Mixture of Gaussians: advanced unsupervised learning

Like k-means, can be used to cluster data without labels.

Assumes that the probability distribution of the data **x** can be described as a weighted sum of Gaussians

$$\mathrm{p}(\mathbf{x}) = \sum_{i=1}^{K} \pi_i \, N(\mathbf{x}|\mu_i, \Sigma_i)$$

p(z) p(**x**|z)

It's a quite rich model, that can describe complicated distributions p(**x**) (very different from a single Gaussian).
It's convenient to describe complex models in terms of "sums of easy Gaussians", and each component *i* is seen as a value of a latent variable (that you do not observe explicitly) that affects locally the distribution of the data around μ_i (like a k-means centroid)

With several observations {**x**_1, …, **x**_N} there is a latent variable z_n for each observation

Maximum likelihood fit of GMM: the EM algorithm

Suppose we have a dataset of observations {**x**_1, . . . , **x**_N}, and we wish to model this data using a mixture of Gaussians.

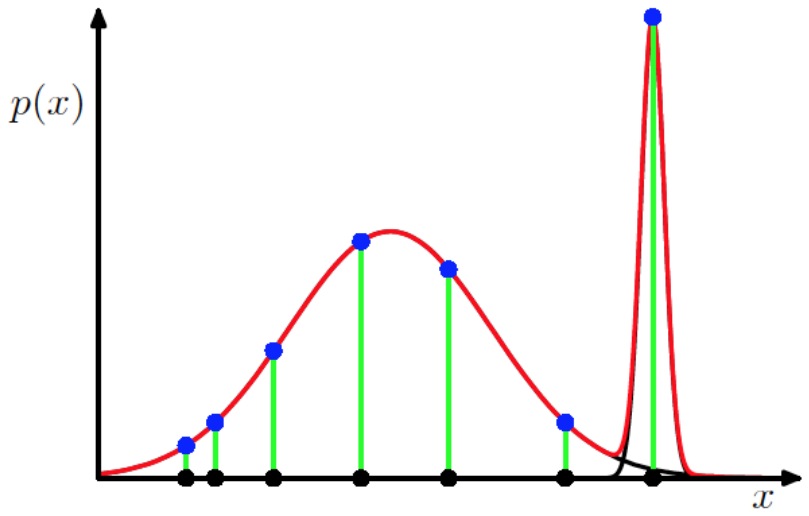
A maximum likelihood approach estimates the parameters of the GMM that best fit data (in the ML sense).

No analytical solution as for one single Gaussian, so EM numerical algorithm: each iteration is made of a

- E step: evaluate posteriors p(component|**x**) from current estimates of model parameters (in k-means this is assigning each data point to closest centroid)
- M step: use current posteriors to re-update model parameters (means, covariances, and components' weights) (in k-means this is updating the centroids)

EM algorithm for GMM takes many more iterations to converge and each iteration requires more computation.
You can first run K-means to initialize the GMM model and then run EM

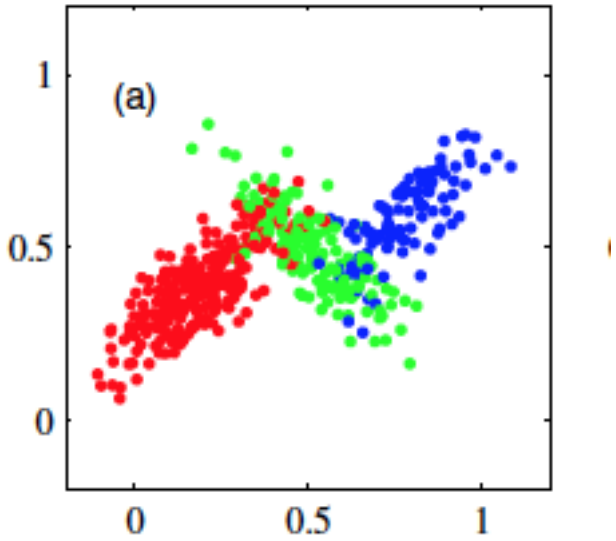
However, in this ML optimization there are singularities whenever one of the Gaussian components 'collapses' onto a specific data point (a case of overfitting).
This does not happen with a single Gaussian (see figure)



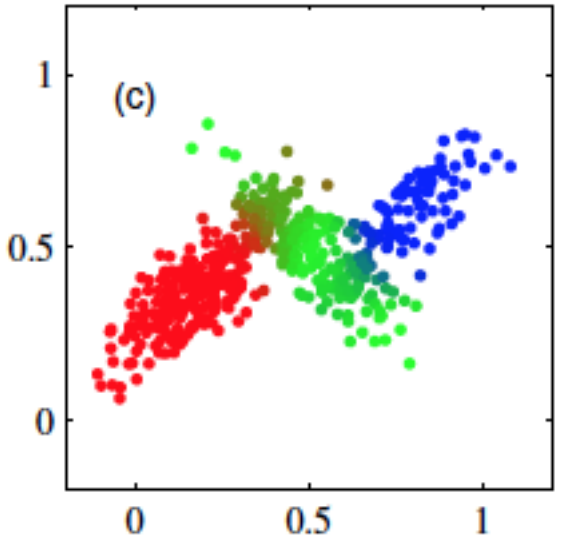
What GMM gives you more than k-means

In each iteration, *K*-means assigns each data point uniquely to one, and only one, of the clusters. This can be unconvenient for points that lie roughly midway between centroids. GMM allows for 'soft' assignments of data points to clusters (components) in a way that reflects the level of uncertainty over the most appropriate assignment.

In practice, while k-means only assigns data to different centroids (type 3 of classification algorithms described in Session 2), GMM also models posterior distributions (type 2 of algorithms, and probably also type 1).



Also done by k-means



Impossible with k-means

k-means as a limiting case of GMM

Take GMM with common fixed variance ε for all components

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2\right\}.$$

The posteriors for each component and data point p(z_k|**x**_n) are then

$$\gamma(z_{nk}) = \frac{\pi_k \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\}}{\sum_j \pi_j \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\}}.$$

In the limit ε->0 (hard assignments, no chance of error) only one p(z_k|**x**_n) (say, k=1) is 1 and all the others are zero: as a consequence, **x**_n is assigned to the component having the closest mean.
Also formula to update means of the components coincides with update of k-means centroids in the limit ε->0.