Clustering using a generative statistical model

Associate each cluster with the same model type, but with different parameters.

Example (Gaussian Mixture Model (GMM)),

$$p(\mathbf{x}|c) = \mathrm{N}(\mathrm{u}_c, \Sigma_c)$$

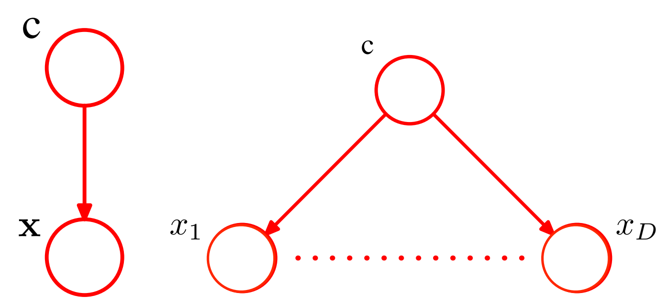
or, assuming feature independence,

$$p(\mathbf{x}|c) = \mathrm{N}(\mathrm{u}_c, \sigma_c^2)$$

$p(\mathbf{x}|c)$ could also be a product of independent multinomials, or, even a product of different distributions (roll your own!).

Clustering using a generative statistical model

Graphical model (and for independent features)



Inference given a clustering

Given a learned clustering model (either supervised or unsupervised), we can compute a posterior probability of which cluster an instance belongs to.

$$p(c|\mathbf{x}) \propto p(\mathbf{x}|c)\, p(c)$$

Easily normalized since the number of clusters is limited:

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)\, p(c)}{\sum_c p(\mathbf{x}|c)\, p(c)}$$

Clustering models representing data statistics

What is the distribution of data best described by clusters? (Example, data coming from a bimodal distribution?)

$$p(\mathbf{x}) = \sum_c p(\mathbf{x}, c) = \sum_c p(c)\, p(\mathbf{x}|c)$$

Generative story:

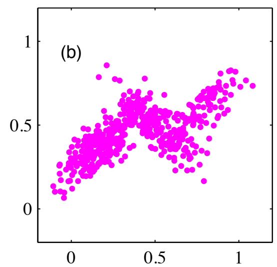
1) choose a cluster with probability, $p(c)$.
2) sample from $p(\mathbf{x}|c)$.
3) rinse and repeat.

Clustering models representing data statistics

Distribution of data described by clusters.

$$p(\mathbf{x}) = \sum_c p(c)\, p(\mathbf{x}|c)$$

Distribution modeled 3 multivariate Gaussians.



Even if we know the exact model, we cannot be sure from the data which point comes from which cluster. We only have the distribution for this.

Learning the parameters from data

For concreteness, assume GMM

Assume K clusters

The goal is to learn mixing coefficients, $p(c)$, and cluster parameters for $p(\mathbf{x}|c)$ for all K clusters indexed by c.

Learning the parameters from data

The goal is to learn mixing coefficients, $p(c)$, and cluster parameters for $p(\mathbf{x}|c)$ for all K clusters indexed by c.

From previous arguments, given $p(\mathbf{x}|c)$, we know the distribution over clusters for each data poing.

Hence we simultaneously cluster and learn a cluster model.

Learning the parameters from data

$$p(\mathbf{x}_i|\theta) = \sum_c p(c)\, p(\mathbf{x}_i|c,\theta_c)$$

Probability of all observed data will be the objective function

$$p(\{\mathbf{x}_i\}|\theta) = \prod_i \left(\sum_c p(c)\, p(\mathbf{x}_i|c,\theta_c) \right)$$ (want this to be large)

or

$$\sum_i \log\left(\sum_c p(c)\, p(\mathbf{x}_i|c,\theta_c) \right)$$ (should be large)

Expectation Maximization (EM)

Operationally this is similar to K-means.

Observe that:

If we knew the cluster assignments,
we could estimate the parameters for $p(\mathbf{x}|c)$.

If we knew $p(\mathbf{x}|c)$, we can make
cluster assignments.

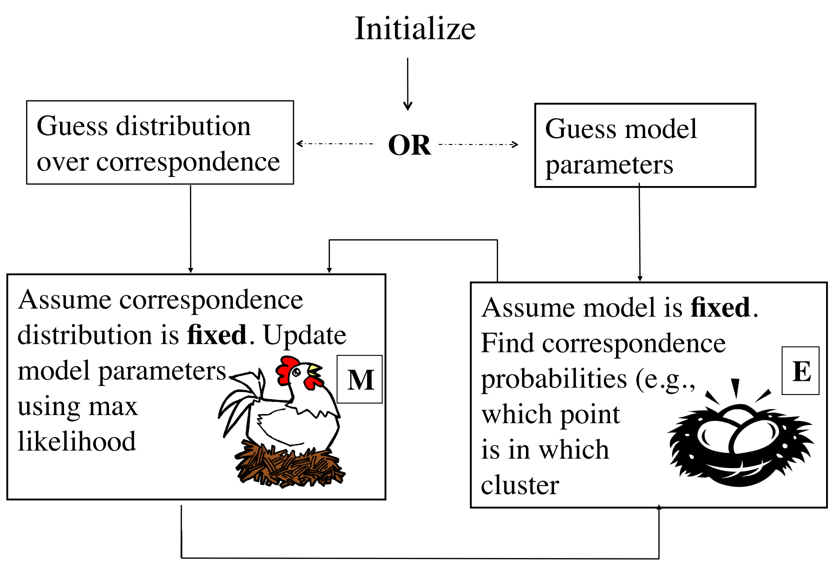
Expectation Maximization (EM)

Difference with K-means.

We have **distributions** over the assignments, $p(c\,|\,\mathbf{x})$.

This leads us to work with expectations.

EM flow chart



EM for GMM

$$p(\mathbf{x}) = \sum_c p(c)\,p(\mathbf{x}\,|\,c) \qquad \text{where} \qquad p(\mathbf{x}\,|\,c) = \mathbb{N}\left(\boldsymbol{\mu}_c, \Sigma\right)$$

$$\Theta = \{\Theta_c\}$$

And, for multiple points

$$p(\{\mathbf{x}_i\}|\theta) = \prod_i \left(\sum_c p(c)\,p(\mathbf{x}\,|\,c) \right)$$

This is our objective function.

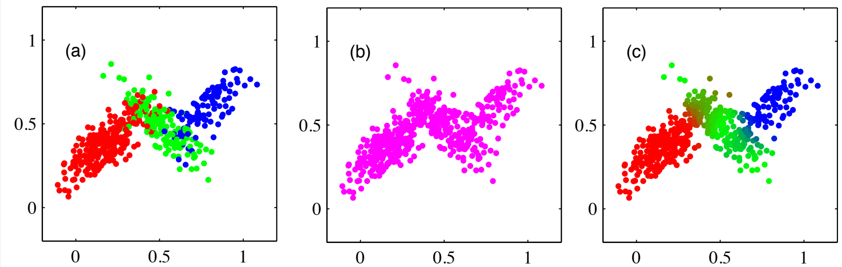
EM for GMM

Assume we have estimates for the probability distribution over clusters for each point (the "egg").

$p(c \mid \mathbf{x}_i, \Theta^{(s)})$ (s indexes interation (step)).

These are called the responsibilities (of the cluster for the point).

Responsibilities illustrated



EM for GMM

- We estimate the mean for each segment naturally by:

Iteration (step)

$$\mu_c^{(s+1)} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \bullet p(c \mid \mathbf{x}_i, \Theta_c^{(s)})}{\sum_{i=1}^{n} p(c \mid \mathbf{x}_i, \Theta_c^{(s)})} \quad \text{(weighted average)}$$

- Variances/covariances work similarly

EM for GMM

- Also, intuitively,

$$p(c) = \frac{\sum_i p\left(c \middle| \mathbf{x}_i, \Theta^{(s)}\right)}{\sum_c \sum_i p\left(c \middle| \mathbf{x}_i, \Theta^{(s)}\right)} = \frac{\sum_i p\left(c \middle| \mathbf{x}_i, \Theta^{(s)}\right)}{N}$$

We can sort out the chicken!

EM for GMM

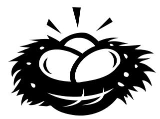
Given the parameters (the chicken), the probability that a given point is associated with each cluster is computed by:

$$p(c \mid \mathbf{x}_i, \Theta^{(s)}) = \frac{\pi_c^{(s)} \bullet p(\mathbf{x}_i \mid \Theta_c^{(s)})}{\sum_{c'} \pi_{c'}^{(s)} \bullet p(\mathbf{x}_i \mid \Theta_{c'}^{(s)})}$$

(Note that we select $\Theta_c^{(s)}$ from $\Theta^{(s)}$.

where $\pi_c^{(s)} = p\left(c \middle| \Theta_c^{(s)}\right)$ i.e., $\pi_c^{(s)}$ is part of $\Theta_c^{(s)}$.

We can do the egg!



This is the cluster membership discussed before,

with less formal notation: $p\left(c \middle| x\right) \propto p\left(c\right) p\left(x \middle| c\right)$

EM illustrated