Mixture conditional density estimation with the EM algorithm

Nikos Vlassis Ben Kröse

RWCP, Autonomous Learning Functions SNN
Dept. of Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
E-mail: {vlassis, krose}@wins.uva.nl

Abstract

It is well-known that training a neural network with least squares corresponds to estimating a parametrized form of the conditional average of targets given inputs. In order to approximate multi-valued mappings, e.g., those occurring in inverse problems, a mixture of conditional densities must be used. In this paper we apply the EM algorithm to fit a mixture of Gaussian conditional densities when the parameters of the mixture, i.e., priors, means, and variances are all functions of the inputs. Our method becomes an interesting alternative to previous approaches based on nonlinear optimization.

1 Introduction

Solving regression problems with neural networks usually involves minimizing a sum-of-squares error between targets and the network outputs, which can be shown [3] that is equivalent to fitting a conditional probability density function of targets given inputs, parametrized on its mean. In case this parametrization is in the form of a generalized linear model, e.g., an RBF network with fixed basis functions, training is carried out in a oneshot step by using matrix techniques [8, 7].

Two useful extensions to the basic model involve parametrizing also the variance of the conditional density [5, 4] and allowing the noise in the targets to be non-Gaussian [2]. In the latter general case, the parameters of the model can be estimated by maximizing the likelihood of the conditional density using some nonlinear optimization method, e.g., the BFGS quasi-Newton algorithm [2].

In this paper we propose an alternative solution to the above problem of estimating a general conditional density function of targets given inputs, in the form of a Gaussian mixture whose parameters, namely, mixing weights, means, and variances are all functions of the input data. We assume a fixed set of basis functions regularly positioned in the input space, and choose appropriate models for the parameters of the mixture which we subsequently fit with the EM algorithm [9]. Our approach borrows all the attractive features of EM, as explained in the following, to become a viable alternative for modeling multi-valued or inverse problems.

2 Modeling the conditional density

Assume a training set of supervised pairs $\{\mathbf{x}_n, y_n\}$, $n = 1, \ldots, N$, where the inputs $\mathbf{x}$ are d-dimensional and the outputs are for simplicity assumed scalars. We model the conditional density of targets given inputs $p(y|\mathbf{x})$ as a mixture of K Gaussians

$$p(y|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) p(y|\mathbf{x}, k), \qquad (1)$$

where

$$\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1, \quad 0 < \pi_k(\mathbf{x}) < 1, \quad \forall \mathbf{x}, \quad (2)$$

and

$$p(y|\mathbf{x}, k) = \frac{1}{s_k(\mathbf{x})\sqrt{2\pi}} \exp\left\{\frac{-[y - f_k(\mathbf{x})]^2}{2s_k^2(\mathbf{x})}\right\}. \quad (3)$$

The priors $\pi_k(\mathbf{x})$, the means $f_k(\mathbf{x})$, and the variances $s_k^2(\mathbf{x})$ of all K components are assumed input-dependent and can be regarded as outputs of the same neural network. For the parametrization we choose a set of M radial basis functions $\phi_m(\mathbf{x})$ with parameters, namely, centers and spreads, either fixed in advance or approximated by the centers and covariance matrices of the components of a second mixture density applied on the inputs and trained independently [3].

We choose to parametrize the means with a weighted sum of the basis functions, the variances with an exponential applied on a second weighted sum, and the priors with a sigmoid function applied on a third weighted sum to ensure that variances are always positive and priors between one and zero, thus

$$f_k(\mathbf{x}) = \sum_{m=1}^{M} a_{km}\phi_m(\mathbf{x}), \quad (4)$$

$$s_k^2(\mathbf{x}) = \exp\left[\sum_{m=1}^{M} b_{km}\phi_m(\mathbf{x})\right], \quad (5)$$

$$\pi_k(\mathbf{x}) = \frac{1}{1 + \exp\left[\sum_{m=1}^{M} c_{km}\phi_m(\mathbf{x})\right]}. \quad (6)$$

Our task is to estimate from the training set the parameter vector $\boldsymbol{\theta} = [a_{km}, b_{km}, c_{km}]$, with $k = 1, \ldots, K$ and $m = 1, \ldots, M$. We start by maximizing the log-likelihood of the training set with respect to $\boldsymbol{\theta}$

$$\mathcal{L} = \sum_n \log p(y_n|\mathbf{x}_n) = \sum_n \log \sum_k \pi_k(\mathbf{x}_n) p(y_n|\mathbf{x}_n, k). \quad (7)$$

The appearance of the logarithm between the two sums complicates the direct maximization of the above quantity, so we try step-wise maximization using the EM algorithm [9]. Assuming an estimate of the parameter vector from the previous step, in each step we first use the Bayes' rule to compute the posterior probability $P(k|\mathbf{x}_n, y_n)$ that a target y_n originates from the k-component of the mixture (1)

$$P(k|\mathbf{x}_n, y_n) = \frac{\pi_k(\mathbf{x}_n) p(y_n|\mathbf{x}_n, k)}{\sum_{l=1}^{K} \pi_l(\mathbf{x}_n) p(y_n|\mathbf{x}_n, l)}, \quad (8)$$

and then find a new parameter vector $\hat{\boldsymbol{\theta}}$ that maximizes the expected log-likelihood of the training set, where the expectation is taken with respect to $P(k|\mathbf{x}_n, y_n)$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_n \sum_k [P(k|\mathbf{x}_n, y_n) \log \pi_k(\mathbf{x}_n) + P(k|\mathbf{x}_n, y_n) \log p(y_n|\mathbf{x}_n, k)]. \quad (9)$$

The new parameters $\hat{\boldsymbol{\theta}}$ are re-introduced in (8) to lead to a new estimate from (9) and so on. The whole procedure is repeated until the log-likelihood gets no significant improvement.

2.1 Estimating the means

Maximizing the second term in (9) is equivalent to minimizing the cost

$$C = \sum_n \sum_k \{P(k|\mathbf{x}_n, y_n) \log s_k(\mathbf{x}_n) + P(k|\mathbf{x}_n, y_n) \frac{[y_n - f_k(\mathbf{x}_n)]^2}{2s_k^2(\mathbf{x}_n)}\}, \quad (10)$$

with $f_k(\mathbf{x})$ and $s_k(\mathbf{x})$ from (4) and (5), respectively. Differentiating with respect to a_{kl} and setting to zero we get

$$\sum_n \frac{P(k|\mathbf{x}_n, y_n)}{s_k^2(\mathbf{x}_n)} \phi_l(\mathbf{x}_n) \left[y_n - \sum_m a_{km}\phi_m(\mathbf{x}_n)\right] = 0 \quad (11)$$

which by changing the order of summations gives

$$\sum_n \frac{P(k|\mathbf{x}_n, y_n)}{s_k^2(\mathbf{x}_n)} \phi_l(\mathbf{x}_n) y_n = \sum_m a_{km} \sum_n \frac{P(k|\mathbf{x}_n, y_n)}{s_k^2(\mathbf{x}_n)} \phi_l(\mathbf{x}_n)\phi_m(\mathbf{x}_n). \quad (12)$$

Now if we denote by $\mathbf{y}_k$ the $N \times 1$ target vector with elements

$$\mathbf{y}_k(n) = \frac{\sqrt{P(k|\mathbf{x}_n, y_n)}}{s_k(\mathbf{x}_n)} y_n, \quad (13)$$

$\mathbf{F}_k$ the $N \times M$ design matrix with elements

$$\mathbf{F}_k(n, m) = \frac{\sqrt{P(k|\mathbf{x}_n, y_n)}}{s_k(\mathbf{x}_n)} \phi_m(\mathbf{x}_n), \quad (14)$$

and $\mathbf{a}_k$ the parameter vector for the k-th mean, then the above system of equations can be written in the matrix form

$$\mathbf{F}_k^T \mathbf{y}_k = \mathbf{F}_k^T \mathbf{F}_k \mathbf{a}_k. \tag{15}$$

Note that the quantities $s_k(\mathbf{x}_n)$ appearing in the above matrices are computed based on the estimates of the previous EM step.

The matrices $\mathbf{F}_k$ are the design matrices of the K fitting problems, and since the quantities $\sqrt{P(k|\mathbf{x}_n, y_n)}$ and $s_k(\mathbf{x}_n)$ form part of their elements, the whole procedure can be regarded as a *weighted chi-square fitting* in analogy to the chi-square fitting [8] when the posterior quantities are missing. The solution to (15) is

$$\mathbf{a}_k = (\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T \mathbf{y}_k, \tag{16}$$

which can be found by singular value decomposition to account for possible ill-conditioning. The variance matrices of the mean vectors estimates are $(\mathbf{F}_k^T \mathbf{F}_k)^{-1}$.

2.2 Estimating the variances and the priors

In principle the parameters for the new variances $s_k^2(\mathbf{x})$ in (5) and priors $\pi_k(\mathbf{x})$ in (6) can also be estimated by directly maximizing the expected log-likelihood (9). However this results in nonlinear formulas that have to be maximized with some nonlinear optimization method like in [2].

Alternatively, we propose here a different approach. Since the means have already been estimated from the previous step, we use this information to compute the input-dependent variances as *weighted squared distances* between the targets and the means of each component, weighted by the posterior probabilities $P(k|\mathbf{x}_n, y_n)$ already computed from the E-step. The priors are also computed based on the posterior values.

It can be shown [10, 12] that when inputs arrive sequentially in a Gaussian mixture fitting problem, then the variances and priors can be estimated by the iterative formulas

$$s_k^2(\mathbf{x}_n) := s_k^2(\mathbf{x}_n) + \lambda \frac{P(k|\mathbf{x}_n, y_n)}{\pi_k(\mathbf{x}_n)} \{[y_n - f_k(\mathbf{x}_n)]^2 - s_k^2(\mathbf{x}_n)\}, \tag{17}$$

$$\pi_k(\mathbf{x}_n) := \pi_k(\mathbf{x}_n) + \lambda [P(k|\mathbf{x}_n, y_n) - \pi_k(\mathbf{x}_n)], \tag{18}$$

where λ plays the role of a 'learning rate'. Introducing this learning rate also has the effect of updating the variances and priors in a slower time-scale than the means, which is essential when inferring an input-dependent variance as also pointed out in [5, 4]. The above sequential formulas correspond to the result that at the maximum of the likelihood function the priors converge to the average of the posteriors and the variances to the average, with respect to the posteriors, of the squared distances of targets to means [9]. Note that the necessary conditions (2) for the priors are satisfied since the posteriors $P(k|\mathbf{x}_n, y_n)$ are by construction in $(0, 1)$ and sum to one.

Finally we estimate b_{km} and c_{km} from the new $s_k^2(\mathbf{x}_n)$ and $\pi_k(\mathbf{x}_n)$, respectively, by solving a new set of K least squares problems to get

$$\mathbf{b}_k = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{s}_k, \tag{19}$$

$$\mathbf{c}_k = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\pi}_k, \tag{20}$$

where

$$\mathbf{s}_k = [\log s_k^2(\mathbf{x}_1), \ldots, \log s_k^2(\mathbf{x}_N)]^T, \tag{21}$$

$$\boldsymbol{\pi}_k = [\log(\frac{1}{\pi_k(\mathbf{x}_1)} - 1), \ldots, \log(\frac{1}{\pi_k(\mathbf{x}_N)} - 1)]^T, \tag{22}$$

and

$$\mathbf{H}(n, m) = \phi_m(\mathbf{x}_n) \tag{23}$$

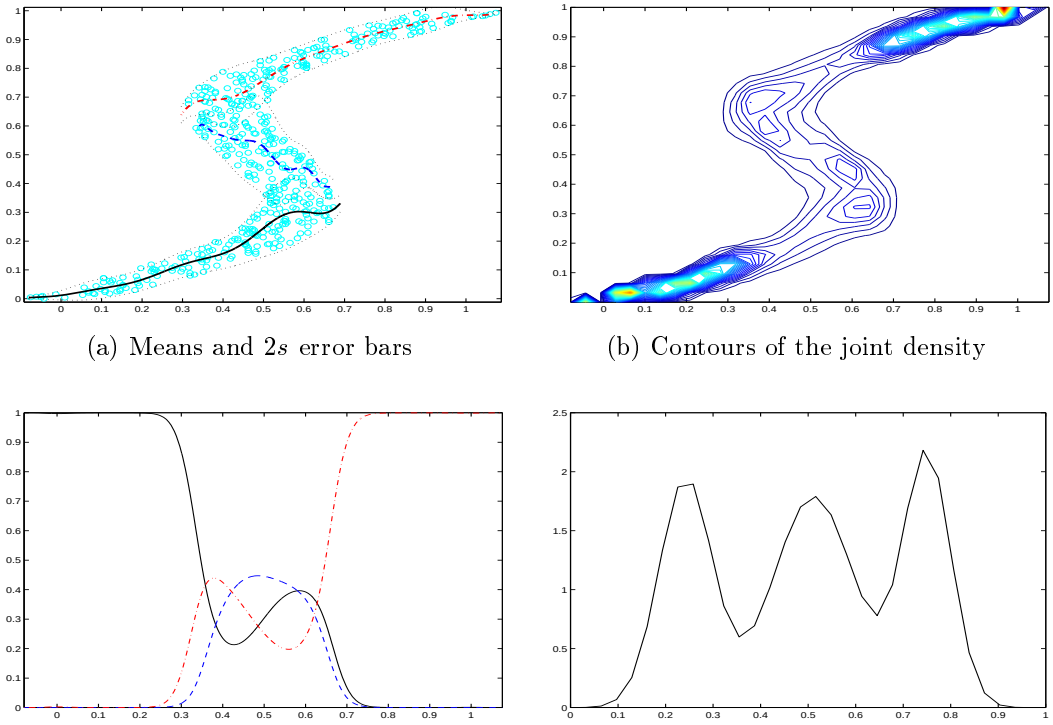
is the design matrix of the new least squares problems.

3 Demonstration and discussion

We tested the proposed method on the same example as in [2]. The training set consists of 1000 one-dimensional pairs (x_n, y_n) that satisfy the relation

$$x = y + 0.3 \sin(2\pi y) + \epsilon, \tag{24}$$

with ϵ a random number uniform in the interval $(-0.1, 0.1)$. We modeled the mixture conditional density (1) with $K = 3$ components, while for the basis functions in (4)–(6) we chose $M = 10$ Gaussians with fixed centers and spreads uniformly distributed in the x space. For the learning rate in (17), (18) we chose the value of $\lambda = 0.1$.



(a) Means and $2s$ error bars

(b) Contours of the joint density

(c) Priors functionals

(d) Plot of $p(y|x = 0.5)$

Figure 1: Results after applying our method to the multi-valued problem in [2].

We initialized the means assuming that all targets were equal to three different levels of their range, the variances to some small values, and the priors to equal values 1/3. We ran the EM algorithm for 20 steps and got the results that are shown in Fig. 1. We should note, however, that almost after 5 steps the approximation was very satisfactory.

In (a) we show the means estimates with $2s$ error bars of the three Gaussian components in those areas of the input space where the prior probabilities of the components are over 0.2. In (b) we show the contours of the joint density, while in (c) we show the functional form of the priors for the three Gaussian components. For input values near the center of their range we see that all three components have substantial contribution to the conditional density, while near the edges two of them are always inactive. Finally in (d) we show a 1-dimensional plot of the conditional likelihood $p(y|x)$ for input $x = 0.5$ which clearly shows the multimodality of the distribution.

It is interesting here to draw a comparison between our proposed EM algorithm for mixture conditional density estimation and a nonlinear optimization method like that in [2].

Concerning convergence, the most appealing property of EM is that it produces sequences of estimates which monotonically increase the log-likelihood [9]. This fact renders EM a better tool for seeking global maxima (although not always achievable) than its nonlinear optimization counterparts like Newton or quasi-Newton methods. In general EM needs relatively few steps in order to capture most of the relevant features of the underlying distribution, especially when the components are sufficiently separated and a good initialization point is chosen. Further speed-up can be achieved using an incremental version of the algorithm [6].

Concerning performance, in the case of univariate Gaussian mixtures EM requires $O(NK)$ arithmetic operations per iteration, compared, e.g., to $O(NK) + O(K^2)$ of a quasi-Newton method, where N is the size of the training set and K is the number of mixing components. In practice, the nice thing about EM is that it requires very little storage and its implementation is trivial.

4 Conclusions and future work

We presented a method for estimating a mixture conditional density function of targets given inputs in a multi-valued mapping problem, and where the parameters of the mixture, i.e., means, variances, and mixing priors, are all functions of the input data. We assumed a fixed set of basis functions and derived an EM algorithm for fitting the parameters. Our method compares favorably to nonlinear optimization methods that exist for this problem since it exploits all the attractive features of EM like monotonic convergence, low computational and storage requirements, simplicity of the implementation, etc.

Regularization and model selection techniques based on cross-validation [7] can easily be incorporated in our approach by appropriately extending the least squares problems. We are currently working on combining our method with a recent result [1] which allows radial basis functions networks to iteratively learn the centers of the basis functions, a result which seems to fit well within our iterative framework. Also we are seeking automatic ways for estimating the learning rate λ in (18) which controls the variances and priors updates, and also the number K of mixing components [11]. Finally, an interesting extension to the problem is to allow missing values in the training set.

References

[1] D. Barber and B. Schottky. Radial basis functions: a Bayesian treatment. In *Proc. NIPS'10*, 1998.

[2] C. M. Bishop. Mixture density networks. Technical report, Aston University, Birmingham, UK, NCRG/94/001, Neural Computing Research Group, 1994.

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

[4] C. M. Bishop and C. S. Qazaz. Regression with input-dependent noise: A Bayesian treatment. In *Proc. NIPS 9*, 1997.

[5] D. J. C. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1991.

[6] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in graphical models*, pages 355–368. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[7] M. J. L. Orr. Introduction to radial basis function networks. Technical report, Center for Cognitive Science, University of Edinburgh, 1996.

[8] W. H. Press, S. A. Teukolsky, B. P. Flannery, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[9] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, Apr. 1984.

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

[11] N. Vlassis and A. Likas. A kurtosis-based dynamic approach to Gaussian mixture modeling. *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, 29(4), July 1999.

[12] N. Vlassis, G. Papakonstantinou, and P. Tsanakas. Mixture density estimation based on maximum likelihood and test statistics. *Neural Processing Letters*, 9(1):63–76, Feb. 1999.