 is the information about the mutual exclusivity of the recognized numerals which is not available in case of properties. As the a priori probabilities of numerals are nearly identical, the resulting tables reflect the net gain provided by the Bayes formula.

[2] A validation set would be necessary to optimize the trade-off between the false negative and false positive decisions and also the underlying mixture complexity.

6 Concluding Remark

We propose to identify properties by means of one-class-classifier based on the log-likelihood ratio which compares the conditional probability distribution of the "target" property with the product of univariate marginals of the unconditional background distribution. The only information about the "negative" properties is contained in the global univariate marginals of involved variables. In the numerical example we compare the problem of identification of properties with the standard "multiclass" Bayes decision function. The proposed identification of properties performs slightly worse than Bayes rule because of the ignored mutual exclusivity of classes. On the other hand recognition of properties should be more advantageous in case of non-exclusive classes. The method is applicable both to the non-exclusive and exclusive properties and provides a unified approach to recognition of properties and feature extraction in the framework of probabilistic neural networks.

Acknowledgement. Supported by the project GAČR No. 102/07/1594 of Czech Grant Agency and by the projects 2C06019 ZIMOLEZ and MŠMT 1M0572 DAR.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2001)
2. Dietterich, T.G., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. Journal of Artif. Intell. Res. 2, 263–286 (1995)
3. Grim, J.: Neuromorphic features of probabilistic neural networks. Kybernetika 43(5), 697–712 (2007)
4. Grim, J., Hora, J.: Iterative principles of recognition in probabilistic neural networks. Neural Networks, Special Issue 21(6), 838–846 (2008)
5. Grim, J., Kittler, J., Pudil, P., Somol, P.: Information analysis of multiple classifier fusion. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 168–177. Springer, Heidelberg (2001)
6. Grim, J.: Extraction of binary features by probabilistic neural networks. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008,, Part II. LNCS, vol. 5164, pp. 52–61. Springer, Heidelberg (2008)
7. Grother, P.J.: NIST special database 19: handprinted forms and characters database, Technical Report and CD ROM (1995)
8. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. The Annals of Statistics 2(26), 451–471 (1998)
9. Haykin, S.: Neural Networks: a comprehensive foundation. Morgan Kaufman, San Francisco (1993)
10. Specht, D.F.: Probabilistic neural networks for classification, mapping or associative memory. Proc. IEEE Int. Conf. on Neural Networks I, 525–532 (1988)
11. Tax, D.M.J.: One-class classification. PhD thesis, Delft University of Technology, The Netherlands (2001)
12. Tax, D.M.J., Duin, R.P.W.: Combining one-class classifiers. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 299–308. Springer, Heidelberg (2001)