Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

PERGAMON

Pattern Recognition 37 (2004) 409–419

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A semiparametric density estimation approach to pattern classification

Fabian Hoti[1], Lasse Holmström*

Rolf Nevanlinna Institute, P.O. Box 4, 00014 University of Helsinki, Finland

Received 13 February 2003; received in revised form 30 June 2003

Abstract

A new multivariate density estimator suitable for pattern classifier design is proposed. The data are first transformed so that the pattern vector components with the most non-Gaussian structure are separated from the Gaussian components. Nonparametric density estimation is then used to capture the non-Gaussian structure of the data while parametric Gaussian conditional density estimation is applied to the rest of the components. Both simulated and real data sets are used to demonstrate the potential usefulness of the proposed approach.

Keywords: Semiparametric density estimation; Kernel estimation; Classification; Handwritten digit data; Satellite data; Microarray data

1. Introduction

Probability density estimation is a classical approach to pattern classifier design. The pattern class densities are estimated from the training data and combined with the estimated or known prior class probabilities to construct discrimination rules that perform optimally, provided that the density estimates are accurate [1]. While it is true that optimal density estimates are not necessary for good discrimination rules and that it is in fact sometimes better to forego the density estimation step altogether and, for example, to estimate the class posterior probabilities directly using regression, good class density estimates certainly will produce low-error classifiers. The purpose of this paper is to propose a new density estimation procedure that is particularly suitable for classifier design.

The two principal approaches to probability density estimation are the *parametric* and the *nonparametric* paradigms. In parametric estimation one assumes that the density f underlying the data $X_1,\ldots,X_n$ belongs to some rather restricted family of functions $f(\cdot;\theta)$ indexed by a small number of parameters $\theta=(\theta_1,\ldots,\theta_k)$. An example is the family of multivariate normal densities which is parameterized by the mean vector and the covariance matrix. A density estimate in the parametric approach is obtained by computing from the data an estimate $\hat{\theta}$ of θ and setting $\hat{f}=f(\cdot;\hat{\theta})$. Such an approach is statistically and computationally very efficient but can lead to poor results if none of the family members $f(\cdot;\theta)$ is close to f.

In nonparametric density estimation (e.g. Ref. [2]) no parametric assumptions about f are made and one assumes instead that f, for example, has some smoothness properties (e.g. two continuous derivatives) or that it is square integrable. The shape of the density estimate is determined by the data and, in principle, given enough data, arbitrary densities f can be estimated accurately. The most popular method is the kernel or Parzen estimator based on local smoothing of the data (the definition is given in Section 2.1). Unfortunately, in practice such nonparametric methods often suffer from the curse of dimensionality: in high dimensions huge data sets may be needed for good estimation

* Corresponding author. Tel.: +358-505637465; fax: +358-9-191-22779.
E-mail address: llh@rolf.helsinki.fi (L. Holmström).
[1] Partially supported by the Finnish Graduate School in Stochastics.

0031-3203/$30.00
doi:10.1016/j.patcog.2003.08.004

results. In the case of the kernel estimator, sparseness of the high-dimensional data requires heavy smoothing to keep the variance in check and this results in a large bias. Also, nonparametric density estimators are typically computationally expensive making them possibly useless in on-line pattern recognition applications.

The idea proposed here is to use nonparametric density estimation only in the data subspace where it matters most and to use simple parametric density estimation to capture the residual structure in the remaining dimensions. To facilitate the implementation of this idea, a transformation is first applied to the original data in order to make its possibly complex nonparametric structure concentrate in the first components of the transformed data vector and with the remaining components exhibiting less complex, Gaussian-like features (at least conditionally). The full density is then modeled as the product of nonparametric and (conditionally) parametric factors. Cross-validation can be used to select the point along the data vector at which this nonparametric–parametric split occurs thus letting the data decide how much nonparametric modeling is feasible for the current problem, given the available data set. The proposed method is intermediate between fully nonparametric and fully parametric approaches and can therefore be characterized as *semi-parametric*.

The proposed approach is related to projection pursuit density estimation [3] in the sense that both methods use linear subspace based feature extraction to effectively combine nonparametric and parametric estimation. In projection pursuit one iteratively updates a multidimensional density estimate by multiplying by a function of a carefully selected one-dimensional feature.

Some similarities can also been found with Ref. [4] where first a nonparametric and a parametric estimate are calculated based on all dimensions and then the final estimate is taken as a linear combination of the two. The authors in Ref. [5] propose updating a parametric estimate iteratively by nonparametric correction functions. Here also all functions are defined using all dimensions.

In the pattern recognition literature, our method resembles the product rule for combining classifiers based on independent features [6,7]. However, in our work no independence assumption is required.

2. The proposed method

2.1. Density estimation

Let d and s be positive integers, $d \geqslant 2$, $1 \leqslant s \leqslant d-1$, and split the d-dimensional vector $x \in \mathbb{R}^d$ into s and $(d-s)$-dimensional parts as $x=(y,z)$, where $y \in \mathbb{R}^s$, $z \in \mathbb{R}^{d-s}$. A d-dimensional continuous random vector X can correspondingly be split into s and $(d-s)$-dimensional parts, $X=(Y,Z)$, and the density functions then satisfy

$$f_X(x) = f_{(Y,Z)}(y,z) = f_Y(y) f_{Z|Y=y}(z),$$
$$x=(y,z) \in \mathbb{R}^d. \tag{1}$$

Here f_X and f_Y are the densities of X and Y, respectively, and $f_{Z|Y=y}$ is the density of Z, conditional on $Y=y$. Assume that the conditional densities $f_{Z|Y=y}$ are Gaussian, that is, multivariate normal, but that the density f_Y does not belong to any simple parametric family. Then a natural estimator of f_X is obtained by estimating f_Y nonparametrically and fitting a multivariate normal density for each $f_{Z|Y=y}$. Thus, given a sample $X_i=(Y_i,Z_i)$, $i=1,\ldots,n$, of independent and identically distributed random vectors with density f_X, one may use for example kernel estimation [8–10] and set

$$\hat{f}_Y(y) = \frac{1}{n}\sum_{i=1}^{n} K_{h_1}(y-Y_i), \qquad y \in \mathbb{R}^s, \tag{2}$$

where K is a symmetric nonnegative kernel function that integrates to 1 (i.e. a symmetric probability density), $K_{h_1}(y)=h_1^{-s}K(y/h_1)$, and $h_1>0$ is the smoothing parameter. In this paper, the Gaussian kernel (the standard multivariate normal density) was used in all kernel estimates.

Conditional on $Y=y$, the vector Z is assumed to have a multivariate normal distribution, $Z|Y=y \sim N(m(y),C(y))$, where the conditional mean and the conditional covariance matrix are defined by

$$m(y) = \mathbb{E}(Z|Y=y), \qquad y \in \mathbb{R}^s, \tag{3}$$

$$C(y) = \mathbb{E}[(Z-m(y))^{\mathrm{T}}(Z-m(y))|Y=y], \qquad y \in \mathbb{R}^s, \tag{4}$$

and where T denotes the transpose operation. We propose to use kernel smoothing for the estimation of $m(y)$ and $C(y)$, too. Thus, set

$$\hat{m}(y) = \frac{\sum_{i=1}^{n} K_{h_2}(y-Y_i)Z_i}{\sum_{j=1}^{n} K_{h_2}(y-Y_j)} = \sum_{i=1}^{n} W_{h_2}(y-Y_i)Z_i, \qquad y \in \mathbb{R}^s, \tag{5}$$

where the weights

$$W_{h_2}(y-Y_i) = \frac{K_{h_2}(y-Y_i)}{\sum_{j=1}^{n} K_{h_2}(y-Y_j)} \tag{6}$$

sum to 1. Formula (5) can be recognized as a multivariate Nadaraya–Watson regression estimator [11,12] of the conditional mean function m. Similarly, the conditional covariance can be estimated as

$$\hat{C}(y) = \sum_{i=1}^{n} W_{h_3}(y-Y_i)(Z_i-\hat{m}(y))^{\mathrm{T}}(Z_i-\hat{m}(y)),$$
$$y \in \mathbb{R}^s. \tag{7}$$

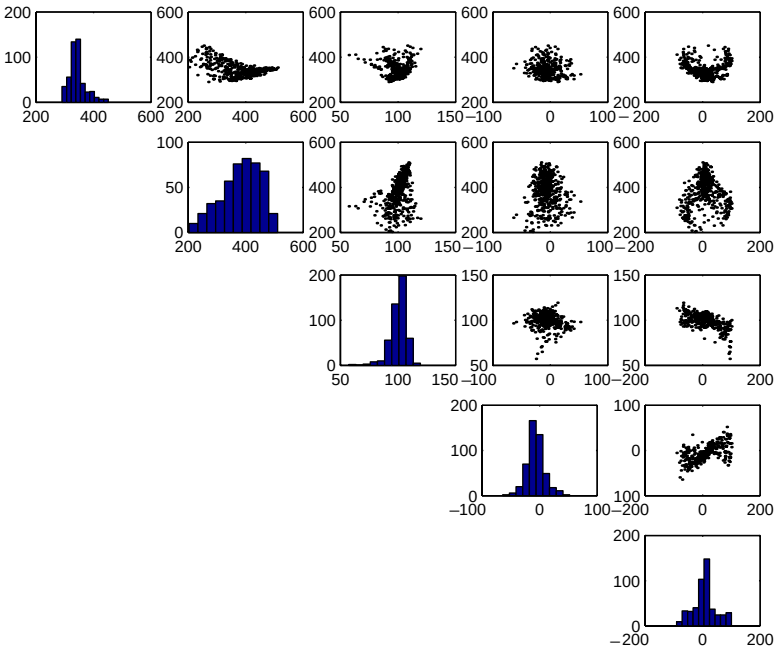

Fig. 1. Satellite data: scatter plots of the first five features in class 2 after a PCA transformation based on all data. For description of the data, see Section 3.2.2.

A parametric estimator for the conditional density $f_{Z|Y=y}$ is now given by

$$\hat{f}_{Z|Y=y}(z) = [(2\pi)^{d-s} \det \hat{C}(y)]^{-1/2} \mathrm{e}^{-(1/2)(z-\hat{m}(y))\hat{C}(y)^{-1}(z-\hat{m}(y))^{\mathrm{T}}}, \quad z \in \mathbb{R}^{d-s}. \tag{8}$$

The estimator for the whole density f_X is then

$$\hat{f}_X(x) = \hat{f}_{(Y,Z)}(y,z) = \hat{f}_Y(y)\hat{f}_{Z|Y=y}(z), \quad x = (y,z) \in \mathbb{R}^d. \tag{9}$$

We call the above procedure *semiparametric kernel density estimation*, sKDE for short. In practice, even if the conditional normality assumption of some of the data vector components is reasonable, for good estimation results one must still select the splitting dimension s and smoothing parameters h_1, h_2 and h_3 appropriately. One possibility is to use least squares or likelihood based cross-validation (e.g. Ref. [13]). However, instead of considering general density estimation, our main interest is in classification and in that context training error cross-validation is a natural parameter optimization criterion (Section 2.3).

2.2. Transforming the data

The crucial assumption in the proposed density estimation method is that the random vector underlying the data can be split into non-Gaussian and conditionally Gaussian parts. Obviously, for arbitrary raw data one can hardly expect that such a split is readily available. However, a suitable transformation of the data might make this assumption more reasonable. Note that for example the Karhunen–Loéve transformation, or principal components analysis (PCA), produces d variables that are often arranged in an order of decreasing complexity of distribution structure. The first principal components capture most of the variation in data and this may be associated with interesting non-Gaussian structure (Fig. 1). The least important components come last and can exhibit simple Gaussian type distribution (Fig. 2).

Thus, let $T : \mathbb{R}^d \to \mathbb{R}^d$ be a transformation such that for some $1 \leqslant s \leqslant d-1$ the transformed random variable $U = T(X)$ can be split into s and $(d-s)$-dimensional parts, $U = (V, W)$, in such a way that, conditional on $V = v$, W is (at least approximately) multivariate normal. One can then use the transformed data $U_i = T(X_i)$, $i = 1, \ldots, n$, to construct a kernel estimator $\hat{f}_V$ and a multivariate normal density estimator $\hat{f}_{W|V=v}$ as in Eqs. (2) and (8), and then define

$$\hat{f}_U(u) = \hat{f}_V(v)\hat{f}_{W|V=v}(w). \tag{10}$$

Provided that the transformation T is sufficiently regular, one can then define an estimator for the original density f_X by

$$\hat{f}_X(x) = |J_T(x)|\hat{f}_U(T(x)), \tag{11}$$

where $J_T(x)$ is the Jacobian of T.

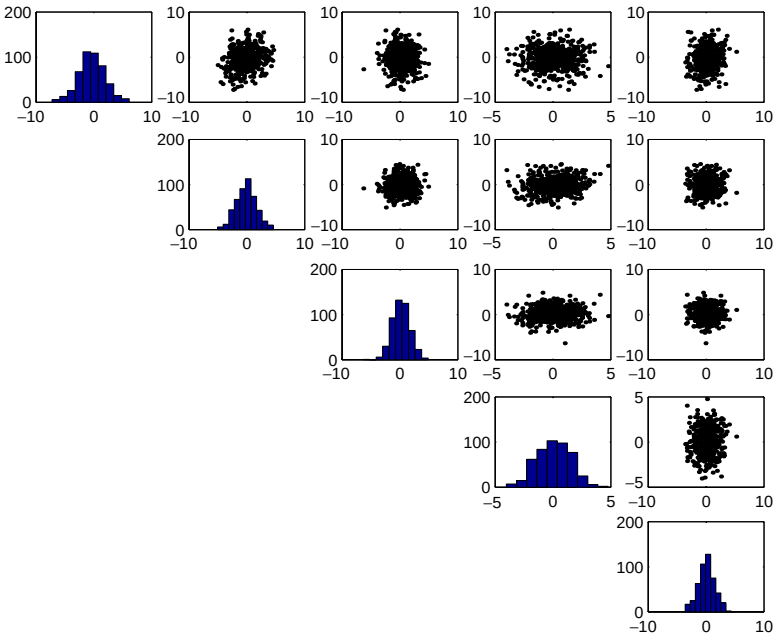

Fig. 2. Satellite data: scatter plots of the last five features in class 2 after a PCA transformation based on all data. For description of the data, see Section 3.2.2.

Although the PCA transformation appears to work well in many cases, it depends only on the second-order statistics of the data and may therefore fail to capture the most interesting directions. For example, if the data come from such a mixture of two Gaussian distributions as in Fig. 3, then the first PCA component has a Gaussian distribution and the second PCA component, conditional on the first one, is highly non-Gaussian. However, if we consider independent component analysis (ICA) [14,15], a method specially geared towards uncovering non-Gaussian structure, then we get the desired transformation, as the first ICA component captures the non-Gaussian structure while the conditional distributions of the second component are approximately normal (Fig. 3). We therefore tried both PCA and ICA in our experiments but, despite the potential shortcomings of PCA, use of ICA did not appear to offer any advantage in practice.

2.3. Classification

Based on the above density estimation method, a pattern classifier can be designed in a straightforward fashion. Let X originate from a mixture of c pattern classes denoted by the integers $1,\ldots,c$. Denote by J the class label, a discrete random variable with values in $\{1,\ldots,c\}$, so that a labeled pattern is represented by the pair (X,J). Let $P^{(j)}=\mathbb{P}(J=j)$ be the prior probability of the class $j\in\{1,\ldots,c\}$, and denote by $f_X^{(j)}$ the density of the class j, that is, $X|J=j\sim f_X^{(j)}$. The optimal Bayes classifier would then assign X to the class j that maximizes the product $P^{(j)}f_X^{(j)}(X)$. We assume that the prior probabilities are known or estimated from the training data by some estimates $\hat{P}^{(j)}$ and propose to construct a classifier as follows.

(1) Using exploratory data analysis of the training data, examine the distributions of the pattern classes.
(2) If deemed necessary, for each class j use the training data to find a suitable transformation $T^{(j)}$ so as to make $T^{(j)}(X)|J=j$ exhibit non-Gaussian structure mostly only in its first components.
(3) Using the density estimation method of Section 2.1 and the transformation formula (11), find estimates $\hat{f}_X^{(j)}$ for the class conditional densities. The splitting dimensions and the required smoothing parameters for the estimates $\hat{f}_X^{(j)}$ are optimized by minimizing the cross-validated training set error of the approximate Bayes classifier that maximizes $\hat{P}^{(j)}\hat{f}_X^{(j)}(X)$.
(4) The final discrimination rule is the approximate Bayes decision that classifies a pattern X to the class j that maximizes the product $\hat{P}^{(j)}\hat{f}_X^{(j)}(X)$, where $\hat{f}_X^{(j)}$ uses the optimal parameters.

We call the above classification rule semiparametric kernel discriminant analysis (sKDA).

In practice, both the size of the training set and the available computing power may require that some of the above steps are simplified. For example, there might not be enough data to estimate separate transformations for each class and

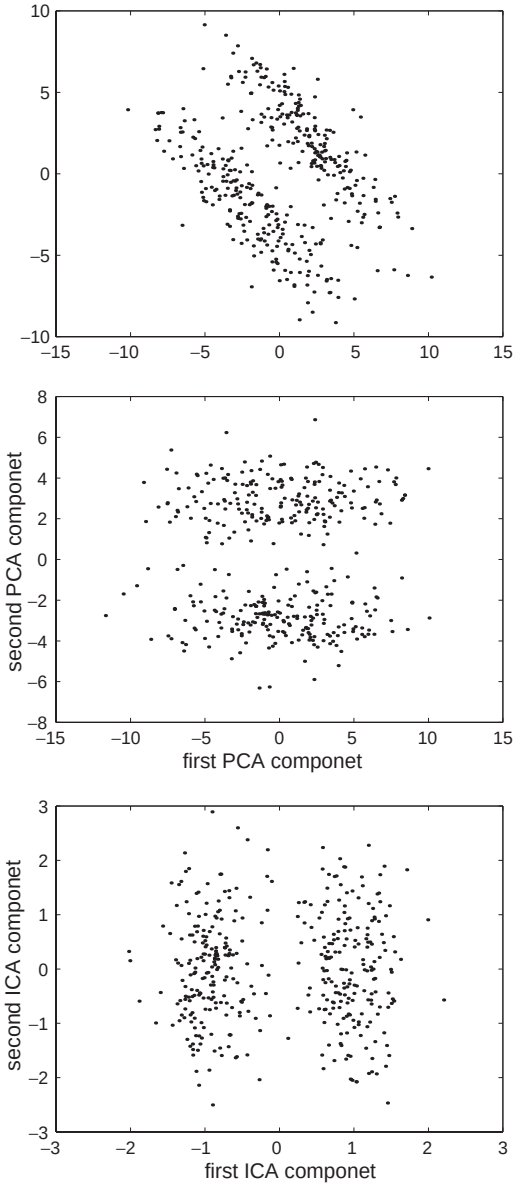

Fig. 3. A comparison of two different transformations. Starting from the top: a sample from a mixture of two normal distributions in the original space, the data after PCA transformation and the data after ICA transformation.

a single transformation must be used instead. A natural computational simplification in Eq. (7) would be to take $h_3 = \alpha h_2$, for some $\alpha > 0$. If insufficient amount of data requires even more simplification, one can try to use an unconditional covariance estimate $\hat{C} = \hat{C}(y)$ in Eq. (8) obtained by taking $W_{h_3}(y - Y_i) = 1/n$, $\hat{m}(y) = (1/n)\sum_{i=1}^{n} Z_i$ in Eq. (7). In fact, this was the approach taken in all the examples presented in Section 3 because using separate h_2 and h_3 was computationally too heavy and taking h_3 proportional to h_2 did not seem to improve on results obtained using an unconditional covariance matrix. The fact that an unconditional covariance matrix worked well can be partly explained by the decorrelating effect of PCA used to transform the data. Note also that having separate splitting dimensions (s) and smoothing parameters (h_1, h_2, h_3) for each class rapidly makes the training set error cross-validation computationally infeasible when the number of classes increases. Also, with fine optimization search grids and large data sets, even a two-class problem may be computationally heavy. It may therefore sometimes be necessary to use for example a single splitting dimension or the same smoothing parameters for all classes.

Throughout this section we assumed that $1 \leqslant s \leqslant d - 1$ so that the kernel estimator (2) and the multivariate normal estimator (8) both really are present in Eq. (9). In the following examples we will also allow the values $s = 0$ and $s = d$ for the splitting dimension so that both a pure multivariate normal estimator and a pure kernel density estimator (for transformed data) are included in our model. In particular, the proposed classifier design algorithm can therefore produce pure kernel discriminant analysis (KDA) and pure quadratic discriminant analysis (QDA) as special cases.

3. Examples

3.1. Simulations

3.1.1. Density estimation

First, we considered probability density estimation using randomly generated example densities f_X. Each f_X is a random mixture of three multivariate normal densities with randomly picked means and random covariance matrices of the form $\sigma^2 I_d$, where $\sigma^2 > 0$ and I_d is the $d \times d$ unit matrix. The three means were drawn from the standard normal distribution and multiplied componentwise by the scaling factors $\sqrt{(d-k)/(d-1)}$, $k = 1, \ldots, d$, so that the kth component of each mean approaches 0 as k increases from 1 to d. The variance σ^2 was drawn from the uniform distribution on the interval [0.1, 1.9] and mixing proportions of the three components were selected randomly by using two uniformly generated random numbers between 0 and 1 to divide the interval [0, 1] into three parts. Note that, by design, the density f_X is expected to have non-Gaussian structure in the first dimensions whereas the last dimensions tend to be unimodal and Gaussian-like. Therefore, no data transformation was used in density estimation.

We compared the performance of the estimator defined in Section 2.1 to both standard kernel density estimation and multivariate normal maximum likelihood fitting that uses the sample mean and the sample covariance matrix as parameter estimates. Three cases were considered: $d = 5$ with sample sizes $n = 50$ and $n = 500$, and $d = 10$ with sample size $n = 500$. For each case, 100 random densities f_X were generated. Given a density estimate $\hat{f}_X$ based on a sample $X_1, \ldots, X_n$,

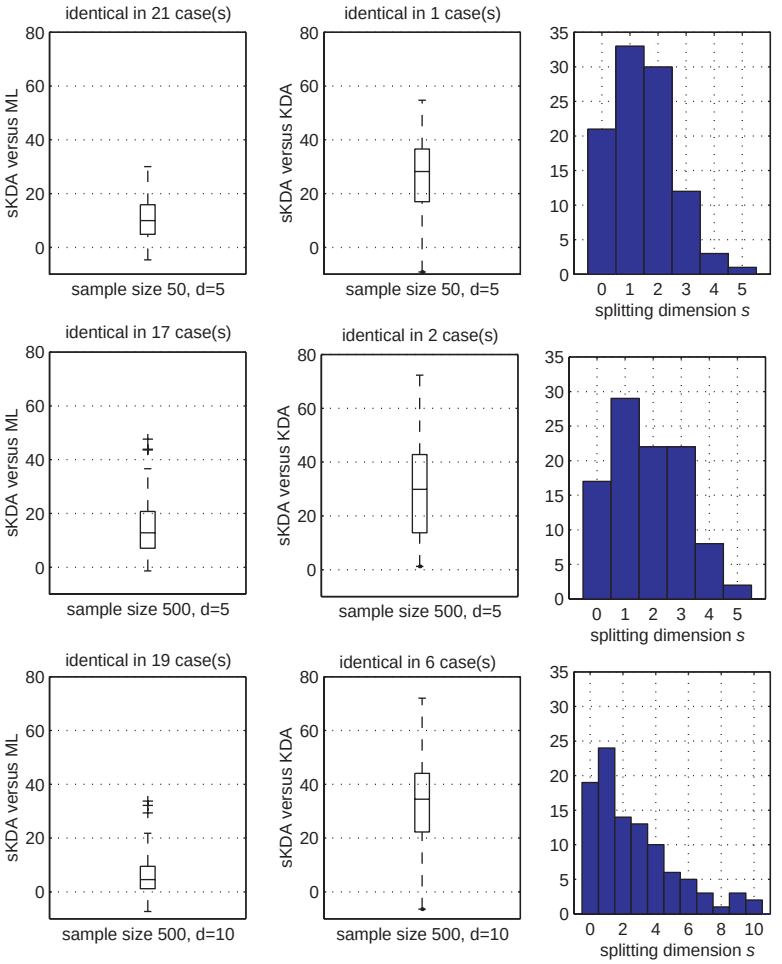

Fig. 4. The first two columns show box and whisker plots of the relative improvement (%) achieved by sKDE over ML and KDE, respectively. The number of cases (out of 100) where sKDE reduced to either KDE or ML is given on top of each plot and these cases were not included in the box plots. Different rows correspond to a different sample size and dimension indicated under each plot. The whiskers are defined using the inter-quartile range multiplied by 1.5. A dot on the lower whisker implies that all data points lie within the whiskers. In each row the rightmost column gives the distribution of the selected splitting dimension.

a natural measure of estimation performance is the L^1-error

$$L(\hat{f}_X) = \int_{\mathbb{R}^d} |f_X - \hat{f}_X| \tag{12}$$

(cf. [16]). Note that since

$$\int_{\mathbb{R}^d} |f_X - \hat{f}_X| = \int_{\mathbb{R}^d} |1 - \hat{f}_X/f_X| f_X$$
$$= \mathbb{E}[|1 - \hat{f}_X(X)/f_X(X)|], \tag{13}$$

a Monte-Carlo estimator of $L(\hat{f}_X)$ is obtained by generating a large auxiliary sample $\tilde{X}_1, \ldots, \tilde{X}_m$ from f_X and setting

$$\hat{L}(\hat{f}_X) = \frac{1}{m} \sum_{j=1}^{m} |1 - \hat{f}_X(\tilde{X}_j)/f_X(\tilde{X}_j)|. \tag{14}$$

We selected the smoothing parameters and the splitting dimension by minimizing $\hat{L}(\hat{f}_X)$ over a discrete search grid using an auxiliary sample of size $m = 5000$ in Eq. (14). The multivariate normal density was fitted using the original sample only. After preliminary exploration of the dependence of $\hat{L}(\hat{f}_X)$ on the tuning parameters, a suitable search grid for the smoothing parameters of the kernel density estimators (the standard kernel estimator used in the comparisons and the nonparametric part of the semiparametric method) was found to be $0.2, 0.4, \ldots, 1.2$. For conditional mean estimation the smoothing parameter grid was $0.4, 0.8, \ldots, 2.4$, and for the splitting dimension all values between 0 and d were allowed.

The final performance of each estimator was evaluated by computing $\hat{L}(\hat{f}_X)$ using still another sample of size $m = 5000$ in Eq. (14) and the results are shown in Fig. 4. The box plots show the relative percentage improvement of sKDE over multivariate normal based maximum likelihood estimation

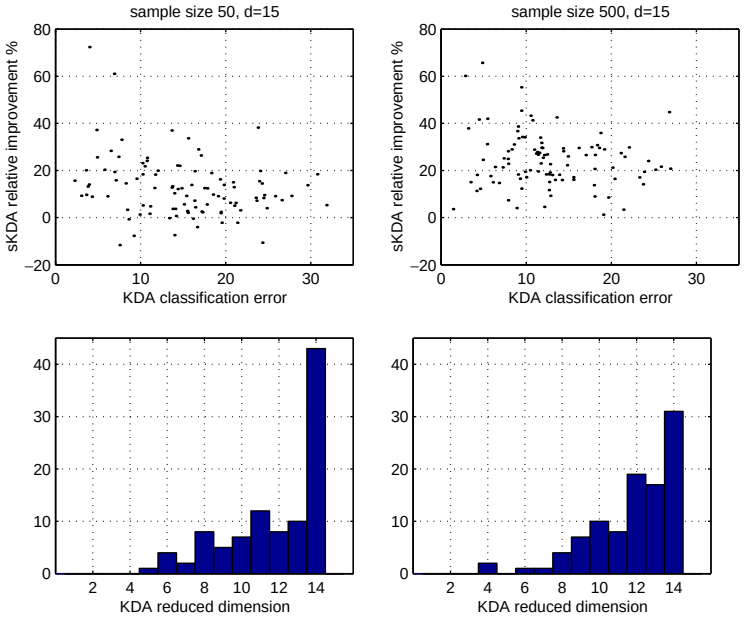

Fig. 5. Comparison of classification performance of KDA with dimension reduction and sKDA. The columns correspond to different sample sizes. Top row: scatterplots of the KDA classification error using dimension reduction and the relative improvement achieved when sKDA is used. Bottom row: the distribution of the reduced dimension in KDA.

(ML) and standard kernel density estimation (KDE). The relative improvement obtained with sKDE is significant. The fact that the improvement over KDE is large shows that the sample sizes used are insufficient for efficient kernel estimation in the 5- and 10-dimensional spaces and that it helps a lot to model some of the dimensions parametrically. The distributions of the selected splitting dimensions are shown in the rightmost column.

3.1.2. Classification

Next we considered a simulated two-class discrimination task. The prior probabilities of the two classes were taken equal and each of the class-conditional densities were generated as in the density estimation example above, with two minor modifications. First, for a d-dimensional problem, $(d+5)$-dimensional densities were produced and the last five-dimensions were dropped from the generated data. This was done because the last components of the class mean vectors are almost zero and therefore contain little discrimination power. Second, to increase class overlap we translated the two class densities towards the origin by subtracting one third of the class mean vectors.

We compared the performance of the proposed sKDA method (Section 2.3) with kernel discriminant analysis (KDA). To make the comparison fairer for KDA, dimension reduction was allowed in its design. This was done by keeping only the first t components of the data vector where $1 \leqslant t \leqslant d$ was selected to optimize classification performance. Two cases were considered: $d = 15$ with $n = 50$ training data from each class and $d = 15$ with $n = 500$ training data from each class. In both cases 100 random classification tasks were generated. Auxiliary samples of size 5000 from each class were used to choose the model parameters by minimizing classification error of the approximate Bayes classifier. Yet other samples of size 5000 from each class were used to evaluate the final classification performance.

The results (Fig. 5) show that the simulated classification tasks have very different levels of difficulty as the classification error of KDA varies from almost zero to over 30%. However, sKDA almost always achieves significant improvement. Fig. 6 shows two-dimensional histograms of the splitting dimensions of the two classes. As one can see, at least in this example there does not seem to be much correlation and it is therefore important to select the dimensions independently. As pointed out earlier, for computational reasons we sometimes must resort to using a common splitting dimension for all classes. Note also that, perhaps a little surprisingly, for $n = 500$ dimension reduction favors keeping almost all components (Fig. 5) while the splitting dimension favors small values (Fig. 6). This suggests that although the last data vector components hold information that is helpful in classification and should therefore be included, they are more effectively used when modeled parametrically.

3.2. Real data sets

When considering the use of sKDA for real data one should bear in mind the basic premises for its success. Thus,

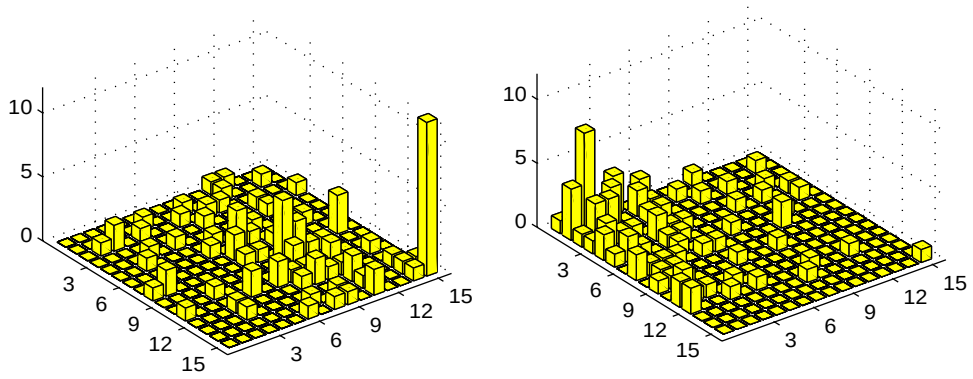

Fig. 6. Histograms of the class density splitting dimensions in the case $n = 50$ and $n = 500$, respectively.

the data should have both Gaussian and non-Gaussian dimensions (at least after a suitable transformation) and the sample size should be large enough to allow for nonparametric density estimation. Except for the data sets in Section 3.2.3, we selected our examples with these preconditions in mind. We did in fact experiment also with data sets which do not meet these requirements with predictable results. The phoneme data set from the Elena project [17] consists of 5404 five-dimensional measurements from two classes with highly non-Gaussian structure and, as expected, the best results for these data were achieved with standard KDA. The Wisconsin breast cancer data set [18] (obtained from Ref. [19]) is an example of a data set where presence of both Gaussian and non-Gaussian structure is suggested but due to the small sample sizes the best results were achieved by combining QDA with dimension reduction.

3.2.1. Handwritten digit data

The handwritten digit data set originates from a comparison study of neural network and statistical classifiers [20]. A total number of 894 forms were filled out by randomly chosen Finnish people. Each form contained two handwritten examples of each of the 10 digits. After several pre-processing steps (see Ref. [20]) each handwritten digit was coded as a 1024-dimensional binary vector. The whole sample of 17 880 binary vectors was divided into two sets of equal size, one for training and one for testing the classifiers. The pattern vectors were transformed into 64-dimensional real feature vectors by a PCA transformation that was estimated using only the training data.

Due to the high number of classes, the same splitting dimensions were used in all classes. Experiments indicated that better results could be achieved by performing an additional dimension reduction on the 64-dimensional feature vector before sKDA was applied. The splitting dimension s was chosen from the grid $2, 4, 6, 8, 10, 15, 20, \ldots, d' - 10$ where d' is the reduced dimension chosen from the grid $20, 30, \ldots, 60$. The smoothing parameter in the kernel estimate (2) and that of the conditional mean estimate (5) were chosen from the grid $1, 1.5, \ldots, 5$.

The best parameter combination was selected from the resulting four-dimensional grid by 10-fold cross-validation. Thus, the training data were first divided randomly into 10 parts of equal size. Then one part at a time was left out, the remaining nine parts were used to train a classifier and the classification error for the part left out was computed. The performance of the parameter combination was then measured by the average classification error over the 10 cases. The best parameters together with all the training data were then used to set up the classifier and the final performance was estimated using the separate testing data. The optimal parameter values were $d' = 50, s = 15, h_1 = 1.5$, and $h_2 = 4$. The classification error for the separate testing data was 2.6%, lower than for all the 16 individual classifiers tried in Ref. [20]. The classification error of both KDA and QDA was 3.7%. Using a so-called convex local subspace classifier, a smaller error rate of 2.1% was reported in Ref. [21].

3.2.2. Public data sets 1: satellite image and handwritten digits

Next we consider two public data sets obtained from the UCI Machine Learning Repository [19]. The first example is a satellite image data set with 4435 training vectors and 2000 testing vectors. The feature vectors are 36-dimensional and consist of digital images (3×3 pixels) of the same scene in four different spectral bands. The feature values vary from 0 (white) to 255 (black) and the samples originate from six different classes (soil types).

The second example is a handwritten digit data set that consists of 3823 training vectors and 1797 testing vectors of the 10 digits. The training and testing sets were produced by 30 and 13 different individuals, respectively. Each digit was coded as a 64-dimensional feature vector obtained by dividing a 32×32 bitmap into 64 nonoverlapping 4×4 bitmap blocks and counting in each block the number of "on" pixels.

Table 1
Summary of results from two case studies using sKDA1 which is based on a common PCA transformation and 10-fold cross-validation

Data	Dimensional params.	Smoothing params.	Error(%)	cv params.
Satellite	$d' = 6, 12, \ldots, 36$	$h_1 = 0.1, 0.2, \ldots, 0.7$	8.35	$d' = 18, s = 9$
	$s = 0, 3, \ldots, d'$	$h_2 = 0.2, 0.4, \ldots, 1.4$		$h_1 = 0.3, h_2 = 0.6$
Digits	$d' = 10, 20, \ldots, 60$	$h_1 = 0.3, 0.6, \ldots, 1.5$	3.06	$d' = 40, s = 25$
	$s = 0, 5, \ldots, d'$	$h_2 = 0.6, 1.2, \ldots, 3$		$h_1 = 0.6, h_2 = 1.2$

Table 2
Summary of results from two case studies using sKDA2 which is based on class-wise PCA transformations and five-fold cross-validation

Data	Dimensional params.	Smoothing params.	Error(%)	cv params.
Satellite	$d' = 6, 12, \ldots, 36$	$h_1 = 0.1, 0.2, \ldots, 0.7$	9.5	$d' = 18, s = 10$
	$s = 0, 3, \ldots, d'$	$h_2 = 0.2, 0.4, \ldots, 1.4$		$h_1 = 0.4, h_2 = 0.6$
Digits	$d' = 10, 20, \ldots, 60$	$h_1 = 0.3, 0.6, \ldots, 1.5$	4.12	$d' = 40, s = 35$
	$s = 0, 5, \ldots, d'$	$h_2 = 0.6, 1.2, \ldots, 3$		$h_1 = 0.6, h_2 = 3$

The data were normalized by dividing each feature with the sample standard deviation estimated from the training data, unless the feature had a constant value in which case it was not changed. We applied two versions of our method to the satellite and digit data sets. In the first version, sKDA1, a single common PCA transformation computed from the training data and dimension reduction was used for all classes. In the second version, sKDA2, first the common PCA transformation and dimension reduction was applied and then further class-wise PCA transformations were performed on the lower dimensional data. Also, the computational task for both methods was reduced by using the same splitting dimension (s) and the same smoothing parameters (h_1, h_2) in all class density estimates.

The results are given in Table 1 for sKDA1 and in Table 2 for sKDA2. For comparison, we calculated the classification error for both sets using KDA and QDA. A PCA common to all classes and dimension reduction was used for both classifiers and, in KDA, the same smoothing parameter was used in all class density estimates. The reduced dimension was chosen from the grid $1, 2, \ldots, d$ and the smoothing parameter in KDA was chosen from the same grid as h_1 in Table 1. For both classifiers, 10-fold cross-validation was used. The test errors with the satellite data were 9.05% for KDA and 14.4% for QDA. For digit data the errors were 3.39% (KDA) and 4.40% (QDA), respectively. For both data sets, sKDA1 outperforms these two classifiers. The satellite data were also used in the EC (ESPRIT) StatLog project [22] that compared a large selection of different classifiers and the error rate of sKDA1 is 1% lower than that of the best classifier in that study ($k-$NN with 9.4% error). The digit data were analyzed in Ref. [23] and although our results are not quite comparable due to the different training set used, the performance of sKDA1 is close to the best classifiers reported there.

Although the results for sKDA2 are not as good they clearly indicate that the method based on class-wise PCA transformations also works. However, probably there is not quite enough of data to allow the use of this more complicated approach effectively.

3.2.3. Public data sets 2: microarray data sets

As our final example we consider microarray data. Classification of microarray expression level patterns of thousands of genes is an important problem in bioinformatics. The properties of expression level data are quite different from the examples considered so far as the number n of available patterns is usually less than 100 while their dimension d can easily be several thousands. To investigate the performance of sKDA in gene expression data classification we considered two high-density oligonucleotide cancer data sets produced using Affymetrix microarrays (both data sets with supplementary information can be obtained from www-genome.wi.mit.edu/cancer). The first example [24] is a leukemia data set where the microarray profiles are divided into two groups, 47 cases of acute lymphoblastic leukemia (ALL) and 25 cases of acute myeloid leukemia (AML). The second example [25] consists of profiles of two types of lymphomas, 58 cases of Diffuse large B-cell lymphoma (BLBCL) and 19 cases of follicular lymphoma (FL). All profiles include 7129 expression measurements.

Both data sets (obtained by Affymetrix software) have undergone a linear scaling to correct for minor differences in microarray intensity (see the supplementary information of Refs. [24,25]). We followed the preprocessing steps used in Refs. [24,25]. Thus, giving in parentheses the value for the leukemia data set if it differs from that used with the lymphoma data, gene-expression values smaller than 20 (100) were replaced with 20 (100) and values higher than 16000 were set to 16000. Further, if x_{ij} is the expression

Table 3
Mean classification errors for the two microarray data sets calculated over 200 random partitions. In parentheses we give the sample standard deviation of the mean value.

Data	ML (%)	KDE (%)	sKDE (%)
Leukemia	24.86 (0.56)	25.65 (0.55)	26.10 (0.55)
Lymphoma	11.46 (0.51)	10.13 (0.49)	11.54 (0.52)

level of gene j in the profile i, and if $x_j = (x_{1j}, \ldots, x_{nj})$, then gene j was excluded from further analysis if $\max(x_j)/\min(x_j) < 3(5)$ or $\max(x_j) - \min(x_j) < 100(500)$. Then a base 10 logarithmic transformation was applied to the data.

The final features were formed as follows (cf. [24]). For each gene the statistic

$$S(x_j) = \frac{\hat{\mu}_j^{(1)} - \hat{\mu}_j^{(2)}}{\hat{\sigma}_j^{(1)} + \hat{\sigma}_j^{(2)}} \tag{15}$$

was calculated, where $\hat{\mu}_j^{(k)}$ is the sample mean and $\hat{\sigma}_j^{(k)}$ is the sample standard deviation of the expression values of gene j in class k. Then 30 genes were selected by including 15 genes with the largest S value and 15 genes with the smallest S value.

The comparison of classifiers was performed by dividing the data sets randomly into two parts, 2/3 was reserved for training and 1/3 for testing. The common PCA transformation was calculated using only the training part and the first 10 PCA features were used in the final estimators. A summary of the classification results achieved by 200 independent train/test partitions is given in Table 3. The median value of the reduced dimension for both KDA and QDA was 3. For sKDA the median of the reduced dimension was 5 and the median of the selected split was 3. All parameters were chosen by leave-one-out cross-validation, the extreme version of k-fold cross-validation where one profile at a time is classified using a classifier designed with the rest of the training data. The reduced dimension d' was chosen from the grid $1, \ldots, 10$, the splitting dimension s from the grid $1, \ldots, d'$, the smoothing parameter in KDA and the parameter h_1 in sKDA from the grid $0.1, 0.2, \ldots, 1.0$, and the second smoothing parameter h_2 in sKDA from the grid $0.2, 0.4, \ldots, 2.0$.

Taking into account the standard deviations of the mean classification results in Table 3 we conclude that, for both data sets, all three classifiers perform similarly. We note that the results reported in Ref. [26] are based partly on the same data sets and they in fact indicate that even simple linear and quadratic classifiers with diagonal covariance matrices can work very well with these data. This suggests that, although sKDA is still able to match the performance of its two classical limits, more training data would be needed to make its more complex structure truly useful for these data sets.

4. Summary

Nonparametric density estimation is an effective general purpose tool for flexible classifier design. However, nonparametric methods typically suffer from the curse of dimensionality and with a limited training set size, efficient feature extraction must be used. A common approach is to base the density estimates on a carefully selected subset of the original variables or, alternatively, on the first few principal components of the data. Using kernel density estimation as the nonparametric density estimation method we have demonstrated that, instead of such a dimension reduction approach, the curse of dimensionality can sometimes be more successfully fought by using conditional parametric estimation in the dimensions that do not require all the flexibility of nonparametric estimation. In a sense, our approach is a mixture of kernel discriminant analysis and quadratic discriminant analysis and experiments with simulated and real data sets show that it has potential to improve on both offering a viable tool in many classification problems. We call the method sKDA for semiparametric kernel discriminant analysis.

Before the density estimates can be effectively split into nonparametric and parametric parts, the data are transformed to concentrate the more complex features in the first dimensions with Gaussianity increasing towards the end of the transformed data vector. In our experience, standard PCA can often accomplish this. Other approaches such as ICA can also be tried.

In their most general form, the density estimates in sKDA use several smoothing parameters that need to be tuned for optimal classification results. In practice, computational complexity or sparseness of the training data in a high dimensional space may require that the density estimation model is simplified. This can be done by reducing the number of independent smoothing parameters or, e.g., by giving up smoothing altogether in the estimation of the conditional covariance.

References

[1] L. Devroye, L. Györfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer, Berlin, 1996.
[2] D. Scott, Multivariate Density Estimation, Wiley, New York, 1992.
[3] J. Friedman, W. Stuetzle, A. Schroeder, Projection pursuit density estimation, J. Am. Stat. Assoc. 79 (1984) 599–608.
[4] I. Olkin, C. Spiegelman, A semiparametric approach to density estimation, J. Am. Stat. Assoc. 82 (1987) 858–865.
[5] N. Hjort, I. Glad, Nonparametric density estimation with a parametric start, Ann. Stat. 23 (1995) 882–904.
[6] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[7] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (2000) 226–239.

[8] M. Rosenblatt, Remarks on some nonparametric estimates of a density fuction, Ann. Math. Stat. 27 (1956) 832–835.
[9] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (1962) 1065–1076.
[10] T. Cacoullos, Estimation of a multivariate density, Ann. Inst. Stat. Math. 18 (1966) 179–189.
[11] E. Nadaraya, On estimating regression, Theory Probab. Its Appl. 9 (1964) 141–142.
[12] G. Watson, Smooth regression analysis, Sankhya Ser. A 26 (1964) 359–372.
[13] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, London, 1986.
[14] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (3) (1994) 287–314.
[15] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.
[16] L. Devroye, L. Györfi, Nonparametric Density Estimation: The L_1 View, Wiley, New York, 1985.
[17] A. Guérin-Dugué, et al., Deliverable R3-B1-P—Task B1: Databases, Tech. rep., Elena-NervesII "Enhanced Learning for Evolutive Neural Architecture", ESPRIT-Basic Research Project Number 6891, anonymous FTP: /pub/neural-nets/ELENA/Databases.ps.Z on ftp.dice.ucl.ac.be (June 1995).
[18] O. Mangasarian, W. Wolberg, Cancer diagnosis via linear programming, SIAM News 23 (5) (1990) 1,18.
[19] C. Blake, C. Merz, UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998).
[20] L. Holmström, P. Koistinen, J. Laaksonen, E. Oja, Neural and statistical classifiers—taxonomy and two case studies, IEEE Trans. Neural Networks 8 (1997) 5–17.
[21] J. Laaksonen, Subspace classifiers in recognition of handwritten digits, Acta Polytechnica Scandinavica Ma 84, Doctoral Thesis, Helsinki University of Technology, 1997.
[22] D. Michie, D. Spiegelhalter, C. Taylor (Eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwood, Chichester, UK, 1994.
[23] E. Alpaydin, F. Gürgen, Comparison of statistical and neural classifiers and their applications to optical character recognition and speech classification, in: C. Leondes (Ed.), Image Processing and Pattern Recognition. Neural Network Systems Techniques and Applications, Academic Press, New York, 1998, pp. 61–88.
[24] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[25] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, R. Agular, M. Gaasenbeek, M.R.M. Angelo, G. Pinkus, T. Ray, M. Koval, K. Last, A. Norton, T. Lister, J. Mesirov, D. Neuberg, J.A.E.S. Lander, T. Golub, Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, Nat. Med. 8 (2002) 68–74.
[26] S. Dudoit, J. Fridlyand, T. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. 97 (2002) 77–87.

About the Author—FABIAN HOTI received the M.S. and Phil.Lic. degrees in Applied Mathematics from the University of Helsinki, Finland, in 1997 and 2001, respectively. Presently he is a Ph.D. student in the Division of Mathematical Methods of Information Technology at the Rolf Nevanlinna Institute. His research interests include statistical pattern recognition, bioinformatics, exploratory data analysis and visualization of multivariate data.

About the Author—LASSE HOLMSTRÖM received the B.S., M.S. and Phil.Lic. degrees in Mathematics from the University of Helsinki, Finland, in 1974, 1975 and 1977, respectively. He receieved the Ph.D. degree in Mathematics from Clarkson College of Technology, Potsdam, NY, in 1980. He has held research positions in the University of Helsinki, Helsinki University of Technology, as well as visiting professorships in several universities in the United States. Currently he is the Director of Rolf Nevanlinna Institute, a research institute of Mathematics, Computer Science and Statistics at the University of Helsinki. His research interests include nonparametric estimation of functions, computational methods in statistics and pattern recognition.