Chapter 12

EM algorithms

The Expectation-Maximization (EM) algorithm is a maximum likelihood method for models that have hidden variables eg. Gaussian Mixture Models (GMMs), Linear Dynamic Systems (LDSs) and Hidden Markov Models (HMMs).

12.1 Gaussian Mixture Models

Say we have a variable which is multi-modal ie. it separates into distinct clusters. For such data the mean and variance are not very representative quantities.

In a 1-dimensional Gaussian Mixture Model (GMM) with m-components the likelihood of a data point x_n is given by

$$p(x_n) = \sum_{k=1}^{m} p(x_n|k)p(s^n = k) \tag{12.1}$$

where s_n is an indicator variable indicating which component is selected for which data point. These are chosen probabilistically according to

$$p(s^n = k) = \pi_k \tag{12.2}$$

and each component is a Gaussian

$$p(x_n|k) = \frac{1}{(2\pi\sigma_k^2)^{1/2}} \exp\left(-\frac{(x_n - \mu_k)^2}{2\sigma_k^2}\right) \tag{12.3}$$

To generate data from a GMM we pick a Gaussian at random (according to 12.2) and then sample from that Gaussian. To fit a GMM to a data set we need to estimate π_k, μ_k and σ_k^2. This can be achieved in two steps. In the 'E-Step' we soft-partition the data among the different clusters. This amounts to calculating the probability that data point n belongs to cluster k which, from Baye's rule, is

$$\gamma_k^n \equiv p(s_n = k|x_n) = \frac{p(x_n|k)p(s_n = k)}{\sum_{k'} p(x_n|k')p(s_n = k')} \tag{12.4}$$

In the 'M-Step' we re-estimate the parameters using Maximum Likelihood, but the data points are weighted according to the soft-partitioning

$$\begin{aligned}
\pi_k &= \sum \gamma_k^n \\
\mu_k &= \frac{\sum \gamma_k^n x_n}{\sum \gamma_k^n} \\
\sigma_k^2 &= \frac{\sum \gamma_k^n (x_n - \mu_k)^2}{\sum \gamma_k^n}
\end{aligned} \tag{12.5}$$

These two steps constitute an EM algorithm. Summarizing:

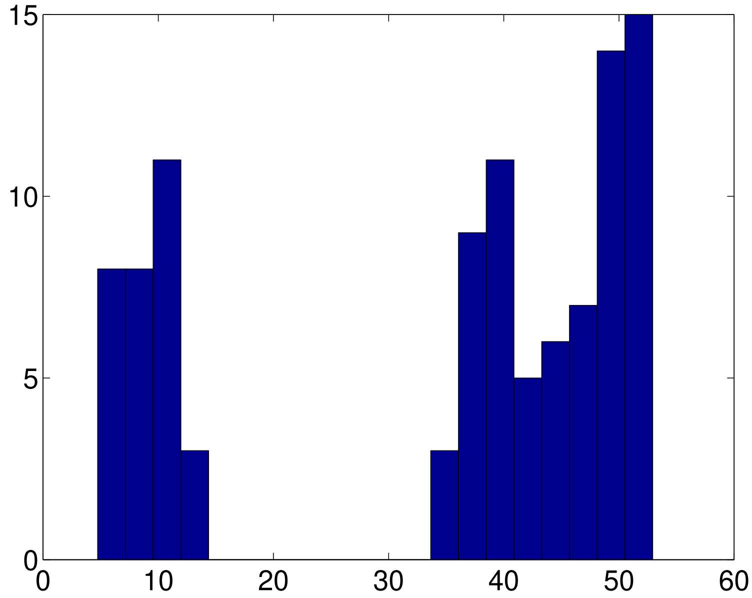

Figure 12.1: A variable with 3 modes. This can be accurately modelled with a 3-component Gaussian Mixture Model.

- E-Step: Soft-partitioning.
- M-Step: Parameter updating.

Application of a 3-component GMM to our example data gives for cluster (i) $\pi_1 = 0.3$, $\mu_1 = 10$, $\sigma_1^2 = 10$, (ii) $\pi_1 = 0.35$, $\mu_1 = 40$, $\sigma_1^2 = 10$, and (iii) $\pi_3 = 0.35$, $\mu_1 = 50$, $\sigma_1^2 = 5$.

GMMs are readily extended to multivariate data by replacing each univariate Gaussian in the mixture with a multivariate Gaussian. See eg. chapter 3 in [3].

12.2 General Approach

If V are visible variables, H are hidden variables and θ are parameters then

1. E-Step: Get $p(H|V,\theta)$
2. M-Step, change θ so as to maximise

$$Q = < \log p(V, H|\theta) > \tag{12.6}$$

where expectation is wrt $p(H|V,\theta)$.

Why does it work ? Maximising Q maximises the likelihood $p(V|\theta)$. This can be proved as follows. Firstly

$$p(V \mid \theta) = \frac{p(H, V \mid \theta)}{p(H \mid V, \theta)} \tag{12.7}$$

This means that the log-likelihood, $L(\theta) \equiv \log p(V \mid \theta)$, can be written

$$L(\theta) = \log p(H, V \mid \theta) - \log p(H \mid V, \theta) \tag{12.8}$$

If we now take expectations with respect to a distribution $p'(H)$ then we get

$$L(\theta) = \int p'(H) \log p(H, V \mid \theta) dH - \int p'(H) \log p(H \mid V, \theta) dH \tag{12.9}$$

The second term is minimised by setting $p'(H) = p(H|V,\theta)$ (we can prove this from Jensen's inequality or the positivity of the KL divergence; see [12] or lecture 4). This takes place in the E-Step. After the E-step the auxiliary function Q is then equal to the log-likelihood. Therefore, when we maximise Q in the M-step we are maximising the likelihood.

12.3 Probabilistic Principal Component Analysis

In an earlier lecture, Principal Component Analysis (PCA) was viewed as a linear transform

$$\boldsymbol{y} = \boldsymbol{Q}^T \boldsymbol{x} \tag{12.10}$$

where the jth column of the matrix $\boldsymbol{Q}$ is the jth eigenvector, $\boldsymbol{q}_j$, of the covariance matrix of the original d-dimensional data $\boldsymbol{x}$. The jth projection

$$y_j = \boldsymbol{q}_j^T \boldsymbol{x} \tag{12.11}$$

has a variance given by the jth eigenvalue λ_j. If the projections are ranked according to variance (ie. eigenvalue) then the M variables that reconstruct the original data with minimum error (and are also linear functions of $\boldsymbol{x}$) are given by $y_1, y_2, ..., y_M$. The remaining variables $y_{M+1}, .., y_d$ can be be discarded with minimal loss of information (in the sense of least squares error). The reconstructed data is given by

$$\hat{\boldsymbol{x}} = \boldsymbol{Q}_{1:M} \boldsymbol{y}_{1:M} \tag{12.12}$$

where $\boldsymbol{Q}_{1:M}$ is a matrix formed from the first M columns of $\boldsymbol{Q}$. Similarly, $\boldsymbol{y}_{1:M} = [y_1, y_2, ..., y_M]^T$.

In probabilistic PCA (pPCA) [60] the PCA transform is converted into a statistical model by explaining the 'discarded' variance as observation noise

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{y} + \boldsymbol{e} \tag{12.13}$$

where the noise is drawn from a zero mean Gaussian distribution with isotropic covariance $\sigma^2 \boldsymbol{I}$. The 'observations' $\boldsymbol{x}$ are generated by transforming the 'sources' $\boldsymbol{y}$ with the 'mixing matrix' $\boldsymbol{W}$ and then adding 'observation noise'. The pPCA model has M sources where $M < d$. For a given M, we have $\boldsymbol{W} = \boldsymbol{Q}_{1:M}$ and

$$\sigma^2 = \frac{1}{M-d} \sum_{j=M+1}^{d} \lambda_j \tag{12.14}$$

which is the average variance of the discarded projections.

There also exists an EM algorithm for finding the mixing matrix which is more efficient than SVD for high dimensional data. This is because it only needs to invert an M-by-M matrix rather than a d-by-d matrix.

If we define $\boldsymbol{S}$ as the sample covariance matrix and

$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I} \tag{12.15}$$

then the log-likelihood of the data under a pPCA model is given by [60]

$$\log p(\boldsymbol{X}) = -\frac{Nd}{2} \log 2\pi - \frac{N}{2} \log |\boldsymbol{C}| - \frac{N}{2} Tr(\boldsymbol{C}^{-1} \boldsymbol{S}) \tag{12.16}$$

where N is the number of data points.

We are now in the position to apply the MDL model order selection criterion. We have

$$MDL(M) = -\log p(\boldsymbol{X}) + \frac{Md}{2} \log N \tag{12.17}$$

This gives us a procedure for choosing the optimal number of sources.

Because pPCA is a probabilistic model (whereas PCA is a transform) it is readily incorporated in larger models. A useful model, for example, is the Mixtures of pPCA model. This is identical to the Gaussian Mixture model except that each Gaussian is decomposed using pPCA (rather than keeping it as a full covariance Gaussian). This can greatly reduce the number of parameters in the model [61].

12.4 Linear Dynamical Systems

A Linear Dynamical System is given by the following 'state-space' equations

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_t &= Cx_t + v_t \end{aligned} \tag{12.18}$$

where the state noise and observation noise are zero mean Gaussian variables with covariances Q and R. Given A,C,Q and R the state can be updated using the Kalman filter.

For real-time applications we can infer the states using a Kalman filter. For retrospective/offline data analysis the state at time t can be determined using data before t *and* after t. This is known as Kalman smoothing. See eg. [21].

Moreover, we can also infer other parameters; eg. state noise covariance Q , state transformation matrix C, etc. See eg. [22]. To do this, the state is regarded as a 'hidden variable' (we do not observe it) and we apply the EM algorithm [15].

For an LDS

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_t &= Cx_t + v_t \end{aligned} \tag{12.19}$$

the hidden variables are the states x_t and the observed variable is the time series y_t. If $x_1^t = [x_1, x_2, .., x_t]$ are the states and $y_1^T = [y_1, y_2, ..., y_t]$ are the observations then the EM algorithm is as follows.

M-Step

In the M-Step we maximise

$$Q = < \log p(y_1^T, x_1^T|\theta) > \tag{12.20}$$

Because of the Markov Property of an LDS (the current state only depends on the last one, and not on ones before that) we have

$$p(y_1^T, x_1^T|\theta) = p(x_1) \prod_{t=2}^{T} p(x_t|x_{t-1}) \prod_{t=1}^{T} p(y_t|x_t) \tag{12.21}$$

and when we we take logs we get

$$\log p(y_1^T, x_1^T|\theta) = \log p(x_1) + \sum_{t=2}^{T} \log p(x_t|x_{t-1}) + \sum_{t=1}^{T} \log p(y_t|x_t) \tag{12.22}$$

where each PDF is a multivariate Gaussian. We now need to take expectations wrt. the distribution over hidden variables

$$\gamma_t \equiv p(x_t|y_1^T) \tag{12.23}$$

This gives

$$< \log p(y_1^T, x_1^T|\theta) >= \gamma_1 \log p(x_1) + \sum_{t=2}^{T} \gamma_t \log p(x_t|x_{t-1}) + \sum_{t=1}^{T} \gamma_t \log p(y_t|x_t) \tag{12.24}$$

By taking derivates wrt each of the parameters and setting them to zero we get update equations for A, Q,C and R. See [22] for details. The distribution over hidden variables is calculated in the E-Step.

E-Step

The E-Step consists of two parts. In the forward-pass the joint probability

$$\alpha_t \equiv p(x_t, y_1^t) = \int \alpha_{t-1} p(x_t|x_{t-1}) p(y_t|x_t) dx_{t-1} \tag{12.25}$$

is recursively evaluated using a Kalman filter. In the backward pass we estimate the conditional probability

$$\beta_t \equiv p(y_t^T|x_t) = \int \beta_{t+1} p(x_{t+1}|x_t) p(y_{t+1}|x_{t+1}) dx_{t+1} \tag{12.26}$$

The two are then combined to produce a smoothed estimate

$$p(x_t|y_1^T) \propto \alpha_t \beta_t \tag{12.27}$$

This E-Step constitutes a Kalman smoother.

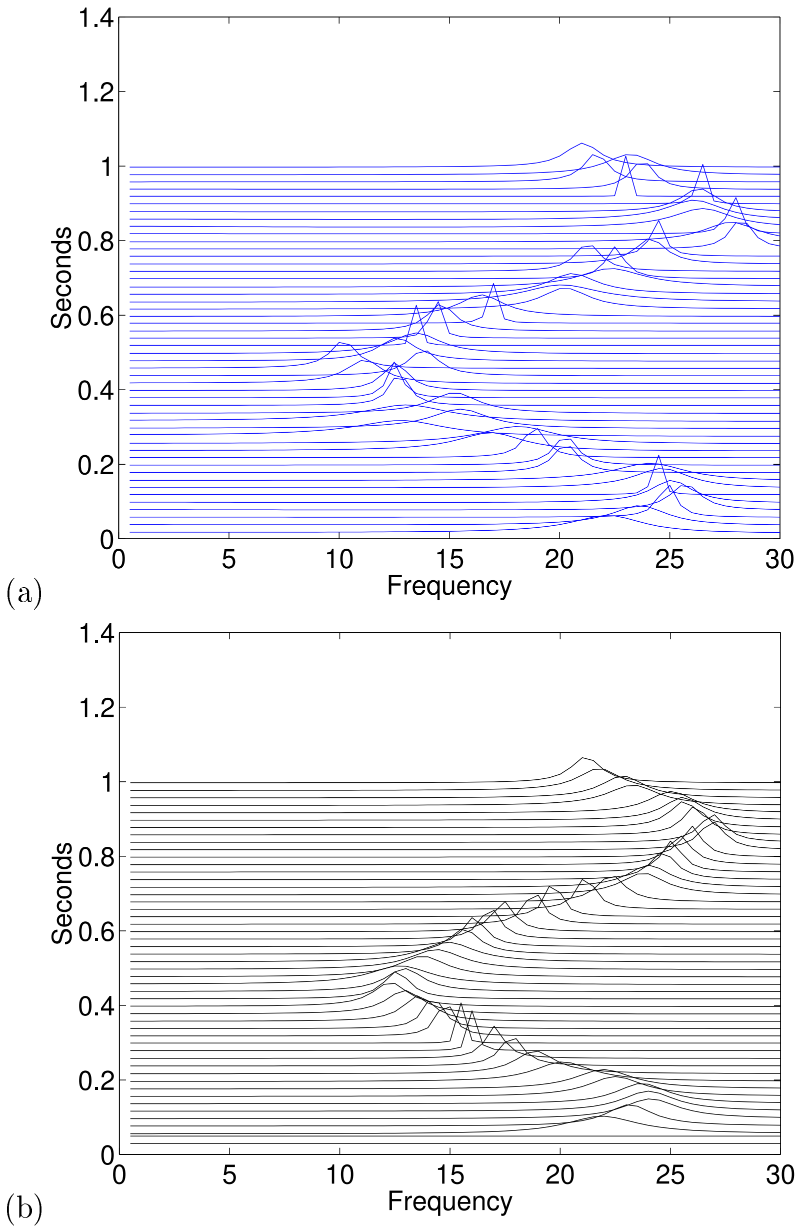

Figure 12.2: (a) Kalman filtering and (b) Kalman smoothing.