

Mixture Models

(Chapter 20)¹

David Barber

(adapted by Miguel Palomino)

¹These slides accompany the book *Bayesian Reasoning and Machine Learning*. The book and demos can be downloaded from www.cs.ucl.ac.uk/staff/D.Barber/brml. Feedback and corrections are also available on the site. Feel free to adapt these slides for your own purposes, but please include a link the above website.

Mixture Models

A mixture model is one in which a set of component models is combined to produce a richer model:

$$p(v) = \sum_{h=1}^H p(v|h)p(h)$$

The variable v is 'visible' or 'observable' and $h = 1, \dots, H$ indexes each component model $p(v|h)$, along with its weight $p(h)$.

Clustering

Mixture models have natural application in clustering data, where h indexes the cluster. This interpretation can be gained from considering how to generate a sample datapoint v from the model. First we sample a cluster h from $p(h)$, and then draw a visible state v from $p(v|h)$.

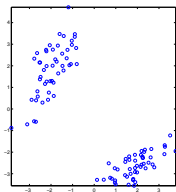


Figure: Two dimensional data which displays clusters. In this case a Gaussian mixture model $1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_1, \mathbf{C}_1) + 1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_2, \mathbf{C}_2)$ would fit the data well for suitable means $\mathbf{m}_1, \mathbf{m}_2$ and covariances $\mathbf{C}_1, \mathbf{C}_2$.

Graphical Model

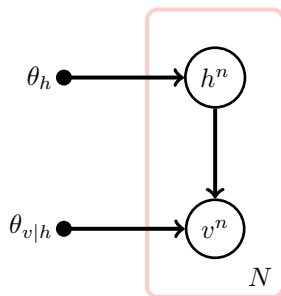


Figure: A mixture model has a trivial graphical representation as a DAG with a single hidden node, which can be in one of H states, $i = 1 \dots, H$. The parameters are assumed common across all datapoints.

Mixture of Independent Bernoulli

A simple model that can be used to cluster a set of binary vectors, $\mathbf{v}^n = (v_1^n, \dots, v_D^n)^\top$, $v_i \in \{0, 1\}$, $n = 1, \dots, N$ is

$$p(\mathbf{v}) = \sum_{h=1}^H p(h) \prod_{i=1}^D p(v_i|h)$$

where each term $p(v_i|h)$ is a Bernoulli distribution.

Parameters

We need to learn $p(v_i = 1|h = h)$ and $p(h = h)$.

To train the model under maximum likelihood, it is convenient to use the EM algorithm.

Remark. EM is very sensitive to initial conditions: parameters must be initialized in a non-symmetric fashion.

EM training

For i.i.d. data, the energy is:

$$\begin{aligned}\sum_n \langle \log p(\mathbf{v}^n, h) \rangle_{p^{old}(h|\mathbf{v}^n)} \\ = \sum_n \sum_i \langle \log p(v_i^n | h) \rangle_{p^{old}(h|\mathbf{v}^n)} + \sum_n \langle \log p(h) \rangle_{p^{old}(h|\mathbf{v}^n)}\end{aligned}$$

(The notation in the book has changed: $p^{old}(h|\mathbf{v})$ used to be denoted as $q(h|\mathbf{v})$.)

Finding the updates

Since the energy terms are simply related to KL divergences, we can immediately write down the EM updates.

Updates

M-step

$$\begin{aligned} p^{new}(v_i = 1|h = j) &= \frac{\sum_n \mathbb{I}[v_i^n = 1] p^{old}(h = j|\mathbf{v}^n)}{\sum_n \mathbb{I}[v_i^n = 1] p^{old}(h = j|\mathbf{v}^n) + \sum_n \mathbb{I}[v_i^n = 0] p^{old}(h = j|\mathbf{v}^n)} \\ p^{new}(h = j) &= \frac{\sum_n p^{old}(h = j|\mathbf{v}^n)}{\sum_{h'} \sum_n p^{old}(h'|\mathbf{v}^n)} = \frac{\sum_n p^{old}(h = j|\mathbf{v}^n)}{N} \end{aligned}$$

E-step

$$p^{old}(h = j|\mathbf{v}^n) \propto p^{new}(h = j) \prod_{i=1}^D p^{new}(v_i^n|h = j)$$

Clustering binary digits

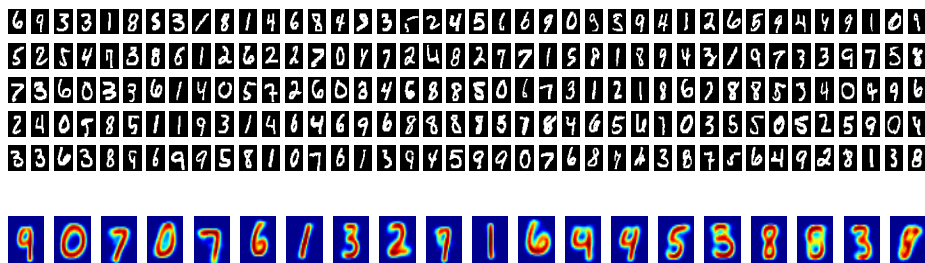


Figure: Top: a selection of 200 of the 5000 handwritten digits in the training set.
Bottom: the trained cluster outputs $p(v_i = 1|h)$ for $h = 1, \dots, 20$ mixtures.

Clustering Questionnaires

If an attribute i is missing for this model one simply removes the corresponding factor $p(v_i^n|h)$ from the algorithm.

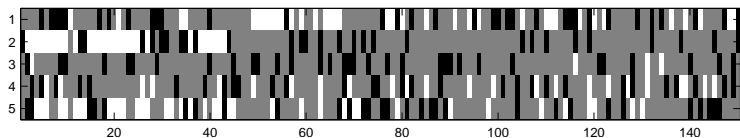


Figure: Data from questionnaire responses. 150 people were each asked 5 questions, with 'yes' (white) and 'no' (gray) answers. Black denotes that the absence of a response (missing data). This training data was generated by two component Binomial mixture. Missing data was simulated by randomly removing values from the dataset.

Gaussian Mixture Model

A D dimensional Gaussian distribution for a continuous variable \mathbf{x} is

$$p(\mathbf{x}|\mathbf{m}, \mathbf{S}) = \frac{1}{\sqrt{\det(2\pi\mathbf{S})}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m}) \right\}$$

where \mathbf{m} is the mean and \mathbf{S} is the covariance matrix. A mixture of Gaussians is then

$$p(\mathbf{x}) = \sum_{i=1}^H p(\mathbf{x}|\mathbf{m}_i, \mathbf{S}_i)p(i)$$

where $p(i)$ is the mixture weight for component i .

Maximum Likelihood

For a set of data $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ and under the usual i.i.d. assumption, the log likelihood is

$$\sum_{n=1}^N \log \sum_{i=1}^H p(i) \frac{1}{\sqrt{\det(2\pi \mathbf{S}_i)}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^n - \mathbf{m}_i)^\top \mathbf{S}_i^{-1} (\mathbf{x}^n - \mathbf{m}_i) \right\}$$

Parameter Constraints

The \mathbf{S}_i must be symmetric positive definite matrices, in addition to $0 \leq p(i) \leq 1$, $\sum_i p(i) = 1$. Gradient based optimisation approaches are feasible under a parameterisation of the \mathbf{S}_i (e.g. Cholesky decomposition) and $p(i)$ (e.g. softmax) that enforce the constraints. An alternative is the EM approach which in this case is particularly convenient since it automatically provides parameter updates that ensure these constraints.

Cálculo Matricial

Para un vector \mathbf{x} y matrices \mathbf{A} y \mathbf{B} :

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{A} \mathbf{x} = \mathbf{A}$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = (\mathbf{A} + \mathbf{A}^T) \mathbf{x}$$

luego, si \mathbf{A} es simétrica:

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = 2\mathbf{A} \mathbf{x} \quad \frac{\partial}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{s})^T \mathbf{A} (\mathbf{x} - \mathbf{s}) = 2\mathbf{A} (\mathbf{x} - \mathbf{s})$$

Además:

$$\frac{\partial}{\partial \mathbf{A}} \text{trace}(\mathbf{A} \mathbf{B}) \equiv \mathbf{B}^T$$

$$\frac{\partial}{\partial \mathbf{A}} \log \det(\mathbf{A}) = \mathbf{A}^{-T}$$

Expectation Maximisation

The energy term is

$$\sum_{n=1}^N \langle \log p(\mathbf{x}^n, i) \rangle_{p^{old}(i|\mathbf{x}^n)} = \sum_{n=1}^N \langle \log [p(\mathbf{x}^n|i)p(i)] \rangle_{p^{old}(i|\mathbf{x}^n)}$$

Plugging in the definition of the Gaussian components, we have

$$\sum_{n=1}^N \sum_{i=1}^H p^{old}(i|\mathbf{x}^n) \left\{ -\frac{1}{2} (\mathbf{x}^n - \mathbf{m}_i)^T \mathbf{S}_i^{-1} (\mathbf{x}^n - \mathbf{m}_i) - \frac{1}{2} \log \det (2\pi \mathbf{S}_i) + \log p(i) \right\}$$

M-step : optimal mixture coefficients

If no constraint is placed on the weights, the update is simply given by

$$p(i) = \frac{1}{N} \sum_{n=1}^N p^{old}(i|\mathbf{x}^n)$$

M-step : optimal \mathbf{m}_i

Maximising the energy with respect to \mathbf{m}_i is equivalent to minimising

$$\sum_{n=1}^N \sum_{i=1}^H p^{old}(i|\mathbf{x}^n) (\mathbf{x}^n - \mathbf{m}_i)^T \mathbf{S}_i^{-1} (\mathbf{x}^n - \mathbf{m}_i)$$

Differentiating with respect to \mathbf{m}_i and equating to zero, optimally

$$\mathbf{m}_i = \frac{\sum_{n=1}^N p^{old}(i|\mathbf{x}^n) \mathbf{x}^n}{\sum_{n=1}^N p^{old}(i|\mathbf{x}^n)}$$

By defining the membership distribution

$$p^{old}(n|i) \equiv \frac{p^{old}(i|\mathbf{x}^n)}{\sum_{n=1}^N p^{old}(i|\mathbf{x}^n)}$$

which quantifies the membership of datapoints to cluster i ,

$$\mathbf{m}_i = \sum_{n=1}^N p^{old}(n|i) \mathbf{x}^n$$

M-step : optimal \mathbf{S}_i

Optimising the energy with respect to \mathbf{S}_i is equivalent to minimising

$$\sum_{n=1}^N \left\langle (\Delta_i^n)^\top \mathbf{S}_i^{-1} \Delta_i^n - \log \det (\mathbf{S}_i^{-1}) \right\rangle_{p^{old}(i|\mathbf{x}^n)}$$

where $\Delta_i^n \equiv \mathbf{x}^n - \mathbf{m}_i$. We isolate the dependency on \mathbf{S}_i to give

$$\text{trace} \left(\mathbf{S}_i^{-1} \sum_{n=1}^N p^{old}(i|\mathbf{x}^n) \Delta_i^n (\Delta_i^n)^\top \right) - \log \det (\mathbf{S}_i^{-1}) \sum_{n=1}^N p^{old}(i|\mathbf{x}^n)$$

Differentiating with respect to \mathbf{S}_i^{-1} and equating to zero, we obtain

$$\sum_{n=1}^N p^{old}(i|\mathbf{x}^n) \Delta_i^n (\Delta_i^n)^\top - \mathbf{S}_i \sum_{n=1}^N p^{old}(i|\mathbf{x}^n) = \mathbf{0}$$

Using the membership $p^{old}(n|i)$, the optimal \mathbf{S}_i is given by

$$\mathbf{S}_i = \sum_{n=1}^N p^{old}(n|i) (\mathbf{x}^n - \mathbf{m}_i) (\mathbf{x}^n - \mathbf{m}_i)^\top$$

This ensures that \mathbf{S}_i is symmetric positive semi-definite.

M-step diagonal covariance

- For high dimensional data, learning the covariance matrix is problematic, and one often restricts the covariance to be diagonal.
- It is straightforward to show that in the diagonal case, the M-step corresponds to taking the diagonal entries of the general covariance update.

E-step

$$p(i|\mathbf{x}^n) \propto p(\mathbf{x}^n|i)p(i)$$

Explicitly, this is given by the responsibility

$$p(i|\mathbf{x}^n) = \frac{p(i) \exp \left\{ -\frac{1}{2}(\mathbf{x}^n - \mathbf{m}_i)^\top \mathbf{S}_i^{-1} (\mathbf{x}^n - \mathbf{m}_i) \right\} \det(\mathbf{S}_i)^{-1/2}}{\sum_{i'} p(i') \exp \left\{ -\frac{1}{2}(\mathbf{x}^n - \mathbf{m}_{i'})^\top \mathbf{S}_{i'}^{-1} (\mathbf{x}^n - \mathbf{m}_{i'}) \right\} \det(\mathbf{S}_{i'})^{-1/2}}$$

Infinite troubles

Consider placing a component $p(\mathbf{x}|\mathbf{m}_i, \mathbf{S}_i)$ with mean \mathbf{m}_i set to one of the datapoints $\mathbf{m}_i = \mathbf{x}^n$. The contribution from that Gaussian will be

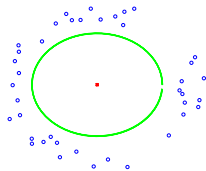
$$p(\mathbf{x}^n|\mathbf{m}_i, \mathbf{S}_i) = \frac{1}{\sqrt{\det(2\pi\mathbf{S}_i)}} e^{-\frac{1}{2}(\mathbf{x}^n - \mathbf{x}^n)^\top \mathbf{S}_i^{-1}(\mathbf{x}^n - \mathbf{x}^n)} = \frac{1}{\sqrt{\det(2\pi\mathbf{S}_i)}}$$

As the covariance goes to zero, this probability density becomes infinite. This means that one can obtain a Maximum Likelihood solution by placing zero-width Gaussians on a selection of the datapoints, resulting in an infinite likelihood.

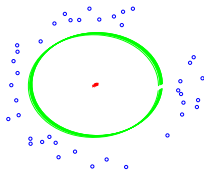
Remedy

Include an additional constraint on the width of the Gaussians, ensuring that they cannot become too small.

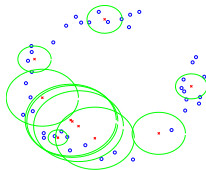
Symmetry Breaking



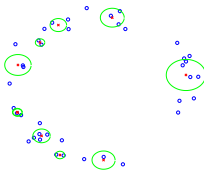
(a) 1 iteration



(b) 50 iterations



(c) 125 iterations



(d) 150 iterations

Figure: Training a mixture of 10 isotropic Gaussians **(a)**: If we start with large variances for the Gaussians, even after one iteration, the Gaussians are centred close to the mean of the data. **(b)**: The Gaussians begin to separate **(c)**: One by one, the Gaussians move towards appropriate parts of the data **(d)**: The final converged solution. The Gaussians are constrained to have variances greater than a set amount.

K-Means

Consider a mixture of K isotropic Gaussians in which each covariance is constrained to be equal to $\sigma^2 \mathbf{I}$,

$$p(\mathbf{x}) = \sum_{i=1}^K p_i \mathcal{N}(\mathbf{x} | \mathbf{m}_i, \sigma^2 \mathbf{I})$$

As $\sigma^2 \rightarrow 0$, the membership distribution becomes deterministic

$$p(i | \mathbf{x}^n) \propto \begin{cases} 1 & \text{if } \mathbf{m}_i \text{ is closest to } \mathbf{x}^n \\ 0 & \text{otherwise} \end{cases}$$

In this limit the EM update for the mean \mathbf{m}_i is given by taking the average of the points closest to \mathbf{m}_i .

Vector Quantisation

K-means is often used as a simple form of data compression. Rather than sending the datapoint \mathbf{x}^n , one sends instead the index of the centre to which it is associated. This is called vector quantisation.