International Journal of Computational Intelligence Systems, Vol.1, No. 2 (May, 2008), 188–201

A COMPARATIVE STUDY OF VARIOUS PROBABILITY DENSITY ESTIMATION METHODS FOR DATA ANALYSIS

ALEX ASSENZA
Università degli Studi di Genova, MUSES (Multi SEnsorS Laboratory)
Via A. Magliotto, 2 - 17100 Savona, Italia
E-mail: alex.assenza@gmail.com

MAURIZIO VALLE
Università degli Studi di Genova, DIBE, MUSES (Multi SEnsorS Laboratory)
Via A. Magliotto, 2 - 17100 Savona, Italia
E-mail: maurizio.valle@.unige.it

MICHEL VERLEYSEN
Université catholique de Louvain, Machine Learning Group
Place du Levant 3, 1348 Louvain-la-Neuve, Belgique
E-mail: verleysen@dice.ucl.ac.be

Received:01-10-2007, Revised: 08-06-2008

Probability density estimation (PDF) is a task of primary importance in many contexts, including Bayesian learning and novelty detection. Despite the wide variety of methods at disposal to estimate PDF, only a few of them are widely used in practice by data analysts. Among the most used methods are the histograms, Parzen windows, vector quantization based Parzen, and finite Gaussian mixtures. This paper compares these estimations methods from a practical point of view, i.e. when the user is faced to various requirements from the applications. In particular it addresses the question of which method to use when the learning sample is large or small, and of the computational complexity resulting from the choice (by cross-validation methods) of external parameters such as the number of kernels and their widths in kernel mixture models, the robustness to initial conditions, etc. Expected behaviour of the estimation algorithms is drawn from an algorithmic perspective; numerical experiments are used to illustrate these results.

Keywords: Probability Density Function estimation, Parzen windows, finite Gaussian mixtures.

1. Introduction

Numerical data are found in many applications of data analysis. Most numerical data come from measurements and experiments, thus resulting from the sampling of a random variable, the mathematical concept that characterizes the numerical results of experiments. To analyse data, one may choose to handle directly the results of the experiments. For example, simple data analysis methods like the linear PCA (Principal Component Analysis), the non-linear MLP (Multi-Layer Perceptron), and many others, work directly on the numerical values of samples. While this way of working may reveal adequate in many situations, other ones require working with the underlying random variable instead of the numerical sample.

A random variable is completely characterized by its Probability Density Functions (PDF), i.e. a function that represents the probability of an event to occur when the random variable is equal to (or contained in an interval around) a specific value. Two examples may be used to illustrate this necessity.

In the Bayesian framework, for example in Bayesian classification, decisions are taken according to Bayes' rule, which directly involves the evaluation of the PDF. In a two-class problem for example, the decision to choose one class or another is taken according to the largest PDF evaluated on the data to classify, after multiplication by some prior. As the PDFs are unknown

in practical situations (only the samples are known), working in such framework necessitates to estimate the PDF of the random variables.

Another example is novelty detection. Let us imagine a sensor giving, in normal conditions, output values distributed according to some PDF. If the sensor characteristics are modified, for example by aging, the PDF of the measurements will vary, even though the conditions have not changed. Assessing the difference between PDFs is thus a way to detect aging, i.e. to assess some novelty in the measurement process.

Estimating PDFs based on a sample is thus of primary importance in many contexts. There exist a lot of methods aimed to estimate PDFs (see for example Ref. 1 for an overview); while all of them are applicable in the univariate case, some of them may also be applied to the multivariate one. However, and despite the vast literature on the topic, even in the univariate case there is no consensus about which method to use, nor about the pros and cons of these methods. In this paper, we will restrict the study to univariate PDF estimation.

The aim of this paper is to give some insights into the methods that are traditionally used by data analysts. Both a priori pros and cons and experimental comparisons will be provided. All questions regarding the use of the methods for estimating PDF will not be answered, nor will all methods be covered. Nevertheless, this paper gives guidelines to the user having to choose between standard PDF estimation methods, when he is faced to various requirements concerning performances, speed and robustness levels. Guidelines should be understood here in a broad sense. It is not the purpose of this study to provide strict rules, nor even to try to generalize numerical results found on illustrating examples to real situations. We simply believe that this is impossible, as the wide variety of situations encountered with various PDF may obviously lead to various conclusions. This paper takes another point of view. By looking to the models and estimation algorithms, it is possible to have insights about their expected behavior, for example what concerns the quality of the estimation, the robustness to small and large samples, the difficulties encountered due to parameters that have to be adjusted, the robustness to these parameters, etc. Most of these expected characteristics are not described in an independent and objective way in the existing literature, in particular when they might be considered as drawbacks rather than as advantages. The originality of this paper is thus to compare widely used PDF estimation models from the point of view of their expected characteristics, rather than from numerical results that would inevitably be obtained on specific examples without convincing generalization power. Nonetheless, in order to validate the assertions, the latter are illustrated on a PDF example chosen for its variety of characteristics; this experimental part of the paper should not be taken as a proof but rather as an illustration of the assertions that form the core of this paper.

After an introduction, Section 2 will describe the most popular methods used to estimate PDF, without aiming at being exhaustive: histograms, Parzen windows, vector quantization based estimators, and Finite Gaussian Mixtures (FGM). Section 3 will comment these methods according to criteria and constraints given by practical applications: expected estimation error, variance of this error, computational effort and memory requirements, number of necessary data for the estimation, number of parameters in the model, etc. The expected model characteristics described in this section form the core of this paper. Next, Section 4 will present the adopted experimental procedure. In Section 5 some selected experiments illustrate the assertions described in Section 3: the expected performances of the methods are compared to the results of the experiments. Finally, Section 6 will conclude the paper by some guidelines for the use of PDF estimation methods.

2. PDF Estimation

Probability density function (PDF) estimation is a fundamental step in statistics as a PDF characterizes completely the “behavior” of a random variable. It provides a natural way to investigate the properties of a given data set, i.e. a realization of this random variable, and to carry out efficient data mining.

When we perform density estimation three alternatives can be considered. The first approach, known as parametric density estimation, assumes the data is drawn from a specific density model. The model parameters are then fitted to the data. Unfortunately, an a priori choice of the PDF model is in practice not suited since it might provide a false representation of the true PDF.

An alternative is to build nonparametric PDF estimators[2,3], as for example the histogram or the Parzen

window estimator, in order to "let the data speak for themselves". A third approach consists in using semi-parametric models[4]. As nonparametric techniques, they do not assume the a priori shape of the PDF to estimate. However, unlike the nonparametric methods, the complexity of the model is fixed in advance, in order to avoid a prohibitive increase of the number of parameters with the size of the data set, and to limit the risk of overfitting. Finite mixture models are commonly used to serve this purpose.

In this section, we briefly recall the histogram and the Parzen window estimator, and show how the kernel width can be selected a priori in the case of Parzen. Next, we present a vector quantisation based version of Parzen, which allows us to reduce its model complexity. Finally, we present Finite Gaussian mixture models.

2.1. *Histograms*

In the following, we refer to X as a continuous random variable, $p_X(x)$ as its PDF and $\{x_n\}_{n=1}^N$ as a sample of X. Nothing prevents the method to be applied, in theory, to multi-dimensional situations; in this case X is a random vector and x_n are vectors. However, we will see that several methods are limited to small dimensions in practical situations.

Given N observations x_n, we approximate the unknown PDF of X by dividing the real line in M bins B_j of width 2σ and counting the number of samples falling into each bin. In order to keep the integral of the estimate equal to one, the latter is multiplied by a normalizing factor:

$$\hat{p}_X(x) = \frac{1}{N2\sigma}\sum_{n=1}^{N}\sum_{j=1}^{M} I(x_n \in B_j) I(x \in B_j) \tag{1}$$

where $I(\cdot)$ is the indicator function defined as follows:

$$I(x \in B_j) = \begin{cases} 1 & \text{if } x \in B_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The histogram appears to be strongly dependent on the choice of σ, as it regulates the smoothness of the estimate. In addition, the choice of the locations of the bin origins may influence the quality of the estimate as well. One way to reduce this dependency is to use an *averaged shifted histogram*[2]. However, in this approach, one should choose additional parameters a priori, i.e. the number of shifts and the shift step.

2.2. *Parzen windows*

Like histograms, the Parzen window estimator[5] does not assume any functional form of the unknown PDF, as it allows its shape to be entirely determined from the data without having to choose a location of the centers. The PDF is estimated by placing a well-defined kernel function centered on each data point and then determining a common width σ, also denoted as the smoothing parameter. In practice, Gaussian kernels are often used:

$$N(x|c,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-c)^2}{2\sigma^2}\right), \tag{3}$$

where c and σ are the kernel centre and width respectively. The estimated PDF is then defined as the sum of all the Gaussian kernels, multiplied by a scaling factor:

$$\hat{p}_X(x) = \frac{1}{N}\sum_{n=1}^{N} N(x|x_n,\sigma) \quad . \tag{4}$$

Next, we present a non-exhaustive list of techniques used for selecting the kernel width, mainly based on the integrated squared error.

2.1.1 *Silverman's plug-in*

Choosing a priori the kernel width σ is definitely not the best way to use Parzen windows, as the optimal value (i.e. the value that minimises the measure of dissimilarity between the true PDF and its estimation) strongly depends on the type of data we are dealing with, their number and the amount of noise they are corrupted by.

Let us first define the *integrated square error* (ISE):

$$\text{ISE} = \int \{\hat{p}_X(x) - p_X(x)\}^2 dx \, . \tag{5}$$

Parzen showed in Ref. 5 that, given a specific standard kernel $K(t)$, the kernel width σ that minimizes the expected ISE should satisfy the following condition (in the univariate PDF estimation case):

$$\sigma_{\text{opt}} = \left(\frac{\int K^2(t)dt}{N\left(\int t^2 K(t)dt\right)^2 \int \frac{\partial^2 p_X(x)}{\partial x^2} dx} \right)^{\frac{1}{5}} \, . \tag{6}$$

Unfortunately, this expression depends on the unknown density itself. Therefore, Silverman proposed in Ref. 3 to plug a Gaussian distribution to approximate $p_X(x)$, leading to the following rule of thumb:

$$\sigma_{\mathrm{SIL}} = 0.9AN^{-\frac{1}{5}}, \text{ with } A = \min\left\{s, \frac{R}{1.34}\right\}. \quad (7)$$

In Eq. (7) s is the empirical standard deviation of the data and R is the sample interquartile range. The motivation behind the introduction of the interquartile range R is to reduce the sensitivity of σ_{SIL} to outliers, as R calculates the sample interquartile range from the 75% quantile to the 25% one.

In theory, the application of the σ_{opt} formula would lead to overfitting, as the optimal value would be computed on the data themselves. In practice however, using the σ_{SIL} estimator instead of σ_{opt} tends to overestimate the optimal kernel width, as already mentioned by Silverman[3]; this is true when dealing with most cases of non-Gaussian distributions, in particular with multimodal populations. Overfitting thus not occurs in this case. However, as it will be discussed in Sections 3 and 4, the overestimation was only found true for relatively large datasets.

2.2.2 *Exhaustive search*

Consider a well-defined error criterion E and suppose we can evaluate it. By letting the kernel width σ vary over a certain range, one can easily determine the optimal kernel width σ_{opt} by selecting σ that minimizes E. Note however that care should be taken to avoid overfitting when evaluating the model performance: part of the data, the learning set, should be used for constructing the Parzen estimate, while a validation set should be kept aside for evaluating its generalization performance and selecting σ_{opt}. Indeed, contrarily to Silverman's plug-in solution, no smoothing estimator enters into the selection of the optimal width in this case. Moreover, resampling techniques should be used in practice, such as for example leave-one-out, K-fold cross-validation, Monte Carlo cross-validation or Bootstrap to obtain reliable estimates. For a detailed overview of these methods, we refer to Ref. 6.

Whereas this approach is expected to outperform Silverman's rule of thumb, the price to pay is its computational complexity. Besides, as we do not know the true PDF in practice, this approach is useless for most error criterions. Yet, this method is very convenient in two particular cases:

- when considering toy problems (thus knowing the density to approximate) in order to assess objectively the quality of the models or to have a reference to compare to;
- when minimizing ISE, because its dependency in the unknown PDF can be eliminated as discussed below.

Consider again the ISE. This error criterion can be rewritten as follows:

$$\mathrm{ISE} = \int \hat{p}_X(x)^2 dx - 2\mathrm{E}\{\hat{p}_X(x)\} + \int p_X^2(x) dx. \quad (8)$$

The last term can be ignored as far as ISE minimization is concerned, as it does not depend on σ. Only the second term depends both on σ and on the unknown density $p_X(x)$. In the second term this unknown density can be approximated by its leave-one-out estimator[2] $\hat{p}_{X,-n}(x_n)$; we may then define the following error criterion:

$$\begin{aligned} \mathrm{E}_{\mathrm{LOO}}(\sigma) &= \int \hat{p}_X(x)^2 dx - 2\mathrm{E}\left\{\hat{p}_{X,-n}(x_n)\right\} \\ &\approx \int \hat{p}_X(x)^2 dx - \frac{2}{N}\sum_{\mathrm{n}=1}^{\mathrm{N}} \hat{p}_{X,-n}(x_n) \end{aligned}, \quad (9)$$

where

$$\hat{p}_{X,-n}(x) = \frac{1}{N-1}\sum_{m=1,m\neq n}^{N} N\left(x|x_m,\sigma\right). \quad (10)$$

Substituting the Parzen estimate of $p_X(x)$ in this expression, we obtain the *leave-one-out cross-validation criterion*[7]:

$$E_{LOO}(\sigma) = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N} N\left(x_m|x_n,\sqrt{2}\sigma\right) - \frac{2}{N(N-1)}\sum_{n=1}^{N}\sum_{m=1,m\neq n}^{N} N\left(x_n|x_m,\sigma\right). \quad (11)$$

Interestingly, this criterion does not require the evaluation of an integral anymore. Finally, by scanning a certain range of σ the optimal width can be selected:

$$\sigma_{\mathrm{LOO}} = \arg\min_{\sigma} E_{LOO}(\sigma). \quad (12)$$

More recently, elaborate techniques, such as the Sheather-Jones plug-in or the smoothed bootstrap, were proposed for selecting σ automatically. For a comprehensive review we refer to Ref. 8 and the references therein. However, these methods are usually

not used by the practitioner due to their technical complexity, while showing little effective improvement in real applications. We therefore excluded these techniques from our discussion.

2.3. *Vector quantization based Parzen*

Whereas the computational complexity of Parzen is relatively small, its model complexity is proportional to the number of data samples. This can rapidly lead to memory storage problems. Furthermore, as the kernel width is common to all kernels, it can be locally mismatched. This in turn can lead to oscillations in the distribution tails or in low-density regions of the input space.

A straightforward way to circumvent these problems is to perform vector quantisation (VQ) beforehand. Once the number of prototypes K is chosen, the VQ algorithm computes their locations in the input space by minimizing a quantization error and associates to each prototype a region of influence, called Voronoi region.

One of the most popular VQ techniques is the *Competitive Learning* (CL)[9-10]. In the remainder we will restrict ourselves to CL as it is sufficiently flexible, while limiting the number of parameters. Other widespread approaches are *K-means*, *Neural Gas*[11] or *Self-Organizing Maps*[12]. The latter usually results in a better quantization of the data, at the price of a slightly increased complexity. When vector quantization performances are the objective to be pursued, such advanced VQ methods are certainly to be preferred to CL. However, in our case, VQ is only used as preprocessing to reduce the number of data and roughly place the centroids at appropriate locations in the data space. Using one VQ method or another does not result in significant differences in the PDF approximation quality; moreover, variations due to the VQ initialization are certainly as large as those due to the choice of the VQ algorithm. The influence of the VQ initialization is discussed in the experimental part of this paper.

CL can be summarized as follows:

(i) Initialise all prototypes $c_k, 1 \le k \le K$

(ii) For each data x_n from the database:

Select the winner:

$$c_{win} = \arg\min_{c_k} \left\| x_n - c_k \right\|^2 . \qquad (13)$$

Update the winner:

$$c_{win} = c_{win} + \alpha \cdot (x_n - c_{win}) . \qquad (14)$$

(iii) Repeat (ii) until convergence.

In Eqs (13) and (14), α is the learning rate. Usually, α decreases exponentially during the iterations of the algorithm.

By placing a Gaussian kernel on each prototype c_k, the unknown PDF can be approximated in a Parzen-like manner. Besides, to each kernel k, we can associate a local kernel width σ_k, corresponding to the experimental standard deviation of the corresponding Voronoi region. This allows taking into account the variations of the PDF, by setting different kernel widths to different parts of the X space or range. Finally, in order to force a certain amount of overlapping between kernels and thereby increase the generalization performance, a common width-scaling factor h is introduced. A similar idea was introduced in Ref. 13 for supervised kernels (in Radial-Basis Function Networks); the width-scaling factor allows controlling the amount of regularization, which will be adjusted to a Leave-One-Out criterion. This leads to the following density estimate:

$$\hat{p}_X(x) = \frac{1}{K} \sum_{k=1}^{K} N\left(x \middle| c_k, h\sigma_k\right). \qquad (15)$$

Similarly to Parzen, where the common kernel width must be optimised, in VQ-based Parzen the width-scaling factor h should be optimised as well. Again, for comparison purposes, h can be selected by exhaustive search by minimizing a well-defined error criterion E. In practice the leave-one-out cross-validation criterion could be used too. Indeed, following the same derivation as the one used for Parzen windows, we may rewrite E_{LOO} using the VQ-based density estimate. The optimal h is then selected by the following rule:

$$h_{\mathrm{LOO}} = \arg\min_h \frac{1}{K^2} \sum_{k=1}^{K}\sum_{l=1}^{K} N\left(c_l \middle| c_k, h\sqrt{\sigma_k^2 + \sigma_l^2}\right) - \frac{2}{K(K-1)} \sum_{k=1}^{K}\sum_{l=1, l\neq k}^{K} N\left(c_k \middle| c_l, h\sigma_l\right) . \qquad (16)$$

2.4. *Finite Gaussian Mixtures (FGM)*

Finite mixture distributions can approximate any continuous PDF, provided the model has a sufficient number of components and provided the parameters of the model are chosen correctly[14]. The true PDF is approximated by a linear combination of K component densities:

$$\hat{p}_X(x) = \sum_{k=1}^{K} P(k)\, p(x|k), \text{ with } K << N \qquad (17)$$

.

In Eq. (17) $p(x|k)$ is the probability of x given the component distribution k and $P(k)$ are the mixture proportions or priors. The priors are non-negative and must sum to one. In practice, Gaussian kernels are often used:

$$p(x|k) = N\left(x|c_k, \sigma_k\right). \qquad (18)$$

A popular technique for approximating iteratively the maximum likelihood estimates of the model parameters $P(k)$, c_k and σ_k is the *expectation-maximization* (EM) algorithm[15]. Let us define the likelihood function:

$$\mathrm{L} = \prod_{n=1}^{N} \hat{p}_X(x_n). \qquad (19)$$

Maximizing the likelihood function is equivalent to finding the most probable PDF estimate provided the data set $\{x_n\}_{n=1}^{N}$.

The EM operates in two stages. First, in the *E-step*, the expected value of some "unobserved" data is computed, using the current parameter estimates and the observed data. Here the "unobserved" data are the data labels of the samples. They correspond to the identification number of the different mixture components and specify which one generated each data. Subsequently, during the *M-step*, the expected values computed in the *E*-step are used to update the model parameters accordingly. Each iteration step i can be summarized as follows[4]:

E-step:

$$P^{(i)}(k|x_n) = \frac{p_X{}^{(i)}(x_n|k)P^{(i)}(k)}{\hat{p}_X{}^{(i)}(x_n)} \quad . \qquad (20)$$

M-step:

$$\boldsymbol{c}_k{}^{(i+1)} = \frac{\sum_{n=1}^{N} P^{(i)}(k|x_n)x_n}{\sum_{n=1}^{N} P^{(i)}(k|x_n)}, \qquad (21)$$

$$(\sigma_k^2)^{(i+1)} = \frac{\sum_{n=1}^{N} P^{(i)}(k|x_n)\left(x_n - c_k{}^{(t+1)}\right)^2}{\sum_{n=1}^{N} P^{(i)}(k|x_n)}, \qquad (22)$$

$$P^{(i+1)}(k) = \frac{1}{N} \cdot \sum_{n=1}^{N} P^{(i)}(k|x_n). \qquad (23)$$

Note that the value $P^{(i)}(k|x_n)$ computed in the *E*-step corresponds to the posterior probability that a known data sample x_n was generated by component k.

It can be mentioned that FGM could be viewed as specific Support Vector Machines in the context of kernel methods[16]. However, in this case, and contrarily to FGM, the kernel widths are fixed in advance and the kernel centres are restricted to the data points. Even if a sparse solution can be obtained, such restrictions on the kernels widths and centres lead to inefficient covering of the effective distribution, contrarily to FGM where kernel centres and widths are optimized.

3. Expected Pros and Cons of PDF Estimation Methods

Not all PDF estimation methods are expected to behave similarly in all situations. There are certainly pros and cons to all methods, making all of them suitable for some kind of applications. The following of this section aims at giving some insights about the expected performances of each method, with respect to several criteria and constraints imposed by the applications. More precisely, the a priori expected performances of the methods are investigated (together with the variance of the estimates). As PDF estimation methods sometimes have computational requirements that are not compatible with the application constraints, computational effort and memory requirements will be discussed too. Finally, all these performances will be related to the number N of data that are necessary to obtain an "adequate" approximation of the underlying PDF, and to the complexity of the models.

Based on the characteristics of the models, the following comments can be made about the methods described in Section 2.

- Histograms suffer from several drawbacks. First, they are not continuous by definition. Using non-continuous PDF estimates might reveal problematic in some contexts, for example in binary Bayesian classification, when the intersection between the PDF of the two classes must be found. Moreover, the choice of the bins (widths and centres) may be too difficult especially in the case of non smooth PDF or of PDF with unbounded support. Except their computational simplicity, histograms have no advantage compared to Parzen estimators, while they have supplementary drawbacks; Parzen estimators may be viewed as a continuous (and derivable) extension to histograms. Finally, let us

note that histograms are not easily extended to multivariate cases, as bins rapidly tend to be empty in average. For all these reasons, histograms will not be considered in the experimental part of this paper.

- Parzen estimators require a correct choice of the smoothing parameter σ. A small σ will result in overfitting: the estimate will show peaks around each data. On the other hand a large σ will result in an excess of regularization: the estimate will be smoother than the true PDF. Often, Silvermann's rule-of-thumb is applied. According to the literature[2-3], this rule overestimates σ for large values of the number N of data, and estimates it more or less correctly when N is small (at least for unimodal distributions). The overestimation mainly results from the non-Gaussian character of the PDF.
- As σ acts as smoothing parameter in Parzen's estimator, σ should be naturally large for small N and small for large N.
- Except if using the Leave-One-Out approximation detailed in Section 2.3, a cross-validation estimate of an optimal σ value is computationally expensive if N is large. Furthermore, such cross-validation is not needed if N is low, as rules-of -thumb provide adequate values.
- Vector Quantisation-based methods do not perform well. They tend to overestimate the tails of the distributions. Indeed in the tails, the number of data available is reduced, leading to poor vector quantisation properties. Furthermore, most vector quantisation methods lead to the so-called "magnification factor" effect[17], which consists in the fact that a PDF after VQ reflects the original PDF elevated to a power lower than one (in other words, the vector quantisation itself, before any PDF estimation, overestimates the tails and underestimates the peaks of a distribution).
- The standard deviation of the estimations is an important concern too. Indeed most methods require some initialisation, leading to different estimates for different initial values of some parameters. Having large variations between the estimates is of course not a good property of the estimation method. Mean and standard deviation are respectively useful in order to assess the real performance of a method and its robustness regarding the initialisation randomness. Parzen estimators do not suffer from this drawback, as there is no initialisation. FGM are based on a powerful optimisation algorithm (EM), therefore usually offer a low standard deviation of the estimate. The standard deviation could increase though: when a large number K of Gaussian components is used, EM may produce different acceptable solutions corresponding to different stable configurations of the component mixtures. Multiple initializations may however be used to reduce the variance, if a higher computational cost is accepted; multiple initializations could be used in the vector quantization based estimator too for the same purpose.
- More dramatically, the FGM optimisation (EM algorithm) may collapse if the number K of Gaussian functions is large, or equivalently, if the number N of data is small[18]. While recent techniques such as maximum penalized FGM[19] or variational FGM[20] can be used to avoid this problem, collapsing quite naturally occurs when one tries to incorporate too much structure into the density model compared to the number of available data. In this situation, Parzen estimate is an interesting alternative. It also has the feature of having a single parameter to optimise (compared to several parameters for each Gaussian function in a FGM), which reveals an advantage when the number N of data available is small (the minimization of the likelihood is trapped in different local minima).
- If this problem regarding a low number N of data available is not encountered, FGM are thus expected to perform well, probably better than other methods. If N is large, one has more knowledge about the underlying PDF; increasing the number K of components in FGM is thus expected to increase the performances of the estimator.

Finally, about memory requirements, histograms and Parzen estimators have the advantage that they do not require any learning. However, if N is large, both the memory requirements and the evaluation of the estimate for any value of x may exceed any reasonable level imposed by the application context. The complexity of the evaluation is a direct consequence of the Parzen formula (it is a sum over all N samples). This is another argument in favour of FGM.

4. Methodology for the Experimental Illustration of Expected Behavior

Section 3 detailed properties and expected behaviour of the PDF estimations methods that directly result from the algorithms. In the remaining of this paper, we will mainly show experimental results that do confirm the

expected behaviour. We will also detail other properties that result from simulation experiments. The latter can be considered valid only for the used PDF and are not expected to draw general properties.

The size N of the sample used to estimate a PDF will be investigated. Indeed, as introduced in Section 3, it is expected that some methods will not perform equally when N is low and when N is large; assessing which method to use in which situation is one of the goals of this study.

Experiments are performed with an artificially generated PDF. Hence it is possible to compare the estimation to the real PDF. The comparison has to rely on a distance (between PDFs) measure that has to be chosen properly too.

4.1. *Reference PDF and sampling*

The reference PDF illustrated in Figure 1 has been chosen for the simulations.

It has been built to include a wide variety of behaviours that influence the performances of the algorithms: flat regions and slopes, sharp and smooth peaks, skewness, and a smooth right-tail and climbing left side. By constructions, it is obviously differentiable.

More precisely, the reference PDF (Figure 1) is a mixture of four Gaussian PDFs, i.e. $N_1(x|3.5,1.6)$, $N_2(x|7.5,2)$, $N_3(x|12,2)$, $N_4(x|16.5,1.5)$, and of one Gamma PDF with parameters equal to 8 and 0.2. Samples with N realizations have been drawn from the reference PDF according to a Monte-Carlo drawing scheme.

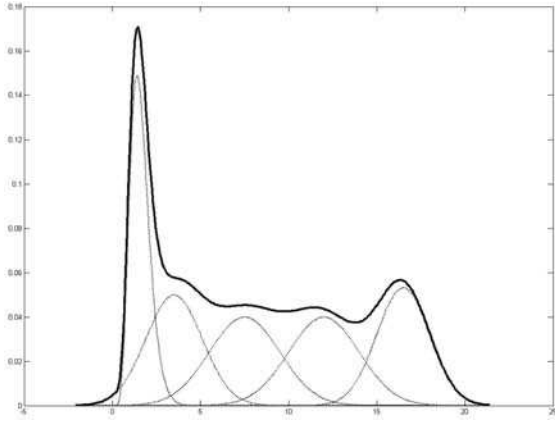

Fig. 1. Reference PDF. Grey lines: PDF components (four Gaussian and one Gamma functions). Black line: reference PDF (sum of components).

4.2. *Distance measures*

Many distance measures could be used to compare PDF, at least in the one-dimensional case[21]. Three widely known ones are mentioned here. Let us define two PDF $p_X(x)$ and $p_Y(x)$, both taking a null value outside the $[a, b]$ interval. The Mean Square Error is defined as

$$\mathrm{MSE}(p_X(x), p_Y(x)) = \frac{1}{b-a}\int_a^b (p_X(x) - p_Y(x))^2\, dx \quad , \tag{24}$$

the Hellinger distance is defined as

$$H^2(p_X(x), p_Y(x)) = \int_a^b \left(\sqrt{p_X(x)} - \sqrt{p_Y(x)}\right)^2 dx \quad , \tag{25}$$

and the Kullback divergence is defined as

$$J(p_X(x), p_Y(x)) = \int_a^b (p_X(x) - p_Y(x)) \log\left(\frac{p_X(x)}{p_Y(x)}\right) dx \quad . \tag{26}$$

Note that the $1/(b\text{-}a)$ factor in the MSE definition is nothing else than a normalisation constant that could be added to the two other measures too.

All distances have been used in experimental conditions. On a qualitative point of view, they all lead to similar conclusions about the use of the PDF estimation methods. For this reason, the Hellinger distance only is used in the following of this paper; the Hellinger distance is symmetric (unlike the Kullback divergence) and not too sensitive to large differences between PDF limited to small regions of the space (unlike the Mean Square Error). It must be insisted on the fact that the Hellinger distance has been chosen for its appropriateness to the goal of this paper (experimental qualitative comparison between PDF estimation methods); other distance measures could however reveal more appropriate in real contexts or more specific situations; for example, if high peaks over a limited range in the difference between the PDF has to be avoided, one would prefer the MSE criterion, because of the larger exponent on the PDF.

5. Experimental Illustration of the Expected Behavior

Parzen windows, vector quantization based Parzen windows, and finite Gaussian mixtures have been used to estimate the reference PDF detailed in Section 4. For

the reasons discussed in Section 3, we will not take into account histograms in the experimental results. The last two models used for the experiments include user-defined parameters (number of kernels) that may have influence on the quality of the results. Moreover, the size *N* of the sample largely influences the performances too. The following experiments show the quality of the estimation methods with regards to the parameters and the size of the sample: the experiments are intended to illustrate the analysis of Section 3. Unless otherwise mentioned, the experiments have been carried out with a size *N* of the sample varying according to Table 1.

Table 1. Number N of data (size of sample) drawn randomly from the reference PDF for all experiments.

Lower bound	Upper Bound	Step
10	250	10
250	1000	250
1000	5000	1000
5000	15000	2500

5.1. *Parzen windows*

Parzen windows are a deterministic method once the kernel width σ is fixed: the performances are deterministic too, and no standard deviation of the results can be computed. For each number *N* of data (size of the sample) as detailed above, σ was varied from 0.15 to 2 by steps of 0.1. Figure 2 shows the results in terms of Hellinger distance between the true PDF and its estimation, with respect to σ and with *N* as parameter.

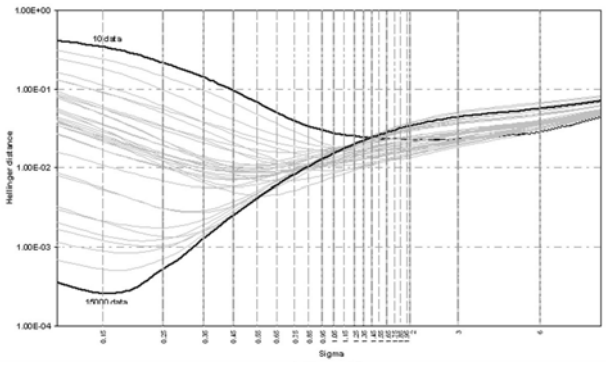

Fig. 2. Distance between the true PDF and its Parzen windows estimation with varying kernel width s. The number of data is a parameter.

As expected, the optimal kernel width σ is small for large values of *N* and large for small values. Indeed the number of kernels used in Parzen windows is equal to the number of data; therefore the kernel variance should increase for small *N* in order to build a smooth PDF estimate.

Other conclusions can be drawn from Figure 2. First, it is seen that whatever is the number *N* of data, the error curve has a minimum. Figure 2 confirms that overfitting occurs for too small σ, and oversmoothing for too large σ.

Another conclusion from Figure 2 is that, for a fixed number *N* of data, the resulting approximation error (at optimal σ) is always lower than the error resulting from a lower *N*. This means that increasing the number of kernels is always beneficial, at least when σ is approximately optimized. However, the same is not true for a fixed σ: in the center and right parts of Figure 2, it may be seen that increasing the number of kernels at fixed σ might increase the approximation error.

What is more surprising is the amount of dependence observed when varying σ, for a fixed number *N* of data. Taking into account the logarithmic scale of Figure 2, one sees that with large *N*, a wrong choice of σ may lead to errors that are two orders of magnitude larger than the optimum. Unexpectedly, the dependency to the kernel width σ is much larger for large samples than for small ones! This result should be put in parallel with ad-hoc heuristics that usually select σ in a non-optimal way, like Silverman's plug-in: in many situations including non-difficult ones, only an exhaustive search (or the associated leave-one-out choice) can lead to an adequate choice of the kernel width σ [13].

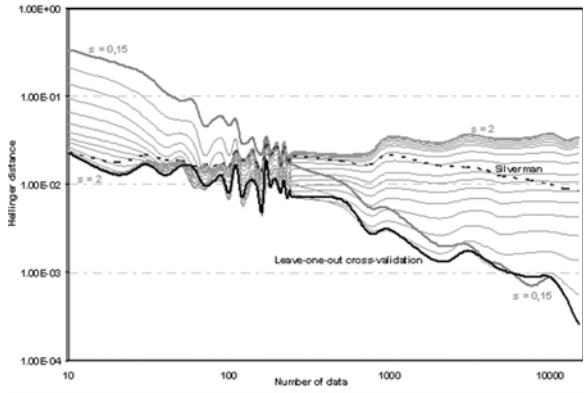

Fig. 3. Distance between the true PDF and its Parzen windows estimation with varying number of data. The kernel width σ (s in the figure) is a parameter.

In order to assess both Silverman's plug-in and the leave-one-out cross-validation procedures for choosing σ, Figure 3 shows the performances (again in terms of Hellinger distance) versus the number *N* of data with σ as parameter. The results of the Silvermann plug-in and the leave-one-out cross validation method are shown too.

Figure 3 shows that Silverman's plug-in is an efficient method to set σ only when a small number of data is considered. For larger samples, Silverman's plug-in results are not acceptable. The leave-one-out cross validation method gives better results almost everywhere; however, because of its higher computational load, it should be used only for large samples.

5.2. *Vector quantization based Parzen*

Contrarily to Parzen windows where only the kernel width σ must be optimized, the Vector quantization based Parzen method necessitates to optimize two parameters: the kernel width through the width scaling factor *h* and the number *K* of kernels. Using a double exhaustive search method embedded in a cross-validation procedure would lead to unaffordable computing times in most applications. Therefore it is necessary to develop heuristic choices to set at least one of the two parameters. In order to assess if the dependency to one of the two parameters is low enough to develop an efficient heuristics, experiments are made by varying the width scaling factor *h* from 1 to 100 and the number *K* of kernels from 5 to 300. As the method uses a random initialization of the kernel centres, each experiment is repeated 20 times and the mean and standard deviations are computed.

Figure 4 shows the mean of the Hellinger distance between the true PDF and its approximation with respect to the width scaling factor *h* and number of kernels *K*. A relative low dependency to the width scaling factor *h* is observed, but only for large number of kernels; the dependency is much higher for low numbers of kernels.

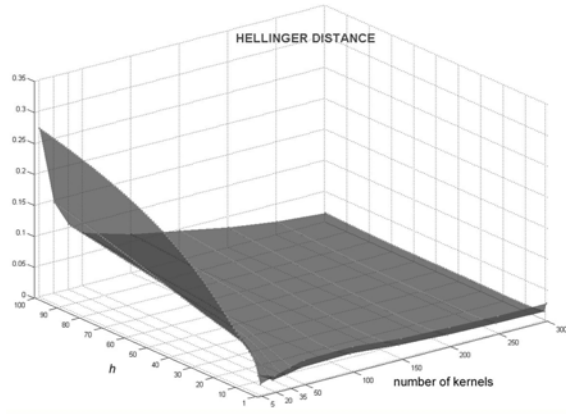

Fig. 4. Mean distance between the true PDF and its vector quantization based Parzen estimation with varying width scaling factor h and number K of kernels.

If for each number *K* of kernels we consider the value of *h* that gives the minimum mean of Hellinger distance, we obtain the results shown in Figure 5. The experimental standard deviation is shown too, together with the optimum value of *h* in each experiment.

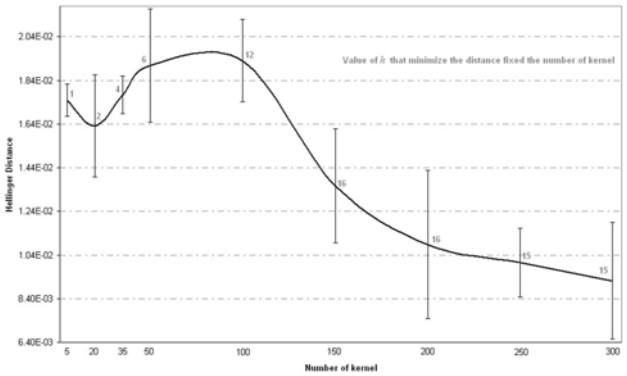

Fig. 5. Mean and standard deviation of the distance between the true PDF and its vector quantization based Parzen estimation with varying number of kernels and optimized value of the width scaling factor *h*.

Figures 4 and 5 show that improved results are obtained when increasing the number of kernels; the optimal width scaling factor *h* should increase in parallel. The best results are obtained with the largest number of kernels used in the experiments; this reflects the fact that, in general, going from the standard Parzen windows estimator to the Vector quantization based Parzen one decreases the quality of the approximation. Furthermore, using a large number of kernels in the

latter method leads to instability, as confirmed by the large observed standard deviation.

Figure 6 shows a detail of the experiments performed with a fixed number of kernels: $K = 300$. It is observed that the standard deviation is always large compared to the improvement that could be obtained by optimizing the width scaling factor h. On one side this is good news for an expected heuristics to set the value of h, but on the other side this result shows that the stochastic nature of the vector-quantization based method leads to a lack of robustness.

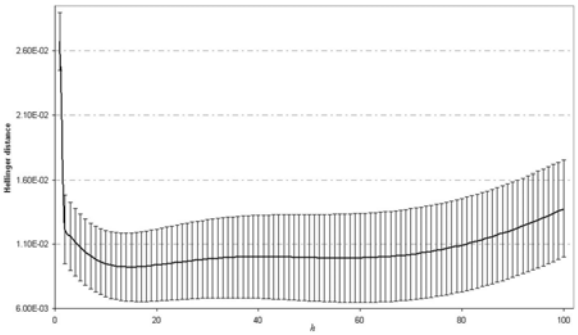

Fig. 6. Mean and standard deviation of the distance between the true PDF and its vector quantization based Parzen estimation with a fixed number of kernels (K = 300) and varying width scaling factor h.

In figure 7 the estimated PDF with the best parameters resulting from the experiments ($K = 300$ and $h = 15$) is shown.

The instability (large standard deviation), the poor results compared to Parzen windows, and the excessive computation time needed to find an optimum value of the two parameters (in a cross-validation scheme) are arguments to use the standard Parzen windows estimator rather than the vector quantization based one, except maybe in situation where the number N of data is so large that using them all without preliminary quantization would be unpractical.

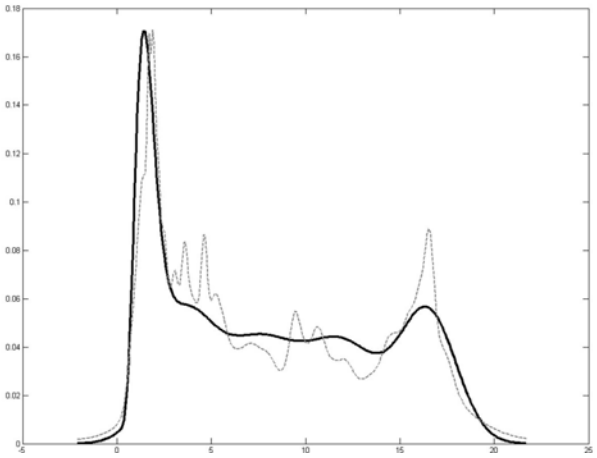

Fig. 7. Reference PDF (solid line) and estimated one (dotted line) through vector quantization based Parzen with K = 300 and h = 15.

5.3. *Finite Gaussian Mixtures*

Like vector quantization based Parzen, Finite Gaussian Mixtures (FGM) rely on two parameters: the number K of kernels and the number of iterations. All other parameters, including the kernel widths, are fixed by learning through the EM algorithm. However, as the latter has no defined stopping criterion, the number of iterations of the E and M steps is usually fixed in advance, therefore constituting a supplementary parameter.

Experiments with 15000 data were performed by varying the number of iterations and the number of kernels. Figure 8 shows the mean value of the Hellinger distance between the true PDF and its approximation, versus these two parameters.

It is not surprising to observe that, in most cases, a larger number of iterations leads to improved results; this is however obtained at the price of an increased computation cost.

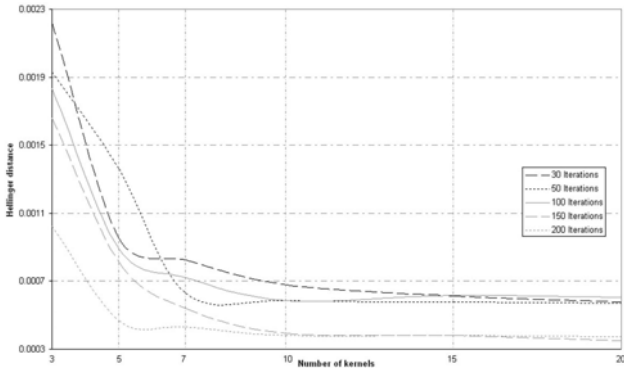

Fig. 8. Mean distance between the true PDF and FGM estimation with varying number of kernels. The number of iterations is a parameter.

In the following experiment, the number of iterations is fixed to 200 in order to observe the dependency to other parameters without interference from the number of iterations. Depending on the number of data and the number of kernels, convergence problems were observed during the run of the EM algorithm. These problems are due to the collapsing of some kernels, as mentioned in the literature[18]. Indeed if the number of data is low compared to the number of kernels, it may happen that only one data is associated to a kernel; the standard deviation of the latter, estimated in the M step of the EM algorithm, then becomes zero, leading to obvious numerical problems. Table 2 shows the configurations (i.e. number of data and number of kernels) where the EM algorithm converges always to a solution, where it never does and where sometimes converges, among the 20 runs made in each configuration.

Table 2. Configurations of the FGM where collapsing of the EM algorithm is observed.

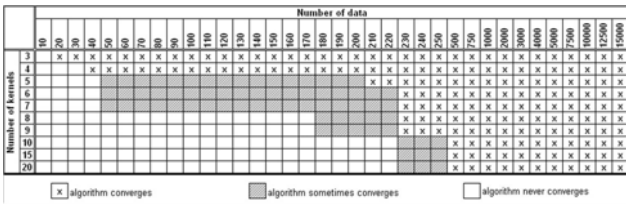

Table 2 clearly shows a problem-free run of the EM algorithm when the number of data is large or equivalently when the number of kernels is low, a lack of convergence in the opposite case, and an intermediate situation for average numbers of kernels and data.

Figure 9 shows the Hellinger distance between the true PDF and its approximation with respect to the number of data, the number of kernels being a parameter.

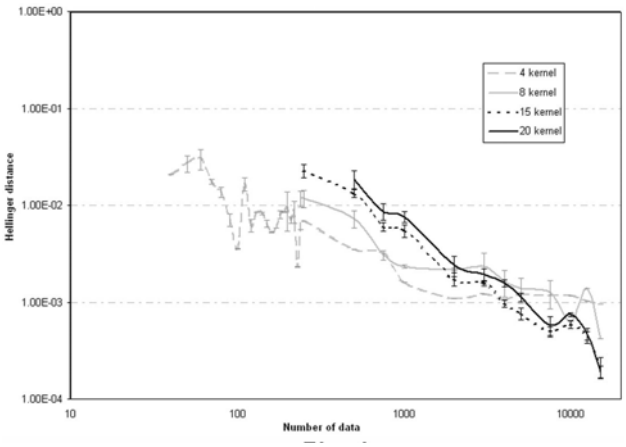

Fig. 9. Mean and standard deviation of the distance between the true PDF and its FGM estimation with varying number of data. The number of kernels is a parameter.

It is interesting to observe the low standard deviation obtained with the EM algorithm, in the situations where it converges. Though the standard deviation could increase in specific situations, this results usually from a particular configuration of the number of data and of kernels, somewhere in the intermediate region of Table 2 where the standard deviation can be computed but includes results with numerical instability.

6. Comments on the Experimental Results

Comparing the methods emphasizes their complementarities. First, as already mentioned in the previous section, vector quantization based Parzen does not bring advantages compared to standard Parzen windows; they require the estimation of a supplementary parameter (leading to increased computation times in a cross-validation scheme), their performance is lower, and finally their stochastic nature (because of the vector quantization initialization) leads to a strong lack of robustness. Compared to standard Parzen windows, vector quantization based Parzen is to recommend only when a very large number of data is available; the vector quantization may then be seen as a pre-processing aimed to reduce the size of the sample, for computational cost reasons. If the sample size is really large, then the vector quantization step will be

harmless regarding the information contained in the data, and beneficial on the computational cost level.

Except in this situation, two methods share the leadership: Finite Gaussian Mixtures (FGM) and Parzen windows. For a low number of data, FGM experience numerical difficulties because of kernel collapsing; Parzen windows are thus preferred. The kernel width should be fixed adequately: despite the sensitivity of the results to the kernel width is lower with a small sample than with a larger one, it remains that the observed results vary by one order of magnitude if the kernel width is chosen adequately. Silverman's rule of thumb may be used in this case, since it provides acceptable results with a low number of samples. For more security, the leave-one-out cross-validation procedure should be used; it provides at least results of the same quality, and largely surpasses Silverman's rule-of-thumb when the number of data is larger.

For large datasets, FGM do not suffer from collapsing problems anymore. They are not too sensitive to the choice of their parameters (number of iterations and number of kernels). Furthermore, in this case Parzen windows would have to be embedded in a leave-one-out cross-validation scheme to set their kernel width parameter; FGM thus offer a good alternative, with a lower computational complexity. This advantage, coupled to a low variance of the results and comparable performances to the Parzen windows, makes FGM more suitable when a large sample is available. These results are summarized in Figure 10.

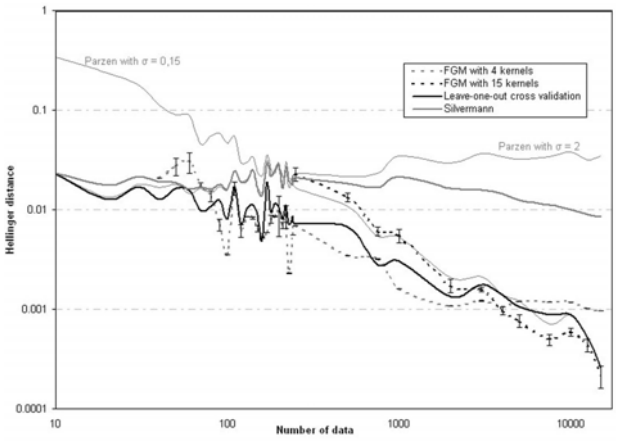

Figure 10. Comparison between the PDF estimation methods with varying number of data.

7. Conclusion

This paper compares widely used PDF estimation models from the point of view of their *expected* characteristics, rather than from numerical results that would inevitably be obtained on specific examples without convincing generalization power. Nonetheless, in order to validate the assertions, the latter are *illustrated* on a PDF example chosen for its variety of characteristics. Other simulations performed on other PDFs show similar qualitative results.

It is shown that Parzen windows is the best estimator when the number of data available is low; low means around 100-200 data in the one-dimensional example shown in this paper, but this number of course varies for different PDF, and increases for higher-dimensional problems. The kernel width parameter in Parzen windows may be set by Silverman's rule-of-thumb if the number of data is definitely low; however the leave-one-out cross-validation procedure should be used when one does not know if the sample is small enough, despite the computational cost increase.

For larger datasets, FGM offer an interesting alternative: their computational complexity becomes smaller than the one of Parzen embedded in a cross-validation scheme, and they show comparable performances and low variance.

Finally, vector quantization based Parzen does not add any advantage, except when the sample is really large and Parzen windows are preferred despite the above conclusion; in this case, the vector quantization may be considered as a pre-processing aiming to reduce the size of the sample, without reducing the contained information.

Acknowledgements

This work has been made possible thanks to the financial support of the bilateral scientific cooperation agreement between the "Communauté Française de Belgique", Belgium, and the Italian Ministry of Foreign Affairs under the project: “Distributed learning algorithms for information processing” 2007 - 2008.

The authors thank Cédric Archambeau and Frédéric Vrins for helpful discussions about pdf estimation and convolution properties of Gaussian distributions respectively.

References

1. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification* (Wiley, New York, 2001).
2. W. Härdle, M. Müller, S. Sperlich and A. Werwatz, *Nonparametric and Semiparametric Models* (Springer, New York, 2004).
3. B. W. Silverman, *Density Estimation* (Chapman & Hall/CRC, London, 1986).
4. G. McLachlan and D. Peel, *Finite Mixture Models* (Wiley, New York, 2000).
5. E. Parzen, On Estimation of a Probability Density Function and Mode, *Annals of Math. Statistics*, **33** (1962) 1065–1076.
6. B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall, London, 1998).
7. A. W. Bowman, An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, **71** (1984) 153-176.
8. M. C. Jones, J. S. Marron and S. J. Sheather, A brief survey of bandwidth selection for density estimation, *Journal of the American Statistical Association* **91**(433) (1996) 401-407.
9. S. C. Ahalt, A. K. Krishnamurthy, P. K. Chen and D. E. Melton, Competitive Learning Algorithms for Vector Quantization, *Neural Networks*, **3**(3) (1990) 277–290.
10. S. Grossberg, Competitive Learning – From Interactive Activation to Adaptive Resonance, *Cognitive Science* **11**(1) (1987) 23–63.
11. T. M. Martinetz, S. G. Berkovich and K. J. Schulten, Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction, *IEEE Trans. Neural Networks*, **4**(4) (1993) 558–569.
12. T. Kohonen, *Self-Organizing Maps* (Springer, Berlin, 1995).
13. N. Benoudjit, C. Archambeau, A. Lendasse, J. A. Lee and M. Verleysen, Width Optimization of the Gaussian Kernels in Radial Basis Function Networks, in *Proc. ESANN '02, Bruges, Belgium, April 24-26, 2002*, pp. 425-432.
14. C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, 1995).
15. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM algorithm, *J. Roy. Stat. Soc. (B)*, **39** (1977) 1–38.
16. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, New-York, 2000).
17. H. U. Bauer, R. Der and M. Herrmann, Controlling the magnification factor of self-organizing feature maps, *Neural Computation*, **8**(4) (1996) 757–771.
18. C. Archambeau, J. A. Lee and M. Verleysen, On Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures, in *Proc. ESANN'03, Bruges, Belgium, April 23-25, 2003*, pp. 99-106.
19. D. Ormoneit and V. Tresp, Averaging, maximum penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates, *IEEE trans. on Neural Networks*, **9**(4) (1998) 639-649.
20. H. Attias, A variational bayesian framework for graphical models. In *NIPS 12*, eds. S. Solla, T. Leen and K.R. Muller (MIT Press, 1999).
21. M. Basseville, Distance measures for signal processing and pattern recognition, *European Journal Signal Processing*, **18**(4) (1989) 349-369.