Robust Nonparametric Probability Density Estimation by Soft Clustering

Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato,
Domingo López-Rodríguez, and María del Carmen Vargas-Gonzalez

School of Computing
University of Málaga
Campus de Teatinos, s/n. 29071 Mlaga
Spain
{ezeqlr,jmortiz}@lcc.uma.es, dlopez@ctima.uma.es

Abstract. A method to estimate the probability density function of multivariate distributions is presented. The classical Parzen window approach builds a spherical Gaussian density around every input sample. This choice of the kernel density yields poor robustness for real input datasets. We use multivariate Student-t distributions in order to improve the adaptation capability of the model. Our method has a first stage where hard neighbourhoods are determined for every sample. Then soft clusters are considered to merge the information coming from several hard neighbourhoods. Hence, a specific mixture component is learned for each soft cluster. This leads to outperform other proposals where the local kernel is not as robust and/or there are no smoothing strategies, like the manifold Parzen windows.

1 Introduction

The estimation of the unknown probability density function(PDF) of a continuous distribution from a set of input data forming a representative sample drawn from the underlying density is a problem of fundamental importance to all aspects of machine learning and pattern recognition (see [2],[11] and [14]).

Parametric methods make a priori assumptions about the unknown distribution. They consider a particular functional form for the PDF and reduce the problem to the estimation of the required functional parameters. On the other hand, nonparametric approaches make less rigid assumptions. Popular nonparametric methods include the histogram, kernel estimation, nearest neighbour methods and restricted maximum likelihood methods, as can be found in [4], [6] and [3].

The kernel density estimator, also commonly referred to as the Parzen window estimator, [9], places a local Gaussian kernel on each data point of the training set. Then, the PDF is approximated by summing all the kernels, which are multiplied by a normalizing factor. Thus, this model can be viewed as a finite mixture model (see [7]) where the number of mixture components equals the number of points in the data sample. Parzen windows estimates are usually built using a

V. Kůrková et al. (Eds.): ICANN 2008, Part I, LNCS 5163, pp. 155–164, 2008.

'spherical Gaussian' with a single scalar variance parameter, which spreads the density mass equally along all input space directions and gives too much probability to irrelevant regions of space and too little along the principal directions of variance of the distribution. This drawback is partially solved in Manifold Parzen Windows algorithm [15], where a different covariance matrix is calculated for each component. The covariance matrix is estimated by considering a hard neighbourhood of each input sample. We propose in Section 2 to build soft clusters to share the information among neighbourhoods. This leads to filter the input noise by smoothing the estimated parameters. Furthermore, we use multivariate Student-t distributions, which have heavier tails than the Gaussians, in order to achieve robustness in the presence of outliers ([10], [12], [16]).

We present in section 3 the mixture of multivariate Student-t distributions which is learnt from the soft clusters. The asymptotical convergence of the proposed method is formally proven in Section 4. We show some experimental results in section 5, where our method produces more precise density estimations than the Manifold Parzen Windows and other approaches. Finally, Section 6 is devoted to conclusions.

2 The Smooth Parzen Windows Method

Let $\boldsymbol{x}$ be a D-dimensional real-valued random variable and $p()$ an arbitrary probability density function over $\boldsymbol{x}$ which is unknown and we want to estimate. The training set of the algorithm is formed by N observations of the random variable. For each training sample $\boldsymbol{x}_i$ we build a hard Q-neighbourhood H_i with the Q nearest neighbours of $\boldsymbol{x}_i$, including itself. Hence H_i is interpreted as a random event which happens iff the input belongs to that neighbourhood. The knowledge about the local structure of the distribution around $\boldsymbol{x}_i$ is obtained when we calculate the mean vector $\boldsymbol{\mu}$ and the correlation matrix $\boldsymbol{R}$:

$$\boldsymbol{\mu}(H_i) = E[\boldsymbol{x}|H_i] = \frac{1}{Q} \sum_{\boldsymbol{x}_j \in H_i} \boldsymbol{x}_j \tag{1}$$

$$\boldsymbol{R}(H_i) = E[\boldsymbol{x}\boldsymbol{x}^T|H_i] = \frac{1}{Q} \sum_{\boldsymbol{x}_j \in H_i} \boldsymbol{x}_j \boldsymbol{x}_j^T \tag{2}$$

Now we present a smoothing procedure to merge the information from different hard neighbourhoods. A soft cluster i is defined by a random event named S_i, which verifies when the input belongs to cluster i. Each hard neighbourhood H_j contributes to S_i with a normalized weight w_{ij}:

$$w_{ij} = P[H_j|S_i] \tag{3}$$

So, we have

$$\forall i \in \{1, 2, \ldots, M\}, \sum_{j=1}^{N} w_{ij} = 1 \tag{4}$$

where the number of soft clusters M may be different from the number of hard neighbourhoods N. We can infer the structure of the soft cluster by merging the information from the hard neighbourhoods:

$$\boldsymbol{\mu}(S_i) = E[\boldsymbol{x}|S_i] = \sum_j P[H_j|S_i]E[\boldsymbol{x}|H_j] = \sum_{j=1}^{N} w_{ij}\boldsymbol{\mu}(H_j) \tag{5}$$

$$\boldsymbol{R}(S_i) = E[\boldsymbol{x}\boldsymbol{x}^T|S_i] = \sum_j P[H_j|S_i]E[\boldsymbol{x}\boldsymbol{x}^T|H_j] = \sum_{j=1}^{N} w_{ij}\boldsymbol{R}(H_j) \tag{6}$$

In order to define a multivariate Student-t distribution we need the estimation of the covariance matrix C for each soft cluster:

$$\boldsymbol{C}(S_i) = E[(\boldsymbol{x} - \boldsymbol{\mu}(S_i))\,(\boldsymbol{x} - \boldsymbol{\mu}(S_i))^T\,|S_i] = \boldsymbol{R}(S_i) - \boldsymbol{\mu}(S_i)\boldsymbol{\mu}(S_i)^T \tag{7}$$

Finally, we need a method to determine the merging weights w_{ij}. We propose two approaches:

a) If $M = N$, we can perform the smoothing by replacing the 'hard' model at the data sample $\boldsymbol{x}_i$ by a weighted average of its neighbours ranked by their distance to $\boldsymbol{x}_i$. Here the model at $\boldsymbol{x}_i$ has the maximum weight, and their neighbours $\boldsymbol{x}_j$ have a weight which is a decreasing function of the distance from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$:

$$\boldsymbol{\omega}_{ij} = \exp\Big(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\boldsymbol{\psi}^2}\Big) \tag{8}$$

$$w_{ij} = \frac{\boldsymbol{\omega}_{ij}}{\sum_{k=1}^{N}\boldsymbol{\omega}_{ik}} \tag{9}$$

where $\boldsymbol{\psi}$ is a parameter to control the width of the smoothing. Please note that $\boldsymbol{\omega}_{ii} = 1$.

b) We may use the fuzzy c-means algorithm [1] to perform a soft clustering. This algorithm partitions the set of training data into M clusters so it minimizes the distance within the cluster. The objective function is:

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}^{\phi} d_{ij}^2 \tag{10}$$

where ϕ is the fuzzy exponent which determines the degree of fuzzyness, and d_{ij} is the distance between training sample $\boldsymbol{x}_j$ and the centroid of cluster i. The degrees of membership of training sample j to soft cluster i are obtained as m_{ij}, which can be regarded as the probability of training sample j belonging to cluster i. In this approach the weights w_{ij} of the local models that we merge to yield the model of cluster i are computed as follows:

$$w_{ij} = \frac{m_{ij}}{\sum_{i=1}^{N} m_{ik}} \tag{11}$$

3 Robust Density Model

Once we have the estimations of the mean vectors $\boldsymbol{\mu}(S_i)$ and covariance matrices $\boldsymbol{C}(S_i)$ for each soft cluster S_i, it is needed to obtain a multivariate Student-t distribution from them. First we define our probability model, which is a mixture of multivariate Student-t distributions. Then we discuss how to make it learn from the data.

3.1 Mixture Model

The proposed algorithm is designed to estimate an unknown density distribution $p()$ from which the N samples of the training dataset are generated. The generated estimator will be formed by a mixture of M multivariate Student-t distributions, one for each soft cluster:

$$\hat{p}(\boldsymbol{x}) = \frac{1}{M} \sum_{i=1}^{M} K_i(\boldsymbol{x}) \tag{12}$$

$$K_i(\boldsymbol{x}) = \frac{\Gamma(\frac{\gamma_i+D}{2}) |\boldsymbol{\Sigma}_i|^{-1/2}}{\left(\Gamma(\frac{1}{2})\right)^D \Gamma(\frac{\gamma_i}{2}) \gamma_i^{D/2}} \left(1 + \frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)}{\gamma_i}\right)^{\frac{\gamma_i+D}{2}} \tag{13}$$

where Γ is the gamma function, D is the dimension of the training samples and γ_i is the degrees of freedom parameter of the i-th Student-t kernel. We require $\gamma_i > 2$ in order that both the mean and the covariance matrix exist.

3.2 Model Learning

As known [16], when $\gamma_i > 1$ the mean of the i-th Student-t distribution exists and is $\boldsymbol{\mu}_i$. We estimate $\boldsymbol{\mu}_i$ by the mean of the i-th soft cluster. On the other hand, if $\gamma_i > 2$ then the covariance matrix exists and is given by $\gamma_i(\gamma_i-2)^{-2}\boldsymbol{\Sigma}_i$. We estimate $\boldsymbol{\Sigma}_i$ with the help of the covariance matrix of the i-th soft cluster. Hence we have:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}(S_i) \tag{14}$$

$$\boldsymbol{\Sigma}_i = \frac{(\gamma_i-2)^2}{\gamma_i} \boldsymbol{C}(S_i) \tag{15}$$

Finally we need to estimate the degrees of freedom parameter γ_i. Since there is no closed formula to obtain its value (see [10]), we follow a maximum likelihood approach here. We choose the value of γ_i which maximizes the log-likelihood of the training set:

$$L = \sum_{j=1}^{N} log\ \hat{p}(\boldsymbol{x}_j) = \sum_{j=1}^{N} log \left(\frac{1}{M} \sum_{i=1}^{M} K_i(\boldsymbol{x})\right) \tag{16}$$

We have only M unknowns to optimize, namely $\gamma_i, i = 1, \ldots, M$. So, the maximization of L with respect to the γ_i's is done by a standard method such as Levenberg-Marquardt. The constraint $\gamma_i > 2$ is enforced for all i (see for example [5]).

3.3 Summary

The training algorithm can be summarized as follows:

1. For each training sample, compute the mean vector $\boldsymbol{\mu}(H_i)$ and correlation matrix $\boldsymbol{R}(H_i)$ of its hard neighbourhood H_i with equations (1) and (2).
2. Estimate the merging weights w_{ij} either by the distance method (9) or the fuzzy c-means algorithm (11).
3. Compute the mean vectors $\boldsymbol{\mu}(S_i)$ and covariance matrices $\boldsymbol{C}(S_i)$ of each soft cluster S_i following (5) and (7).
4. Obtain the optimal values of the degrees-of-freedom parameters γ_i by maximizing the log-likelihood (16). Note that the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are not subject to optimization, because they are computed by equations (14) and (15), respectively.

4 Convergence Proof

In this section we prove that our estimator $\hat{p}()$ converges to the true density function $p()$ in the limit $N \to \infty$ and $M \to \infty$.

Lemma 1. *Every local Student-t kernel $K_i(\boldsymbol{x})$ tends to the D-dimensional Dirac delta function $\delta(\boldsymbol{x} - \boldsymbol{\mu}(S_i))$ as $N \to \infty$ and $M \to \infty$.*

Proof. In the limit $N \to \infty$ and $M \to \infty$ the clusters S_i reduce their volume to zero. This means that $\gamma_i^p \to 0$ for all i and p, where γ_i^p is the p-th eigenvalue of the Mahalanobis distance matrix Σ_i. Hence the kernels $K_i(\boldsymbol{x})$ are confined to a shrinking volume centered at $\boldsymbol{\mu}(S_i)$, because the variances in each direction are γ_i^p , but they continue to integrate to 1. So, we have that $K_i(\boldsymbol{x}) \to \delta(\boldsymbol{x} - \boldsymbol{\mu}(S_i))$. It should be noted that if we had allowed $\gamma_i \to 0$, the tails of the kernels could be so heavy that this property would have not hold, but this is not the case because we require $\gamma_i > 2$.

Theorem 1. *The expected value of the proposed estimation tends to the true probability density function as $N \to \infty$ and $M \to \infty$.*

Proof. The expectation is w.r.t. the underlying distribution of the training samples, which is the true probability density function $p()$:

$$E[\hat{p}(\boldsymbol{x})] = \frac{1}{M} \sum_{i=1}^{M} E[K_i(\boldsymbol{x})] \tag{17}$$

Since $K_i(\boldsymbol{x})$ are independent and identically distributed random variables we get

$$E[\hat{p}(\boldsymbol{x})] = E[K_i(\boldsymbol{x})] = \int p(\boldsymbol{y}) K_{\boldsymbol{y}}(\boldsymbol{x}) d\boldsymbol{y} \tag{18}$$

where $K_{\boldsymbol{y}}()$ is a multivariate Student-t centered in $\boldsymbol{y}$. Then, by Lemma 1, if $N \to \infty$ and $M \to \infty$ then $K_{\boldsymbol{y}}()$ shrinks to a Dirac delta:

$$E[\hat{p}(\boldsymbol{x})] = \int p(\boldsymbol{y}) \delta(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} \tag{19}$$

So, the expectation of the estimation converges to a convolution of the true density with the Dirac delta function. Then,

$$E[\hat{p}(\boldsymbol{x})] \rightarrow p(\boldsymbol{x}) \tag{20}$$

Theorem 2. *The variance of the proposed estimation tends to zero as* $N \rightarrow \infty$ *and* $M \rightarrow \infty$.

Proof. The variance is w.r.t. the underlying distribution of the training samples, which is the true probability density function $p()$:

$$var[\hat{p}(\boldsymbol{x})] = var\left[\frac{1}{M}\sum_{i=1}^{M} K_i(\boldsymbol{x})\right] \tag{21}$$

Since $K_i(\boldsymbol{x})$ are independent and identically distributed random variables we get

$$var[\hat{p}(\boldsymbol{x})] = var\Big[\frac{1}{M} K_i(\boldsymbol{x})\Big] \tag{22}$$

By the properties of variance and (18) we obtain

$$var[\hat{p}(\boldsymbol{x})] = \frac{1}{M}\left(E[(K_i(\boldsymbol{x}))^2] - E[K_i(\boldsymbol{x})]^2\right) = \frac{1}{M}\left(E[(K_i(\boldsymbol{x}))^2] - E[\hat{p}(\boldsymbol{x})]^2\right) \tag{23}$$

By definition of expectation

$$var[\hat{p}(\boldsymbol{x})] = \frac{1}{M}\left(\int p(\boldsymbol{y})\,(K_{\boldsymbol{y}}(\boldsymbol{x}))^2\,d\boldsymbol{y} - E[\hat{p}(\boldsymbol{x})]^2\right) \tag{24}$$

where again $K_{\boldsymbol{y}}()$ is a multivariate Student-t centered in $\boldsymbol{y}$. We can bound the integral of the above equation with the help of (18), and so we get

$$var[\hat{p}(\boldsymbol{x})] \leq \frac{\sup(N(\cdot))E[\hat{p}(\boldsymbol{x})]}{M} \rightarrow 0 \quad \text{as } N \rightarrow \infty \text{ and } M \rightarrow \infty \tag{25}$$

5 Experimental Results

This section shows some experiments we have designed in order to study the quality of the density estimation achieved by our method. We call it SmoothTDist when the distance weighting is used, and SmoothTFuzzy when we use fuzzy c-means. Vincent and Bengio's method is referred as MParzen, the original Parzen windows method (with isotropic Gaussian kernels) is called OParzen, and finally the Mixtures of Probabilistic PCA model of Tipping and Bishop [13] is called MPPCA. For this purpose the performance measure we have chosen is the average negative log likelihood

$$ANLL = -\frac{1}{T}\sum_{i=1}^{T} log\ \hat{p}(\boldsymbol{x}_i) \tag{26}$$

where $\hat{p}()$ is the estimator, and the test dataset is formed by T samples $\boldsymbol{x}_i$.

5.1 Experiment on 2D Artificial Data

We have considered two artificial 2D datasets. The first dataset consists of a training set of 100 points, a validation set of 100 points and a test set of 10000 points, which are generated from the following distribution of two dimensional (x, y) points:

$$x = 0.04t\sin(t) + \epsilon_x, \qquad y = 0.04t\cos(t) + \epsilon_y \tag{27}$$

where $t \sim U(3, 15)$, $\epsilon_x \sim N(0, 0.01)$, $\epsilon_y \sim N(0, 0.01)$, $U(a, b)$ is uniform in the interval (a, b) and $N(\mu, \sigma)$ is a normal density. The second dataset is a capital letter 'S'.

We have optimized separately all the parameters of the five competing models with disjoint training and validation sets. The performance of the optimized models has been computed by 10-fold cross-validation, and the results are shown in Table 1, with the best result marked in bold. It can be seen that our models outperform the other three in density distribution estimation (note that lower is better).

Figures 1 and 2 show density distribution plots corresponding to the five models. Darker areas represent zones with high density mass and lighter ones indicate the estimator has detected a low density area.

Table 1. Quantitative results on the artificial datasets (standard deviations in parentheses)

Method	ANLL on test set (espiral)	ANLL on test set (capital 'S')
SmoothTDist	**-1.0901 (1.1887)**	-1.9384 (1.2157)
SmoothTFuzzy	-1.0880 (1.3858)	**-2.1176 (1.0321)**
OParzen	1.0817 (1.3357)	-0.6929 (0.4003)
MParzen	-0.9505 (0.3301)	-1.0956 (0.0657)
MPPCA	0.2473 (0.0818)	-0.1751 (0.6204)

We can see in the plots that our models have less density holes (light areas) and less 'bumpiness'. This means that our model represents more accurately the true distribution, which has no holes and is completely smooth. We can see that the quantitative ANLL results agree with the plots, because the lowest values of ANLL match the best-looking plots. So, our model outperforms clearly the other three considered approaches.

5.2 Density Estimation Experiment

A density estimation experiment has been designed, where we have chosen nine datasets from the UCI Repository of Machine Learning Databases [8]. As before, we have optimized all the parameters of the five competing models with disjoint training and validation sets. The parameters for the density estimator of each dataset have been optimized separately. Table 2 shows the results of the 10-fold

Fig. 1. Density estimation for the 2D espiral dataset. From left to right and from top to bottom: SmoothDist, SmoothFuzzy, OParzen, MParzen and MPPCA.

Table 2. ANLL on test set (lower is better). The standard deviations are shown in parentheses).

Database	SmoothTDist	SmoothTFuzzy	OrigParzen	ManifParzen	MPPCA
BrCancerWis	**-48.17 (2.48)**	9.54 (1.68)	9.35 (1.51)	10.22 (1.50)	13.19 (0.99)
Glass	**-197.18 (72.35)**	-8.22 (4.90)	-1.75 (5.87)	-1.04 (5.74)	-3.98 (6.24)
Ionosphere	**-21.92 (9.73)**	-8.01 (8.58)	-15.68 (3.24)	-14.83 (3.21)	-1.21 (5.15)
Liver	**17.53 (3.56)**	21.41 (0.63)	20.27 (3.45)	21.09 (3.31)	22.25 (0.83)
Pima	**-304.37 (36.48)**	27.95 (0.31)	32.21 (3.91)	32.85 (3.56)	29.87 (0.89)
Segmentation	65.96 (3.08)	**-71.22 (10.95)**	51.18 (3.93)	51.97 (3.86)	18.77 (4.29)
TAE	**-2.57 (11.70)**	7.59 (1.99)	8.17 (0.55)	9.09 (0.54)	11.98 (0.12)
Wine	**-59.84 (1.58)**	62.78 (18.53)	29.16 (4.55)	31.14 (7.01)	18.98 (0.63)
Yeast	**-319.02 (22.86)**	-9.50 (0.34)	-17.85 (0.33)	-16.96 (0.31)	-11.99 (0.31)

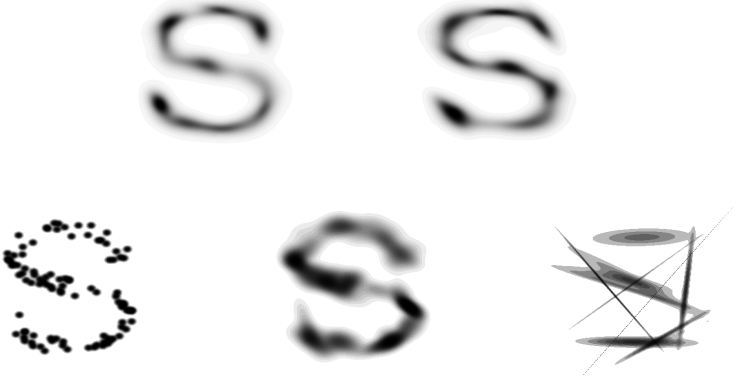

Fig. 2. Density estimation for the 2D capital 'S' dataset. From left to right and from top to bottom: SmoothDist, SmoothFuzzy, OParzen, MParzen and MPPCA.

cross-validation, with the winning models in bold. Our two proposals show a superior performance.

We have used the T-test to check the statistical significance of the difference between the two best performing models for each database. We have considered that the difference is statistically significant if we have less than 0.05 probability that the difference between the means is caused by chance. It has been found that the difference is statistically significant for all the considered databases.

6 Conclusions

We have presented a probability density estimation model. It is based in the Parzen window approach. Our proposal builds local models for a hard neighbourhood of each training sample. Then soft clusters are obtained by merging these local models, and local multivariate Student-t kernels are introduced. This allows our method to represent input distributions more faithfully than three well-known density estimation models. Computational results show the superior performance of our method.

Acknowledgements

The authors acknowledge support from CICYT (Spain) through grant TIN2005-02984 (including FEDER funds). This work is also partially supported by Junta de Andalucía (Spain) through grant TIC-01615.

References

1. Bezdek, J.C.: Numerical taxonomy with fuzzysets. J. Math. Biol. 1, 57–71 (1974)
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Hjort, N.L., Jones, M.C.: Locally Parametric Nonparametric Density Estimation. Annals of Statistics 24(4), 1619–1647 (1996)
4. Izenman, A.J.: Recent developments in nonparametric density estimation. Journal of the American Statistical Association 86(413), 205–224 (1991)
5. Kanzow, C., Yamashita, N., Fukushima, M.: Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties. Journal of Computational and Applied Mathematics 172, 375–397 (2004)
6. Lejeune, M., Sarda, P.: Smooth estimators of distribution and density functions. Computational Statistics and Data Analysis 14, 457–471 (1992)
7. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, Chichester (2000)
8. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
9. Parzen, E.: On the Estimation of a Probability Density Function and Mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)

10. Shoham, S.: Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions. Pattern Recognition 35, 1127–1142 (2002)
11. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York (1986)
12. Svensén, M., Bishop, C.M.: Robust Bayesian mixture modeling. Neurocomputing 64, 235–252 (2005)
13. Tipping, M.E., Bishop, C.M.: Mixtures of Probabilistic Principal Components Analyzers. Neural Computation 11, 443–482 (1999)
14. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
15. Vincent, P., Bengio, Y.: Manifold Parzen Windows. Advances in Neural Information Processing Systems 15, 825–832 (2003)
16. Wang, H., Zhang, Q., Luo, B., Wei, S.: Robust mixture modelling using multivariate t-distribution with missing information. Pattern Recognition Letters 25, 701–710 (2004)