Machine Learning Course - CS-433

Gaussian Mixture Models

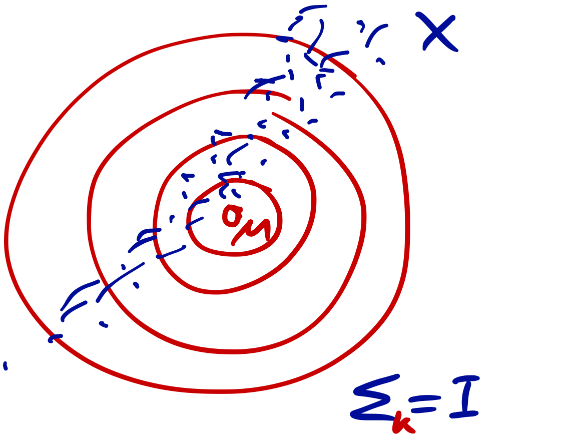
Nov 10, 2016

minor changes by Martin Jaggi 2016

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

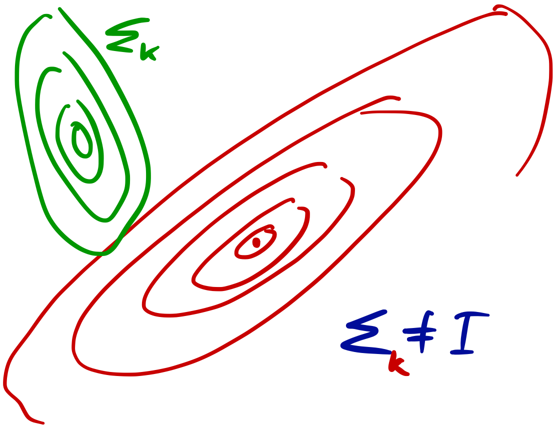
Motivation

K-means forces the clusters to be *spherical*, but sometimes it is desirable to have *elliptical* clusters. Another issue is that, in K-means, each example can only belong to one cluster, but this may not always be a good choice, e.g. for data points that are near the "border". Both of these problems are solved by using Gaussian Mixture Model.



Clustering with Gaussians

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



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

$I^{D\times D}$ for K-means

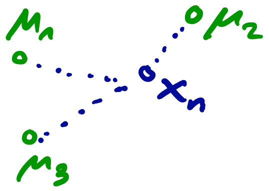
any Σ_k (p.s.d.)

$\Sigma = (\Sigma_1, \ldots, \Sigma_K)$

$\mu = (\mu_1, \ldots, \mu_K)$

Soft-clustering

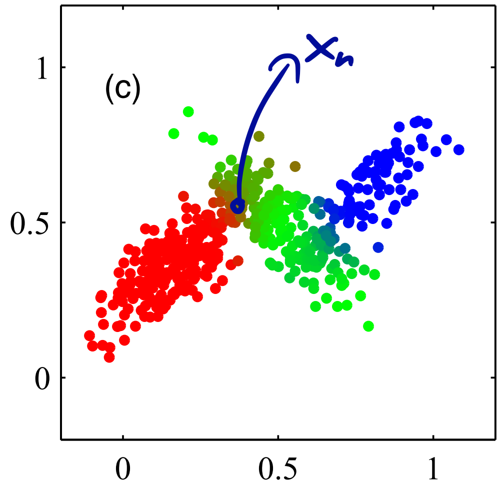
The second issue is resolved by defining z_n to be a random variable. Specifically, define $z_n \in \{1, 2, \ldots, K\}$ that follows a multinomial distribution.

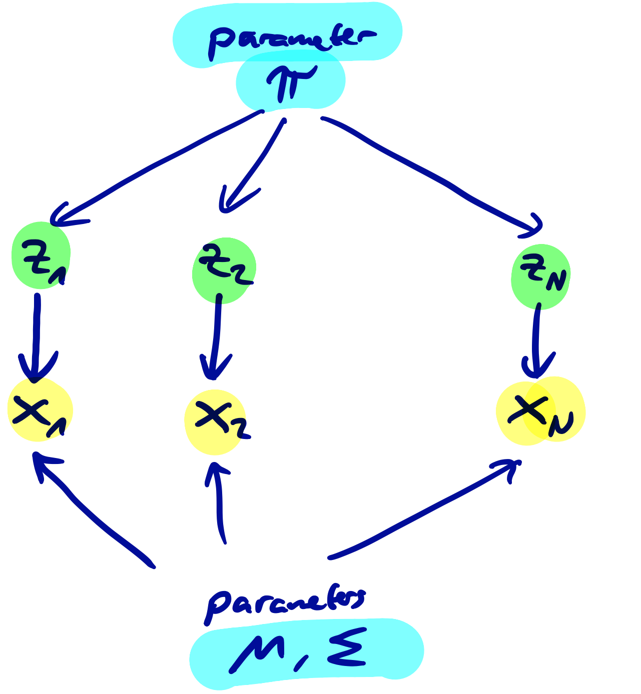


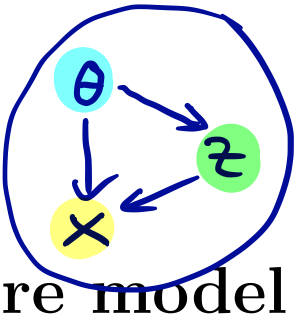
$$\boxed{p(z_n = k)} = \pi_k \text{ where } \pi_k > 0, \forall k \text{ and } \sum_{k=1}^{K} \pi_k = 1$$

$$\underbrace{\begin{array}{ll} n \to 1 & P = 0.1 \\ n \to 2 & P = 0.7 \\ n \to 3 & P = 0.2 \end{array}}_{z_n}$$

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







Gaussian mixture model

$\theta = (\mu, \Sigma, \pi)$

Together, the likelihood and the prior define the joint distribution of Gaussian mixture model (GMM):

Bayes Rule
$p(a, b|\,) = p(a|b\,)\, p(b|\,)$

$$p(\mathbf{X}, \mathbf{z}\,|\,\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = p(X|z, \mu, \Sigma, \pi) \cdot p(z|\mu, \Sigma, \pi)$$

$$= \prod_{n=1}^{N} p(\mathbf{x}_n|z_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \cancel{\pi})\, p(z_n|\boldsymbol{\pi}, \cancel{\mu, \Sigma})$$

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

$z_n: = (0, 1, 0)$
↑
$\pi = (0.1, 0.7, 0.2)$
$\quad\;\; = \pi_1 \;\; = \pi_2 \;\; = \pi_3$

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.

$$\underline{\text{marginal}}$$

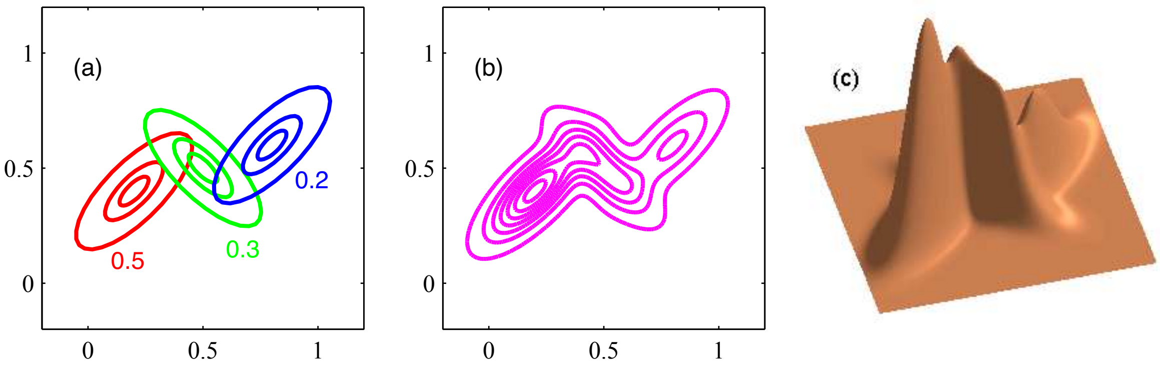
$$p(a,b) \quad \text{joint}$$

$$\overset{\text{marginal}}{p(a)} = \sum_k p(a, b=k)$$

$$= \sum_k p(a|b=k) \cdot p(b=k)$$

Specifically, we get the following marginal likelihood by marginalizing z_n out from the likelihood:

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



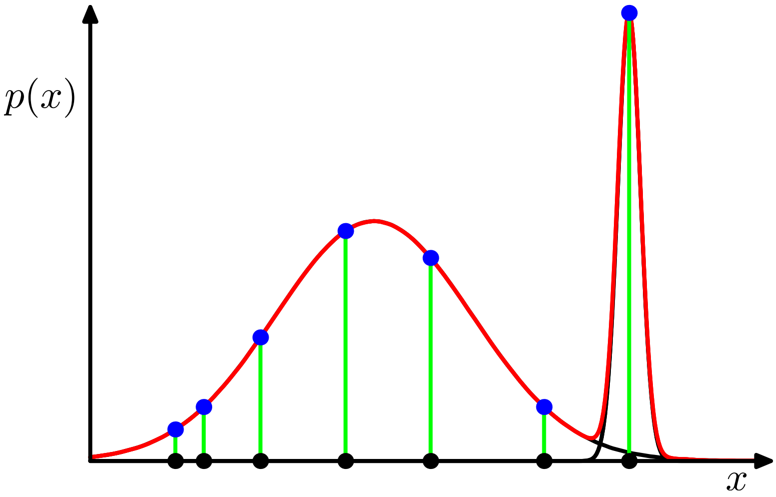
Deriving cost functions this way, is good for *statistical efficiency.* Without a latent variable model, the number of parameters grow at rate $O(N)$. After marginalization, the growth is reduced to $O(D^2K)$ (assuming $D, K \ll N$).

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?



ToDo

1. Understand K-means extension to GMM. Why do we need to treat z_n as a random variable? Identify the joint, likelihood, prior, and marginal distributions, respectively.