Exercises
Machine Learning
HS 2016

Machine Learning Laboratory
Dept. of Computer Science, ETH Zürich
Prof. Joachim M. Buhmann
Web https://ml2.inf.ethz.ch/courses/ml/

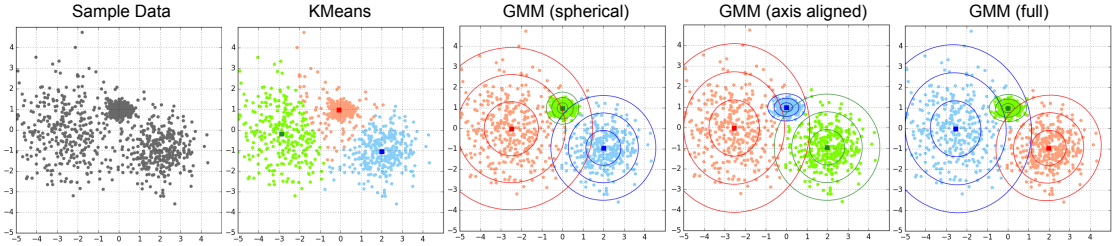
Series 5, Dec 2nd, 2016 (Clustering and Mixture Models)

For questions 1,2,3 : **Andreas Georgiadis**

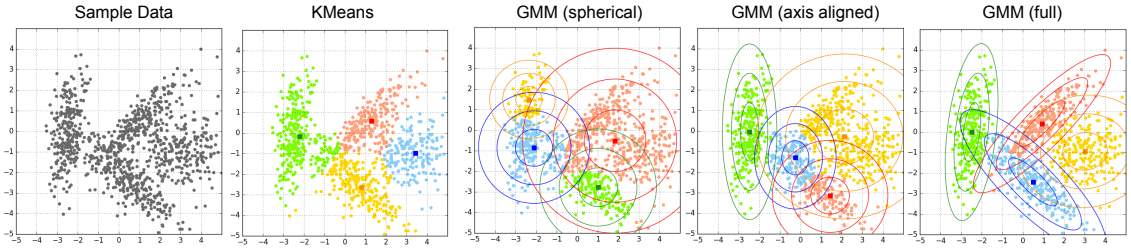
For questions :

Problem 1 (KMeans and GMMs):

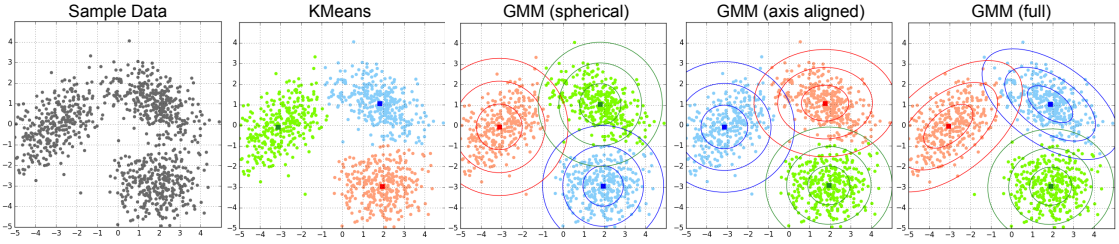
1. (a) In the first data distribution, we observe that the data clusters are all characterized by a spherical but non-uniform spread. As a result, a GMM with a spherical covariance matrix $\Sigma_k = \sigma_k^2 I$ would be the best choice in this case. As we observe in the following simulation, KMeans wrongly assignes some data points at the edges of the clusters, since it assumes spherical and uniform spread for all clusters. Mixture Models with a higher model complexity are not significantly different compared to the solution with spherical covariance.



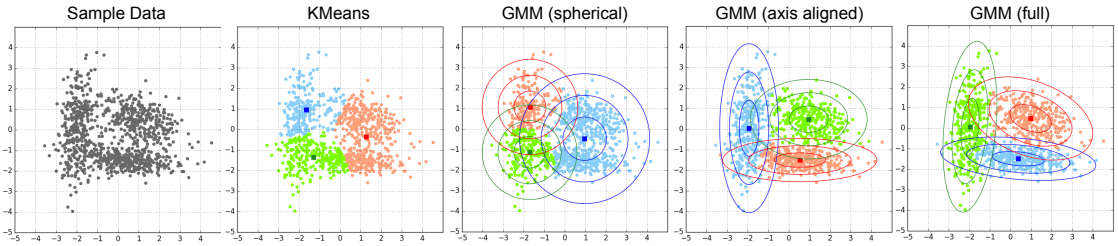
(b) In the second data distribution, we observe that the shapes of the clusters do not follow some general structure. As a result, a GMM with complete covariance matrices has to be trained in order to successfully identify the structure of the data. As we observe in the following simulation, all other alternatives fail because the assumptions they make do not hold for this case of data distribution.



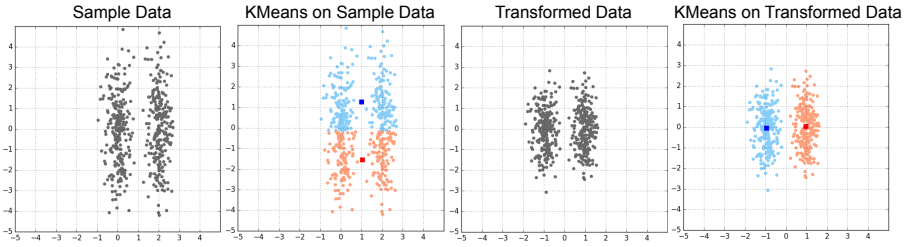
(c) In the third data distribution, the clusters are better separated and exploit an approximately spherical and uniform shape and as a consequence KMeans is the appropriate alternative in this case. In the following simulation, we observe that Gaussian Mixture Models with a greater model complexity are able to discover more accurately the structure of the data, but for the clustering task it does not change much because the data points are assigned to the same cluster anyway.



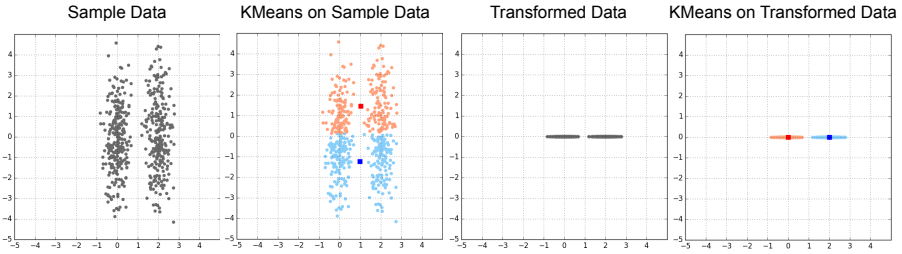
(d) In the fourth data distribution, we observe that the clusters have neither spherical nor uniform shape. Their shape however approximately resembles an ellipse, whose axes are aligned with the axis of the coordinate system. As a result, a GMM with diagonal covariance matrices $\Sigma_k = diag(\sigma^2_{k(1)}, \sigma^2_{k(2)})$ is appropriate in this case, since this type of GMM assumes that the cluster's shapes are ellipses, whose axes are approximately parallel to the coordinate axes. In the following simulation we observe that training a mixture model with full covariance matrices does not show a big difference.



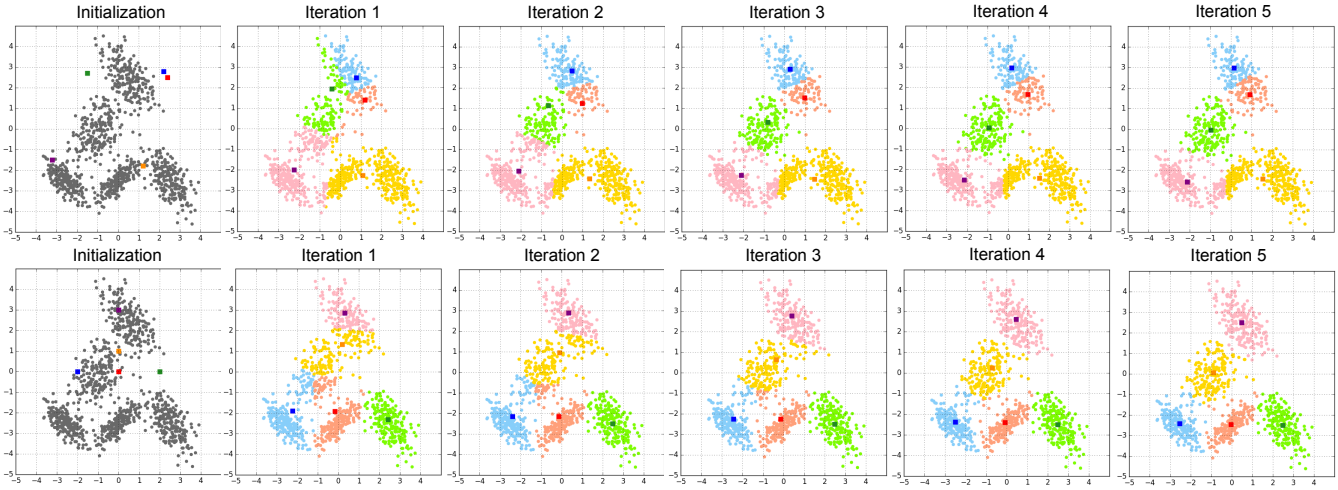
2. The reason why KMeans fails to recover the structure of the data is the significantly higher variance of the data in the y-axis compared to the x-axis. In order to tackle this problem, we propose two solutions. The former covers the normalization of the data to zero mean and unit variance. This way, the variance along the y-axis is decreased and KMeans can partition the transformed data successfully. The following plot illustrates that process.



The latter solution regards to projecting the data on the x-axis. Since we observe that what actually differentiates the data belonging to one cluster or the other is actually their x-value, projecting them on the x-axis will preserve that information while the variance on the y-axis will vanish and KMeans will be able to partition the data successfully. The following plot illustrates that process.



3. In following plot we have illustrated the evolution of the KMeans algorithm for two different initializations of the cluster centroids. It is evident that in those two cases, the EM algorithm is going to converge to a different solution. As a result, in order to obtain a good result, it is essential to run the KMeans algorithm with different initial values for the cluster centroids and keep the solution with the lowest value of the objective function.



Problem 2 (Mixture of Poisson Distributions):

1. $L(\boldsymbol{\pi}, \boldsymbol{\lambda}; \mathcal{D}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k)$ and $l(\boldsymbol{\pi}, \boldsymbol{\lambda}; \mathcal{D}) = \sum_{n=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k) \right]$

2. The latent random variable z_{kn} is a binary random variable $z_{kn} \in \{0, 1\}$, with $\sum_{k=1}^{K} z_{kn} = 1$ and indicates whether a data point m_n was generated by the mixture component k. The marginal distrubiton of the random variables in index and vector notation is respectively:

   $P(z_{kn} = 1) = \pi_k$ and $P(Z_n; \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_{kn}}$

3. If we apply the Bayes Rule, similarly as in the tutorial slides, we obtain:

   $$P(z_{kn} = 1 | m_n) = \frac{P(m_n | z_{kn} = 1) P(z_{kn} = 1)}{\sum_{l=1}^{K} P(m_n | z_{ln} = 1) P(z_{ln} = 1)} = \frac{\pi_k \mathcal{P}(m_n; \lambda_k)}{\sum_{l=1}^{K} \pi_l \mathcal{P}(m_n; \lambda_l)} = \gamma_{kn}$$

4. $l(\boldsymbol{\pi}, \boldsymbol{\lambda}; \mathcal{D}) = \sum_{n=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k) \right] =$

   $$\sum_{n=1}^{N} \log \left[\sum_{k=1}^{K} \frac{q_{kn} \pi_k \mathcal{P}(m_n; \lambda_k)}{q_{kn}} \right] \geq \text{(where } \sum_{k=1}^{K} q_{kn} = 1)$$
   $$\sum_{n=1}^{N} \sum_{k=1}^{K} q_{kn} \left[\log \mathcal{P}(m_n; \lambda_k) + \log \pi_k - \log q_{kn} \right]$$

5. Please note that λ refers to the Lagrange multiplier and λ_k refers to the arrival rates of the individual Poisson distributions. The Lagrangian is decoupled for each data point and has the form:

   $\mathcal{L}(\boldsymbol{q_n}, \lambda) = \sum_{k=1}^{K} q_{kn} \left[\log \mathcal{P}(m_n; \lambda_k) + \log \pi_k - \log q_{kn} \right] + \lambda(\sum_{k=1}^{K} q_{kn} - 1)$

   $\nabla_{q_{kn}} \mathcal{L} = \log \mathcal{P}(m_n; \lambda_k) + \log \pi_k - \log q_{kn} - 1 + \lambda$

   $\nabla_{q_{kn}} \mathcal{L} \stackrel{!}{=} 0 \Rightarrow \log \mathcal{P}(m_n; \lambda_k) + \log \pi_k - \log q_{kn}^* - 1 + \lambda = 0$

   We have to perform the following steps in order to calculate the Lagrange multiplier:

   $\log \left[\pi_k p \mathcal{P}(m_n; \lambda_k) \right] - 1 + \lambda = \log q_{kn}^* \Rightarrow \pi_k \mathcal{P}(m_n; \lambda_k) \exp(-1 + \lambda) = q_{kn}^* \Rightarrow$

$\sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k) \exp(-1+\lambda) = \sum_{k=1}^{K} q_{kn}^* \Rightarrow \exp(-1+\lambda) \sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k) = 1$

$\Rightarrow -1 + \lambda = -\log\big[\sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k)\big] \Rightarrow \lambda = 1 - \log\big[\sum_{k=1}^{K} \pi_k \mathcal{P}(m_n; \lambda_k)\big]$

We plug the expression of λ in the Lagrangian and obtain the optimal value q_{kn}^* which is equal to the posterior probability $P(z_{kn} = 1|m_n) = \gamma_{kn}$:

$$q_{kn}^* = \frac{\pi_k \mathcal{P}(m_n; \lambda_k)}{\sum_{l=1}^{L} \pi_l \mathcal{P}(m_n; \lambda_l)} = P(z_{kn} = 1|m_n) = \gamma_{kn}$$

6. If we replace $q_{kn}^* = \gamma_{kn}$ in the expression of the lower bound of the log-likelihood and taking into account that the inequality becomes tight (you can verify this with some straightforward computations) we get:

   $l(\boldsymbol{\pi}, \boldsymbol{\lambda}; \mathcal{D}) = \sum_{n=1}^{N} \gamma_{kn}\big[\log \mathcal{P}(m_n; \lambda_k) + \log \pi_k\big] - \gamma_{kn} \log \gamma_{kn}$

   We also have that:

   $$\log \mathcal{P}(m_n; \lambda_k) = \log\left[\frac{\lambda_k^m \exp(-\lambda_k)}{m_n!}\right] = m_n \log \lambda_k - \lambda_k - \log m_n!$$

   Subsequently, we calculate the gradient with respect to λ_k and set it to zero:

   $$\nabla_{\lambda_k} l(\boldsymbol{\pi}, \boldsymbol{\lambda}; \mathcal{D}) = \sum_{n=1}^{N} \gamma_{kn}\left[\frac{m_n}{\lambda_k} - 1\right]$$

   $$\nabla_{\lambda_k} \stackrel{!}{=} 0 \Rightarrow \sum_{n=1}^{N} \frac{\gamma_{nk} m_n}{\lambda_k^*} = \sum_{n=1}^{N} \gamma_{nk} \Rightarrow \lambda_k^* = \frac{\sum_{n=1}^{N} \gamma_{nk} m_n}{\sum_{n=1}^{N} \gamma_{nk}}$$

   In order to derive the expression for π_k^* we initially formulate the Lagrangian, which also takes the constraint with respect to π_ks and proceed as presented in the lecture slides, for the case of GMMs:

   $\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{kn}\big[\log \mathcal{P}(\mu_n; \lambda_k) + \log \pi_k\big] - \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{kn} + \lambda(\sum_{k=1}^{K} \pi_k - 1)$

   $$\nabla_{\pi_k} \mathcal{L} = \sum_{n=1}^{N} \gamma_{kn} \frac{1}{\pi_k} + \lambda \Rightarrow \nabla_{\pi_k} \mathcal{L} \stackrel{!}{=} 0 \Rightarrow \sum_{n=1}^{N} \gamma_{kn} = -\lambda \pi_k^*$$

   In order to calculate the Lagrange multiplier, we some both sides of the equation over k and plug this expression in the previous equation, in order to obtain the expression for π_k^*

   $$\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{kn} = -\lambda \sum_{k=1}^{K} \pi_k^* \Rightarrow \lambda = -N \Rightarrow \pi_k^* = \frac{1}{N} \sum_{n=1}^{N} \gamma_{kn}$$

Problem 3 (Singularities in GMMs):

1. $l(\theta; \mathcal{D}) = \sum_{n=1}^{N} \log\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)\right\}$
2. $l(\theta; x_n) = \log\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)\right\}$
3. $p(x_n; \mu_j, \Sigma_j) = \mathcal{N}(x_n; \mu_j, \Sigma_j) = \mathcal{N}(x_n; \mu_j, \Sigma_j) = \mathcal{N}(x_n; x_n, \sigma_j^2 I) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\sigma_j^D}$
4. We see that $l(\theta; \mathcal{D}) \stackrel{\sigma_j \to 0}{\to} \infty$. As a result the maximization of the log-likelihood is not a well-posed problem. This situation is an example of the severe overfitting that can occur while employing the maximum likelihood

approach and consequently leads to a very poor clustering. Such singularities will always be present whenever a Gaussian component collapses onto a specific data point.

5. In this case, the likelihood function takes the following form:

$$L(\theta;\mathcal{D}) = \prod_{n=1}^{N} \mathcal{N}(x_n;\mu,\sigma^2) = \prod_{n=1}^{N} \frac{1}{(2\pi)^{D/2}} \frac{1}{\sigma^D} \exp\left\{\frac{-1}{2\sigma^2}(x_n-\mu)^T(x_n-\mu)\right\}$$

Assuming that $x_{n'} = \mu$ the above expression becomes:

$$L(\theta;\mathcal{D}) = \frac{1}{(2\pi)^{ND/2}} \underbrace{\frac{1}{\sigma^{ND}}}_{\overset{\sigma\to 0}{\to}\infty} \prod_{n\neq n'}^{N} \underbrace{\exp\left\{\frac{-1}{2\sigma^2}(x_n-\mu)^T(x_n-\mu)\right\}}_{\overset{\sigma\to 0}{\to} 0 \text{ exponentially fast}} \overset{\sigma\to 0}{\to} 0$$

We note that in this case we should also account for the contribution of the multiplicative factors arising from the other data points, which go to zero exponentially fast and as a result, in this case the likelihood goes to zero, rather than infinity.

However, once we have (at least) two components in the mixture, one of the components can have a finite variance and therefore assign finite probability to all of the data points while the other component can shrink onto a specific data point and thereby contribute an ever increasing additive value to the log-likelihood.

6. A simple heuristic that can help us avoid the optimization being trapped in situations like this, is to detect whether a mixture component is collapsing to a specific data point. Once we detect such a situation, we can reset its mean to a randomly chosen value and its covariance matrix with some also random, large values and then continue with the optimization.