A Self-Organising Mixture Network for Density Modelling

Hujun Yin and Nigel M. Allinson
Dept. of Electrical Engineering and Electronics, UMIST
P.O. Box 88, Manchester, M60 1QD, UK
h.yin@umist.ac.uk, allinson@umist.ac.uk

Abstract

A completely unsupervised mixture distribution network, namely the self-organising mixture network, is proposed for learning arbitrary density functions. The algorithm minimises the Kullback-Leibler information by means of stochastic approximation methods. The density functions are modelled as mixtures of parametric distributions such as Gaussian and Cauchy. The first layer of the network is similar to the Kohonen's self-organising map (SOM), but with the parameters of the class conditional densities as the learning weights. The winning mechanism is based on maximum posterior probability, and the updating of weights can be limited to a small neighbourhood around the winner. The second layer accumulates the responses of these local nodes, weighted by the learning mixing parameters. The network possesses simple structure and computation, yet yields fast and robust convergence. Experimental results are also presented.

1. Introduction

In a completely unsupervised situation where there is little or no prior knowledge about data properties except for the data samples themselves, the joint pattern distribution can be often considered or modelled as a mixture of some parametric forms such as Gaussians [1]. Such a method has also provided a general strategy for designing and training a complex learning system and has extended the single network approach to a modular architecture approach such as *mixture of experts* networks [2]. This provides a trade-off between simple and limited *parametric* approaches and computational intensive *nonparametric* approaches. In some cases such as pattern classification, there is also a need for solving individual conditional distributions, which neither parametric nor non-parametric approaches are capable. The form of individual conditional densities or components of the mixture is usually assumed to be some popular functions, e.g. Gaussian, Cauchy, Laplace. The parameters for each component density, however, have to be derived solely from the data samples. Xu and Jordan [3] applied the expectation-maximisation (EM) method to this kind of problems, produced an EM algorithm for Gaussian mixtures and showed its advantages over other algorithms. Yin and Allinson have recently proposed a Bayesian SOM for solving Gaussian mixture problems, and have shown additional advantages (e.g. less local minima, and much faster convergence speed) over the EM algorithm [4, 5].

In this paper, we extend and generalise the learning principle in the Bayesian SOM to any kind of mixture distributions. The resulting network, the self-organising mixture network (SOMN), combines the criterion of minimising the Kullback-Leibler information [6], stochastic approximation method, and the SOM [7] structure. The resulting algorithms require simple scalar and local calculation, and hence are computational efficient, converge fast, and have good noise tolerant properties.

2. The Mixture Distribution and Unsupervised Learning

2.1. Mixture distributions

The mixture model has been employed in many practical pattern classification applications, e.g. [1] and [2]. In a mixture model, each sample, $\mathbf{x}$, from a d-dimensional input space, $\Omega\in\mathbf{R}^d$, is assigned to one of K distinct classes, each of which has a prior probability P_i. In each pattern class, samples are distributed according to a prescribed class-conditional probability density. The joint-probability density of data samples is given by [8]

$$p(\mathbf{x}|\Theta)=\sum_{i=1}^{K}p_i(\mathbf{x}|\theta_i)P_i \qquad (1)$$

where $p_i(\mathbf{x}|\theta_i)$ is the i-th class-conditional density, and θ_i are the sufficient statistics or parameter vector, for the i-th class-conditional density, i=1, 2, ... K. Θ=(θ_1, θ_2, ...

0-7803-4859-1/98 $10.00©1998 IEEE

$\theta_K)^T$. P_i is the prior probability of the i-th class and is also called the mixing parameters or weights. For a Gaussian or Cauchy mixture, the conditional density has the following forms respectively,

$$p_i(\mathbf{x}|\boldsymbol{\theta}_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}-\mathbf{m}_i)} \qquad (2)$$

$$q_i(\mathbf{x}|\boldsymbol{\theta}_i) = \frac{1}{\pi|\Sigma_i|^{1/2}\,[1+(\mathbf{x}-\mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}-\mathbf{m}_i)]} \qquad (3)$$

where $\boldsymbol{\theta}_i=\{\theta_{i1}, \theta_{i2}\}=\{\mathbf{m}_i, \Sigma_i\}$ are the mean vector and covariance matrix of the i-th Gaussian or Cauchy conditional density respectively.

2.2. Maximum likelihood estimation and the EM algorithm

For most unsupervised learning applications, only the form of their class-conditional density are known; the other parameters have to be learnt, unsupervised, from a set of N unlabelled independent samples, $\Omega=\{\mathbf{x}(1), \mathbf{x}(2), \ldots\mathbf{x}(N)\}$. In these cases, *maximising* the joint-*likelihood* (ML) of all observed samples, $p\{\Omega|\Theta) = \prod_{k=1}^{N} p(\mathbf{x}(k)|\Theta)$, may lead to a singular solution. When restricted to the largest finite maximum and Gaussian components, it results in some implicit equations for these parameters [8].

The EM algorithm is an iterative maximum likelihood procedure for parameter estimation under incomplete data or missing data situations [9]. Many problems can be viewed as instances of such situations. For example, in the unsupervised learning for the mixture distribution model, the input samples are incomplete, the missing data are the class-labels or indicator functions for each sample. By using the EM procedure, the marginal, or incomplete-data, likelihood is obtained by the average or expectation of the complete-data likelihood respect to the missing data under the current parameter estimates (E-step), then the new parameter estimates are obtained by maximising the marginal likelihood (M-step). The EM algorithm has been shown to be an iterative gradient ascent algorithm, in which the likelihood function exhibits no decrease after each iteration [9].

The EM method has been applied to unsupervised parameter estimation of Gaussian mixtures by Xu and Jordan [3]. The resulting algorithms coincide with Duda and Hart's earlier suggestion ([8], see Section 2.2). It is an extended and generalised k-means algorithm with considerations of class-conditional distribution and priors, thus will generally result in improved clustering than the k-means algorithm. Xu and Jordan [10, 11] have shown that this EM algorithm is a variable metric gradient ascent algorithm with first-order convergence. They have also acknowledged the slow convergence of the algorithm, especially when the mixture components are not well separated, but found that faster methods such as superlinear and Hessian gradient generally performed poorly for this kind of ill-conditioned problems.

The EM algorithm provides a feasible solution to this kind of unsupervised learning problem. But its slow convergence and high computational costs need to be addressed for practical applications.

3. The Self-Organising Mixture Network

3.1. The SOMN structure

Based on the mixture distribution model, i.e. Eqn. (1), the SOMN structure can be illustrated as in Fig. 1. For a mixture of finite components, the network Ξ places K nodes in the input space, Ω. The kernel parameters, e.g. mean vectors, $\hat{\mathbf{m}}_i$ and covariance matrix, $\hat{\Sigma}_i$, are the learning weights. The output of a kernel is the conditional density of that component in the mixture. The upper layer, or the network output, sums the responses of these kernel weighted by the prior probability or mixing weight, $\hat{P}_i$, which are also learning parameters. At each time step, n, a sample, denoted by $\mathbf{x}(n)$, is randomly taken from Ω. A winner is chosen according to its kernel output multiplied by its mixing parameter, i.e. estimated posterior probability. Within a neighbourhood of the winner, η_c, the weights are updated. Thus the SOMN is similar to the SOM in terms of its local learning properties.

The number of nodes, however, needs not to be known *a priori*, but has to be equal to or larger than the number of underlying components in the mixture. That is, one can always use a large number of nodes to learn the mixture, and only the signification ones will remain. Such a number can be an objective factor in the learning. For a smooth estimation of an arbitrary density, one can use a large number of nodes. For mixtures with a known class number or where only a number of major or principal classes need to be traced, a SOMN with this number of components can be used to interpret interested sub-densities of individual classes.

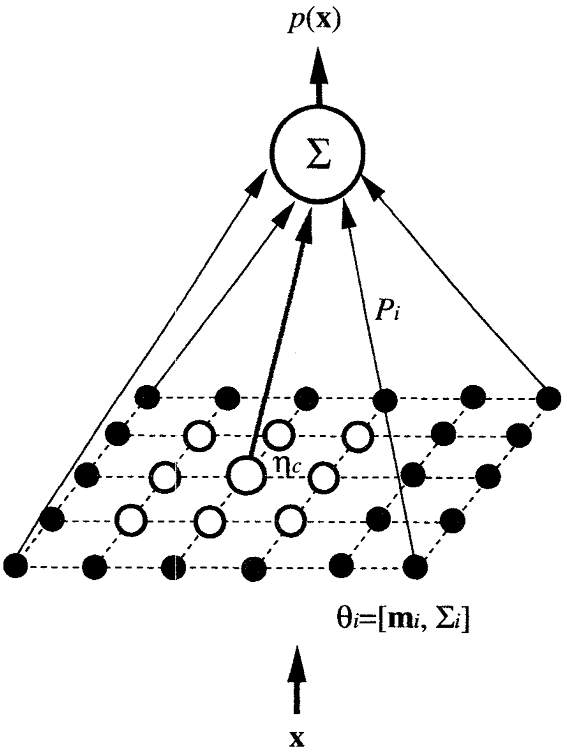


Figure 1. Structure of the self-organising mixture network

3.2. The SOMN updating algorithm

Suppose that the true environmental data density function and the estimated one are $p(\mathbf{x})$ and $\hat{p}(\mathbf{x})$ respectively. Kullback-Leibler information metric [6] measures the divergence or 'distance' between these two, and is defined as:

$$\mathbf{I} = -\int \log \frac{\hat{p}(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x} \qquad (4)$$

It is also referred to as *relative entropy*, and is an expectation of the negative log-likelihood in the limit of an infinite number of data points subtracting a bias which is known as the entropy of the data density. It measures the average information remaining in each data point by the estimator. It is always a positive number, and will be zero if and only if the estimated density equals the true one. It has been shown that this criterion is equivalent to the ML or quasi-ML criterion [1] and has provided an important criterion in density or unsupervised learning [12-14].

When the estimated density is modelled as a mixture distribution, a function of various sub-densities and their parameters, one can seek the optimal estimate of these parameters by minimising **I** via its partial differentials in respect to every model parameter, i.e.

$$\frac{\partial \mathbf{I}}{\partial \theta_{ij}} = -\int [\frac{1}{\hat{p}(\mathbf{x}|\Theta)} \frac{\partial \hat{p}(\mathbf{x}|\Theta)}{\partial \theta_{ij}}] p(\mathbf{x}) d\mathbf{x} \equiv 0, \quad i=1, 2, \ldots K, \quad \textit{and } j=1, 2 \qquad (5)$$

$$\frac{\partial \mathbf{I}}{\partial \hat{P}_i} = -\int [\frac{1}{\hat{p}(\mathbf{x}|\Theta)} \frac{\partial \hat{p}(\mathbf{x}|\Theta)}{\partial \hat{P}_i}] p(\mathbf{x}) d\mathbf{x} + \lambda \frac{\partial}{\partial \hat{P}_i} [\sum_{j=1}^{K} \hat{P}_j - 1]$$

$$= -\frac{1}{\hat{P}_i} \int [\frac{\hat{P}_i \hat{p}_i(\mathbf{x}|\boldsymbol{\theta}_i)}{\hat{p}(\mathbf{x}|\Theta)} - \lambda \hat{P}_i] p(\mathbf{x}) d\mathbf{x} \equiv 0, \quad i=1, 2, \ldots K \qquad (6)$$

where θ_{ij} represents the jth parameter of the ith conditional class density, e.g. mean vector and covariance matrix for j=1 and 2 respectively. In Eqn. (6), the method of Lagrange multipliers with constraint parameter λ is used to ensure the constraint of a valid probability, i.e. $\sum_{i=1}^{K} \hat{P}(\omega_i) = 1$.

The Robbins-Monro stochastic approximation method [15] can be used for solving these non-directly solvable equations, and this results in the following adaptive updating algorithm:

$$\theta_{ij}(n+1) = \theta_{ij}(n) + \alpha(n)[\frac{1}{\hat{p}(\mathbf{x}|\Theta)} \frac{\partial \hat{p}(\mathbf{x}|\Theta)}{\partial \theta_{ij}(n)}]$$

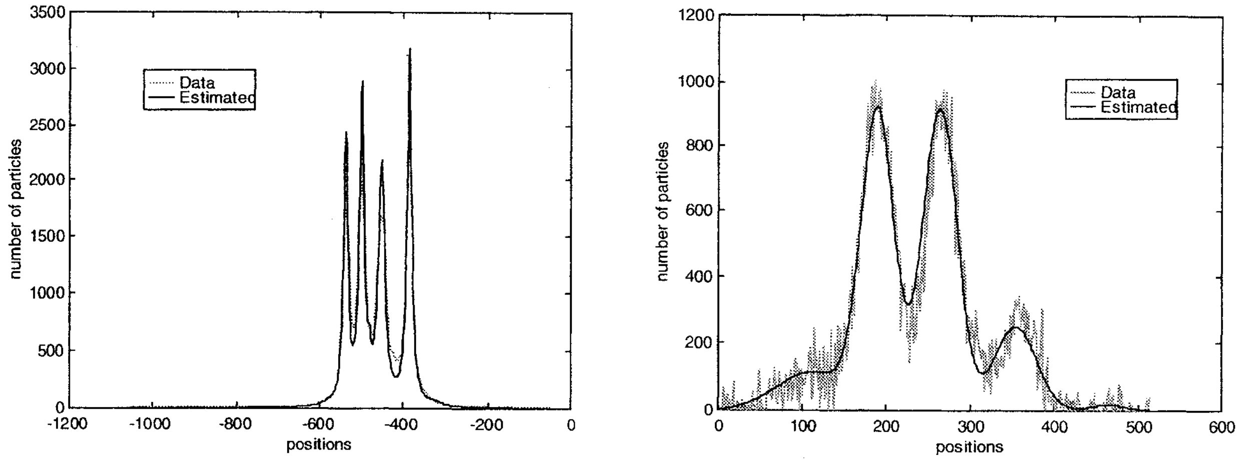
$$= \theta_{ij}(n) + \alpha[\frac{\hat{P}_i(n)}{\hat{p}(\mathbf{x}|\Theta)} \frac{\partial \hat{p}_i(\mathbf{x}|\boldsymbol{\theta}_i)}{\partial \theta_{ij}(n)}] \qquad (7)$$

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \alpha(n)[\frac{\hat{p}_i(\mathbf{x}|\boldsymbol{\theta}_i)\hat{P}_i(n)}{\hat{p}(\mathbf{x}|\Theta)} - \hat{P}_i(n)] \qquad (8)$$

where $\alpha(n)$ is the learning coefficient or rate, and the constraint parameter λ is set to 1.

It is straight forward to calculate the corresponding partial differential terms in Eqn. (7) for a specified conditional model, e.g. Eqn. (2) or (3).

The updating of the above parameters can be limited to a small neighbourhood of a winning node, which has the largest response or posterior probability, due to the spreading properties of the most conditional densities.



(a)

(b)

Figure 2. Density estimates using the Gaussian and Cauchy mixture models.

That is, the density can be approximated by a mixture of a small number of nodes at one time, i.e.

$$p(\mathbf{x}|\Theta) \approx \sum_{i\in\eta_c} \hat{p}_i(\mathbf{x}|\theta_i)\hat{P}_i \qquad (9)$$

where c is the winning node index and η_c is a neighbourhood of the winner.

4. Experimental Results

The SOMN has been applied successfully to various real problems such as clustering, texture classification, density estimation, and peak location [16]. Some typical results on density modelling using various mixture models are shown in Fig. 2.

Fig. 2(a) shows the profile of an x-ray rocking curve (i.e. the profile of an x-ray diffraction peak as a function of photon energy). Though in reality a spectrum, the experimental data points (for the network learning) were sampled randomly to provide an effective density histogram. The network, initially assigned 20 nodes as the number of peaks is assumed unknown, has successfully learnt the four main peaks using the Cauchy mixture (the Gaussian mixture can also be used, but the network requires many more nodes for a smooth interpretation and will result in larger errors). The estimated density (solid line) is after only five epochs. The four main components after five epochs are in positions, -537.8, -498.5, -452.1, and -386.7 with standard variances of 6.25, 6.69, 8.99, and 5.86 and mixing weights of 0.205, 0.253, 0.258, and 0.268 respectively. Other nodes have resulted in very small influence (less than 2%).

In Fig. 2(b), a Gaussian mixture with five components is used to learn a noisy snapshot image from a capillary electrophoresis system. As only two main peaks are clearly visible before the learning, a five-Gaussian-node SOMN is used for this data. The figure shows the results after five learning epochs. Two major peaks have been correctly and accurately located, and other peaks have also been revealed. An important feature of the SOMN is that it can simultaneously provide the width information about each peaks, which are in many cases important.

In the above two experiments, only the winning node and its two neighbouring nodes (one on each side) are updated at each iteration. To prevent the variances and mixing parameters going to singular and zero respectively, or vary drastically between consecutive samples, it is better to use small initial learning rates. However, it has been found that the network converges over a wide range of learning rates.

5. Conclusions

An unsupervised learning structure, based on the criterion of maximising the Kullback-Leibler information entropy, the stochastic approximation method and the SOM principle, is proposed for estimating general densities by means of mixture distribution models. As shown in other papers [4, 5] the Gaussian SOMN outperforms the EM algorithm in both convergence speed and robustness; and can be regarded as a more generalised and adaptive version of the EM algorithm for unsupervised learning and data density modelling. Like the SOM algorithm, the SOMN is a computational simple algorithm and is easy to implement. Its neighbourhood conscience learning and locality of the kernel function provides the network with efficient computation. The algorithm with either Gaussian or Cauchy (or others) mixture models employed) resemble the SOM algorithm in wieght updating, that is, only scalar neighbourhood functions rather than matrix ones are required, though such scalar neighbourhood function may vary form model to model. It has been shown that the neighbourhood learning will provide the SOM with a certain annealing effect in searching for a better estimate [17] and can also result in good noise tolerance[18]. This SOMN algorithm also provides some insights and quantitative analysis to the SOM's neighbourhood functional role. For example, in the Gaussian SOMN, the neighbourhood functions equal the posterior probabilities of the mixture components, and the network will converge to a mixture of Gaussian distributions.

We greatly acknowledge the provision of x-ray data by Prof. K. E. Singer, and capillary electrophoresis data by Dr. B. Pokric and Dr. E. Bergstrom.

6. References

[1] D.M. Titterington, A.F.M. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions.* John Wiley, New York, 1985.

[2] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts". *Neural Computation*, vol. 3, 1991, pp. 79-87,

[3] L. Xu. and M.I. Jordan, "Unsupervised learning by EM algorithm based on finite mixture of Gaussians," *Proceedings of World Congress on Neural Networks* (II), 1993, pp. 431-434).

[4] H. Yin and N. M. Allinson, "Bayesian learning for self-organising maps". *Electronics Letters*, vol. 33, 1997, pp. 304-305.

[5] H. Yin and N. M. Allinson, "Comparison of a Bayesian SOM with the EM algorithm for Gaussian mixtures," *Proc. WSOM'97*, 1997, pp. 304-305.

[6] S. Kullback and R.A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, 1951, pp. 79-86.

[7] T. Kohonen, "Self-organised formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, 1982, pp. 56-69.

[8] R.O. Duda and PE. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete date via the EM algorithm". *Journal of Royal Statistical Society,* B, vol. 39, 1977, pp. 1-38.

[10] L. Xu and M.I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation.* vol. 8, 1996, pp. 129-151.

[11] M.I. Jordan and L. Xu, "Convergence results for the EM approach to mixture of experts architectures," *Neural Networks*, vol. 8(9), 1995, pp. 1409-1431.

[12] M. Benaim and L. Tomasini, "Competitive and self-organising algorithms based on the minimisation of an information criterion," *Proc ICANN'91*, 1991, pp. 391-396.

[13] V.N. Vapnik, *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[14] H. White, "Learning in artificial neural networks," *Neural Computation*, vol. 1, 1989, pp. 425-464.

[15] H. Robbins and S. Monro, "A stochastic approximation method". *The Annals of Mathematical Statistics*, vol. 22, 1951, pp. 400-407.

[16] H. Yin and N.M. Allinson, "A Bayesian self-organising map for unsupervised learning in finite Gaussian mixtures," Submitted to *Neural Networks*, 1997.

[17] T.M. Martinetz, G.B. Stanislav, and K.L. Schulten, "'Neural-gas' network for vector quantisation and its application to time-series prediction," *IEEE Transactions on Neural Networks*, vol. 4(4), 1993, pp. 558-569.

[18] S.P. Luttrell, A Bayesian analysis of self-organising map. *Neural Computation*, vol. 6, ,1994, pp. 767-794.