STA 414/2104: Statistical Methods in Machine Learning II

Week 9 : Variational Inference II/EM algorithm

Murat A. Erdoğdu and **Piotr Zwiernik**

University of Toronto

Overview for the first hour

- Variational Inference
- ELBO and its properties
- Estimating gradients of the ELBO
  - Simple Monte Carlo
  - Reparameterization trick

Recap: Posterior Inference for Latent Variable Models

We encountered a few latent variable models (e.g. the trueskill model).

These models have a factorization $p(x, z) = p(z)p(x|z)$ where:

- x are the observations or data,
- z are the unobserved (latent) variables
- $p(z)$ is usually called the **prior**
- $p(x|z)$ is usually called the **likelihood**
- The conditional distribution of the unobserved variables given the observed variables (aka the **posterior**) is

$$p(z|x) = \frac{p(x, z)}{p(x)} = \frac{p(x, z)}{\int p(x, z)dz}$$

- We assume $p(x) = \int p(x, z)dz$ is hard to compute

Variational methods

Variational inference works as follows:

- Choose a tractable parametric distribution $q_\phi(z)$ with parameters ϕ. This distribution will be used to approximate $p(z|x)$.
    - For example, $q_\phi(z) = \mathcal{N}(z|\mu, \Sigma)$ where $\phi = (\mu, \Sigma)$.
- Encode some notion of "distance" between $p(z|x)$ and $q_\phi(z)$ that can be efficiently estimated. Usually we will use the KL divergence.
- Minimize this distance.

Recall: KL divergence

We will measure the difference between q and p using the **Kullback-Leibler divergence**

$$\begin{aligned}\mathrm{KL}(q_\phi(z)\|p(z|x)) &= \int q_\phi(z)\log\frac{q_\phi(z)}{p(z|x)}dz \\ &= \mathop{\mathbb{E}}_{z\sim q_\phi}\log\frac{q_\phi(z)}{p(z|x)}\end{aligned}$$

Properties of the KL Divergence

- $\mathrm{KL}(q_\phi\|p)\geq 0$
- $\mathrm{KL}(q_\phi\|p)=0 \quad\Leftrightarrow\quad q_\phi=p$
- $\mathrm{KL}(q_\phi\|p)\neq\mathrm{KL}(p\|q_\phi)$
- KL divergence is not a metric, since it is not symmetric

ELBO: Evidence Lower Bound

- Evaluating $\mathrm{KL}(q_\phi(z)\|p(z|x))$ is intractable because of the integral over z and the term $p(z|x)$, which is intractable to normalize.

- We can still "optimize" this KL without knowing the normalization constant $p(x)$.

- We solve a surrogate optimization problem: maximize the **evidence lower bound (ELBO)**.

- Maximizing the ELBO is equivalent to minimizing

$$\mathrm{KL}(q_\phi(z)\|p(z|x)).$$

ELBO: Evidence Lower Bound

Maximizing the ELBO is the same as minimizing $\mathrm{KL}(q_\phi(z) \| p(z|x))$.

$$\begin{aligned}
\mathrm{KL}(q_\phi(z) \| p(z|x)) &= \mathop{\mathbb{E}}_{z \sim q_\phi} \log \frac{q_\phi(z)}{p(z|x)} \\
&= \mathop{\mathbb{E}}_{z \sim q_\phi} \left[\log \left(q_\phi(z) \cdot \frac{p(x)}{p(z,x)} \right) \right] \\
&= \mathop{\mathbb{E}}_{z \sim q_\phi} \left[\log \frac{q_\phi(z)}{p(z,x)} \right] + \mathop{\mathbb{E}}_{z \sim q_\phi} \log p(x) \\
&:= -\mathcal{L}(\phi) + \log p(x)
\end{aligned}$$

Where $\mathcal{L}(\phi)$ is the **ELBO**:

$$\mathcal{L}(\phi) = \mathop{\mathbb{E}}_{z \sim q_\phi} \Big[\log p(z,x) - \log q_\phi(z) \Big]$$

ELBO: Evidence Lower Bound

Recall: $\mathrm{KL}(q_\phi(z) \| p(z|x)) = -\mathcal{L}(\phi) + \log p(x)$.

- Rearranging, we get

$$\mathcal{L}(\phi) + \mathrm{KL}(q_\phi(z) \| p(z|x)) = \log p(x)$$

- Because $\mathrm{KL}(q_\phi(z) \| p(z|x)) \geq 0$,

$$\mathcal{L}(\phi) \leq \log p(x)$$

- maximizing the ELBO ⇒ minimizing $\mathrm{KL}(q_\phi(z) \| p(z|x))$.
- Note: $\mathcal{L}(\phi) = \underset{z \sim q_\phi}{\mathbb{E}}\Big[\log p(z, x)\Big] + \underset{z \sim q_\phi}{\mathbb{E}}\Big[-\log q_\phi(z)\Big]$, so

  ELBO = expected log-join+entropy

Maximizing ELBO

Recall: $\nabla \mathcal{L}(\phi)$ gives the direction of the steepest ascent of $\mathcal{L}(\phi)$.

Gradient descent (GD) methods: $\phi_{t+1} = \phi_t + s_t \nabla \mathcal{L}(\phi_t)$.

- We have that $\mathcal{L}(\phi) = \underset{z\sim q_\phi}{\mathbb{E}}\Big[\log p(x,z) - \log q_\phi(z)\Big]$.
- We need $\nabla_\phi \mathcal{L}(\phi)$ or its unbiased estimate (stochastic GD).

Approximating the gradient of some $\mathbb{E}(f(Y,\phi))$:

- If the distribution of Y independent of ϕ then

$$\nabla_\phi \mathbb{E}(f(Y,\phi)) = \mathbb{E}(\nabla_\phi f(Y,\phi)).$$

- We then have $\nabla_\phi \mathbb{E}(f(Y,\phi)) \approx \frac{1}{n}\sum_{i=1}^n \nabla_\phi f(y_i,\phi)$.
- Problem: In our case the distribution of z depends on ϕ.

The reparameterization trick

Problem:

$$\nabla_\phi \mathop{\mathbb{E}}_{z\sim q_\phi}\Big[\log p(x,z) - \log q_\phi(z)\Big] \neq \mathop{\mathbb{E}}_{z\sim q_\phi}\Big[\nabla_\phi \left(\log p(x,z) - \log q_\phi(z)\right)\Big].$$

But in some situations there is a trick:

- We break this sampling process into two parts:
  - Sample a random variable ϵ that has fixed (or no) parameters, such as a uniform distribution or standard normal.
  - Determinsitically compute z's as a function ϕ and ϵ, such that:
    - $\epsilon \sim p_0(\epsilon)$
    - $z = T(\epsilon, \phi)$
    - $\implies$
    - $z \sim q_\phi(z)$

The reparameterization trick

- For example, $z = \mu + \sigma\epsilon = T(\phi, \epsilon)$ (here $\phi = (\mu, \sigma)$).
  - $\epsilon \sim \mathcal{N}(0, 1)$
  - $z = \mu + \epsilon\sigma$
  - $\implies$
  - $z \sim \mathcal{N}(\mu, \sigma)$
- This makes the density independent of the parameter ϕ, which will let us use simple Monte Carlo: $z = T(\phi, \epsilon)$

$$\begin{aligned}
\nabla_\phi \mathcal{L}(\phi) &= \nabla_\phi \mathbb{E}_{z \sim q_\phi(z)} \Big[\log p(x, z) - \log q_\phi(z) \Big] \\
&= \nabla_\phi \mathbb{E}_{\epsilon \sim p_0(\epsilon)} \Big[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon)) \Big] \\
&= \mathbb{E}_{\epsilon \sim p_0(\epsilon)} \nabla_\phi \Big[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon)) \Big]
\end{aligned}$$

Example: Bayesian Neural Networks

- z are weights of neural network
- x are all observed outputs: $y_1, y_2, ...$ Assume inputs $\mathbf{x}_i$ are fixed.
- $p(z)$ prior on weights, usually standard normal (hard to set)
- $p(x|z) = \prod_i p(y_i|\mathbf{x}_i, z)$
    - for regression: $p(y_i|\mathbf{x}_i, z) = \mathcal{N}(nnet(\mathbf{x}_i, z), \sigma^2)$
    - for classification: $p(y_i|\mathbf{x}_i, z) = \text{Categorical}(y_i|\text{softargmax}(nnet(\mathbf{x}_i, z)))$
- $p(z|\mathbf{x}, y)$ is a collection of plausible sets of parameters that all fit the data (and have some probability under the prior). Certain where the data is, uncertain where extrapolation is ambiguous.

SVI: Stochastic Variational Inference

- Instead of computing the full gradient (which is in general not possible), we compute a simple Monte Carlo estimate of it.
- For example, instead of

$$\mathbb{E}_{\epsilon \sim p_0(\epsilon)} \nabla_\phi \Big[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon)) \Big]$$

we work with a mini-batch of size m

$$\begin{aligned} &\widehat{\mathbb{E}}_{\epsilon \sim p_0(\epsilon)} \nabla_\phi \Big[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon)) \Big] \\ &\approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\phi \Big[\log p(x, T(\phi, \epsilon_i)) - \log q_\phi(T(\phi, \epsilon_i)) \Big] \end{aligned}$$

MCMC: Pros & Cons

Pros of MCMC:

- Accurate results (at least asymptotically)
- Flexibility
- No approximation
- Handles multimodal distributions

Cons of MCMC:

- High computational cost
- Requires tuning of hyperparameters
- Convergence issues
- Inefficient in sampling complex dependencies

SVI: Pros & Cons

Pros of SVI:

- Faster convergence
- Scalability
- Ease of use

Cons of SVI:

- Approximate results
- Limited flexibility
- Mode seeking
- Sensitive to choice of hyperparameters

Summary of the first hour

We covered the basics of gradient-based stochastic variational inference.

More specifically:

- ELBO
- Reparametrization trick
- Stochastic VI

Overview of the second hour

- Gaussian mixture models
- EM-algorithm
- Clustering

Mixture of Gaussians

We combine simple models into a complex model by taking a mixture of K multivariate Gaussian densities of the form:

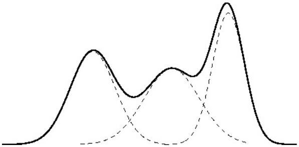
$$p(x) \;=\; \sum_{k=1}^{K} \pi_k N_m(x|\mu_k, \Sigma_k),$$

where $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$.

- Each Gaussian component has its own mean vector μ_k and covariance matrix Σ_k.
- The parameters π_k are called the mixing coefficients.

Example:

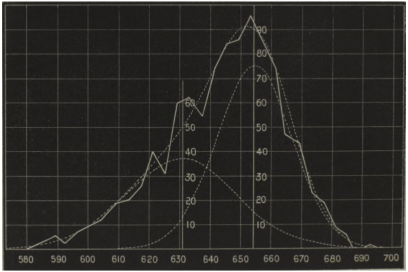
- $K = 3$ (three Gaussian components)
- $m = 1$ (univariate Gaussians)



The crabs from Naples bay

In 1892, scientists collected data on populations of the crab, Carcinus Moenas, and observed that the ratio of forehead width to the body length actually showed a highly skewed distribution.

On Certain Correlated Variations in Carcinus maenas (1893) W. F. Weldon

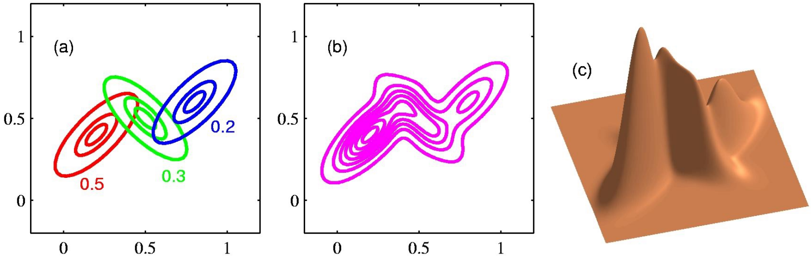


They wondered whether this distribution could be the result of the population being a mix of two different normal distributions (two sub-species).

In **1894**, Karl Pearson proposed a method to fit this model (read here), whose modern version is the "method of moments".

Mixture of Gaussians

- Illustration of a mixture of 3 Gaussians in a 2-dimensional space:



(a) Contours of constant density of each of the mixture components, along with the mixing coefficients

(b) Contours of marginal probability density $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

(c) A surface plot of the distribution p(x).

Mixture of Gaussians as a latent variable model

Recall: $p(x) = \sum_{k=1}^{K} \pi_k N_m(x|\mu_k, \Sigma_k)$.

- Consider a latent variable z with K states $z \in \{1, \ldots, K\}$.
- The distribution of z given by the mixing coefficients:

$$p(z = k) = \pi_k.$$

- Specify the conditional as $p(x|z = k) = N_m(x|\mu_k, \Sigma_k)$ with joint:

$$p(x, z = k) = p(z = k)p(x|z = k) = \pi_k N_m(x|\mu_k, \Sigma_k).$$

- Then the marginal $p(x)$ satisfies

$$p(x) = \sum_{k=1}^{K} p(x, z = k) = \sum_{k=1}^{K} \pi_k N_m(x|\mu_k, \Sigma_k).$$

Mixture of Gaussians: inference

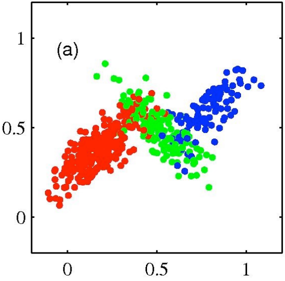
- If we have several observations $x_1, \dots, x_N$, for every observed data point x_n there is a corresponding latent z_n.
- Consider the conditional $p(z|x)$

$$
\begin{aligned}
p(z = k|x) &= \frac{p(z = k)p(x|z = k)}{\sum_{j=1}^{K} p(z = j)p(x|z = j)} \\
&= \frac{\pi_k N_m(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N_m(x|\mu_j, \Sigma_j)}
\end{aligned}
$$

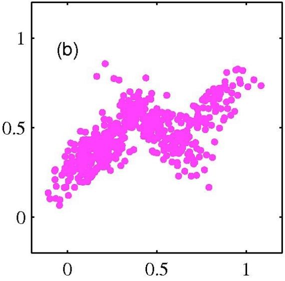
- We view π_k as prior probability that $z = k$, and $p(z = k|x)$ is the corresponding posterior once we have observed the data.

Example

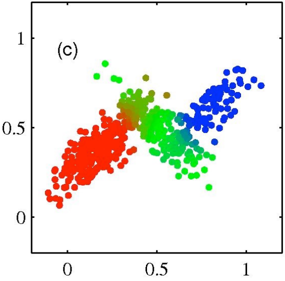
- 500 points drawn from a mixture of 3 Gaussians.



Samples from the joint distribution p(x,z).



Samples from the marginal distribution p(x).



Same samples where colors represent the value of responsibilities.

The Likelihood function

Parameters: $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K)$.
Recall: $p(x|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k N_m(x|\mu_k, \Sigma_k)$

- Represent the dataset $\{x_1, \ldots, x_N\}$ as $\boldsymbol{X} \in \mathbb{R}^{N \times m}$.
- The latent variable is represented by a vector $\boldsymbol{z} \in \mathbb{R}^N$.
- The log-likelihood takes the form

$$\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N_m(x_n|\mu_k, \Sigma_k) \right)$$

Maximum Likelihood ($\boldsymbol{\mu}$)

Recall: $\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N_m(x_n|\mu_k, \Sigma_k) \right)$.

- Differentiating wrt μ_k and setting to zero gives:

$$\begin{aligned} 0 &= \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j, \Sigma_j)} \Sigma_k^{-1}(x_n - \mu_k) \\ &= \sum_{n=1}^{N} p(z_n = k|x_n) \Sigma_k^{-1}(x_n - \mu_k). \end{aligned}$$

- Equivalently (as Σ_k is positive definite)

$$\mu_k = \sum_n \frac{p(z = k|x_n)}{N_k} x_n, \qquad N_k = \sum_n p(z = k|x_n).$$

- Simple interpretation: the MLE given by the weighted mean of the data weighted by the posterior $p(z = k|x_n)$.

Maximum Likelihood $(\boldsymbol{\Sigma}, \boldsymbol{\pi})$

Recall: $\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N_m(x_n|\mu_k, \Sigma_k) \right)$.

- Differentiating wrt Σ_k and setting to zero gives:

$$\Sigma_k = \sum_n \frac{p(z=k|x_n)}{N_k}(x_n - \mu_k)(x_n - \mu_k)^\top.$$

- Again data points weighted by posterior probabilities.

- Finally, for the weights π_k the MLE is

$$\pi_k = \frac{N_k}{\sum_{j=1}^{K} N_j} = \frac{N_k}{N}, \qquad N_k = \sum_n p(z=k|x_n).$$

Motivating the EM algorithm

- The MLE does not have a closed form solution.
- The estimates depend on the posterior probabilities $p(z = k|x_n)$, which themselves depend on those parameters.
- Indeed, recall that

$$p(z = k|x_n) \;=\; \frac{\pi_k N_m(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N_m(x_n|\mu_j, \Sigma_j)}.$$

- Iterative solution (EM algorithm):
  - Initialize the parameters to some values.
  - E-step Update the posteriors $p(z = k|x_n)$.
  - M-step Update model parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.
  - Repeat.

EM algorithm for Gaussian mixtures

- Initialize $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.
- E-step: for each k, n compute the posterior probabilities

$$p(z = k|x_n) \;=\; \frac{\pi_k N_m(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N_m(x_n|\mu_j, \Sigma_j)}.$$

- M-step: Re-estimate model parameters

$$\mu_k^{\text{new}} = \sum_{n=1}^{N} \frac{p(z = k|x_n)}{N_k} x_n, \qquad N_k = \sum_{n=1}^{N} p(z = k|x_n),$$

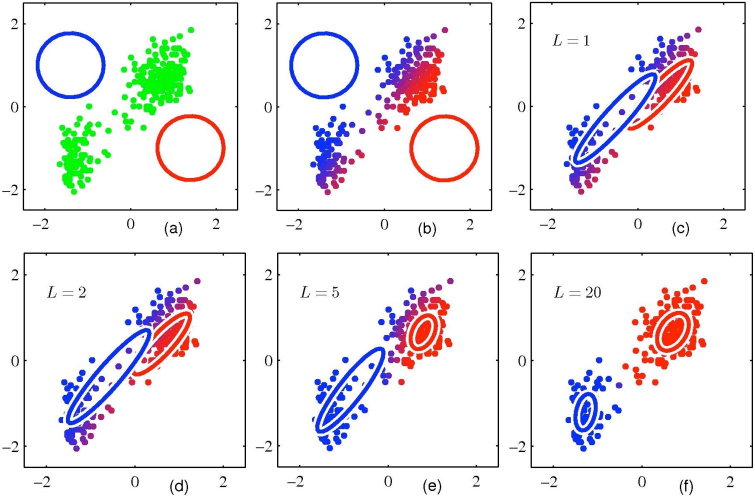
$$\Sigma_k^{\text{new}} = \sum_{n=1}^{N} \frac{p(z = k|x_n)}{N_k} (x_n - \mu_k^{\text{new}})(x_n - \mu_k^{\text{new}})^\top,$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}.$$

- Evaluate the log-likelihood and check for convergence.

Mixture of Gaussians: Example

- Illustration of the EM algorithm (much slower convergence compared to K-means)



The General EM algorithm

Consider a general setting with latent variables.

- Observed dataset $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, latent variables $\boldsymbol{Z} \in \mathbb{R}^{N \times K}$.

Maximize the log-likelihood $\log p(\boldsymbol{X}|\theta) = \log\left(\sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\theta)\right)$.

- Initialize parameters θ^{old}.
- **E-step**: use θ^{old} to compute the posterior $p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{\text{old}})$.
- **M-step**: $\theta^{\text{new}} = \arg\max_{\theta} Q(\theta, \theta^{\text{old}})$, where

$$
\begin{aligned}
Q(\theta, \theta^{\text{old}}) &= \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{\text{old}}) \log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) \\
&= \mathbb{E}\Big(\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) \Big| \boldsymbol{X}, \theta^{\text{old}} \Big)
\end{aligned}
$$

  which is tractable in many applications.

- Replace $\theta^{\text{old}} \leftarrow \theta^{\text{new}}$. Repeat until convergence.

Example: Gaussian mixture

- If z was observed, the MLE would be trivial

$$\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) = \sum_{n=1}^{N} \log p(x_n, z_n|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{1}(z_n = k) \log \left(\pi_k N(x_n|\mu_k, \Sigma_k)\right)$$

For the E-step: $p(\boldsymbol{Z}|\boldsymbol{X}, \theta) = \prod_{n=1}^{N} p(z_n|\boldsymbol{X}, \theta)$ we have

$$p(z_n = k|\boldsymbol{X}, \theta) = p(z_n = k|x_n, \theta) = \frac{\pi_k N_m(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N_m(x_n|\mu_j, \Sigma_j)}.$$

For the M-step: $\mathbb{E}(\mathbb{1}(z_n = k)|\boldsymbol{X}, \theta^{\text{old}}) = p(z_n = k|\boldsymbol{X}, \theta^{\text{old}})$ and so

$$\mathbb{E}\left(\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) \middle| \boldsymbol{X}, \theta^{\text{old}} \right) = \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k|\boldsymbol{X}, \theta^{\text{old}}) \log \left(\pi_k N(x_n|\mu_k, \Sigma_k)\right).$$

Maximizing gives the formulas on Slide 28.

Relationship to K-Means (STA 314)

- Consider a Gaussian mixture, s.t. $\Sigma_k = \epsilon I$ for all $k = 1, \dots, K$.
- We have

$$p(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi\epsilon)^{m/2}} \exp\left(-\frac{1}{2\epsilon}\|x - \mu_k\|^2\right).$$

- Consider the EM algorithm in this special case, $\theta = (\boldsymbol{\pi}, \boldsymbol{\mu})$.
- The posterior probabilities take the form

$$p(z_n = k|\boldsymbol{X}, \theta) = \frac{\pi_k \exp(-\|x_n - \mu_k\|^2/2\epsilon)}{\sum_{j=1}^{K} \pi_j \exp(-\|x_n - \mu_j\|^2/2\epsilon)}.$$

- If $\epsilon \to 0$, the term with smallest $\|x_n - \mu_j\|$ tends to zero most slowly.
- Thus $p(z_n = k|\boldsymbol{X}, \theta) \;\to\; r_{nk} \;=\; \begin{cases} 1 & \text{if } k = \arg\min_j \|x_n - \mu_j\| \\ 0 & \text{otherwise} \end{cases}$

Relationship to K-Means

Recall: $\mathbb{E}\Big(\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) \Big| \boldsymbol{X}, \theta^{\text{old}} \Big) = \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k|\boldsymbol{X}, \theta^{\text{old}}) \log \left(\pi_k N(x_n|\mu_k, \Sigma_k) \right).$

As $\epsilon \to 0$, we have

$$p(z_n = k|\boldsymbol{X}, \theta) \to r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_n - \mu_j\| \\ 0 & \text{otherwise} \end{cases}$$

which gives

$$\mathbb{E}\Big(\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta) \Big| \boldsymbol{X}, \theta^{\text{old}} \Big) \to -\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2 + \text{const.}$$

- In the limit, maximizing the expected log-likelihood is equivalent to minimizing the distortion measure in the K-means algorithm.
- The EM-algorithm is slower but more flexible and accurate.

Summary

- EM algorithm is a classical method in statistics.
- It can be used in the presence of latent variables.
- When applied to Gaussian mixtures, compared to k-means, it captures the covariance structure of the data.