Neural networks to Estimate ML Multi-class Constrained Conditional Probability Density Functions

Juan Ignacio Arribas*, Jesus Cid-Sueiro*, Tulay Adali** and Anibal R. Figueiras-Vidal***
*Dept. of Teoría de la Señal, Comunicaciones e Ing. Telemática,
Universidad de Valladolid, Valladolid, Spain.
**Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, USA
***Dept. of Tecnologías de las Comunicaciones,
Universidad Carlos III de Madrid, Madrid, Spain.
email: jarribas@tel.uva.es, jescid@tel.uva.es, adali@engr.umbc.edu, arfv@ing.uc3m.es

Abstract

In this paper, a new algorithm, the Joint Network and Data Density Estimation (JNDDE), is proposed to estimate the 'a posteriori' probabilities of the targets with neural networks in multiple classes problems. It is based on the estimation of conditional density functions for each class with some restrictions or constraints imposed by the classifier structure and the use Bayes rule to force the a posteriori probabilities at the output of the network, known here as a implicit set. The method is applied to train perceptrons by means of Gaussian mixture inputs, as a particular example for the Generalized Softmax Perceptron (GSP) network. The method has the advantage of providing a clear distinction between the network architecture and the model of the data constraints, giving network parameters or weights on one side and data over parameters on the other. MLE stochastic gradient based rules are obtained for JNDDE. This algorithm can be applied to hybrid labeled and unlabeled learning in a natural fashion.

Introduction

It is a well known fact that when a neural network is trained in order to minimize the mean square error or the cross entropy between the target and the network outputs, it provides after learning, estimates of the 'a posteriori' probabilities of the classes. The resulting algorithm establishes some bridge between parametric and non-parametric techniques of a posteriori probability estimation. Applications such as medical diagnosis, financial data analysis and communications can exploit this property.

Following [3], we will say that a classifier is *Strict Sense Bayesian* (SSB) if its outputs are estimates of the a posteriori probabilities of the classes. In a similar way, for a training viewpoint, we will say that a cost function is SSB if it is minimized when the classifier is SSB.

It is well-known, that the quadratic and cross entropy costs are SSB. Several authors, [1] and [2], have deduced general expressions for the SSB cost functions for binary and multi-class problems. However, up to the knowledge of the authors, there is little work on the comparison between SSB costs and on the comparison between this and other Bayesian approaches to probability estimation.

In [3] and [4], a SSB cost function is defined so as to have a unique minimum when output **y** coincides with a posteriori class probabilities, which we are trying to determine. There, it is also demonstrated that it is necessary and sufficient for a cost function to be SSB to have the following form

$$C(\mathbf{y},\mathbf{d}) = \sum_{i=1}^{L} \int_{d_i}^{y_i} g(\alpha)(\alpha - d_i)d\alpha + r(\mathbf{d})$$

where $g(\alpha)$ is any positive function which does not depend on **d**, and $r(\mathbf{d})$ is an arbitrary function which does not depend on **y**. Additional constraints must be imposed to cost C in order to guarantee its concavity, as proved in [3], which could be a desirable property of the cost function.

In [3], [4],[5] and [7], any cost function providing a posteriori class probabilities is called *Strict Sense Bayesian*

0-7803-5529-6/99/$10.00

(SSB). When the a posteriori probabilities are estimated using SSB cost functions, no assumptions about the data distribution are required: probabilities are estimated without estimating the conditional density functions of the different data classes. It is known, however, that this states some problems:

- The estimation of a posteriori probabilities in well-separated data requires very large training sets.
- No previous knowledge about the data distribution can be used.
- It is difficult to use unlabeled data to improve learning.

The main idea is searching the conditional data density functions that best fit the data, between a large set of implicit densities defined by network weights **w** and over-parameters **o** of the data density model.

The use of over-parametric densities is essential to generalize the data density model without increasing the complexity of the classifier structure Among the big number of imaginable neural network structures we have considered the *Generalized Softmax Perceptron* (GSP) here. In [6] and [7] we considered others. Let us now center our discussion in the GSP network, defined as follows.

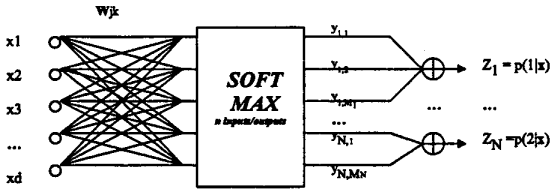

Fig 1.- The GSP Network

The Generalized Softmax Perceptron (GSP)

In the way stated in [6], some not general enough networks have a limited classification capability: for instance, in a binary class problem, only hyper-plane decision boundaries are possible. A more general classifier can be obtained by computing the class outputs as the sum of several softmax outputs (see Fig. 1).

Applying Bayes formula we have,

$$p(j|\mathbf{x}) = \frac{p(j)f(\mathbf{x}|j)}{\sum_{i=1}^{N} p(i)f(\mathbf{x}|i)} \qquad (1)$$

where N is the number of classes, $p(j)$ is the *a priori* probability of class j, $\mathbf{x}$ is an arbitrary point of the sample space and f denotes a density function.

We are considering this particular network architecture, because it is specially appropriate under some conditions. First of all, it is general enough, so we can obtain any kind of boundary and not only hyper-planes. If we suppose he have classes which are Gaussian mixtures and we use the right dimensionality of the GSP net, i.e. enough number of elements in the mixture, we could compute the a posteriori class probabilities for those distributions.

Even more, given that one can approximate any distribution with a general enough Gaussian mixture, we could estimate more general distribution probabilities with this network, so GSP has a potential ability of estimating general class distributions.

As we want that outputs Z_j of GSP network are estimations of a posteriori class probabilities, we arrive to:

$$Z_j = g_j(\mathbf{x},\mathbf{w}) = \frac{p(j)f(\mathbf{x}|j)}{\sum_{i=1}^{N} p(i)f(\mathbf{x}|i)} = p(j|\mathbf{x}) \quad j=1,\ldots,N \qquad (2)$$

where Z_j are the outputs of GSP network, $g(\mathbf{x},\mathbf{w})$ represents the function which links inputs and outputs in the network and **w** is the network parameter vector.

***DEFINITION**: Conditional density functions {f(x\|j), j=1...N} form an implicit set for a network given by outputs $g_j(x,w)$ if equation (2) is satisfied for some parameter vector w.*

If we want outputs of GSP to be an implicit set, then we immediately arrive to:

$$Z_j = \frac{\sum_{k=1}^{M_j} e^{\mathbf{w}_{j,k}{}^T\mathbf{x}}}{\sum_{i=1}^{N}\sum_{k=1}^{M_i} e^{\mathbf{w}_{i,k}{}^T\mathbf{x}}} = \frac{p_j f(\mathbf{x}|i)}{\sum_{i=1}^{N} p_i f(\mathbf{x}|j)} \quad j=1,\ldots,N \qquad (3)$$

where M_j are the number of elements within j-th class, also called subclass number, and N is the number of classes.

***THEOREM**: Density functions $f_j(x)$ form an implicit set for the GSP network, if and only if*

$$f_j(\mathbf{x}) = \frac{f_c(\mathbf{x})\sum_{k=1}^{M_j} e^{\mathbf{w}_{j,k}{}^T\mathbf{x}}}{\int f_c(\mathbf{x})\sum_{k=1}^{M_j} e^{\mathbf{w}_{j,k}{}^T\mathbf{x}}d\mathbf{x}} \quad j=1,\ldots,N \tag{4}$$

where j is class index, M_j is the number of elements within j-th class, x the input sample, w the network weights and $f_c(x)$ an arbitrary probability density function, called Central Density.

Proof: The proof is easy and omitted. ■

Similar theorems can be obtained for different networks, like those based in the Softmax function described in [6].

It is important to remark, that function $f_c(x)$ is a density function because it has unit area and is positive semi-definite but it does not represent any real data distribution, neither a joint data density nor any conditional density, though for obvious reasons real data densities will depend on it.

In the next two sections, we are going to study the JNDDE for the GSP network, and the problem of estimating a posterior probabilities, for a particular case of the *Central Density*, $f_c(x)$: the Gaussian Mixture.

Example: the Gaussian Mixture

Some authors, have already fully investigated Mixtures , [8] and [9] are two good examples, and have obtained some important results. As the Mixture literature is gaining importance each and every day, that is the reason why we show and example concerning Gaussian Mixtures.

Let us use a special case of *Central Density* function, $f_C(\mathbf{x})$, the *Gaussian Mixture* for the case of multi-dimensional input samples, where σ_m represents the standard deviation:

$$f_c(\mathbf{x}) = \sum_{m=1}^{L} q_m N(\mathbf{x}-\mathbf{z}_m, \sigma_m{}^2)$$
$$N(\mathbf{x}-\mathbf{z}_m, \sigma_m{}^2) = \frac{1}{\sqrt{2\pi}\sigma_m} e^{-\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{z}_m\|^2}{\sigma_m{}^2}} \tag{5}$$

where L is the number of mixture components, N is the well known Normal Distribution, with restrictions as follow

$$\sum_{m=1}^{L} q_m = 1 \quad q_m > 0 \tag{6}$$

We are now interested in calculating the precise formulae for the conditional class probability density functions of each class. To do so, we must first compute the following expressions, (7) and (8), whose elementary calculus details have been omitted for the shake of clarity:

$$f_c(\mathbf{x})e^{\mathbf{w}_{j,k}^T\mathbf{x}} = \sum_{m}^{L} q_m e^{\frac{1}{2}\sigma_m{}^2\mathbf{w}_{j,k}^T\mathbf{w}_{j,k}+\mathbf{w}_{j,k}^T\mathbf{z}_m} N(\mathbf{x}-\mathbf{v}_{j,k,m}, \sigma_m{}^2) \tag{7}$$

$$\int f_c(\mathbf{x})e^{\mathbf{w}_{j,k}^T\mathbf{x}}d\mathbf{x} = \sum_{m=1}^{L} q_m e^{\frac{1}{2}\sigma_m{}^2\mathbf{w}_{j,k}^T\mathbf{w}_{j,k}+\mathbf{w}_{j,k}^T\mathbf{z}_m} \tag{8}$$

and let us define probability scalar $\pi_{j,k,m}$ and the modified vector mean term $\mathbf{v}_{j,k,m}$ for convenience in (9) and (10) respectively, where again j is the class index, k the sub-class index and m the mixture component index:

$$\pi_{j,k,m} \equiv \frac{q_m e^{\frac{1}{2}\sigma_m{}^2\mathbf{w}_{j,k}^T\mathbf{w}_{j,k}+\mathbf{w}_{j,k}^T\mathbf{z}_m}}{\sum_{i}^{M_j}\sum_{n}^{L} q_n e^{\frac{1}{2}\sigma_m{}^2\mathbf{w}_{j,k}^T\mathbf{w}_{j,k}+\mathbf{w}_{j,k}^T\mathbf{z}_m}} \tag{9}$$

$$\mathbf{v}_{j,k,m} = \mathbf{z}_m + \sigma_m{}^2\mathbf{w}_{j,k} \tag{10}$$

Using these newly defined terms, we can now express in an appropriate closed form the conditional probability density functions of each class as follows:

$$f_{\mathbf{w},\mathbf{o}}(\mathbf{x}\,|\,j) = \sum_{k}^{M_j}\sum_{m}^{L} \pi_{j,k,m}(\mathbf{w},\mathbf{o})N(\mathbf{x}-\mathbf{v}_{j,k,m}(\mathbf{w},\mathbf{o}), \sigma_m{}^2) \tag{11}$$

where we have explicitly shown the dependence of density functions over network parameters **w** as well as data model over-parameters **o** in the spirit of JNDDE algorithm.

Gradient based learning rule

Now that we have obtained models for conditional densities in (11) as well as for joint density, computed from conditionals, what we do next is to estimate the parameters for these densities using Maximum Likelihood.

It is good to recall that in the parameter set we distinguish two subsets, one for the network model (**w**) and another data density model (**o**) but only first subset is going to affect to the probabilities once convergence is achieved.

We wish to find the partial derivatives of the conditional class probabilities, with respect to each of the parameters, in order to determine the learning rule to be used in the simulations, for the case of the GSP network with a Gaussian Mixture as Central Density, making use of

previously defined probability $\pi_{j,k,m}$, and defining two new probability terms, $\beta_{j,k}$ and γ_m:

$$\beta_{j,k} \equiv \frac{e^{\mathbf{w}_{j,k}^T \mathbf{x}}}{\sum_i^{M_j} e^{\mathbf{w}_{j,k}^T \mathbf{x}}} \qquad \gamma_m \equiv \frac{\frac{q_m}{\sigma_m} e^{\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{z}_m\|^2}{\sigma_m^2}}}{\sum_n^L \frac{q_n}{\sigma_n} e^{\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{z}_n\|^2}{\sigma_n^2}}} \tag{12}$$

The partial derivatives which result, are the following:

$$\nabla_{\mathbf{w}_{j,k}} f(\mathbf{x} \mid j) = \beta_{j,k}\mathbf{x} - \sum_{m=1}^{L} \pi_{j,k,m}\mathbf{v}_{j,k,m} \tag{13}$$

$$\nabla_{q_p} f(\mathbf{x} \mid j) = \frac{\gamma_m}{q_m} - \sum_k^{M_j} \frac{\pi_{j,k,m}}{q_m} \tag{14}$$

$$\nabla_{\mathbf{z}_m} f(\mathbf{x} \mid j) = \frac{\mathbf{x}-\mathbf{z}_m}{\sigma_m^2}\gamma_m - \sum_k^{M_j} \pi_{j,k,m}\mathbf{w}_{j,k} \tag{15}$$

$$\nabla_{\sigma_m} f(\mathbf{x} \mid j) = \gamma_m \left[\frac{\|\mathbf{x}-\mathbf{z}_m\|^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right] - \sigma_m \sum_k^{M_j} \pi_{j,k,m}\mathbf{w}_{j,k}^T\mathbf{w}_{j,k} \tag{16}$$

To obtain these set of equations, we have made use of some basic algebra and calculus. All previous formulae, can be generalized to multi-dimensional inputs straightforward.

Conclusions

A new parametric density estimation algorithm has been proposed to train neural classifiers. We have developed the theoretical formulation in order to establish the MLE gradient learning rules for several cases of the OPDE method. The method requires making some hypothesis about implicit density functions, but this can be assumed as general as desired. The resulting method is an alternative to Strict Sense Bayesian (SSB) Probability Estimation method in several situations, showing more robust convergence, as demonstrated through computer simulation.

Acknowledgements

This work has been partially supported by the Spanish Government under C.I.C.Y.T. grant numbers TIC96-0500-C10-03, TIC96-0500-C10-09 and TIC97-0772. T. Adali's work was partially supported by the National Science Foundation Career Award, NSFNCR-9703161.

References

[1] D.W.Ruck, S.K.Rogers, M.Kabrisky, M.E.Oxley and B.W.Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function", **IEEE Transactions on Neural Networks**, vol. 1, no. 4, pp. 296-298, 1990.

[2] P. Smyth, J.W. Miller, R. Goodman, "Objective functions for probability estimation", **Proceedings of the Int. Joint Confefernce on Neural Networks**, pp. 881-886, 1991.

[3] J. Cid-Sueiro, J. I. Arribas, S. Urban-Muñoz, and A. R. Figueiras-Vidal , "Cost Functions to Estimate A Posteriori Probabilities in Multiclass Problems" **IEEE Transactions on Neural Networks**, May 1999, Volume 10, Number 03, p. 645.

[4] J.Cid-Sueiro, A.R. Figueiras-Vidal, "Cost Functions to Estimate Class Probabilities", **Proceedings of the European Conference on Signal Analysis and Prediction (ECSAP'97)**, pp. 113-116, Praha, 24-27 June, 1997.

[5] J. I. Arribas , J. Cid-Sueiro, "Bayesian Approaches to the Estimation of A Posteriori Probabilities", **Proceedings of the IASTED International Conference on Signal Processing and Communications**, pp. 107-110, Canary Islands, Spain, February 1998.

[6] J.Ignacio Arribas, Jesus Cid-Suerio, Tülay Adali and Anibal R. Figuerias-Vidal, "Neural architectures for parametric estimation of a posteriori probabilities by constrained conditional density functions", **Proceedings of the IEEE Workshop on Neural Networks**, Madison, Wisconsin, USA, 1999.

[7] Juan I. Arribas, Jesús Cid-Sueiro and Anibal R. Figueiras-Vidal, "Estimation of Posteriori Class Probabilities by Means of Constrained Probability Density Functions", **International Workshop Learning'98**, Getafe, Madrid, Spain, 1998.

[8] David Miller and Hasan S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data", **Neural Information Processing Systems 9**, pp. 571-577, 1996.

[9] Michael I. Jordan, "Why the logistic function? A turotial discussion on probabilities and neural networks", **M.I.T., Computational Cognitive Science Technical Report 9503**, August 13, 1995.