The Q function of the EM algorithm for hidden variable structure learning

Michalis K. Titsias
Institute for Adaptive and Neural Computation
Division of Informatics, University of Edinburgh
Edinburgh, 5 Forrest Hill, EH1 2QL, UK
M.Titsias@sms.ed.ac.uk

Abstract

It is well known that maximum likelihood fails to account for model complexity. We show that in graphical models with hidden variables, the expected complete data log likelihood, or Q function, used in the operation of the EM algorithm can penalise too complex hidden structures and thus it may be used either as a model selection criterion or as an objective function that we wish to maximise. We apply the Q function as a model selection criterion to the mixture models and factor analysis model fitted using EM. Also we maximise the Q function itself in the case of Gaussian mixture models and derive an EM-type algorithm that automatically adjusts model parameters and number of mixture components. We demonstrate these ideas to some data sets.

1 Introduction

We address the problem of learning graphical models with hidden variables from data, where the hidden structure is unknown. The typical way of training a model with a fixed hidden structure is by maximising the likelihood using the EM algorithm (Dempster et al., 1977). However, a fundamental problem of maximum likelihood method is that it cannot account for model complexity since more flexible models can always increase the likelihood of the data, but the model is prone to ovefitting. The Bayesian method (Mackay, 1992; Heckerman et al., 1995) can in principle overcome these difficulties by integrating over parameters (models) and computing the marginal likelihood. This computation is generally intractable and is approximated according to several schemes. Chickering and Heckerman (1997) provide a comparison over several approximations of the marginal likelihood including the Laplace and BIC/MDL approximation. More recently Attias (1999) and Ghahramani and Beal (2000) have applied variational approximations.

In this paper we consider a different than the Bayesian approach to the hidden structure learning that is rather simple and easy to apply. We have made the observation that the expected complete data log likelihood used in the operation of the EM algorithm

can account for model complexity concerning the hidden variables. This is because the Q function decomposes into two terms: the log likelihood term and ii) the conditional entropy of the hidden variables given the observations. On one hand the log likelihood expresses how well the model fits the data, so it can account for a misfit term. On the other hand the entropic term measures how well the hidden variables are determined by the observations and thus it can account for a model complexity term by penalising over-flexible hidden structures. We consider the Q function as a criterion for selecting among models trained by maximum likelihood and we apply this framework to mixture models and factor analysis. Furthermore, we consider the Q function as an objective function that we wish to maximise and we derive an EM-type algorithm for Gaussian mixture models that automatically adjusts model parameters and number of mixture components.

The remainder of the paper is structured as follows: In section 2 we discuss the Q function as a criterion for selecting the hidden structure and we apply this to mixture models and factor analysis. In section 3 we derive an EM-type algorithm for Gaussian mixtures that directly maximises the Q function. In section 4 we provide experiments with Gaussian mixtures and factor analysers to some data sets and we conclude with a discussion in section 5.

2 The Q function

Let $X = \{x^i\}$ denote all the data and $Z = \{z^i\}$ the set of hidden variables. Also assume that the joint distribution $p(X, Z|M)$ is computed by making use of a set of conditional independences represented by a graphical model (either with directed or undirected links). M stands for the model and refers to any structural choice we made concerning the hidden variables, such as the connection links, number of hidden variables and how many values a discrete hidden variable can take.

Fitting the model to the data by maximising the likelihood can little say about how suitable was the original choice of the hidden structure M. The maximum likelihood approach cannot account for model complexity since more flexible hidden structures can always provide higher likelihood value.

A way to penalise flexible hidden structures is to measure how the data change the distribution of the hidden variables; if many alternative hidden values appear to be plausible explanations of the observations, then the model probably has a too complex hidden structure and a simpler hidden structure can be found. Such a measure is the entropy of the hidden variables Z conditioned on the data X:

$$H_M = -\sum_Z p(Z|X, M) \ \log p(Z|X, M), \tag{1}$$

where the sum can when appropriate be replaced by an integral. Next we refer to the above quantity as the posterior entropy of the hidden variables or simply posterior entropy. When the hidden values are well determined by the observations, the conditional distribution $p(Z|X, M)$ should be fairly deterministic, which would result in low posterior entropy. On the other hand an ambiguous distribution $p(Z|X, M)$ would result in high posterior entropy. Generally, we expect models with simple hidden structures to

have lower posterior entropy than more flexible models. H_M by itself cannot indicate which model is appropriate for the data, since models with too simple hidden structure, that do not fit the data, might give low entropy but are inappropriate. A reasonable approach for model selection should penalise also underfitting of the data by considering a misfit measure. Such a measure can be the log likelihood

$$L_M = \log \sum_Z p(X, Z|M), \tag{2}$$

which we would like to be maximised. Finally since we wish the log likelihood to be maximised and the posterior entropy to be minimised a criterion for selecting the hidden structure can be the following

$$Q_M = L_M - H_M \tag{3}$$

or

$$Q_M = \sum_Z p(Z|X, M) \log p(X, Z|M). \tag{4}$$

Clearly 3 is the expected complete data log likelihood, or Q function, used in the operation of the EM algorithm. The above justifies the use of the Q function as a criterion for hidden structure selection. A framework for applying the above idea would be to first find a parameter value $\hat{\theta}$ that locally maximises the log likelihood for any candidate hidden structure M and then assign a preference to that model by the value

$$C_M = \hat{L}_M - \hat{H}_M, \tag{5}$$

where $\hat{L}_M$ and $\hat{H}_M$ are estimated for the parameter values $\hat{\theta}$. Preferred models are those that have the highest values C_M. In section 2.1 and 2.2 we apply this criterion to mixture models and factor analysis.

2.1 Mixture models

A mixture model is a weighted average of densities:

$$p(x) = \sum_{z=1}^{J} \pi_z p(x|z) \tag{6}$$

where z is a discrete hidden variable and π_z the corresponding prior probability. The number J of mixture components is the structural choice that we have to make about the model. For fixed J and given a set of i.i.d. data $X = \{x^1, \ldots, x^N\}$ we can maximise the log likelihood using the EM algorithm in order to deal with the hidden variables $Z = \{z^1, \ldots, z^N\}$. However as expected the log likelihood increases with the number of components J. According to our model selection approach, if $\hat{\theta}$ is a set of parameters obtained by maximising the log likelihood for a mixture model with J components, then we can assign a preference to that model by

$$C_J = \sum_{n=1}^{N} \log \hat{p}(x^n) + \sum_{n=1}^{N} \sum_{z=1}^{J} \hat{P}(z|x^n) \log \hat{P}(z|x^n), \tag{7}$$

where $P(z|x)$ is the responsibility

$$P(z|x) = \frac{\pi_z p(x|z)}{p(x)}. \tag{8}$$

The first term in 7 is the log likelihood and the second is the negative posterior entropy. The above equation is very intuitive; the log likelihood term expresses how well the model explains the data while the entropic term expresses how well the model separates the data to the J assumed clusters. As J becomes larger than the number of true clusters in our data, the components will overlap causing an increase of the posterior entropy that will penalise the log likelihood.

However the candidate values of J must be much less than the number of training points N, since if $J = N$ and each component fit a data point, the entropy will become zero and the log likelihood infinitely large. Roughly speaking this is because the Q function does not explicitly penalise the parameter space.

2.2 Factor analysis

In the factor analysis model (Everitt, 1984), a d-dimensional vector of real values x is considered to be generated by taking the linear transformation of a k-dimensional vector z, where k is smaller that d, and adding some Gaussian independent noise. The z values are the hidden variables or factors. The data point x is generated by

$$x = Az + n \tag{9}$$

where A is the factor loading matrix, the factors z are assumed to be $N(z; 0, I_k)$ distributed and the random variable n is distributed according to $N(n; 0, \Psi)$ with Ψ being diagonal. By marginalising out z the resulting distribution of x is Gaussian with zero mean and covariance $AA^T + \Psi$. The learning objective in factor analysis is given a set of i.i.d. data $X = \{x^1, \ldots, x^N\}$ to find the parameters A and Ψ that best explain these data. The EM algorithm can be used to deal with the hidden variables $Z = \{z^1 \ldots, z^N\}$ and provide a local maximum of the likelihood (Rubin and Thayer, 1982). The posterior distribution of the hidden factor z given the observation x is

$$p(z|x) = N(A^T(\Psi + AA^T)^{-1}x, (A^T\Psi^{-1}A + I)^{-1}) \tag{10}$$

where the covariance matrix is independent on the observation x.

The structural parameter that we have to specify when we apply the above model is the dimensionality k of the factors. Ghahramani and Beal (2000) have applied the Bayesian method with variational approximation to infer the dimensionality of the factors. Next we apply the Q function as the criterion for selecting the dimensionality of z. The log likelihood in the factor analysis model is $L_k = \sum_{n=1}^{N} \log N(x^n; 0, AA^T + \Psi)$, while the entropy of the factors conditioned on the observations is

$$H_k = -\sum_{n=1}^{N} \int p(z|x^n) \log p(z|x^n). \tag{11}$$

Since $p(z|x)$ is Gaussian, its differential entropy has a nice closed form and equals $\frac{1}{2}\log\left((2\pi)^k|(A^T\Psi^{-1}A+I)^{-1}|\right)$, thus we finally obtain

$$H_k = \frac{N}{2}\log\left((2\pi)^k|(A^T\Psi^{-1}A+I)^{-1}|\right). \tag{12}$$

We can now plug in the above entropy and the log likelihood in the equation 5 and obtain the criterion for selecting the dimensionality k. The parameter values for a certain k can be estimated by maximising the log likelihood using the EM algorithm. The intuition behind the above criterion lies in the behaviour of the entropic term as that varies with respect to the dimensionality k. The entropy expresses how broad the posterior Gaussian $p(z|x)$ is. As we overfit the number of factors we expect the factors to overlap each other in the light of data, which causes the posterior $p(z|x)$ to become uncertain with large entropy.

3 The Q function as an objective function

If the Q function can account for model complexity of the hidden structure, then we can make a step further and attempt to maximise the Q function itself instead of maximising the log likelihood. Potentially this can allow us to optimise simultaneously parameters and hidden structure. Below we apply this to Gaussian mixtures.

The EM algorithm in the M-step of each iteration maximises the Q function by considering fixed the distribution of the hidden variables given the observations, which do not maximise the Q function in general. We can see the general problem of this maximisation as penalised maximum likelihood

$$Q_M(\theta) = L_M(\theta) - H_M(\theta), \tag{13}$$

where we explicitly view all the quantities as a function of the parameters θ. If the entropic term is a highly non linear function over θ, then the above maximisation generally would be much more difficult that maximising the log likelihood.

Assume a mixture model of the form 6 where the component densities are Gaussians: $p(x|z) = N(x;\mu_z,\Sigma_z)$. According to the above framework we wish to train the model by maximising the function

$$Q(\theta) = \sum_{n=1}^{N}\log\sum_{z=1}^{J}\pi_z N(x^n;\mu_z,\Sigma_z) + \sum_{n=1}^{N}\sum_{z=1}^{J}P(z|x^n)\log P(z|x^n). \tag{14}$$

By differentiating with respect to the parameters $\theta = \{\mu_j,\Sigma_j,\pi_j\}_{j=1}^{J}$ and setting to zero we can arrive at the following fixed point equations

$$\mu_z^{(t+1)} = \frac{\sum_{n=1}^{N}[P^{(t)}(z|x^n) + (s_z^n)^{(t)}]x^n}{\sum_{n=1}^{N}[P^{(t)}(z|x^n) + (s_z^n)^{(t)}]}, \tag{15}$$

$$\Sigma_z^{(t+1)} = \frac{\sum_{n=1}^{N}[P^{(t)}(z|x^n) + (s_z^n)^{(t)}](x^n-\mu_z^{(t+1)})(x^n-\mu_z^{(t+1)})^T}{\sum_{n=1}^{N}[P^{(t)}(z|x^n) + (s_z^n)^{(t)}]}, \tag{16}$$

$$\pi_z^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N}[P^{(t)}(z|x^n) + (s_z^n)^{(t)}], \tag{17}$$

where

$$s_z^n = P(z|x^n)\left[\log P(z|x^n) - \sum_{i=1}^{J} P(i|x^n)\log P(i|x^n)\right]. \tag{18}$$

The above equations differ from the respective EM updates for maximum likelihood only that now we have the term s_j^n instead of the responsibility $P(z|x^n)$ itself. The s_j^n values can be positive or negative and satisfy $\sum_{z=1}^{J} s_z^n = 0$ for each training point[1]. The value s_z^n becomes large when the corresponding component explains the data point x^n with large responsibility. For instance, if we have four components and for some x^n the responsibilities are: 0.6, 0.3, 0.1 and 0 respectively, the corresponding s_z^n would be: 0.23, -0.09, -0.14 and 0. Also note that the regularisation mechanism is only active when the model fails to separate well the data to J clusters; if for a data point x^n one component has responsibility value equal to 1, then all the s_z^n values are zero.

We apply iteratively equations 15-17 by starting from a large number of components and as the algorithm evolves some components die and are discarded from the mixture model. A component z dies if its prior π_z becomes zero or negative. So in each iteration of the algorithm we identify the alive components z having $\pi_z > 0$, we discard the rest of them from the mixture model and also renormalise the priors if at least one component is discarded. Additionally, in each iteration as in the maximum likelihood training we must ensure that the covariance matrices are positive definite[2].

Someone can ask what is the difference between applying the Q function as a model selection criterion for models trained by maximum likelihood and maximising directly this function. Clearly, even if the selected hidden structure is the same for both frameworks the parameter estimates might differ. This is because the latter framework estimates the parameters by simultaneously maximising the log likelihood and minimising the posterior entropy.

4 Experiments

We tested the methods on two clustering problems and one factor analysis problem. In each case Q is applied as a model selection criterion, we maximise the likelihood 5 times for each candidate model under different parameter initialisations and select the parameters that give the larger likelihood value to stand for that model. In addition, in the case of Gaussian mixtures where we also maximise the Q function directly, we perform 10 maximisations of the Q function by starting always from 50 components and select the parameters that give the larger Q value.

The first problem is a synthetic data set of 500 two-dimensional data points forming 5 clusters (Figure 1a, left). The candidate models are from 1 to 15 components. Figure

[1]Because of that property the denominator in π_z update remains the number of training points N, as in the EM for maximising the log likelihood.

[2]Note that since the value $P(z|x) + s_j^n$ can be also negative a covariance matrix can also take negative eigen values. However this occurs for the dying components that receive negative s_j^n values from many different data points.

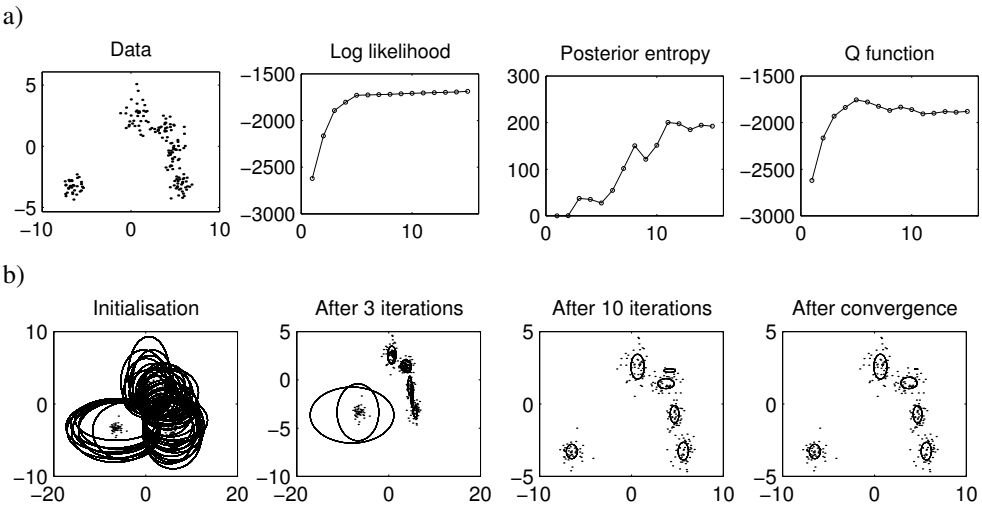

Figure 1: a) From left to right: the synthetic 2-dimensional data, the log likelihood, the posterior entropy and the Q function with respect to the number of components. b) The state of the algorithm for maximising the Q function after: initialisation, 3 iterations, 10 iterations and convergence.

1a displays the log likelihood, the posterior entropy and the Q function with respect to the number of components. Clearly the Q function achieves the maximum at 5 components. Figure 1b displays the solutions found by maximising the Q function using the algorithm of section 3. Clearly the 5 clusters are well represented, however there are additionally two other 'singular' components that fit two and three data points respectively. This is because, as mentioned in section 2.1., the Q function does not penalise the situation when a component fits one or few spurious data points since the contribution to the total entropy will probably be zero, while the log likelihood increases. However the singular components can be easily cut down after training (and then refine the other parameters) since they will have almost zero π_z.

The second data set is the spiral data from Ueda et al. (1999) consisting of 800 3 dimensional data points (Figure 2a, left). Here the data are uniform and there is not 'right' number of components. The Q function applied according to the framework of section 2 is not peaked at any number of components and appears to increase with a smaller rate than the log likelihood. On the other hand maximising directly the Q function by starting from 50 components we found a solution with 14 components as displayed in Figure 2b. We had no problem with singular components in that case and the algorithm running time was about 20 seconds in a 360MHz processor.

Finally we generated a synthetic data set of 300 points of intrinsic dimensionality 5 embedded in 30 dimensions. To generate this data set we randomly selected each element of the factor loading matrix A from the standard Gaussian, while the noise variances in Ψ were randomly selected from the interval $(0\ 2]$. We consider all the possible dimensions of the factors (from 1 to 30). Figure 3a displays the log likelihood,

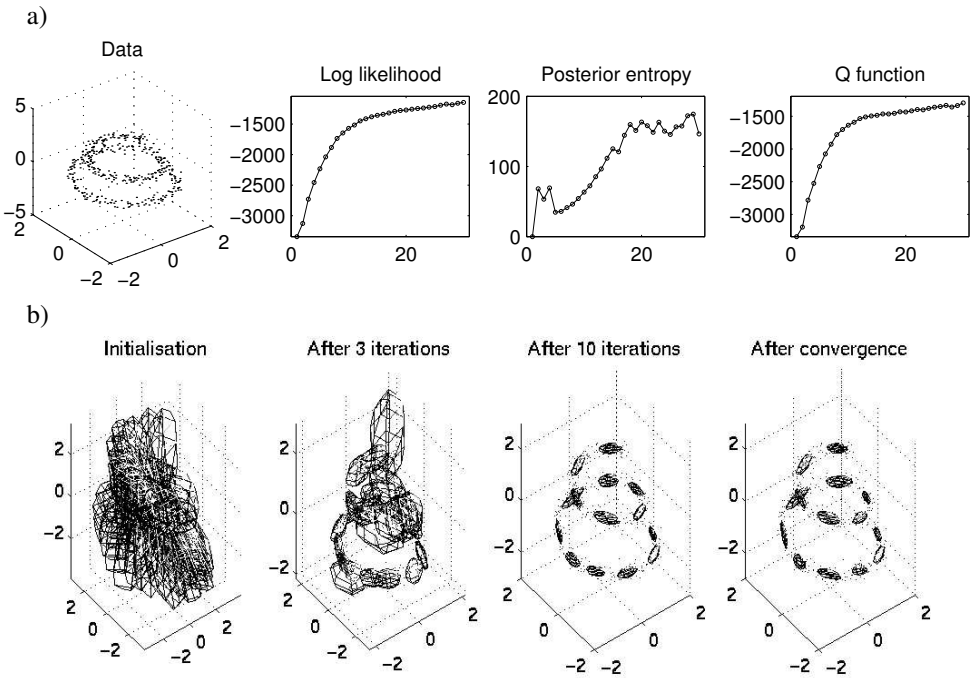

Figure 2: a) From left to right: the spiral data, the log likelihood, the posterior entropy and the Q function with respect to the number of components. b) The state of the algorithm for maximising the Q function after: initialisation, 3 iterations, 10 iterations and convergence. Note that the algorithm has almost converged in 10 iterations.

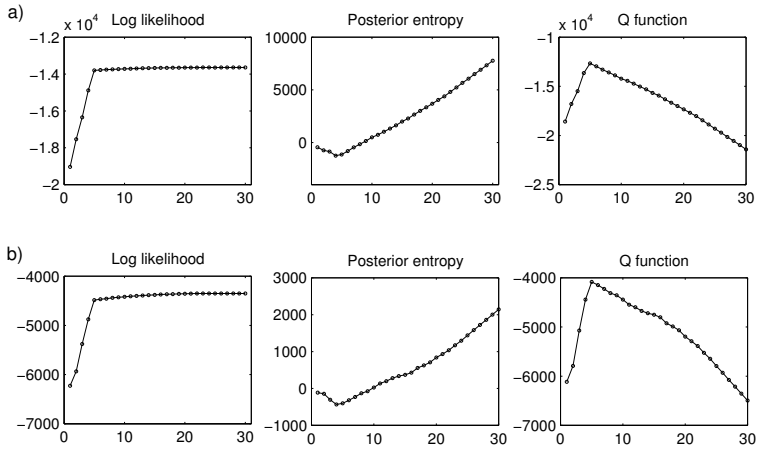

Figure 3: Factor analysis data with 30 data dimensions and 5 intrinsic dimension. a) Training using 300 data points. From left to right: the log likelihood, the posterior entropy and the Q function with respect to the number of factors. b) the respective plots considering 100 training points.

posterior entropy and the Q function with respect to the number of factors. The Q function is sharply peaked in 5 dimensions. Figure 3b shows the respective results by considering only 100 training points. Note that the posterior entropy increases almost linearly once the factors overfit the data.

5 Discussion

In this paper we showed that the Q function of the EM algorithm can be used for learning the hidden structure in models with hidden variables. Our key observation was that the entropy of the hidden variables given the observations can penalise over-flexible hidden structures. Thus, combining this term with a data misfit term (the log likelihood) we derived the Q function. We demonstrated this method in the mixture models and factor analysis, however we consider it of broader applicability.

The posterior entropy can measure the flexibility of the model by considering how uncertain the unknown hidden variables are given the known data. A generalisation would be to consider all the unknown configurations of the model, parameters and hidden variables, and measure the complexity as the entropy of all unknown configurations of the model given the data. This also will directly penalise the parameter space. Our main future research focus is to generalise the current framework in order to consider parameter together with hidden variables.

Acknowledgements: MT thanks Chris Williams, David Barber and Felix Agakov for helpful discussions on the ideas of this paper. Also we thank Chris Williams for comments on the manuscript and Zoubin Ghahramani for making available the code of the EM algorithm for factor analysis.

References

Attias, H. (1999). Inferring parameters structure of latent variable models by variational Bayes. *In Proc. 15th Conf. on Uncertainty in Artificial Intelligence*.

Chickering, D. M. and Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29:181–212.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38.

Everitt, B. S. (1984). *An introduction to latent variable models*. Chapman Hall, London.

Ghahramani, Z. and Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems*, volume 12, Cambridge, MA. MIT Press.

Heckerman, D., Gieger, D., and Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

Mackay, D. J. C. (1992). Bayesian interpolation. *Neural Compuation*, 4:405–447.

Rubin, D. and Thayer, D. (1982). EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76.

Ueda, N., Nakano, Z., Ghahramani, Z., and Hinton, G. E. (1999). SMEM algorithm for mixture models. In *Advances in Neural Information Processing Systems*, volume 11, Cambridge, MA. MIT Press.