A BAYESIAN REGULARIZATION METHOD FOR THE PROBABILISTIC RBF NETWORK

C. Constantinopoulos, M. Titsias, A. Likas

28 – 2001

Preprint, no 28 – 01 / 2001

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

A Bayesian Regularization Method for the Probabilistic RBF Network

Constantinos Constantinopoulos
Michalis K. Titsias
Aristidis Likas

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece
e-mail: {ccostas,mtitsias,arly}@cs.uoi.gr

Abstract

The Probabilistic RBF (PRBF) network constitutes a recently proposed classification network that employs Gaussian mixture models for class conditional density estimation. The particular characteristic of this model is that it allows the sharing of the Gaussian components of the mixture models among all classes, in the same spirit that the hidden units of a classification RBF network feed all output units. Training of the PRBF network is a likelihood maximization procedure based on the Expectation - Maximization (EM) algorithm. In this work, we propose a Bayesian regularization approach for training the PRBF network that takes into account the existence of ovelapping among classes in the region where a Gaussian component has been placed. We also propose a fast and iterative training procedure (based on the EM algorithm) to adjust the component parameters. Experimental results on well-known classification data sets indicate that the proposed method leads to superior generalization performance compared to the original PRBF network with the same number of kernels.

1 Introduction

In pattern recognition it is well-known that a convenient way to construct a classifier is on the basis of inferring the posterior probability of each class. From the statistical point of view this inference can be achieved by first evaluating the class conditional densities $p(x|C_k)$ and the corresponding prior probabilities $P(C_k)$ and then making optimal decisions for new data points by combining these quantities through the Bayes theorem

$$P(C_k|x) = \frac{p(x|C_k)P(C_k)}{\sum_{k'} p(x|C_{k'})P(C_{k'})}, \tag{1}$$

and then selecting the class with maximum $P(C_k|x)$. In the traditional statistical approach each class density $p(x|C_k)$ is estimated using a separate mixture model and considering only

the data points of the specific class, therefore the density of each class is estimated independently from the other classes. We will refer to this approach as the *separate mixtures* model.

The probabilistic RBF network [3, 4] constitutes an alternative approach for class conditional density estimation. It is an RBF-like neural network adapted to provide output values corresponding to the class conditional densities $p(x|C_k)$. Since the network is RBF, the kernels (hidden units) are shared among classes and each class conditional density is evaluated using not only the corresponding class data points (as in the traditional statistical approach), but using all the available data points. In order to train the PRBF network, an Expectation - Maximization (EM) algorithm can be applied [4]. The treatment of the training procedure as a likelihood maximization problem provides the opportunity to define Bayesian priors on the network parameters. The priors we propose tend to favor solutions that avoid the placement of a kernel in a region with weak overlap among classes. We provide an iterative EM-based procedure for finding the maximum a posteriori probability (MAP) [2] PRBF parameters. The effectiveness of the proposed method is demonstrated using several data sets and the experimental results indicate that the method leads to performance improvement over the classical PRBF training method.

2 The Probabilistic RBF Network

Consider a classification problem with K classes and a training set $X = \{(x^{(n)}, y^{(n)}),\ n = 1, \ldots, N\}$ where $x^{(n)}$ is a d-dimensional pattern and $y^{(n)}$ is a label C_k $(k = 1, \ldots, K)$ indicating the class of pattern $x^{(n)}$. The original set X can be easily partitioned into K independent subsets X_k, so that each subset contains only the data of the corresponding class. Let N_k denote the number of patterns of class C_k, ie. $N_k = |X_k|$.

Assume that we have a number of M kernel functions (hidden units), which are probability densities, and we would like to utilize them for estimating the conditional densities of all classes by considering the kernels as a common pool [3, 4]. Thus, each class conditional density function $p(x|C_k)$ is modeled as

$$p(x|C_k) = \sum_{j=1}^{M} \pi_{jk} p(x|j), \qquad k = 1, \ldots, K \tag{2}$$

where $p(x|j)$ denotes the kernel function j, while the mixing coefficient π_{jk} represents the prior probability that a pattern has been generated from kernel j, given that it belongs to class C_k. The priors take positive values and satisfy the following constraint:

$$\sum_{j=1}^{M} \pi_{jk} = 1, \qquad k = 1, \ldots, K. \tag{3}$$

It is also useful to introduce the posterior probabilities expressing our posterior belief that kernel j generated a pattern x given its class C_k. This probability is obtained using the Bayes' theorem

$$P(j|C_k, x) = \frac{\pi_{jk} p(x|j)}{\sum_{j'} \pi_{j'k} p(x|j')}. \tag{4}$$

In the following, we assume that the kernel densities are Gaussians of the general form

$$p(x|j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right\} \tag{5}$$

where $\mu_j \in R^d$ is a vector representing the center of kernel j, while Σ_j represents the corresponding $d \times d$ covariance matrix. The whole adjustable parameter vector of the model consists of the priors and the kernel parameters (means and covariances) and we denote it by θ.

Training of the PRBF network may efficiently achieved with the EM algorithm [3, 4]. It consists of the iterative application of the following processing steps:

1. E-step: For each training point $(x^{(n)}, y^{(n)}) \in X$ compute the posterior probabilities $P^{(t)}(j|C_k, x^{(n)})$, for $j = 1, \ldots, M$ and $k = 1, \ldots, K$, from (4) using the current parameters $\theta^{(t)}$.

2. M-step: Find the new parameter vector $\theta^{(t+1)}$ using the following equations:

$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} P^{(t)}(j|C_k, x) x}{\sum_{k=1}^{K} \sum_{x \in X_k} P^{(t)}(j|C_k, x)} \tag{6}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} P^{(t)}(j|C_k, x)(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{k=1}^{K} \sum_{x \in X_k} P^{(t)}(j|C_k, x)} \tag{7}$$

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|} \sum_{x \in X_k} P^{(t)}(j|C_k, x), \;\; k = 1, \ldots, K \tag{8}$$

It is apparent that the PRBF model is a special case of the RBF network where the outputs correspond to probability density functions and the second layer weights are constrained to represent prior probabilities. Furthermore, it can be shown that the separate mixtures model can be derived as a special case of PRBF.

As discussed in [5] both the PRBF model (trained in a typical manner) and the separate mixtures model, in some cases provide solutions that are not very effective from the classification point view. More specifically, it has been observed [5] that the PRBF network provides inferior classification solutions when there exist kernels that have been placed on regions where there exists weak overlapping among classes. The motivation behind this work

is to exploit Bayesian regularization, by specifying appropriate priors on the PRBF parameters, in order to guide the training process to avoid solutions exhibiting the above undesirable characteristic.

3 Bayesian Regularization

Let $P(C_k)$ denote the prior probability of class C_k. In order to use Bayes rule (1) for unlabeled input data we have to find first appropriate values for both class prior probabilities and parameter vector θ. Thus, the whole adjustable parameter vector is $\Theta = (\theta, P(C_1), \ldots, P(C_k))$. We can utilize Bayes theorem once again to estimate the a posteriori distribution of the parameter vector Θ according to

$$p(\Theta \mid X) = \frac{p(X \mid \Theta)p(\Theta)}{\int p(X \mid \Theta)p(\Theta)\mathrm{d}\Theta} \tag{9}$$

where $p(X \mid \Theta)$ is the density of the observations X given Θ, and $p(\Theta)$ is the prior density on Θ. The configuration $\hat{\Theta}$ that maximizes $p(X \mid \Theta)p(\Theta)$ also maximizes $p(\Theta \mid X)$, and is known as the maximum a posteriori (MAP) estimation of Θ

$$\hat{\Theta} = \arg\max_{\Theta} \; p(X \mid \Theta)p(\Theta) \tag{10}$$

In order to proceed with the MAP estimation, we have to define a proper prior $p(\Theta)$ for PRBF training. At first we introduce the variables μ_{jk} and Σ_{jk}, for $j = 1, \ldots, M$ and $k = 1, \ldots, K$. These represent means and covariance matrices respectively as follows:

$$\mu_{jk} = \frac{\sum_{x \in X_k} P(j|C_k, x)x}{\sum_{x \in X_k} P(j|C_k, x)} \tag{11}$$

$$\Sigma_{jk} = \frac{\sum_{x \in X_k} P(j|C_k, x)(x - \mu_{jk})(x - \mu_{jk})^T}{\sum_{x \in X_k} P(j|C_k, x)}. \tag{12}$$

As shown in [4], μ_{jk} and Σ_{jk} constitute an estimation of the parameters of kernel j, when only data of class C_k are considered. It has been shown [4] that during PRBF training, the parameters of kernel j computed at any EM iteration can be written as

$$\mu_j = \sum_{k=1}^{K} P(C_k|j)\mu_{jk} \tag{13}$$

$$\Sigma_j = \sum_{k=1}^{K} P(C_k \mid j)\Sigma_{jk} \tag{14}$$

where

$$P(C_k \mid j) = \frac{\pi_{jk}p(C_k)}{\sum_{k'=1}^{K} \pi_{jk'}P(C_{k'})} \tag{15}$$

is the probability that pattern x belongs to class C_k, given that it has been generated from kernel j. The above equations indicate that the parameters μ_j, Σ_j of kernel j actually correspond to the mean values of the variables μ_{jk} and Σ_{jk}, for $k = 1, \ldots, K$. For convenience we will refer to a 'component' with parameters μ_{jk} and Σ_{jk} as *subkernel* jk. In other words each subkernel jk defines a distribution $p(x \mid jk)$ with mean μ_{jk} and covariance Σ_{jk}. Now we can quantify the overlapping among classes in the region of a kernel using measures of the distance among distributions. The expected value of the distance between the kernel j and its subkernels jk can be used as a measure of class ovelapping in the region of kernel j. Using the *Bhattacharya distance* between $p(x \mid j)$ and $p(x \mid jk)$ we obtain the desirable measure:

$$\delta_j = \sum_{k=1}^{K} P(C_k \mid j) \left\{ -\ln \int [p(x \mid j) p(x \mid jk)]^{1/2} \, \mathrm{dx} \right\} \tag{16}$$

In the case of complete overlapping among classes δ_j equals zero, and the same holds if only one class exists in the region of the kernel j.

Based on this property of δ_j, we define the prior on Θ as

$$p(\Theta) = \prod_{j=1}^{M} \exp\{-\alpha \delta_j\} \tag{17}$$

Apparently there is no a priori assumption about class priors, and each factor of the product refers to a kernel of the model. According to the above discussion, solutions where kernels are placed in regions with high overlapping or no overlapping at all are prefered. With this choice of $p(\Theta)$, it is expected that in the case where a subkernel jk exhibits weak overlapping with the remaining subkernels jl, the training algorithm will force π_{jk} to become zero.

3.1 The EM training procedure

The posterior log likelihood function of the data set X is

$$L(\Theta) = \sum_{n=1}^{N} \log p(x^{(n)}, y^{(n)} \mid \Theta) + \log p(\Theta) \tag{18}$$

Using that $p(x, C_k \mid \Theta) = p(x \mid C_k, \Theta) P(C_k \mid \Theta)$ and the fact that the data set X consists of K independent subsets X_k, the above equation takes the form

$$\begin{aligned} L(\Theta) &= \sum_{k=1}^{K} |X_k| \log p(C_k \mid \Theta) \\ &+ \sum_{k=1}^{K} \sum_{x \in X_k} p(x \mid C_k, \Theta) + \log p(\Theta) \end{aligned} \tag{19}$$

To simplify the procedure, we maximize the first term of (19) separately, and then use the resulting solution in the maximization of the remaining terms. Maximization of the first term yields

$$P(C_k) = \frac{|X_k|}{|X|}, k = 1, \ldots, K \tag{20}$$

while the maximization of the reamaining terms is equivalent to PRBF training using regularization. Consequently the a posteriori log likelihood function suitable for training of the PRBF network is given by

$$L(\theta) = \sum_{k=1}^{K} \sum_{x \in X_k} p(x \mid C_k, \theta) + \log p(\theta) \tag{21}$$

and assuming Gaussian mixture models the above equation can be written as

$$L(\theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_{jk} p(x \mid j) - \alpha \sum_{j=1}^{M} \sum_{k=1}^{K} P(C_k \mid j) \beta_{jk} \tag{22}$$

where β_{jk} is the Bhattacharya distance between Gaussian distributions $p(x \mid j)$ and $p(x \mid jk)$

$$\begin{aligned} \beta_{jk} &= \frac{1}{8}(\mu_j - \mu_{jk})^T \left[\frac{\Sigma_j + \Sigma_{jk}}{2}\right]^{-1} (\mu_j - \mu_{jk}) \\ &+ \frac{1}{2} \ln \frac{\left|\frac{1}{2}(\Sigma_j + \Sigma_{jk})\right|}{|\Sigma_j|^{1/2} |\Sigma_{jk}|^{1/2}} \end{aligned} \tag{23}$$

In order to maximize $L(\theta)$ we employ the EM algorithm [1] and show that PRBF regularization can be performed with a fast, effective and easily implementable scheme.

The Expectation-Maximization (EM) algorithm is a general technique for maximum likelihood estimates in the case where hidden information exists. Given the corresponding incomplete data set X, the complete data set is defined as $X_C = \{(x^{(n)}, y^{(n)}, z^{(n)}),\ n = 1, \ldots, N\}$ where the hidden variable z is a M-dimensional vector of zero-one values, indicating the kernel that generated x. If kernel j is responsible for generating x then $z_j = 1$, otherwise $z_j = 0$. The expected value of z equals the a posteriori probability $P(j \mid x, C_k)$ that kernel j generated x given the class label C_k, defined as

$$P(j \mid x, C_k) = \frac{\pi_{jk} p(x \mid j)}{\sum_{i=1}^{M} \pi_{ik} p(x \mid i)} \tag{24}$$

Following the common procedure, we define the expected complete a posteriori log likelihood as

$$L_C(\theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} P(j \mid x, C_k) \log \pi_{jk} p(x \mid j) - \alpha \sum_{j=1}^{M} \sum_{k=1}^{K} P(C_k \mid j) \beta_{jk} \tag{25}$$

We make the reasonable assumption that $\Sigma_{jk} = \Sigma_j$, and concentrate on the centers of the subkernels. So at iteration $t+1$ of the algorithm, the quantity to be maximized at the M-step

is:

$$Q(\theta;\theta^{(t)}) = \sum_{k=1}^{K}\sum_{x\in X_k}\sum_{j=1}^{M} P^{(t)}(j \mid x, C_k)\log \pi_{jk} p(x \mid j) - \frac{\alpha}{8}\sum_{k=1}^{K}\sum_{j=1}^{M} P(C_k \mid j)(\mu_j - \mu_{jk}^{(t)})^T \left[\Sigma_j^{(t)}\right]^{-1}(\mu_j - \mu_{jk}^{(t)}) \quad (26)$$

Based on several algebraic manipulations, it can be shown that the above maximization can be performed analytically thus leading to the following update equations:

$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^{K}\sum_{x\in X_k} p^{(t)}(j \mid x, C_k)x + \frac{\alpha}{4}\sum_{k=1}^{K} P^{(t)}(C_k \mid j)\mu_{jk}^{(t)}}{\sum_{k=1}^{K}\sum_{x\in X_k} P^{(t)}(j \mid x, C_k) + \frac{\alpha}{4}} \quad (27)$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^{K}\sum_{x\in X_k} P^{(t)}(j \mid x, C_k)(x - \mu_j^{(t+1)})(x - \mu_j^{(t+1)})^T}{\sum_{k=1}^{K}\sum_{x\in X_k} P^{(t)}(j \mid x, C_k)} \quad (28)$$

$$\pi_{jk}^{(t+1)} = \frac{\sum_{x\in X_k} P^{(t)}(j \mid x, C_k) + \frac{\alpha}{8}P^{(t)}(C_k \mid j)\left\{\sum_{l=1}^{K} P^{(t)}(C_l \mid j)\delta_{jl}^{(t)} - \delta_{jk}^{(t)}\right\}}{|X_k| + \frac{\alpha}{8}\sum_{i=1}^{M} P^{(t)}(C_k \mid i)\left\{\sum_{l=1}^{K} P^{(t)}(C_l \mid i)\delta_{il}^{(t)} - \delta_{ik}^{(t)}\right\}} \quad (29)$$

where

$$\delta_{rs}^{(t+1)} = (\mu_r^{(t+1)} - \mu_{rs}^{(t)})^T \left[\Sigma_r^{(t+1)}\right]^{-1}(\mu_r - \mu_{rs}^{(t)}) \quad (30)$$

It is worthwile to examine the regularization term in (29). Notice that for any kernel j, the regularization terms corresponding to the subkernels jk $(k = 1, \ldots, K)$ sum to zero:

$$\sum_{k=1}^{K} P(C_k \mid j)\left\{\sum_{l=1}^{K} P(C_l \mid j)\delta_{jl} - \delta_{jk}\right\} = 0 \quad (31)$$

This equations indicate that there is competition among the subkernels. If the distance between the kernel j and one subkernel jk' is less than the average, then the corresponding regularization term is positive, otherwise it is negative. In that way the remote subkernel is penalized, and eventually rejected if the prior $\pi_{jk'}$ becomes zero.

A computational problem that we experience is that sometimes the negative regularization term becomes too high and results in negative priors. To avoid this situation, at each iteration if the minimum prior of any class becomes negative we set it equal to zero, and normalize the remaining priors in order to satisfy (3).

4 Experimental Results and Conclusions

In this section we compare the proposed training method with the typical PRBF training method [3, 4]. We considered four well-known data sets from the UCI repository, namely the Phoneme, Satimage, Pima Indians Diabetes and Ionosphere data sets. For each data set,

	Number of kernels				
Algorithm	6	8	10	12	14
PRBF	30.33	30.07	28.26	28.00	27.35
$\alpha = 5$	31.25	28.25	27.21	26.82	25.78
$\alpha = 10$	28.91	26.30	**26.56**	27.60	26.82
$\alpha = 15$	28.00	27.73	27.99	**26.81**	26.43
$\alpha = 20$	**27.47**	**25.64**	27.86	28.39	**25.38**

Table 1: Generalization error on the Pima Indians Diabetes data set.

	Number of kernels				
Algorithm	6	9	12	15	18
PRBF	24.12	17.09	17.01	16.08	16.16
$\alpha = 5$	23.92	17.20	16.92	16.05	15.69
$\alpha = 10$	24.10	17.08	16.41	15.99	15.71
$\alpha = 15$	22.74	16.58	**15.80**	15.85	15.68
$\alpha = 20$	**22.55**	**15.88**	15.99	**15.76**	**15.48**

Table 2: Generalization error on the Satimage data set.

in order to obtain an estimation of the generalization error, we have employed 5-fold cross-validation. In every experiment all training algorithms started from the same initial state. Tables 1-3 provide the obtained results for both methods, for several values of the number of kernel functions M, and the hyperparameter α. The values of α we used were multiples of the quantity $\frac{N}{KM}$.

The results indicate that the proposed regularization technique provides networks with superior performance compared to typical PRBF training. It must also be noted that the method is fast, since in all experiments 100 EM iterations were sufficient for reaching the final solution.

In what concerns future enhancement of the method, our current work focuses on the utilization of alternative distance measures, averaging PRBF networks obtained for different values of the hyperperameter α, and developing an approach for dynamicaly adjusting the number of kernels M. In the last case our aim is to exploit recent results for adjusting the number of kernels in a Gaussian mixture that have been developed in the framework of pdf estimation [6].

	Number of kernels				
Algorithm	8	10	12	14	16
PRBF	21.12	21.58	21.23	21.57	21.33
$\alpha = 5$	**20.97**	21.66	21.27	21.64	21.10
$\alpha = 10$	21.08	21.44	**20.81**	20.70	20.62
$\alpha = 15$	21.34	21.16	20.99	20.75	**20.38**
$\alpha = 20$	21.03	**20.81**	20.97	**20.55**	20.57

Table 3: Generalization error on the Phoneme data set.

	Number of kernels				
Algorithm	4	6	8	10	12
PRBF	24.49	17.37	12.83	11.69	9.42
$\alpha = 5$	19.11	12.26	**10.25**	9.40	9.41
$\alpha = 10$	**14.52**	**10.26**	10.80	9.11	9.70
$\alpha = 15$	14.80	12.83	10.82	9.40	9.70
$\alpha = 20$	14.80	12.83	10.25	**8.83**	**9.11**

Table 4: Generalization error on the Ionosphere data set.

References

[1] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society B*, vol. 39, pp. 1-38, 1977.

[2] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[3] M. Titsias, A. Likas, "A Probabilistic RBF network for Classification", *Proc. of International Joint Conference on Neural Networks*, Como, Italy, July 2000.

[4] M. Titsias, A. Likas, "Shared Kernel Models for Class Conditional Density Estimation", *IEEE Trans. on Neural Networks*, vol. 12, no. 5, pp. 987-997, Sept. 2001.

[5] M. Titsias, A. Likas, "Class Conditional Density Estimation Using Mixtures with Constraint Component Sharing", Tech. Rep. 8-2001, Dept. of Computer Science, Univ. of Ioannina, 2001.

[6] N. A. Vlassis and A. Likas, "A Greedy-EM Algorithm for Gaussian Mixture Learning", *Neural Processing Letters*, to appear.