Probabilistic Graphical Models - Tutorial 8

Fernando García Sanz

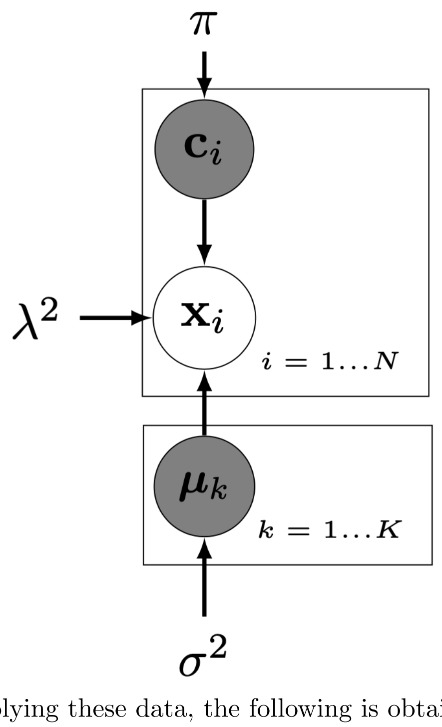
February 25, 2020

Assignment 1

Compute ELBO in closed form (by plugging in all distributions and taking expectations under the approximate distribution). This result will be later used to track the convergence of the model and compare different runs of optimization.

$$\mathcal{L}(\mathbf{x}|\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi}) = \sum_{k=1}^{K} E_q[\log p(\boldsymbol{\mu}_k)] + \sum_{i=1}^{N} E_q[\log p(c_i)] + \sum_{i=1}^{N} E_q[\log p(\mathbf{x}_i|c_i, \boldsymbol{\mu})] \\ - \sum_{k=1}^{K} E_q[\log q(\boldsymbol{\mu}_k)] - \sum_{i=1}^{N} E_q[\log q(c_i)]$$

We are given the following information:



$\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\alpha}, \sigma^2\mathbf{I}) \qquad k = 1, ..., K$

$c_i \sim Categorical\left(\frac{1}{K}, ..., \frac{1}{K}\right) \qquad i = 1, ..., N$

$\mathbf{x}_i|c_i, \boldsymbol{\mu} \sim \mathcal{N}(c_i^T\boldsymbol{\mu}, \lambda^2\mathbf{I}) \qquad i = 1, ..., N$

$q(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, s_k^2\mathbf{I})$

$q(c_i) \sim Categorical(\boldsymbol{\phi}_i)$

c_i is an indicator vector.

Data are p-dimensional.

Applying these data, the following is obtained:

$$\mathcal{L}(\mathbf{x}|\mathbf{m}, \mathbf{s}^2, \boldsymbol{\phi}) = \sum_{k=1}^{K} E_q[\log \mathcal{N}(\boldsymbol{\alpha}, \sigma^2\mathbf{I})] + \sum_{i=1}^{N} E_q\left[\log Categorical\left(\frac{1}{K}, ..., \frac{1}{K}\right)\right] \\ + \sum_{i=1}^{N} E_q[\log \mathcal{N}(c_i^T\boldsymbol{\mu}, \lambda^2\mathbf{I})] - \sum_{k=1}^{K} E_q[\log \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, s_k^2\mathbf{I})] - \sum_{i=1}^{N} E_q[\log Categorical(\boldsymbol{\phi}_i)]$$

From here, we will focus on each one of the terms individually:

- First term:

$$\sum_{k=1}^{K} E_q[\log \mathcal{N}(\boldsymbol{\alpha}, \sigma^2 \mathbf{I})] \rightarrow \text{Substituting by the p.d.f.}$$
$$\rightarrow \sum_{k=1}^{K} E_q \left[\log \left((2\pi)^{-\frac{p}{2}} |\sigma^2 \mathbf{I}|^{-\frac{1}{2}} \exp \left[-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\alpha})^T (\sigma^2 \mathbf{I})^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\alpha})\right]\right)\right]$$
$$= \sum_{k=1}^{K} E_q \left[\log((2\pi)^{-\frac{p}{2}} |\sigma^2 \mathbf{I}|^{-\frac{1}{2}}) + \left(-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\alpha})^T (\sigma^2 \mathbf{I})^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\alpha})\right)\right]$$
$$= \sum_{k=1}^{K} \log((2\pi)^{-\frac{p}{2}} \sigma^{-p}) - \frac{1}{2}(\sigma^2)^{-1} E_q[(\boldsymbol{\mu}_k - \boldsymbol{\alpha})^T (\boldsymbol{\mu}_k - \boldsymbol{\alpha})]$$

From here, analyzing the expectations:

$$\begin{aligned} E_q[(\boldsymbol{\mu}_k - \boldsymbol{\alpha})^T (\boldsymbol{\mu}_k - \boldsymbol{\alpha})] &= E_q[\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k] - E_q[\boldsymbol{\mu}_k^T]\boldsymbol{\alpha} - \boldsymbol{\alpha}^T E_q[\boldsymbol{\mu}_k] + \boldsymbol{\alpha}^T \boldsymbol{\alpha} \\ &= (*)\ s_k^2 p + \mathbf{m}_k^T \mathbf{m}_k - \mathbf{m}_k^T \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{m}_k + \boldsymbol{\alpha}^T \boldsymbol{\alpha} \end{aligned}$$

$(*)$ Sum of second moments (inner product per dimension).

So the final expression is equal to:

$$-\frac{1}{2}\left[Kp \log((2\pi\sigma^2)) - \sum_{k=1}^{K} \frac{1}{\sigma^2}(s_k^2 p + \mathbf{m}_k^T \mathbf{m}_k - \mathbf{m}_k^T \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{m}_k + \boldsymbol{\alpha}^T \boldsymbol{\alpha})\right]$$

- Second term:

$$\sum_{i=1}^{N} E_q \left[\log Cat. \left(\frac{1}{K}, ..., \frac{1}{K}\right)\right] \implies \log Categorical \left(\frac{1}{K}, ..., \frac{1}{K}\right) = \log \prod_{k=1}^{K} \left(\frac{1}{K}\right)^{(c_i)_k}$$
$$= \sum_{i=1}^{N} E_q \left[\sum_{k=1}^{K} (c_i)_k (-\log(K))\right] \implies \sum_{i=1}^{N} \underbrace{\sum_{k=1}^{K} (\phi_i)_k}_{1} (-\log(K))$$
$$= \sum_{i=1}^{N} (-\log(K)) = N(-\log(K))$$

- Third term:

$$\begin{aligned} \sum_{i=1}^{N} E_q[\log p(\mathbf{x}_i | c_i, \boldsymbol{\mu})] &= \sum_{i=1}^{N} E_q \left[\log \prod_{k=1}^{K} p(\mathbf{x}_i | \boldsymbol{\mu}_k)^{c_{i,k}}\right] = \sum_{i=1}^{N} E_q \left[\sum_{k=1}^{K} c_{i,k} \log p(\mathbf{x}_i | \boldsymbol{\mu}_k)\right] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \boldsymbol{\phi}_{i,k} E_q[\log p(\mathbf{x}_i | \boldsymbol{\mu}_k)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \boldsymbol{\phi}_{i,k} E_q[\log \mathcal{N}(\boldsymbol{\mu}_k | \lambda^2 \mathbf{I})] \\ &\implies E_q \left[(2\pi)^{-\frac{p}{2}} |\lambda^2 \mathbf{I}|^{-\frac{1}{2}}\right] - \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\lambda^2 \mathbf{I})^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \\ &= -\frac{p}{2} \log(2\pi) - p \log(\lambda) - \frac{1}{2\lambda^2} E_q[(\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{x}_i - \boldsymbol{\mu}_k)] \end{aligned}$$

Analyzing the expectations:

$$\begin{aligned} E_q[(\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{x}_i - \boldsymbol{\mu}_k)] &= E_q[\mathbf{x}_i^T \mathbf{x}_i - \mathbf{x}_i \boldsymbol{\mu}_k - \boldsymbol{\mu}_k^T \mathbf{x}_i + \boldsymbol{\mu}_k^T \boldsymbol{\mu}_k] \\ &= E_q[\mathbf{x}_i^T \mathbf{x}_i] - E_q[\mathbf{x}_i \boldsymbol{\mu}_k] - E_q[\boldsymbol{\mu}_k^T \mathbf{x}_i] + E_q[\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k] \\ &= \mathbf{x}_i^T \mathbf{x}_i - \mathbf{x}_i^T \mathbf{m}_k - \mathbf{m}_k^T \mathbf{x}_i + s_k^2 p + \mathbf{m}_k^T \mathbf{m}_k \end{aligned}$$

Substituting this in the previous expression, the following is obtained:

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\boldsymbol{\phi}_{i,k}\left(-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]\right)$$

• Fourth term:

$$\sum_{k=1}^{K}E_q[\log\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k,s_k^2\mathbf{I})] \implies E_q\left[\log[(2\pi)^{-\frac{1}{2}}|s_k^2\mathbf{I}|^{-\frac{1}{2}}]-\frac{1}{2}(\boldsymbol{\mu}_k-\mathbf{m}_k)^T(s_k^2\mathbf{I})^{-1}(\boldsymbol{\mu}_k-\mathbf{m}_k)\right]$$

$$=-\frac{p}{2}\log(2\pi)-p\log(s_k)-\frac{1}{2s_k^2}E_q[(\boldsymbol{\mu}_k-\mathbf{m}_k)^T(\boldsymbol{\mu}_k-\mathbf{m}_k)]$$

Analyzing the expectation:

$$E_q[\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k]-E_q[\boldsymbol{\mu}_k^T\mathbf{m}_k]-E_q[\boldsymbol{\mu}_k^T\mathbf{m}_k]+E_q[\mathbf{m}_k^T\mathbf{m}_k]$$
$$=s_k^2p+\mathbf{m}_k^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{m}_k+\mathbf{m}_k^T\mathbf{m}_k=s_k^2p$$

Substituting in the previous expression:

$$-\frac{p}{2}\log(2\pi)-p\log(s_k)-\frac{p}{2}$$

Finally, the following is obtained:

$$\sum_{k=1}^{K}E_q[\log\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k,s_k^2\mathbf{I})]=\sum_{k=1}^{K}-\frac{p}{2}\left[\log(2\pi)+2\log(s_k)+1\right]$$

• Fifth term:

$$\sum_{i=1}^{K}E_q[\log Cat.(\boldsymbol{\phi}_i)] \implies \log Categorical(\phi_i)=\sum_{i=1}^{N}E_q\left[\log\prod_{k=1}^{K}\phi_{i,k}^{c_{i,k}}\right]$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}E_q[c_{i,k}\log\phi_{i,k}]=\sum_{i=1}^{N}\sum_{k=1}^{K}\log(\phi_{i,k})\phi_{i,k}$$

So, substituting each one of the expressions obtained in the original expression $\mathcal{L}(\mathbf{x}|\mathbf{m},\mathbf{s}^2,\boldsymbol{\phi})$ returns the final result of the derivation.

Assignment 2

Show that the variational update for i-th cluster assignment is:

$$\phi_{i,k}\propto\exp\left\{\frac{\mathbf{x}_i^TE[\boldsymbol{\mu}_k]}{\lambda^2}-\frac{E[\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k]}{2\lambda^2}\right\}$$

It is known that $p(x)=KL+ELBO$. Maximizing $ELBO$ will imply that we are minimizing KL, and this fact will provide a better approximation since we are reducing the distance calculated in KL.

As we want to explore over $\phi_{i,k}$, we can perform the derivative over it in the expression $\mathcal{L}(\mathbf{x}|\mathbf{m},\mathbf{s}^2,\boldsymbol{\phi})$. This will return the following result:

• First term:

$$-\frac{1}{2}\left[Kp\log((2\pi\sigma^2))-\sum_{k=1}^{K}\frac{1}{\sigma^2}(s_k^2p+\mathbf{m}_k^T\mathbf{m}_k-\mathbf{m}_k^T\boldsymbol{\alpha}-\boldsymbol{\alpha}^T\mathbf{m}_k+\boldsymbol{\alpha}^T\boldsymbol{\alpha})\right]$$

This term is constant over $\phi_{i,k}$, so the derivative is zero.

- Second term:

$$N(-\log(K))$$

This term is constant over $\phi_{i,k}$, so the derivative is zero.

- Third term:

$$\frac{\partial\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\boldsymbol{\phi}_{i,k}\left(-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]\right)\right]}{\partial\phi_{i,k}}$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}\underbrace{\boldsymbol{\phi}_{i,k}}_{1}\underbrace{\left(-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]\right)}_{\text{Constant over }\phi_{i,k}}$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]$$

- Fourth term:

$$\sum_{k=1}^{K}-\frac{p}{2}\left[\log(2\pi)+2\log(s_k)+1\right]$$

This term is constant over $\phi_{i,k}$, so the derivative is zero.

- Fifth term:

$$\frac{\partial\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log(\phi_{i,k})\phi_{i,k}\right]}{\partial\phi_{i,k}}=\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{1}{\phi_{i,k}}\phi_{i,k}+\log\phi_{i,k}=\sum_{i=1}^{N}\sum_{k=1}^{K}1+\log\phi_{i,k}$$

From here, we know that the final expression is:

$$\sum_{i=1}^{N}\sum_{k=1}^{K}-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]-(1+\log\phi_{i,k})=0$$

We can work directly with the inner terms of the summations, since we want to prove this expression for $\phi_{i,k}$. Therefore:

$$\log\phi_{i,k}=-\frac{p}{2}\log(2\pi\lambda^2)-\frac{1}{2\lambda^2}[\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+s_k^2p+\mathbf{m}_k^T\mathbf{m}_k]-1$$

Working with the terms inside the brackets, we have:

$$\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i+\underbrace{s_k^2p+\mathbf{m}_k^T\mathbf{m}_k}_{E_q[\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k]}$$

$$\implies\mathbf{x}_i^T\mathbf{x}_i-\mathbf{x}_i^T\mathbf{m}_k-\mathbf{m}_k^T\mathbf{x}_i=\mathbf{x}_i^T(\mathbf{x}_i-\mathbf{m}_k)-\underbrace{\mathbf{m}_k^T\mathbf{x}_i}_{\mathbf{x}_i^T\mathbf{m}_k}$$

$$=\mathbf{x}_i^T(\mathbf{x}_i-2\mathbf{m}_k)=\mathbf{x}_i^T\mathbf{x}_i-2\mathbf{x}_i^T\mathbf{m}_k$$

Now, substituting in the previous expression:

$$\log\phi_{i,k}=\underbrace{-\frac{p}{2}\log(2\pi\lambda^2)-1-\frac{\mathbf{x}_i\mathbf{x}_i^T}{2\lambda^2}}_{\text{Constant}}+\frac{2\mathbf{x}_i^T\mathbf{m}_k}{2\lambda^2}-\frac{s_k^2p+\mathbf{m}_k^T\mathbf{m}_k}{2\lambda^2}$$

$$=\text{constant}+\frac{\mathbf{x}_i^TE_q[\boldsymbol{\mu}_k]}{\lambda^2}-\frac{E_q[\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k]}{2\lambda^2}$$

Finally obtaining:

$$\phi_{i,k}\propto\exp\left[\frac{\mathbf{x}_i^TE_q[\boldsymbol{\mu}_k]}{\lambda^2}-\frac{E_q[\boldsymbol{\mu}_kT\boldsymbol{\mu}_k]}{2\lambda^2}\right]$$

Assignment 3

Complete the square to find the parameters of the optimal Gaussian $\boldsymbol{\mu}_k \sim \mathcal{N}(\mathbf{m}_k, s_k^2\mathbf{I})$. Those parameters will be used for variational updates of the posterior of the mixture component means.

It is known that:

$$
\begin{aligned}
q^*(\boldsymbol{\mu}_k) &\propto -\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2\sigma^2} + \frac{\boldsymbol{\mu}_k^T\alpha}{\sigma^2} + \sum_{i=1}^N \phi_{i,k}\left(-\frac{1}{2\lambda^2}\right)(\mathbf{x}_i^T - \boldsymbol{\mu}_k)^T(\mathbf{x}_i - \boldsymbol{\mu}_k) + const \\
&= -\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2\sigma^2} + \frac{\boldsymbol{\mu}_k^T\alpha}{\sigma^2} + \sum_{i=1}^N \phi_{i,k}\left(-\frac{1}{2\lambda^2}\right)(\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^T\mathbf{x}_i + \boldsymbol{\mu}_k^T\boldsymbol{\mu}_k) \\
&= -\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2\sigma^2} + \frac{\boldsymbol{\mu}_k^T\alpha}{\sigma^2} + \sum_{i=1}^N \phi_{i,k}\left(-\frac{1}{2\lambda^2}\right)(\mathbf{x}_i^T\mathbf{x}_i - 2\boldsymbol{\mu}_k^T\mathbf{x}_i + \boldsymbol{\mu}_k^T\boldsymbol{\mu}_k) \\
&= -\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2}\underbrace{\left(\frac{1}{\sigma^2} + \sum_{i=1}^N \phi_{i,k}\left(\frac{1}{\lambda^2}\right)\right)}_{\text{Term 1}} + \boldsymbol{\mu}_k^T\underbrace{\left(\frac{\alpha}{\sigma^2} + \frac{1}{\lambda^2}\sum_{i=1}^N \phi_{i,k}\mathbf{x}_i\right)}_{\text{Term 2}} + \underbrace{\sum_{i=1}^N -\phi_{i,k}\frac{\mathbf{x}_i^T\mathbf{x}_i}{2\lambda^2} + const}_{\text{Const. over } \boldsymbol{\mu}_k}
\end{aligned}
$$

Expanding now the expression $q^*(\boldsymbol{\mu}_k)$:

$$
\begin{aligned}
q^*(\boldsymbol{\mu}_k) = &\underbrace{\frac{1}{\sqrt{2\pi|s_k^2\mathbf{I}|}}}_{\text{Const. over } \boldsymbol{\mu}_k} \exp\left[-\frac{1}{2}(\boldsymbol{\mu}_k - \mathbf{m}_k)^T(s_k^2\mathbf{I})^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_k)\right] \\
\implies &\underbrace{\exp}_{Const.}\left[-\frac{1}{2s_k^2}(\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^T\mathbf{m}_k - \mathbf{m}_k^T\boldsymbol{\mu}_k + \mathbf{m}_k^T\mathbf{m}_k)\right] \\
\implies &-\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2s_k^2} - \frac{1}{2s_k^2}(-\boldsymbol{\mu}_k^T\mathbf{m}_k - \underbrace{\mathbf{m}_k^T\boldsymbol{\mu}_k}_{\boldsymbol{\mu}_k^T\mathbf{m}_k} + \mathbf{m}_k^T\mathbf{m}_k) \\
\implies &-\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2s_k^2} + \frac{\boldsymbol{\mu}_k^T}{2s_k^2}(2\mathbf{m}_k) - \frac{1}{2s_k^2}(\mathbf{m}_k^T\mathbf{m}_k) \\
\implies &-\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2s_k^2} + \boldsymbol{\mu}_k^T\frac{\mathbf{m}_k}{s_k^2} - \underbrace{\frac{1}{2s_k^2}(\mathbf{m}_k^T\mathbf{m}_k)}_{\text{Const.}}
\end{aligned}
$$

Therefore, using the previously obtained terms and this outcome:

$$
\begin{aligned}
-\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2s_k^2} &= -\frac{\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k}{2}\left(\frac{1}{\sigma^2} + \sum_{i=1}^N \phi_{i,k}\left(\frac{1}{\lambda^2}\right)\right) \implies s_k^2 = (\text{Term 1})^{-1} \\
\boldsymbol{\mu}_k^T\frac{\mathbf{m}_k}{s_k^2} &= \boldsymbol{\mu}_k^T\left(\frac{\alpha}{\sigma^2} + \frac{1}{\lambda^2}\sum_{i=1}^N \phi_{i,k}\mathbf{x}_i\right) \implies \mathbf{m}_k = \text{Term 2} \cdot s_k^2
\end{aligned}
$$

Assignment 4

Using results of the previous assignments implement missing parts of the algorithm in the provided python code.

The code can be found in the appendix.

The results obtained from the completed code are the following:

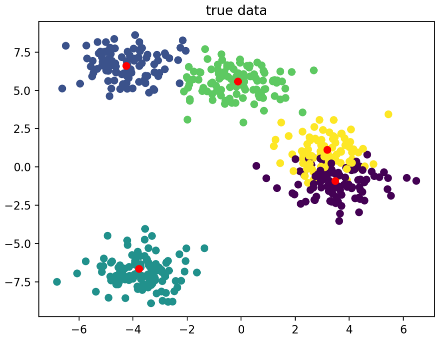


Figure 1: True data variational inference.

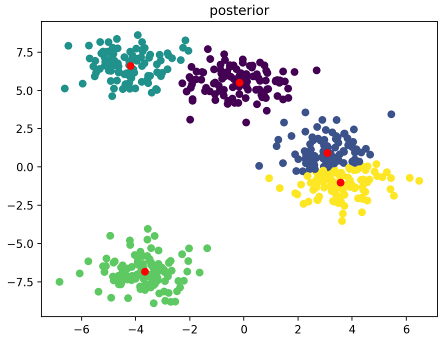


Figure 2: Posterior variational inference.

By means of these images can be observed that the predicted posterior is a really good approximation of each one of the clusters defined by the real data.

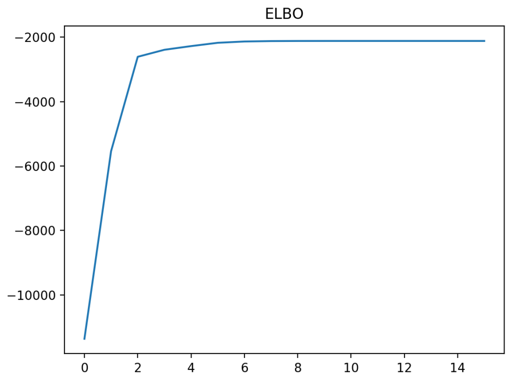


Figure 3: *ELBO* value.

As it can be seen, the *ELBO* value ends up converging around the value -2000.

Variational-Inference

```python
# author Olga Mikheeva olgamik@kth.se
# PGM tutorial on Variational Inference
# Bayesian Mixture of Gaussians

import numpy as np
import matplotlib.pyplot as plt
import math


def generate_data(std, k, n, dim=1):
    means = np.random.normal(0.0, std, size=(k, dim))
    data = []
    categories = []
    for i in range(n):
        cat = np.random.choice(k)  # sample component assignment
        categories.append(cat)
        data.append(np.random.multivariate_normal(means[cat, :], np.eye(dim)))  #
    sample data point from the Gaussian
    return np.stack(data), categories, means


def plot(x, y, c, means, title):
    plt.scatter(x, y, c=c)
    plt.scatter(means[:, 0], means[:, 1], c='r')
    plt.title(title)
    plt.show()


def plot_elbo(elbo):
    plt.plot(elbo)
    plt.title('ELBO')
    plt.show()


def compute_elbo(data, phi, m, s2, sigma2, mu0):
    """ Computes ELBO """
    n, p = data.shape
    k = m.shape[0]

    elbo = 0

    # TODO: compute ELBO
    # expected log prior over mixture assignments
    elbo += (-0.5) * (k * p * np.log(2 * np.pi * sigma2))
    aux = 0
    for i in range(k):
        aux += (s2[i] * p + m[i].T @ m[i] - m[i].T @ mu0 - mu0.T @ m[i] + mu0.T @
    mu0)
    aux *= 0.5 * (1 / sigma2)
    elbo += aux

    # expected log prior over mixture locations
    elbo += n * (-np.log(k))

    # expected log likelihood
    # lambda = 1 --> (Discussion)
    lmb = 1
    for i in range(n):
        for j in range(k):
            elbo += phi[i, j] * ((-p / 2) * np.log(2 * np.pi * pow(lmb, 2)) - (1 /
    (2 * pow(lmb, 2))) * (data[i].T @ data[i] -
            data[i].T @ m[j] - m[j].T @ data[i] + s2[j] * p + m[j].T @ m[j]))

    # entropy of variational location posterior
    aux = 0
    for i in range(n):
        for j in range(k):
            aux += np.log(phi[i, j]) * phi[i, j]
    elbo -= aux
```

```

    # entropy of the variational assignment posterior
    aux = 0
    for i in range(k):
        aux += np.log(2 * np.pi) + 2 * np.log(np.sqrt(s2[i])) + 1
    aux *= -p / 2
    elbo -= aux

    return elbo


def cavi(data, k, sigma2, m0, eps=1e-15):
    """ Coordinate ascent Variational Inference for Bayesian Mixture of Gaussians
    :param data: data
    :param k: number of components
    :param sigma2: prior variance
    :param m0: prior mean
    :param eps: stopping condition
    :return (m_k, s2_k, psi_i)
    """
    n, p = data.shape
    # initialize randomly
    m = np.random.normal(0., 1., size=(k, p))
    s2 = np.square(np.random.normal(0., 1., size=(k, 1)))
    phi = np.random.dirichlet(np.ones(k), size=n)
    lbm = 1

    # compute ELBO
    elbo = [compute_elbo(data, phi, m, s2, sigma2, m0)]
    convergence = 1.
    while convergence > eps:  # while ELBO not converged
        # TODO: update categorical
        for i in range(n):
            for j in range(k):
                phi[i, j] = np.exp((data[i].T @ m[j]) / pow(lbm, 2) - (s2[j] * p +
    m[j].T @ m[j]) / (2 * pow(lbm, 2)))
            phi[i] /= np.sum(phi[i])

        # TODO: update posterior parameters for the component means
        for j in range(k):
            aux = 0
            for i in range(n):
                aux += phi[i, j] * (1 / pow(lbm, 2))
            aux += 1 / sigma2
            s2[j] = 1 / aux

        for j in range(k):
            aux = 0
            for i in range(n):
                aux += phi[i, j] * data[i]
            aux *= 1 / pow(lbm, 2)
            aux += m0 / sigma2
            m[j] = aux * s2[j]

        # compute ELBO
        elbo.append(compute_elbo(data, phi, m, s2, sigma2, m0))
        convergence = elbo[-1] - elbo[-2]

    return m, s2, phi, elbo


def main():
    # parameters
    p = 2
    k = 5
    sigma = 5.

    data, categories, means = generate_data(std=sigma, k=k, n=500, dim=p)
    m = list()
    s2 = list()
```

```
    psi = list()
    elbo = list()
    best_i = 0
    for i in range(10):
        m_i, s2_i, psi_i, elbo_i = cavi(data, k=k, sigma2=sigma, m0=np.zeros(p))
        m.append(m_i)
        s2.append(s2_i)
        psi.append(psi_i)
        elbo.append(elbo_i)
        if i > 0 and elbo[-1][-1] > elbo[best_i][-1]:
            best_i = i
    class_pred = np.argmax(psi[best_i], axis=1)
    plot(data[:, 0], data[:, 1], categories, means, title='true data')
    plot(data[:, 0], data[:, 1], class_pred, m[best_i], title='posterior')
    plot_elbo(elbo[best_i])

if __name__ == '__main__':
    main()
```