Machine Learning Course - CS-433

Gaussian Mixture Models

Nov 15, 2018

minor changes by Martin Jaggi 2016

minor changes by Martin Jaggi 2017

changes by Ruediger Urbanke 2018

Last updated: November 9, 2018

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

Recall that K-means was equivalent to assuming that the data came from K spherically symmetric Gaussians and then maximing the likelihood of the data. In other words, we built into the model the assumption that the clusters are have a *spherical* symmetry. But sometimes it is desirable to have *elliptical* clusters. Further, in K-means, each sample belongs to exactly one cluster. This may not always be a good choice. E.g. think of data points that are near the "boundary". Both of these problems are easily solved by using Gaussian Mixture Models.

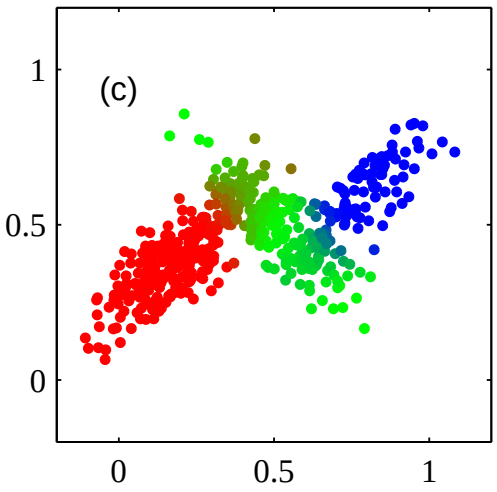
Clustering with Gaussians

The first issue is trivially resolved by using full covariance matrices $\mathbf{\Sigma}_k$ instead of *isotropic* covariances.

$$p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} [\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)]^{z_{nk}}$$

Soft-clustering

The second issue is resolved by allowing fractional assignment. We can interpret these fractional assignments as probabilities.
I.e., think of z_n as a random variable taking values in $\{1, 2, \ldots, K\}$. And assume that the prior distribution of z_n follows a multi-



nomial distribution,

$$p(z_n = k) = \pi_k \text{ where } \pi_k > 0 \forall k, \text{ and } \sum_{k=1}^{K} \pi_k = 1.$$

This leads to soft-clustering as opposed to having "hard" assignments.

Gaussian mixture model

If we combine now both of these idea we get Gaussian mixture models (GMMs):

$$\begin{aligned} & p(\mathbf{X}, \mathbf{z} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ &= \prod_{n=1}^{N} p(\mathbf{x}_n | z_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(z_n | \boldsymbol{\pi}) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} [\mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{z_{nk}} \prod_{k=1}^{K} [\pi_k]^{z_{nk}} \end{aligned}$$

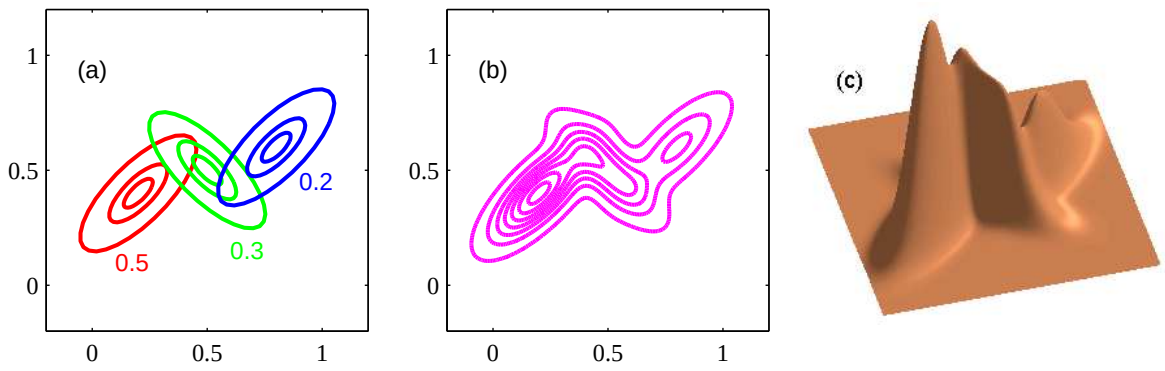
Here, $\mathbf{x}_n$ are observed data vectors, z_n are *latent* unobserved variables, and the unknown *parameters* are given by $\boldsymbol{\theta} :=$

$\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K, \boldsymbol{\pi}\}$.

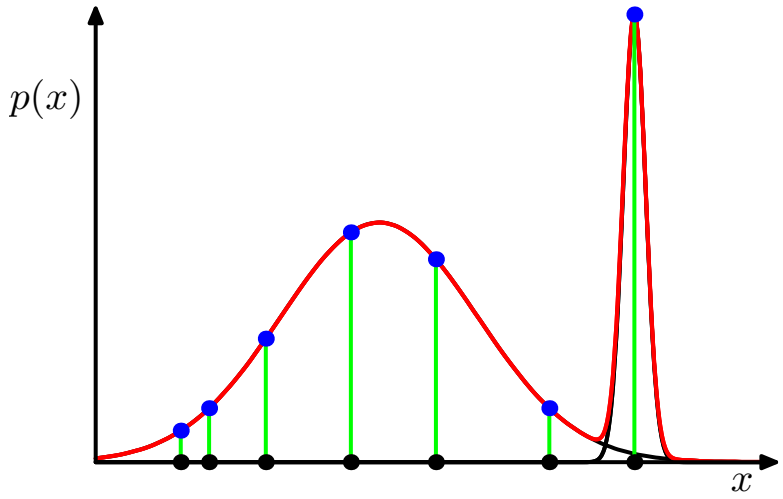
Marginal likelihood

GMM is a latent variable model with z_n being the unobserved (latent) variables. An advantage of treating z_n as latent variables instead of *parameters* is that we can *marginalize* them out to get a cost function that does not depend on z_n, i.e. as if z_n never existed.
Specifically, we get the following marginal likelihood by marginalizing z_n out from the likelihood:

$$p(\mathbf{x}_n|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Deriving cost functions this way, is good for *statistical efficiency*. Without a latent variable model, the number of parameters grow at a rate $O(N)$. After marginalization, the growth is reduced to $O(D^2K)$ (assuming $D, K \ll N$).



Maximum likelihood

To get a maximum (marginal) likelihood estimate of $\boldsymbol{\theta}$, we maximize the following:

$$\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Is this cost convex? Identifiable? Bounded?