Local Selection of Model Parameters in Probability Density Function Estimation

Ezequiel López-Rubio[1], Juan Miguel Ortiz-de-Lazcano-Lobato[1],
Domingo López-Rodríguez[2], Enrique Mérida-Casermeiro[2],
and María del Carmen Vargas-González[1]

[1] Department of Computer Science and Artificial Intelligence, University of Málaga,
Campus Teatinos, s/n, 29071 Málaga, Spain
{ezeqlr, jmortiz}@lcc.uma.es
http://www.lcc.uma.es

[2] Department of Applied Mathematics, University of Málaga, Campus Teatinos, s/n,
29071 Málaga, Spain
{dlopez, merida}@ctima.uma.es
http://www.satd.uma.es/matap

Abstract. Here we present a novel probability density estimation model. The classical Parzen window approach builds a spherical Gaussian density around every input sample. Our proposal selects a Gaussian specifically tuned for each sample, with an automated estimation of the local intrinsic dimensionality of the embedded manifold and the local noise variance. This leads to outperform other proposals where local parameter selection is not allowed, like the manifold Parzen windows.

1 Introduction

The estimation of the unknown probability density function (PDF) of a continuous distribution from a set of data points 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 [1], [2] and [3]).

Parametric approaches make assumptions about the unknown distribution. They consider, a priori, a particular functional form for the PDF and reduce the problem to the estimation of the required functional parameters. On the other hand, nonparametric methods make less rigid assumptions. Thus they are more flexible and they usually provide better results. Popular nonparametric methods include the histogram, kernel estimation, nearest neighbor methods and restricted maximum likelihood methods, as can be found in [4], [5], [6] and [7].

The kernel density estimator, also commonly referred as the Parzen window estimator, [9], places a 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 [8]) where the number of mixture components will equal the number of points in the data sample. The parameter which defines the shape of those components, i.e. the covariance of the Gaussian kernel, is the same for all of them and the estimation of the arbitrary distribution is, therefore, penalized because of the poor adaptation to local structures of the

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 292 – 301, 2006.

data. Besides, most of the time, Parzen windows estimates are built using a "spherical Gaussian" with a single scalar variance parameter σ^2, 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, [10], where a different covariance matrix is calculated for each component. On the other hand, this model considers that the true density mass of the dataset is concentrated in a non-linear lower dimensional manifold embedded in the higher dimensional input space. In this sense, only information about directions of the lower dimensional manifold will be preserved in order to reduce the memory cost of the model. There is also a unique regularization parameter which is used to represent the variance in the discarded directions of the components, as it will be explained more detailed in section 2.

We present, in section 3, a model that selects automatically the adequate values for some parameters of the Manifold Parzen Windows model. Our method chooses the right dimensionality of the manifold according to a quality criterion specified by the user, which is the percentage of neighbourhood variance we want to be retained in each component. In a similar way, the regularization variance parameter will be selected by the method itself without the aid of human knowledge. Therefore the time invested in tuning the parameters to obtain good density estimations will be diminished. We show some experimental results, in section 4, where the selection achieved by our method produces more precise estimations that the Manifold Parzen Windows one.

2 The Manifold Parzen Windows Method

Let X be an n-dimensional random variable and $p_X()$ an arbitrary probability density function over X which is unknown and we want to estimate. The training set of the algorithm is formed by l samples of the random variable and the density estimator has the form of a mixture of Gaussians, whose covariances C_i may be identical or not:

$$\hat{p}_{mp}(x) = \frac{1}{l}\sum_{i=1}^{l} N_{x_i,C_i}(x)\,. \tag{1}$$

with $N_{\mu,C}(x)$ the multivariate Gaussian density:

$$N_{\mu,C}(x) = \frac{1}{\sqrt{(2\pi)^n |C|}} e^{-\frac{1}{2}(x-\mu)' C^{-1}(x-\mu)} \tag{2}$$

where μ is the mean vector, C is the covariance matrix and $|C|$ the determinant of C.

The density mass is expected to concentrate close to an underlying non-linear lower dimensional manifold and, thus, the Gaussians would be "pancakes" aligned with the plane locally tangent to that manifold. Without prior knowledge about the distribution $p_X()$ *t*he information about the tangent plane is provided by the samples of the training set. Thus the principal directions of the samples in the neighbourhood of each sample x_i will be computed. The local knowledge about the principal directions will be obtained when we calculate the weighted covariance matrix C_{κ_i} for each sample:

$$C_{\kappa_i} = \frac{\sum_{j=1..l,j\neq i} \kappa(x_j;x_i)(x_j - x_i)'(x_j - x_i)}{\sum_{j=1..l,j\neq i} \kappa(x_j;x_i)} \tag{3}$$

where *$(x_j$-$x_i)$' $(x_j$-$x_i)$* denotes the outer product and *$K(x, x_i)$* is a neighbourhood kernel centered in x_i which will associate an influence weight to any point x in the vicinity of x_i.

Vincent and Bengio propose in [10] the utilization of a hard k-neighbourhood which assigns a weight of 1 to any point no further than the k-th nearest neighbour of the sample x_i among the training set, according to some metric such as the Euclidean distance in input space, and setting the weight to 0 to those points further than the k-neighbour. This approach usually involves C_{κ_i} to be ill-conditioned so it is slightly modified by adding a small isotropic Gaussian noise of variance σ^2

$$C_i = C_{\kappa_i} + \sigma^2 I \tag{4}$$

When we deal with high dimensional training datasets it would be prohibitive in computation time and storage to keep and use each full covariance matrix C_i. Therefore, a compacted representation of them is preserved, storing only the eigenvectors associated with the first d largest eigenvalues of them, where d is chosen by the user of the algorithm and is fixed for each covariance matrix. The eigenvectors related to the largest eigenvalues of the covariance matrix correspond to the principal directions of the local neighbourhood, i.e. the high variance local directions of the supposed underlying d-dimensional manifold. The last few eigenvalues and eigenvectors are but noise directions with a small variance and a same low noise level, which is also the same σ^2 it was used before, is employed for them.

Once the model has been trained any sample of the distribution may be tested. The probability density estimation for the sample will be computed by the average of the probability density provided by the l local Gaussians as was mentioned in (**1**).

3 Dynamic Parameter Selection in Manifold Parzen Windows Algorithm

We extend the training of Vincent and Bengio's method [10], by providing a more automatic way to estimate density functions.

First we incorporate the capacity of estimating the intrinsic dimensionality, i.e. the needed number of principal directions d, of the underlying manifold for each neighbourhood. The cause is that we use a qualitative parameter, α, which represents the explained variance by the principal directions of the local manifold. Then, the method will be able to choose by itself the minimum number of eigenvectors which retain a particular amount of the variance presented in the vicinity of each training sample. This method has been employed in [11] and [12] with good results.

A second level of automated adaptation to the data will be added by means of a parameter γ. This parameter will enable the method to select the right noise level for discarded directions. So, a better adaptation of the model to the unknown distribution will be achieved.

3.1 The Explained Variance Method

The explained variance method considers a variable number, D_i, of eigenvalues and their corresponding eigenvectors to be kept which is computed independently for each sample x_i. This number reflects the intrinsic dimensionality of the lower dimensional manifold where the data lies for the neighbourhood of x_i. Through the training process the method ensures that a minimum amount of variance is conserved in order to satisfy the level of accuracy, $\alpha \in [0,1]$, chosen by the user. The number of principal directions which are preserved is set consequently to the minimum value which allows us to reach that level at least.

The most precise estimation of the data in the neighbourhood of a sample can be achieved if we conserve the full covariance matrix, i. e. we keep information about every direction, because it will be more likely to discover the right dimensionality of the underlying manifold. On the other hand, the worst estimation will be obtained when all the directions are ignored and the sample x_i is the only statistical information which is kept, i.e. when we lose all the variance relative to the directions of the embedded manifold. Thus, the lost variance when no directions are kept, V_0, can be defined as:

$$V_0 = \sum_{p=1}^{D} \lambda_i^p \tag{5}$$

with λ_i^p the p eigenvalue of the covariance matrix C_{κ_i}, which are supposed to be sorted in decreasing order, and D the dimension of the training samples.

In any other situation the discarded variance, V_Z, depends on the number, Z, of principal directions conserved:

$$V_Z = \sum_{p=Z+1}^{D} \lambda_i^p \tag{6}$$

Our goal is to obtain the most compressed representation of the covariance C_{κ_i}, while the model maintains a minimum level of quality α. with respect to the maximum accuracy the method can achieve. Let $V_0 - V_Z$ be the amount of error (we must remember that the more variance is lost the less precise the estimation will be) eliminated when we conserve information about the Z principal directions. Then

$$D_i = \min\{Z \in \{0,1,\ldots,D\} \mid V_0 - V_Z \geq \alpha \ V_0\} \tag{7}$$

Substitution of (**5**) and (**6**) into (**7**) yields

$$D_i = \min\left\{Z \in \{0,1,\ldots,D\} \mid \sum_{p=1}^{D} \lambda_i^p - \sum_{p=Z+1}^{D} \lambda_i^p \geq \alpha \ \sum_{p=1}^{D} \lambda_i^p\right\} \tag{8}$$

It is well known that the sum of variances of a dataset equals the trace of the covariance matrix for this dataset, therefore equation **(8)** can be simplified as follows:

$$D_i = \min\left\{Z \in \{0,1,\ldots,D\} \mid \sum_{p=1}^{Z} \lambda_i^p \geq \alpha \ \ trace(C_{\kappa_i})\right\} \tag{9}$$

The quotient between λ_i^p and $trace(C_{K_i})$ is the amount of variance explained by the *p*th principal direction of the estimated manifold. Thus, if we sum these quotients for all the retained directions, we can see the parameter α as the amount of variance which we want to be retained in each neighbourhood. Hence, we select D$_i$ so that the amount of variance explained by the directions associated to the D$_i$ largest eigenvalues is at least α.

3.2 The Qualitative Parameter *γ*

With the variance explained parameter our aim is to add the model the ability to adapt by itself to the local properties of the distribution. Thus, it saves memory space which is not required, i. e. only the necessary information of the covariance of each neighbourhood will be stored.

The same idea was applied to deal with the parameter σ^2, which controls the width of the Gaussians in Manifold Parzen Windows method. In order to take into consideration the local structure and to obtain better estimators, the noise variance for each neighbourhood is determined by

$$\sigma_i^2 = \gamma \cdot \lambda_i^{D_i} \tag{10}$$

where $\gamma \in [0,1]$ and $\lambda_i^{D_i}$ is the last of the preserved eigenvalues, i.e. the smallest of the first D_i largest eigenvalues.

As can be noticed there is a close relation between α and γ. If we use a value for α near to 0 then we likely retain only the first eigenvalue, which is associated to the first principal direction of the data. Therefore, it encompasses a great percentage of the total variance of the distribution. This means that $\lambda_i^{D_i}$ will be large and the noise variance will be set to a relatively large value. This implies that the Gaussian for the i neighbourhood will be widened along the discarded directions. In the opposite case, if a value near to 1 is assigned to α, then we store nearly all the eigenvalues and eigenvectors of the covariance matrix. The last retained eigenvalue will be very small and independently of the value of γ the noise variance will be set to a value near 0. This is in consonance with the fact that if we conserve all the information about the directions of change then there is not noise variance, because there is not any discarded dimension. In subsection 4.2, we present some plots where the fact just commented can be observed.

3.3 Parzen Manifold Windows with Qualitative Parameters

The proposed algorithm is designed to estimate an unknown density distribution $p_X()$ which the l samples of the training dataset are generated from. The generated estimator will be formed by a mixture of l Gaussians, one for each sample. Their shapes are adapted to the adequate local structure of the neighbourhoods through the training process and rely on the user specified qualitative parameters. The user chooses both the quality of the estimation, expressed by the explained variance parameter α; and γ, which means the width of the Gaussians in the discarded directions relative to the width in the last conserved direction.

The training method can be summarized as follows:

1. Take the training sample x_i with $i \in \{1,2,...,l\}$. Initially the first sample x_1 is selected.
2. Compute the covariance matrix C_{κ_i} following (**3**) where only the k nearest neighbours x_j of x_i are considered.
3. Extract the eigenvalues and eigenvectors from C_{κ_i} and estimate the dimensionality of the underlying manifold D_i, by means of (**9**)
4. Use (**10**) to calculate σ_i^2, the noise variance for the discarded directions.
5. Store the local model, i. e., the first D_i eigenvectors and eigenvalues, the local noise level σ_i^2, the l samples and the number of neighbours k.
6. Go to step 1, and continue the training process for the next sample. If there are not more samples to process, the algorithm finishes.

4 Experimental Results

This section shows some experiments we have designed in order to compare quality of density estimation presented by our method, we term MparzenQuality throughout this whole section and by the Vincent and Bengio's one, which will be referred as MParzen. For this purpose the measure used was the average negative log likelihood

$$ANLL = -\frac{1}{m}\sum_{i=1}^{m} \log \hat{p}(x_i) \tag{11}$$

where $\hat{p}(x)$ is the estimator, and the training dataset is formed by m examples x_i.

4.1 Experiment on 2D Artificial Data

A training set of 300 points, a validation set of 300 points and a test set of 10000 points were generated from the following distribution of two dimensional *(x,y)* points:

$$x = 0.04t\sin(t) + \varepsilon_x, \quad y = 0.04t\cos(t) + \varepsilon_y$$

where $t \sim U(3,15), \varepsilon_x \sim N(0,0.01), \varepsilon_y \sim N(0,0.01), U(a,b)$ is uniform in the interval *(a,b)* and $N(\mu,\sigma)$ is a normal density.

We trained a MParzenQuality model with explained variance 0.1 and 0.9 on the training set. The parameters k and γ were tuned to achieve the best performance on the validation test. On the other hand, MParzen with d = 1 and d = 2 was trained and the rest of its parameters were also tuned.

Quantitative comparative results of the two models are reported en Table 1, where it can be seen that our model outperforms the previous one in density distribution estimation. Figure 1 shows the results obtained when we applied the models on the test set. Darker areas represent zones with high density mass and lighter ones indicates the estimator has detected a low density area.

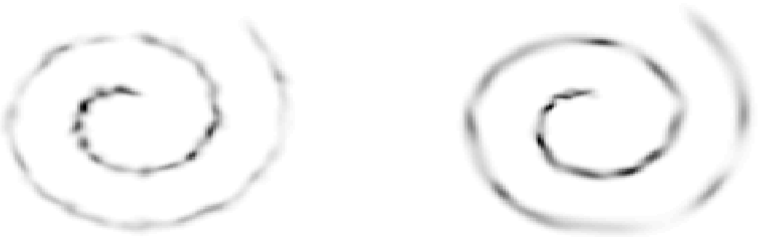

Fig. 1. Density estimation for the 2D artificial dataset, MParzen model (left) and MParzenQuality (right)

We can see in the plots that our model has 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. So, our model outperforms clearly the MParzen approach.

Table 1. Comparative results on the espiral dataset

Algorithm	Parameters used	ANLL on test-set
MParzen	$d = 1,\ k = 11,\ \sigma^2 = 0.009$	-1.466
MParzen	$d = 2,\ k = 10,\ \sigma^2 = 0.00001$	-1.419
MParzenQuality	$\alpha = 0.1,\ k = 10,\ \gamma = 0.1$	-2.204
MParzenQuality	$\alpha = 0.9,\ k = 10,\ \gamma = 0.1$	-2.116

4.2 Density Estimation on Astronomical Data

The dataset comes from the VizieR service [13], which is an information system for astronomical data. In particular, we have selected the Table 6 of the Complete near-infrared and optical photometric CDFS Catalog from Las Campanas Infrared Survey [14]. We have extracted 22 numerical features from 10,000 stars. Hence, we have 10,000 sample vectors. These data have been normalized in order to cope with the variability and the very heterogeneous scaling of the original data. This dataset has been split randomly in a training set (10% of the dataset), validation set (10%) and test set (the remaining 80%).

We have carried out simulation runs for MParzen with the number of dimensions retained from 1 to 6. For each of those values we have tried the following noise levels: σ^2 = 0.09, 0.1, 0.11, 0.13, 0.15, 0.17, 0.19, 0.3 and 0.5 (values near 0.11, which generates good performance). The simulations with the MParzenQuality model have been carried out with the following parameter values: α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9; γ = 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.1, 0.2, 0.3, 0.4 and 0.5. In both models we have tried the following numbers of neighbours: 10, 15 and 20.

In Figure 2 the ANLL of the models is plotted versus the number of retained principal directions. For each value of d or α, only the best performing combination of the rest of the parameters is shown in the plot. Please note that for the MParzenQuality model the average of the principal directions retained is averaged over all the samples, so fractional values of dimensionality are shown. It can be observed that our proposal is clearly superior in all conditions.

It should also be noted that with the MParzen model we have detected serious problems with the outliers. The original VizieR dataset is fairly uniform, but there are 3 outliers. These data samples caused the MParzen model to completely fail the ANLL performance test, because the model assigned a zero probability to these samples, up to double precision calculations, yielding a plus infinite ANLL. Our MParzenQuality model did not suffer from this problem, showing a better probability density allocation. These outliers have been removed in order to perform the tests corresponding to Figure 2.

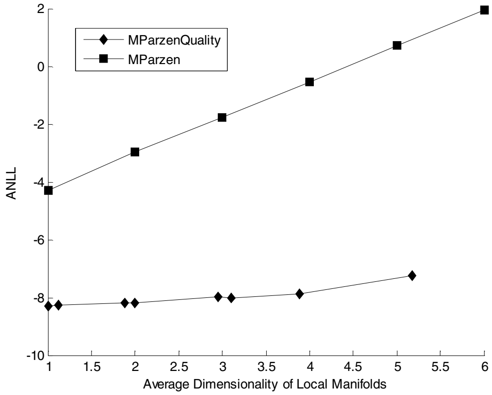

Fig. 2. Results with the VizieR astronomical dataset

A set of curves which represents the contribution of the qualitative parameters when we employ 15 neighbours for each data sample is presented in Figure 3.

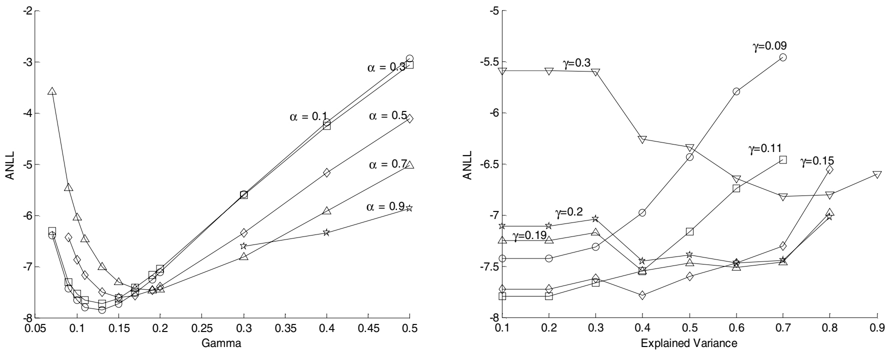

Fig. 3. Relationship of the qualitative parameters and the quality of the results

Similar conclusions may be extracted for both plots. First when the explained variance is fixed to a small percentage, then smaller values for parameter γ produces more adequate width for the "pancakes" and, thus, better results (see the minimum values for the curves of the left plot). On the other hand, if parameter α is greater than 0.5 then the last preserved eigenvalue is small, and the width of the Gaussians will be too narrow if the value assigned to γ is not chosen high enough. A compromise value γ is 0.2, which maintains an average performance, although it does not achieve the best results.

5 Conclusions

We have presented a probability density estimation model. It is based in the Parzen window approach. Our proposal builds a local Gaussian density by selecting independently for each training sample the best number of retained dimensions and the best estimation of noise variance. This allows our method to represent input distributions more faithfully than the manifold Parzen window model, which is an improvement of the original Parzen window method. Computational results show the superior performance of our method, and its robustness against outliers in the test set.

References

1. Bishop, C., Neural Networks for Pattern Recognition, Oxford University Press (1995)
2. Silverman, B., Density Estimation for Statistics and Data Analysis, Chapman and Hall, New York (1986)
3. Vapnik, V. N., Statistical Learning Theory, John Wiley & Sons, New York (1998)
4. Izenman, A. J., Recent developments in nonparametric density estimation. Journal of the American Statistical Association, 86(413) (1991) 205-224
5. Lejeune, M., Sarda, P., Smooth estimators of distribution and density functions. Computational Statistics & Data Analysis, 14 (1992) 457-471.
6. Hjort, N.L., Jones, M.C., Locally Parametric Nonparametric Density Estimation, Annals of Statistics, 24, 4 (1996) 1619-1647
7. Hastie, T., Loader, C., Local regression: Automatic kernel carpentry, Statistical Science, 8 (1993) 120-143
8. McLachlan, G., Peel, D., Finite Mixture Models, Wiley, 2000)
9. Parzen, E., On the Estimation of a Probability Density Function and Mode, Annals of Mathematical Statistics, 33 (1962) 1065-1076
10. Vincent, P., Bengio, Y., Manifold Parzen Windows, Advances in Neural Information Processing Systems, 15 (2003) 825-832
11. López-Rubio, E., Ortiz-de-Lazcano-Lobato, J. M., Vargas-González, M. C., López-Rubio, J. M., Dynamic Selection of Model Parameters in Principal Components Analysis Neural Networks, Proceedings fo the 16th European Conference on Artificial Intelligence (ECAI 2004) 618-622
12. López-Rubio, E., Ortiz-de-Lazcano-Lobato, J. M., Vargas-González, M. C., Competitive Networks of Probabilistic Principal Components Analysis Neurons, 9th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2005) 141-146

13. VizieR service [online], Available at: http://vizier.cfa.harvard.edu/viz-bin/VizieR/ (March 29, 2004)
14. Chen, H.-W., et al., Early-type galaxy progenitors beyond z=1, *Astrophysical Journal, 560*, 2001, L131