PNN model for Parzen's pdf estimation of image histograms

E.M. Aitnouri, Q. Jiang, S. Wang and D. Ziou
Faculty of Sciences, DMI, University of Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1
E-mail: {aitnouri, jiang, wang, ziou}@dmi.usherb.ca

Abstract

In this paper, we describe two applications of the Probabilistic Neural Network (PNN) to image histograms. The PNN is used to implement Parzen's window *pdf* estimator. The first application uses the PNN to smooth image histograms. This is a pre-processing operation which increases the robusteness of any post-processing algorithms. The second application deals with the non-parametric *pdf* estimation of image histograms. This is used in ship detection in RADAR-SAR imagery.

Keywords: *pdf* estimation, PNN model, non-parametric methods, histogram smoothing, RADAR-SAR image histogram, chip detection.

1 Introduction

In the literature, we find three alternative approaches of *pdf* estimation, each of which has its merits and limitations. The parametric method in which a specific functional form of the density model is assumed, approaches the *pdf* using only one distribution [1][2]. By contrast, the second approach, called semi-parametric method [3] (one variant of this method is the mixture model) tries to approach each class of the histogram by one distribution [4][5]. This leads to mixture models which are well suited to pre-process tasks as image segmentation [6]. Finally, the third approach of non-parametric estimation does not assume a particular form, but allows the form of the density to be determined entirely by the input data. The image histogram is considered as a set of points, rather than a set of modes, where each point is approached by a kernel function. The problem with this technique lies in both the choice of the width of kernels and the computation complexity. We are interested in this paper in the latest technique to estimate image histogram's *pdf*.

A particular form of non-parametric methods is the Parzen's window [7] estimator which suggests to superpose on each input data point a kernel function. Thus, the *pdf* results of the combination of these kernels. The problems with the Parzen method are however the computational complexity and the value of the kernel's width σ denoted "smoothing parameter", and which controls the smoothness of the estimated *pdf*. As solution to the complexity, we develop in this paper an algorithm based on the Probabilistic Neural Network (PNN) [8]. The algorithm uses a look-up-table which permits to reduce the complexity by half time. Concerning the choice of the best value of the smoothing parameter, we adopt in this paper a new experimental approach based on the "cross validation" technique [9]. The approach implements a regulation technique to select the best smoothing parameter from a set of possible values. Thus, we will describe how to compute an average value of the smoothing parameter for a given class of image histograms. The experiment is first applied to smooth (artificial) histograms. In order to have artificial histograms as close as real histograms, we make use of the algorithm described in [10] to generate such histograms. Then, the algorithm is applied to real histograms to verify if experimental results are in accommodation with the reality.

As applications, we will show some results about the increase of robustness of some algorithms when data are preprocessed with the PNN. The algorithm is also applied to ship target detection in SAR imagery using CFAR (Constant False Alarm Rate) technique. The CFAR technique uses the histogram *pdf* to compute possible targets. Thus, an accurate estimation of the *pdf* is strongly needed. We will show in this paper that the use of the PNN gives better results in terms of accuracy, than standard techniques using a parametric *pdf*.

This paper is organized as follow. In section 2 is presented the non-parametric *pdf* estimation using Parzen window *pdf* estimator. Section 3 deals with the PNN model as learning algorithm of the Parzen window method. An experimental method to estimate the width σ is also presented in this section. Experimental results of the application of the algorithm to both histogram's smoothing and ship target detection in RADAR-SAR images are presented in section 4. Finally, the conclusion is given in section 5.

2 *Kernel-based* methods and Parzen's Window estimator

Suppose we have a set of N data points noted $x_k / k = 1, .., N$, and we want to estimate the unknown *pdf* responsible of the generation of x_k. The basic idea of *Kernel-based* methods is that each input data point is considered as an elementary kernel, around which a region of volume V is defined. Thus, the superposition of these kernels (all the points) gives the unknown *pdf*. Parzen [7] uses as kernel a weight function, denoted by $W(d)$, centered on each point of the training set, and which has its largest value for $d = 0$. Moreover, $W(d)$ decreases rapidly as d increases. For a given point x of the training sample, the values of $W(d)$ are determined by the distances $d_k = (x - x_k) / k = 1, .., N$, from x to all the training points. In this way, the estimated density function can be given by:

$$p(x) = \frac{1}{VN} \sum_{k=1}^{N} W(x - x_k) \quad (1)$$

In [8] is presented the conditions that $W(d)$ should satisfy. Since the Gaussian distribution satisfies all those conditions, it is the commonly used weighted function. In this case, the Parzen's estimated *pdf* is given by:

$$p(x) = \frac{1}{N} \sum_{k=1}^{N} G_\sigma(x - x_k) \quad (2)$$

where $G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\frac{x}{2\sigma})^2}$. Here we use a normalized Gaussian which makes $V = 1$. If we analyze Eq.2, we find that the *pdf* depends only on one parameter σ, which represents the width of the Gaussian. Thus, a good choice of σ leads to a good *pdf* estimation. Two questions are to be mentioned however. Is there any algorithm which performs the Parzen's *pdf* rapidly, and how much is the value of σ in order to get smooth and accurate *pdf*? To answer these two questions, let us introduce the PNN learning algorithm.

3 The PNN as learning algorithm of the Parzen's *pdf* estimator

In many applications, *pdf* estimation is used for classification. A simple classifier can be built in the sense of Bayes classification for a data of c classes $C_j / j = 1, .., c$ as:

$$x_k \in C_i \;\; if \; p_i(x_k) > p_j(x_k) \; i \neq j \;\; /i, j = 1, .., c \quad (3)$$

where $p_i(x)$ is the probability that x belongs to C_i. Thus, it is necessary to estimate the conditional density $p(x_k, C_i)$ of each point in relation with each class C_j. Sprecht [8] used the Parzen's *pdf* estimator to build a neural network, called PNN (Probabilistic Neural Network). When there is only one class data, the PNN architecture for a small network is illustrated in Fig. 1.

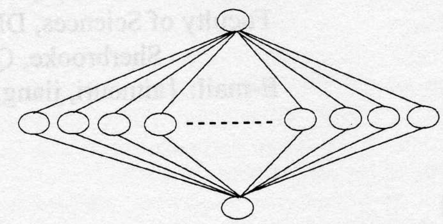

Figure 1: A simple PNN architecture for data belonging to one class

Fig 1 illustrates a simple PNN architecture to compute one data point in relation with all data points as presented in Eq.2. The point is presented to the network in the input layer. The pattern layer stores the learning set which corresponds to a large set of random points of the original image. The neurons of the pattern layer compute the value of the Gaussian with respect to the width σ and the distances between both the input and pattern layer values. The output layer sums the values of the pattern layer. This realizes a linear combination of the Gaussians. The architecture of Fig. 1 is capable of extremely high-speed operations. A review of the PNN model can be found in [11].

In practice, the algorithm is composed of two modules. The first one deals with a look-up table where the values of Gaussians of all the possible distances are stored. The second step realizes the linear combination of values selected from the look-up table. The size of the look-up table depends on the size of the learning set N. In the case of gray-level images, N have the same order as image size, thus a kernel function is approximatively superposed on each data pixels. This is really expensive. Fortunately, when histograms are used, as in our case, the complexity of the look-up table is reduced since a set of pixel may have the same gray-level. Indeed, consider a histogram represented by a function, $h(x)$, $x \in Gl_M$ of the gray-level frequencies of the image, where $Gl_M = \{0, 1, ..., M-1\}$ corresponds to the gray levels of the image. $h(x)$ gives the number of pixels having gray level x. Thus, in introducing $h(x)$ in Eq.2 we will have:

$$p(\mathbf{x}) = \sum_{k=1}^{M} \frac{h(k)}{N} \; G_\sigma(x - x_k) \quad (4)$$

The quantity $\frac{h(x)}{N}$ is the normalized histogram denoted by $h_N(x)$. We can remark that the complexity of the look-up table is significantly reduced. Since the Gaussian is an even function, the size of the look-up table is implicitly reduced from M^2 to $M(\frac{M-2}{2})$.

3.1 Estimation of σ

The problem of choosing an appropriate value of σ has been investigated by many researches [12][13], but it still a problem which has not been completely resolved since results are obtained under various assumptions. A particular formal approach of choosing σ has been introduced by Duin [14], where Eq.4 is considered as a specific parametric form of density function $p(x, \sigma)$, of which the parameter σ is unknown. A general solution can be obtained using the maximum likelihood technique. However it is true only when the sample size M approach to infinity. Otherwise, the value of σ approaches zero. Kraaijveld [15] used approximations of Duin's solution in order escape from the zero σ solution. His method is based on the use of both training and test data. This method is restrictive since it requires a large set of test data.

In this paper, we adopt an experimental approach for choosing σ. Since we want to apply the PNN model to estimate image histogram's *pdf*, we make use of our knowledge of the application data. The aim of the experiment is the development of an automatic procedure of choosing the best value of the smoothing parameter σ. For image histograms, the best value of σ is for which:

- the resulted *pdf* is accurate,
- the resulted *pdf* is suffisently smooth.

For this purpose, we define an error function to measure the effectiveness of the estimated *pdf*. The error, denoted by E_p is given by:

$$E_p = E_q + \eta E_d \qquad (5)$$

where E_q is the quadratic error, and $E_d = \frac{\delta p^r(x)}{\delta x^r}$ is some derivative of the estimated *pdf*. E_p implements a regulation technique. Indeed, the first term of E_p measures the deviation between the input histogram and the estimated *pdf*, denoted by "goodness of fit". However, the second term introduces a term of penalty. It punishes any estimated *pdf* which tends to oscillate. Thus, it favors smooth estimate. η expresses the penalty rate.

The regulation method defined by E_p realizes a model choice criterion, where each model is defined by the value of the smoothing parameter σ, with $\sigma_{min} \leq \sigma \leq \sigma_{max}$. The model with the best σ is for which E_p is minimum.

Now, what are the suitable values of the bounds, σ_{min} and σ_{max}? For this purpose, we present in this paper an experimental method to determine these bounds. The experiment which is based on the "cross-validation" technique, aims to determine an average value of the smoothing parameter σ for a class of image histograms. The cross validation technique suggests the use of two types of data, training and validation data. We use the training data to learn the PNN. This will result in an estimation of a *pdf*. Then we compare with the validation data. The best σ results with the minimum error observed between the estimated *pdf* and the validation data. Note that validation data must not be used to train the network. The experiment is performed first on artificial histograms. Using artificial data allows to set some parameters as the penalty rate η. Then we will apply the algorithm on a set of real image histograms.

3.1.1 Estimation of σ using artificial histograms

Artificial histograms used in this experiment are generated as follows:
step1: We generate smooth histograms using a mixture model. Each component of the mixture, *versus* mode is assumed to be normal. In order to have significant results when applying the algorithm to artificial histograms, we should provide artificial histograms as close as possible to real histograms. For this purpose, we use the algorithm describe in [10] to generate such histograms, denoted by h_{smooth}.
step2: We add to the smooth histogram a Gaussian white noise of width σ_n. To do so, we re-scale h_{smooth} so that $0 \leq h_{smooth} \leq Gl$. Figure 2 shows a smooth and noisy histograms with $\sigma_n = 3$.

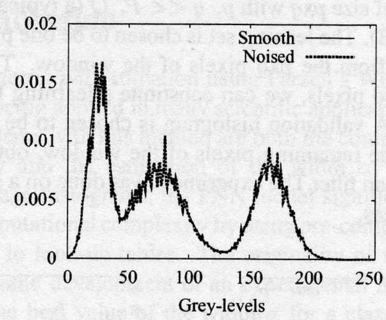

Figure 2: Smooth and noisy histograms, with $\sigma_n = 3$

We generate L noisy histograms, denoted by h^j_{noised} $/j = 0,..,L$, by adding different Gaussian white noise of width $0 \leq \sigma_n \leq L$ to h_{smooth}. One of those noisy histograms, denoted by h_{learn} and corresponding to σ_{learn} is used as learning set. The remaining noisy histograms are used to compute the validation set, denoted by h_{val}. h_{val} is the mean of the remaining noisy histograms and given by $h_{val} = \frac{1}{L-1}\sum^{L}_{i\neq\sigma_{learn}} h^j_{noised}$. For the histogram of figure 2, we have studied the behavior of the PNN for different values of the noise σ_n $/\sigma_n = 0,..,10$, the penalty η $/\eta = 0,..,2$ and the smoothing parameters σ $/\sigma = 0.5,..,10$.

mean error is observed around $\sigma = 2.5$. Fig. 4.b shows, for any given value of σ, the number of images $N(\sigma)$ that gave rise to the minimum error. Fig. 4.b provides information about the distribution of the number of images *versus* the best choice of σ. In the case of this experiment, the concentration of this distribution is around $\sigma = 2.5$ which is in accommodation with the result of Fig.4.a . From the experiment, we can see that $\sigma = 2.5$ is a good choice in general, although it is not the best choice for all the images in the database. It is to be pointed out that this result is specific to a set of radar images. The optimal value σ for optical images is different. Thus, for radar images, the bounds σ_{min} and σ_{max} should be chosen around $\sigma = 2.5$.

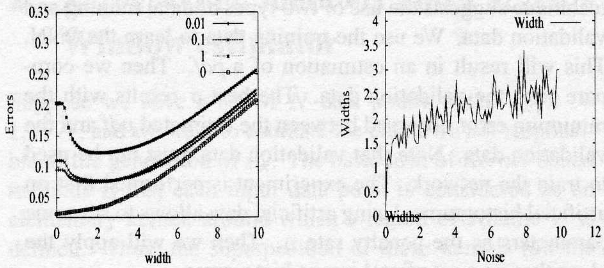

Figure 3: a. The Error E_p for different values of the penalty η, with $\sigma_n = 8$, b. σ in relation with σ_n, with $\eta = 0.1$

Figure 3 shows the error E_p in relation with σ for different values of η, and $\sigma_n = 8$. We see that the best smoothing parameters is around $\sigma = 2.5$ what ever the value of η. Moreover, when $\eta = 0$, there is no regulation. We choose for the rest of the paper $\eta = 0.1$. Figure 3.b shows the values of σ in relation with the noise σ_n, with $\eta = 0.1$. We see that what ever the value of σ_n, the best value is observed around $\sigma = 2.5$. This behavior was observed for different artificial histograms. Did the same behavior is observed for real images?

3.1.2 Estimation of σ for real images

Consider an image of size PxQ which is divided in small windows of size pxq with $p,\ q << P,\ Q$ (a typical value of p and q is 3). The leaning set is chosen to be one pixel taken randomly from the pxq pixels of the window. Thus, with the selected pixels, we can constitute a learning histogram (h_{learn}). A validation histogram is chosen to be a median value of the remaining pixels of the window, obtained using a median filter.The experiment was done on a set of 200 radar images.

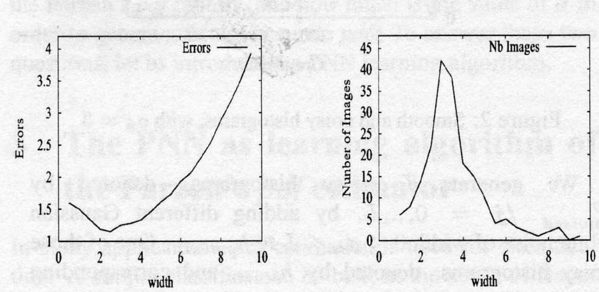

Figure 4: Results of the experiment for estimating the best value of the width σ. The experiment has been done for $\sigma = 0.5$ to 9.5 with a step of 0.5 and $\eta = 0.1$. a) The mean error. b) Number of images that gave rise to the minimum error.

Fig. 4.a illustrates the mean error $M(E_p)$ calculated over 200 radar images for each value of σ. The minimum

4 Application of the PNN to image histograms

In this section we will describe the use of the PNN algorithm to two image histogram applications. These applications involve image histograms, thus we make use of the experimental results, namely the best smoothing parameter is likely around $\sigma = 2.5$. Since the complexity of the PNN algorithm is reduced with the introduction of the look-up table, this allows us to explore different values of σ around $\sigma = 2.5$. Indeed, we will use the bound $0 \leq \sigma \leq 9.5$, with a step of 0.5.

4.1 PNN algorithm as image histogram smoother

In [5] is presented the EFC algorithm for estimating the number of modes of grey-level image histograms. The model begins with smoothing the input histogram. This is done using the PNN. While this operation is not essential, it however increases the robustness of the model.

The EFC algorithm was applied to a set of artificial histograms. The experiment is described in [16], where the EFC was compared to an Akaike's criterion based algorithm. The aim of the experiment is the computation of the ability of the EFC to estimate the exact number of components in each histogram of the set. The EFC was applied to a set of 1000 three-mode histograms. Here, we present results with the use of the PNN and without the use of the PNN. Table 1 illustrates the comparison of the two procedures. Table 1 gives the percentage of cases the EFC finds the exact number of modes, which is three. We see that the EFC overestimate the number of modes when the input data are not smoothed with the PNN. Thus pre-processing input data with the PNN, provides more robustness.

EFC	without PNN	with PNN
1	0	0
2	6	10
3	50	66
4	30	12
5	12	10
6	2	2
7	0	0

Table 1: The EFC with and without the PNN

4.2 Ship targets detection in SAR imagery

Ship target detection in SAR imagery, using the Constant False Alarm Rate (CFAR) [17] is a technic which can be classified in the domain of pattern recognition. Indeed, CFAR technique suggests the examination of the image histogram in order to estimate a threshold I_c. I_c depends on the image histogram's *pdf* and is given by:

$$\eta_c = \int_0^{I_c} p(x)dx \qquad (6)$$

where η_c is the required significance level ($\eta_c = 0.995$). The first step is the estimation of the image histogram *pdf*. Vachon and Ray [18] used a parametric approach to estimate the *pdf*. Since RADAR-SAR image histograms do not behave as a normal distribution (Gaussian), K-distribution model was used. However, complicated mathematical expression of the K-distribution makes it impossible to have a formal solution of I_c from Eq.6. With the use of the PNN described by Eq. 4, it is possible to design automatic computation of the threshold I_c fro Eq. 6. This is done using dichotomy in [17].

For this application, we have used the PNN to compute the *pdf*. Since accurate *pdf* is needed, we have designed an automatic procedure of searching the best smoothing parameter σ among six different values, all around $\sigma = 2.5$. The error E_p defined in Eq. 5 have been used as choice criterion. The values of σ were $\sigma = 1.6,\ 2.0,\ 2.4,\ 2.8,\ 3.2$.

Here we present comparison results between the use of both PNN model and K-distribution model to estimate the SAR-image histogram *pdf* by illustrating three examples given respectively in Tables 2, 3 and 4. The related comparisons are CPU time, which is the total detection time; the value of the estimated threshold I_c; the mean square error (MSE) of the estimated *pdf*. Experimental results show clearly that the PNN model in most time is suitable than K-distribution model for ship target detection. Indeed, it requires less computational time than K-distribution model. Secondly, small values of the threshold involve more targets detected. Finally, the PNN's *pdf* is mode accurate than the K-distribution *pdf*. This algorithm however offers the possibility to obtain a totally automatic detection algorithm since there is no need to human intervention to set parameters.

	PNN-model	K-model
CPU time	0.002048	0.012338
Threshold I_c	198	224
MSE	2.55	3.35

Table 2: First example of the comparison

	PNN-model	K-model
CPU time	0.000848	0.006953
Threshold I_c	92	93
MSE	2,24	4.2

Table 3: second example of the comparison

	PNN-model	K-model
CPU time	0.03016	0.049056
Threshold I_c	24	25
MSE	3.18	13.5

Table 4: third example of the comparison

5 Conclusion

Non-parametric *pdf* estimation using Parzen's Window approach is being applied in many scientific applications. The main problems with this method are both the computational complexity and the estimation of the width σ. For the case of image histograms, the PNN model significantly reduces computational complexity by using pre-computed values stored in look-up tables. The originality of this work however is the development of an experimental method to compute the best value of the width σ for a class of data. This experimental method can be used for any class of image data, although it has been used in this paper only for RADAR-SAR images.

The application of the PNN model to ship target detection has given promising results in terms of computational time and accuracy. However, the PNN can also be used to smooth image histograms. The algorithm described in this paper provides a good structure for possible hardware implementation of automatic applications.

References

[1] D.B. Lawrence and J.T. Gene Hwang. How to Approximate a Histogram by a Normal Density. *The American*

Statistician, 47(4), 1993.

[2] Q. Jiang and S. Wang and D. Ziou and A. El Zaart and G.B. Bénié. Ship Detection in RADARSAT SAR Imagery. In *Proc. of CMS'98*, San Diego, 1998.

[3] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford University Press, Oxford, 1995.

[4] R. Samadani. A finite mixtures algorithm for finding proportions in SAR images. 4(8):750–754, August 1995.

[5] E.M. Aitnouri and S. Wang and D. Ziou and J. Vaillancourt and L. Gagnon. Estimation of a Multi-modal Histogram's *pdf* using a Mixture Model. *Neural, Parallel & Scientific Computation*, 7(1):103–118, 1999.

[6] L. Gupta and T. Sortrakul. A Gaussian-Mixture-Based image segmentation Algorithm. *Pattern Recognition*, 31(3):315–325, 1998.

[7] E. Parzen. On the Estimation of a Probability Density Function and Mode. *Annals of Math. Stat.*, 33:1065–1076, 1962.

[8] D.F. Sprecht. Probabilistic Neural Network. *Neural Network*, 3(1):109–118, 1990.

[9] M. Stone. Cross-Validatory choice and assessment of Statistical Predictions. *the Royal Statistical Society*, 36(1):111–147, 1974.

[10] E.M. Aitnouri and F. Dubeau and S. Wang and D. Ziou . Generating Test Vectors for Clustering Algorithms. Technical Report N^o 236, DMI, Univ. of Sherbrooke, Sherbrooke, Sep. 1999.

[11] F. Vicino. The Probabilistic Neural Network. *Substance Use and Misuse*, 33(2):335–352, 1998.

[12] K. Fukunaga. *Introduction to Statistical Pattern recognition*. Academic Press, New York , 1990.

[13] L. Devroye. Automatic pattern recognition, a study of a probability of an error. *IEEE Trans. on Pattern Anal. Mach. Intell*, 4, 1988.

[14] R.P.W Duin. On the choice of smoothing parameters for Parzen estimators of probability density function. *IEEE Trans. on Computer*, 25:1175–1179, 1976.

[15] M.A. Kraaijveld. A Parzen classifier with an improved robusteness against deviations between training and test data. *Pattern Recognition Letters*, 17:679–689, 1996.

[16] E.M. Aitnouri and S. Wang and D. Ziou . On Comparison of Clustering Techniques for Histogram *pdf* Estimation. *to appear in jounal of Pattern Recognition & Image Analysis* , 10(2), 2000.

[17] Q. Jiang and E.M. Aitnouri and S. Wang and D. Ziou. Ship Detection Using PNN model. In *Proc. of ADRO'98*, Montreal, 1998.

[18] P.W. Vachon and J. Campbell and C. Bjerklund and F. Dobson andM. Rey. Ship detection by the RADARSAT SAR: Validation of detection. *Canadian remote sensing*, 23(1), 1997.