

# Comparison of Three Nonparametric Density Estimation Techniques Using Bayes' Classifiers Applied to Microarray Data Analysis

Christopher A. Peters  
Computational Science Research Center  
San Diego State University  
San Diego, CA 92182  
cpeters@sciences.sdsu.edu

Faramarz Valafar  
Department of Computer Science  
San Diego State University  
San Diego, CA 92182  
faramarz@sciences.sdsu.edu

**Abstract** *In this study, we attempt to distinguish between acute myeloid leukemia (AML) and acute lymphoid leukemia (ALL) using microarray gene expression data. Bayes' classification is used with three different density estimation techniques: Parzen,  $k$  nearest neighbors ( $k$ -NN), and a new hybrid method, called  $k$ -neighborhood Parzen ( $k$ -NP), that combines properties of the other two. The classifiers are applied both to single genes and pairs of genes. The highest testing accuracy of the three classifiers were within 3% of one another in both the one-gene and two-gene cases. Our overall best classifier used  $k$ -NN with two genes, and achieved a testing accuracy of 94%. A comparison with other published results shows that our methods performed very well, especially considering the low number of genes used.*

**Keywords:** Parzen density estimation,  $k$  nearest neighbors, Bayes' classifier, microarray, leukemia, cancer

## 1 Introduction

Sequencing the human genome [1] provided a tremendous amount of new information for biological and medical research. However, identifying all of the genes is not enough to solve most problems. The function of genes and their relationship to one another must be determined in order to make this knowledge applicable to medical and biological problems. Knowledge of gene function is useful in many areas of research, such as in understanding diseases, drug effects, mutations, development, and evolution. DNA microarrays, also known as DNA chips, allow researchers to monitor the expression levels of thousands of genes at once, making it far more practical to study genetic cause-effect rela-

tionships. On a DNA microarray, there are thousands of spots, each less than 200  $\mu\text{m}$  in diameter [2]. Each spot represents a gene in the DNA of the organism being studied. Before the microarray experiment, the sample DNA is labelled with fluorescent dye in one of two ways depending on the type of study. In some studies, pairs of samples are compared, such as healthy and diseased tissue samples from the same patient, or samples before and after a drug is administered. In this case, the two samples are stained with different colors so that fewer chips are necessary. For studies where the samples do not naturally fall into pairs, only one color is used. After labelling is complete, the sample DNA is hybridized to the microarrays, which are then scanned for the wavelength(s) emitted by the dye(s). Following the scan, image analysis is then used to determine numerical values for the expression levels of the genes [3]. Once the experimental data has been converted into numerical form, it can either be analyzed using classical pattern recognition techniques for tasks such as class discovery and class prediction, or to find genes relevant to a specific biological or medical problem. Due to the novelty and practical value of microarray data, an enormous body of research has been devoted to these problems in recent years [4]. Methods applied to microarray data include clustering [5, 6], support vector machines [5, 7, 8], and artificial neural networks [9, 10, 11, 12, 13]. In this study, we apply nonparametric density estimation using Bayes' classification (a classical pattern recognition technique) to a set of microarray data collected from leukemia patients.

The data used in this study was introduced by Golub *et al.* in 1999 [10]. The data is from two groups of patients, one with acute myeloid leukemia

(AML) and the other with acute lymphoid leukemia (ALL). Tumors from the two types of cancer have a similar appearance, but respond very differently to treatment [14, 15]. All the samples were taken from patients at the time they were diagnosed with either ALL or AML. There is a training set of 38 patients and an independent set of 34 patients for testing. Expression levels of 7,129 genes are given for each patient. This data set has become somewhat of a standard, and many authors have used it to test new techniques. Our results are compared to those of similar studies in section 4.3.

This paper is organized as follows. Section 2 is a discussion of three nonparametric density estimation techniques. Section 3 explains how an estimated density function is used to build a Bayes' classifier. Computational details for the study are found in section 4. Numerical experiments involving the leukemia data set appear in section 5, along with comparison to other studies. Finally, section 6 contains conclusions and plans for future work.

## 2 Kernel-based density estimation

The aim of density estimation is to approximate the true probability density function of a random phenomenon using a set of sampled data. There are two main types of density estimation: parametric and nonparametric. The distinction between the two is that parametric methods assume prior knowledge about the distribution of the data (typically the type of distribution, e.g. uniform, normal, etc.), while nonparametric methods do not. The methods used in this study are all kernel-based methods, one family of nonparametric density estimators.

In kernel-based density estimation, a kernel function is generated around each data point  $x_i$  in the training set. An estimate of the density function,  $\hat{p}(x)$ , is found by adding these kernel functions together and scaling the sum:

$$\hat{p}(x) = \frac{1}{V} \sum_{i=1}^n K_i(x), \quad (1)$$

where  $K_i(x)$  is the kernel function around  $x_i$  and  $V$  is a scaling factor so that the area under  $\hat{p}(x)$  is 1. The choice of kernel functions is left to the researcher. In this study, we used the commonly chosen Gaussian kernels. The equation for a Gaus-

sian kernel is as follows:

$$G(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \quad (2)$$

where  $d$  is the dimensionality of the data,  $\mu$  is the mean or center of the kernel, and  $\Sigma$  is the covariance matrix.

Since the Golub leukemia data set is divided into two patient classes (ALL and AML), a density function must be estimated for each class in order to build a Bayes' classifier. This is explained more thoroughly in section 3. The kernels of two common methods of density function estimation, as well as that of  $k$ -neighborhood Parzen ( $k$ -NP), a new hybrid method which combines aspects of the other two, are discussed below.

### 2.1 Parzen density estimation

For Parzen density estimation [16], the same fixed  $\Sigma$  is used for all the kernel functions. Gaussian kernels centered at each data point were used in this study, in the form  $K_i(x) = G(x; x_i, \Sigma)$ .

### 2.2 $k$ nearest neighbors density estimation

Rather than a fixed kernel shape,  $k$ -NN density estimation [17] uses a different  $\Sigma$  for each data point in an attempt to better reflect the local distribution. Building a kernel function begins with finding the  $k$ -neighborhood of each data point in the training set. The  $k$ -neighborhood consists of the point itself and its  $k-1$  nearest neighbors. Then a kernel function is fit to the  $k$ -neighborhood. In this study, we use two different Gaussian kernels for  $k-NN$  density estimation. The first kernel type ( $k$ -NN1) is centered at  $x_i$  and given by  $K_i(x) = G(x; x_i, \Sigma_i^{(1)})$ , where

$$\begin{aligned} \Sigma_i^{(1)} &= E((X - x_i)(X - x_i)^T) \\ &= \frac{1}{n} \sum_{j=1}^k (x_j x_j^T) - \mu_i \mu_i^T - x_i \mu_i^T + x_i x_i^T. \end{aligned} \quad (3)$$

The summation in equation (3) is over the points in the  $k$ -neighborhood. The second type of kernel ( $k$ -NN2) is centered at  $\mu_i$ , the mean of the  $k$ -neighborhood of  $x_i$ , and is given by  $K_i(x) = G(x; \mu_i, \Sigma_i^{(1)})$ , where

$$\begin{aligned} \Sigma_i^{(2)} &= E((X - \mu_i)(X - \mu_i)^T) \\ &= \frac{1}{n} \sum_{j=1}^k (x_j x_j^T) - \mu_i \mu_i^T, \end{aligned} \quad (4)$$

which sums over the points in the  $k$ -neighborhood.

Finally, the kernel function is assigned to the class that the majority of the  $k$ -neighborhood belongs to. It is important to note that there could be a tie for the majority class if there are two classes and  $k$  is even. Odd values of  $k$  were always used in this study to avoid resolving ties.

### 2.3 $k$ -neighborhood Parzen density estimation

Depending on the particular data set, either the  $k$ -NN or Parzen technique may work better than the other. For example, in regions where the data is sparse,  $k$ -NN can give a smoother estimate of the distribution, which may or may not be more representative of the true distribution. Our hybrid  $k$ -NP estimation technique was developed to combine the properties of the two methods in order to have a method that performs well in a wider variety of situations.

In the  $k$ -NP technique, the  $k$ -neighborhood of each data point is found as in the  $k$ -NN technique. Instead of fitting a single kernel function to each  $k$ -neighborhood, a Parzen density function is built for the  $k$ -neighborhood. As in the case of Parzen density estimation, the Gaussian kernels for the points in the  $k$ -neighborhood all use the same fixed  $\Sigma$ . The Parzen density function for each neighborhood is given by

$$K_i(x) = \frac{1}{k} \sum_{j=1}^k G(x; x_j, \Sigma). \quad (5)$$

Like the  $k$ -NN method, the kernel function is assigned to the majority class of the  $k$ -neighborhood.

## 3 Bayes' classifier

Hypothesis-based classification compares the probabilities of different hypotheses in order to make a classification. Bayes' classification [16] is one such method. A Bayes' classifier assigns a data point to the class with the highest probability of membership for that data point. The classification rule for the two-class case is as follows:

$$x \in \begin{cases} C_1, & \text{if } P(C_1|x) > P(C_2|x) \\ C_2, & \text{otherwise} \end{cases}, \quad (6)$$

where  $C_1$  and  $C_2$  are the two classes, and  $x$  is the observed data. To compute the posterior probabilities  $P(C_i|x)$ , Bayes' theorem is used:

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)}, \quad (7)$$

where  $P(C_i)$  is the a priori unconditional probability that a data point is in class  $i$ ,  $p(x|C_i)$  is the probability for  $x$  to be observed given the assumption that the data is sampled from class  $i$ , and  $p(x)$  is the unconditional probability of observing  $x$ . Combining Bayes' theorem with the classification rule gives:

$$\begin{aligned} P(C_1|x) > P(C_2|x) \\ \iff \frac{p(x|C_1)P(C_1)}{p(x)} > \frac{p(x|C_2)P(C_2)}{p(x)} \quad (8) \\ \iff p(x|C_1)P(C_1) > p(x|C_2)P(C_2), \end{aligned}$$

as long as  $p(x)$  is never zero in the domain [16]. This assumption is reasonable for most continuous problem spaces.

## 4 Computational methods

All computations were done with MATLAB or C code generated by the MATLAB compiler. For each gene (or pair of genes), the best choice of parameters and the corresponding accuracy on the training set were recorded. Then all classifiers with an accuracy of 95% or higher on the training set were tested on the 34 independent testing samples.

The Euclidean distance metric was used to find  $k$ -neighborhoods. The 1-D density functions were scaled using the trapezoid integration technique, and the midpoint rule was used for 2-D functions. This was done to ensure an area of 1 under the density curves.

The classification boundary is defined as the set of points where  $P(\text{AML}|x) = P(\text{ALL}|x)$ . Using Bayes' rule, this boundary can be expressed in terms of the a priori probabilities as  $P(x|\text{AML})P(\text{AML}) = P(x|\text{ALL})P(\text{ALL})$ . Finding this boundary can be very complicated. In one dimension, it can be represented as one or more points where the density functions cross each other, but in two dimensions it can be a set of both open and closed curves, making it difficult to represent either analytically or numerically.

Since there are only 38 points in the training set, it is computationally inexpensive to save the normalization factors from the density estimators and to evaluate the kernel functions at the points in the independent set. Then the values of the kernel functions can be added together and normalized. This results in an exact implicit boundary rather than an estimated explicit boundary.

In the  $k$ -NP density estimator, some data points are in more than one  $k$ -neighborhood. Since the

kernel functions are summed to build the estimated density functions, this results in Gaussian functions centered at some points being computed and added multiple times. To reduce computational expense, we created an array to count the number of times a Gaussian centered at each point in the training set is added to the conditional density function for each class. This array was filled as each  $k$ -neighborhood and majority class was found. Then each kernel function was only added once to the density function for each class, multiplied by the corresponding entry in the counting array.

## 5 Experiments and results

### 5.1 One-gene classifiers

The data for each gene was preprocessed by subtracting the across-patient mean and dividing by the standard deviation. This does not affect the results of density estimation, but allows  $\sigma$  to be chosen for the kernel functions without considering the range of values of the individual genes. For the Parzen density estimator, the following values were used for  $\sigma$  in the kernel function: {0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50}. The  $k$ -NN method used the following values of  $k$ : {3, 5, 7, 9, 11, 13, 15, 17, 19, 21}. The  $k$ -NP method used all possible combinations of the parameters used in the other methods. The best results from the classifiers are summarized in Table 1.

In the single gene case, the  $k$ -NN techniques performed better than the other two. It is also interesting to note that there were 11 genes with training accuracy  $\geq 95\%$  and testing accuracy  $\geq 74\%$  for both the Parzen and  $k$ -NN classifiers. All of these genes but one (gene 3252) also met the same performance criteria in the  $k$ -NP method. Descriptions of these genes appear in Table 2.

### 5.2 Two-gene classifiers

When multiple genes are considered, subtracting the mean of each gene does not affect the estimated density functions, but dividing by the standard deviation can affect the results. This is because it eliminates the information that comes from the relative expression levels of various genes for each patient. However, in the special case of Parzen density estimation with the assumption of zero covariance between genes, dividing by the standard deviation has no effect.

The values used for  $k$  are the same as those used for the one-gene case. The  $k$ -NN methods were

Table 1: Best classification results

method	gene1	gene2	accuracy (%)	
			training	testing
one-gene				
Parzen	6041		97	88
$k$ -NN1	3252		95	91
$k$ -NN2	3252		95	91
$k$ -NP	6041		95	88
two-gene				
Parzen	6041	3252	97	91
	4847	6855	100	91
	6041	6855	100	91
$k$ -NN1 (u)	6041	3252	95	91
	1882	6855	100	91
$k$ -NN1 (p)	4847	6855	100	91
	6041	6855	97	94
$k$ -NN2 (u)	1882	6855	100	91
$k$ -NN2 (p)	3252	2020	95	91
	4847	6855	100	91
	6041	6855	97	94
$k$ -NP	6041	6855	97	91

\*(p) denotes preprocessed data  
(u) denotes unprocessed data

applied to both the unprocessed data set and to the preprocessed data set, using the same preprocessing technique as the one-gene case. The Parzen and  $k$ -NP techniques were run only on the preprocessed data, and used the following covariance matrix:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}. \quad (9)$$

The diagonal structure of  $\Sigma$  represents an assumption of zero covariance. The following values were used for  $\sigma_1$  and  $\sigma_2$ : {0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40}.

Two-gene classifiers were built for all possible pairs of the 11 genes listed in Table 2. This subset of the data set was used to maintain a high level of testing accuracy while completing all the computations in a reasonable amount of time. The best performance of these classifiers is summarized in Table 1.

Table 2: Genes with high training and testing accuracy in both 1-D Parzen and  $k$ -NN methods

number	description
173	PRKCD Protein kinase C, delta
1745	LYN V-yes-1 Yamaguchi sarcoma viral related oncogene homolog
1882	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
1926	PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta
2020	FAH Fumarylacetoacetate
3252	GLUTATHIONE S-TRANSFERASE, MICROSOMAL
4847	Zyxin
6041	APLP2 Amyloid beta (A4) precursor-like protein 2
6218	ELA2 Elastase 2, neutrophil
6376	PFC Properdin P factor, complement
6855	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)

### 5.3 Comparison with other studies

We have also solved the classification problem another way by extending the work of Pollard *et al.* [6]. The original paper uses Partitioning Around Medoids (PAM) to cluster the training set, but does not perform any validation on the testing set. In our work, their methods were followed to select the 2000 genes with highest across-patient variance. Then each gene was preprocessed by subtracting the mean and dividing by the standard deviation. The PAM algorithm with the Euclidean distance metric was used to divide the training samples into two clusters and find a representative sample (cluster center) for each cluster. Next, the distance from each sample in the testing set to the representative sample of each cluster was computed. Each training sample was classified by assigning it to the cluster with the closest cluster center.

Table 3 shows the results of our PAM classification (line 3), compared to results that other researchers have achieved using pattern recognition techniques on the Golub leukemia data set.

The simple classifiers employed in this paper performed better than other, more sophisticated methods, as well as the results in the original paper. Also, the testing accuracy in this study using one or two genes is comparable to techniques that use many more genes. Our best two-gene classifier achieved the same level of accuracy as Li's one-gene logistic regression classifier, and our one-gene classifiers did almost as well. The gene used in Li's best classifier was gene 4847, which is also one of the genes that performed well in our one-gene methods (Table 2). Finally, the only classifier we could find that achieved 100% accuracy was Cho's ensemble classifier, which was arguably the most sophisticated of all the classifiers we encountered, combining seven different methods.

Many other authors, including Golub *et al.*, noted that sample 66 was often misclassified or unclassifiable [10, 18, 21, 22]. All of our best-performing classifiers also misclassified sample 66, except for the two-gene  $k$ -NN methods. In addition, samples 58 and 59 were misclassified by all of our classifiers.

## 6 Conclusions

As a rule of thumb, nonparametric density estimation usually requires a large amount of training data to provide a good estimate of the true distribution of a data set [16]. Because of this property and the small size of the Golub data set, the high testing accuracy we achieved was unexpected. However, the most important factor that is often overlooked is not the amount of training data, but rather how well the training set represents the actual distribution of the data. Due to the accuracy of our classifiers, it appears that the genes with the highest performance are both highly related to the leukemia conditions and highly representative of the overall population. In order to verify this, it would be necessary to validate our results using a new independent data set. Also, the density estimates could be improved by finding more accurate estimates of the a priori probabilities  $P(\text{AML})$  and  $P(\text{ALL})$  by sampling the patient population.

There are several ways this study can be improved and expanded. First, we only used one best parameter (or set of parameters) for each gene (or pair of genes) to perform testing. Instead of this, we could find all the parameters with high training accuracy for a given gene and test them all on the independent samples. This would be important if some parameter choices result in the same training accuracy but different testing accuracies.

Table 3: Testing accuracy from other studies

author	method	# genes	testing accuracy (%)
Bicciato <i>et al.</i> [9]	AANN	50	65
Furey <i>et al.</i> [7]	SVM	2000	76
Peters <i>et al.</i>	PAM	2000	79
Golub <i>et al.</i> [10]	Weighted voting	50	85
Li <i>et al.</i> [18]	Logistic regression	1	94
Dudoit <i>et al.</i> [19]	Nearest neighbor	40	95
	Diagonal LDA	40	95
	BoostCART	40	95
Ben-Dor <i>et al.</i> [20]	AdaBoost	7129	96
Nguyen <i>et al.</i> [21]	PCA/logistic discriminant	50	97
	PLS/logistic discriminant	50	97
Cho <i>et al.</i> [22]	classifier ensemble	172	100

A larger set of genes can also be screened for two-dimensional density estimation. For example, the genes in Table 2 could be paired with all other genes in the data set, rather than just the genes that performed well by themselves. Higher dimensional classifiers or one- and two-dimensional classifiers that use sums or ratios of expression levels could also be created to model more complex relationships between genes.

To improve the Parzen and  $k$ -NP density estimators, covariance could be included in the model. This would eliminate the zero covariance assumption and more accurately describe relationships between genes that work together in some way. Different distance metrics, such as Manhattan and Mahalanobis distance could be used in the  $k$ -NN and  $k$ -NP methods. The appropriate distance metric is often problem-dependent, and Mahalanobis distance has been successfully applied with  $k$ -means clustering on microarray data [4]. Weighted voting [23] could also be applied in these methods for comparison. Katkovnik and Shmulevich [24] developed a new kernel density estimation technique that uses varying adaptive window width. It would also be interesting to compare our results with this method.

In addition, it is imperative to determine whether the genes in Table 2, especially the ones with the best accuracy in the two-gene studies, have biologically significant relationships with ALL and/or AML. Furthermore, variable selection techniques could be used to produce alternative genes to use in building classifiers. It would be interesting to see whether such techniques could validate the results shown in Table 2. Finally, these methods could be applied to other publicly available microarray data to determine whether they are gen-

erally useful for microarray classification problems.

## References

- [1] J.C. Venter *et al.*, "The sequence of the human genome," *Science*, 291:1304–1351, 2001.
- [2] D. Gershon, "Microarray technology: an array of opportunities," *Nature*, 416:885–891, 2002.
- [3] S. Dudoit and R. Gentleman, "Statistics & genomics short course," <http://www.bioconductor.org/workshops/ShortCourse012302/>, Department of Biostatistics, Harvard School of Public Health, 2002.
- [4] F. Valafar, "Pattern recognition techniques in microarray data analysis: a survey," *Special Issue of Annals of New York Academy of Sciences, Techniques in Bioinformatics and Medical Informatics*, 980:41–64, 2002.
- [5] J. Quackenbush, "Computational analysis of microarray data," *Nature Genetics*, 2:418–427, 2001.
- [6] K.S. Pollard and M.J. van der Laan, "Statistical inference for simultaneous clustering of gene expression data," *Mathematical Biosciences*, 176:99–121, 2002.
- [7] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, 16(10):906–914, 2000.
- [8] M.P.S. Brown *et al.*, "Knowledge-based analysis of microarray gene expression data using support vector machines," *Proceedings of the National Academy of Sciences of the United States of America*, 97:262–267, 2000.
- [9] S. Bicciato, M. Pandin, G. Didonè, and C. Di Bello, "Pattern identification and classification in gene expression data using an autoassociative neural network model," *Biotechnology and Bioengineering*, 81(5):594–606, 2003.

- [10] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, 286:531–537, 1999.
- [11] F.M. Selaru, Y. Xu, J. Yin *et al.*, "Artificial neural networks distinguish among subtypes of neoplastic colorectal lesions," *Gastroenterology*, 122:606–613, 2002.
- [12] P. Toronen *et al.*, "Analysis of gene expression data using self organizing maps," *FEBS Letters*, 451:142–146, 1999.
- [13] P. Tamayo *et al.*, "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the United States of America*, 96:2907–2912, 1999.
- [14] C.H Pui and W.E. Evans, "Drug Therapy: Acute Lymphoblastic Leukemia." *New England Journal of Medicine*, 339:605–615, 1998.
- [15] J.F. Bishop, "Adult acute myeloid leukaemia: update on treatment." *Medical Journal of Australia*, 170:39–43, 1999.
- [16] C.M. Bishop, *Neural networks for pattern recognition*, New York: Oxford University Press, 1995.
- [17] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Boston, San Diego, New York: Academic Press, 1990.
- [18] W. Li and Y. Yang, "How many genes are needed for a discriminant microarray data analysis," in *Proc. Critical Assessment of Techniques for Microarray Data Mining Workshop*, Dec. 2000.
- [19] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," Department of Statistics, University of California, Berkeley, Tech. Rep. 576, June 2000.
- [20] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and N. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, 7:559–584, 2000.
- [21] D.V. Nguyen and D.M. Rocke, "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics*, 18(1):39–50, 2002.
- [22] S. Cho and J. Ryu, "Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features," *Proceedings of the IEEE*, 95(11):1744–1753, 2002.
- [23] J. Zavrel, "An empirical re-examination of weighted voting for  $k$ -NN," *Proc. of the 7th Belgian-Dutch Conference on Machine Learning*, SMC-6:325–327, 1997.
- [24] V. Katkovnik and I. Shmulevich, "Nonparametric density estimation with adaptive varying window size," *SPIE Conference on Image and Signal Processing for Remote Sensing VI, European Symposium on Remote Sensing*, Barcelona, Spain, 2000.