Variational Inference

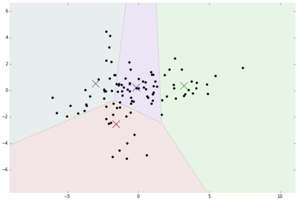
Hugh Salimbeni

29th February 2015

Recommended reading: Bishop PRML ch 9.2, 10.1, 10.2

Clustering: From K-means to Gaussian Mixtures

- Aim: find K clusters in the data
- Objective function:

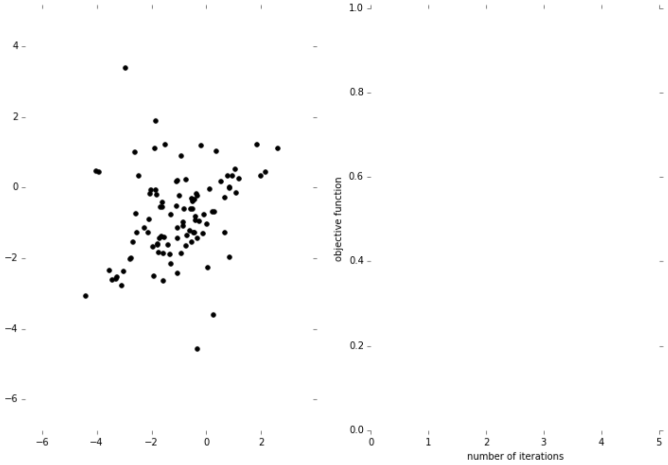


$$J = \sum_{n=1,j=1}^{N,K} z_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

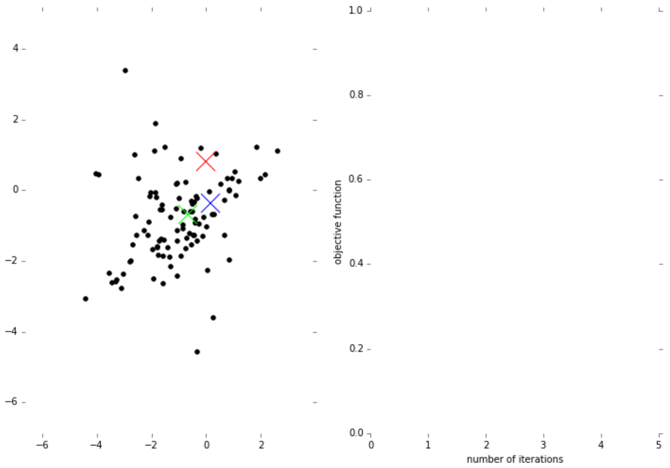
where $z_{nk} = 1$ if the nth point is in the kth cluster, 0 otherwise

- Difficult optimization problem ($N + KD$ parameters)
- Easy to find a local optimum:
  1. Fix cluster centers $\boldsymbol{\mu}_k$. Then the best option is to assign points to the closest center
  2. Fix assignments z_{nk}. The best choice for the centers is the mean of the points assigned to each cluster
  3. Repeat until converged

K-means demo

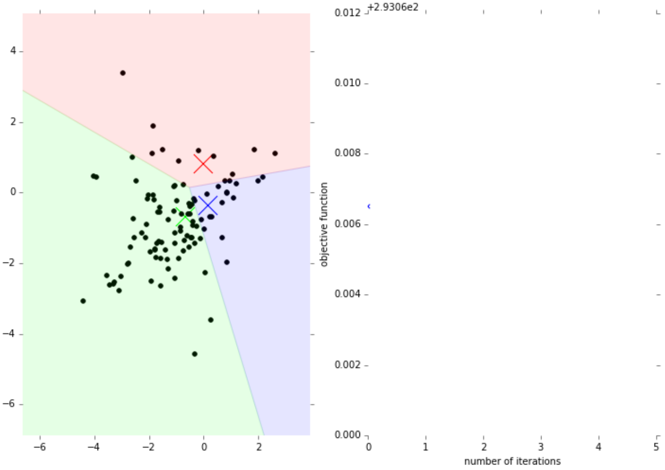


K-means demo



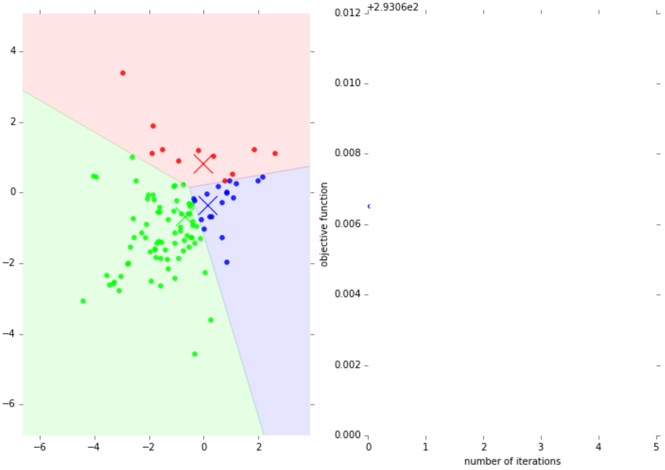
4
2
0
−2
−4
−6
−6
−4
−2
0
2
1.0
0.8
0.6
0.4
0.2
0.0
objective function
0
1
2
3
4
5
number of iterations

K-means demo



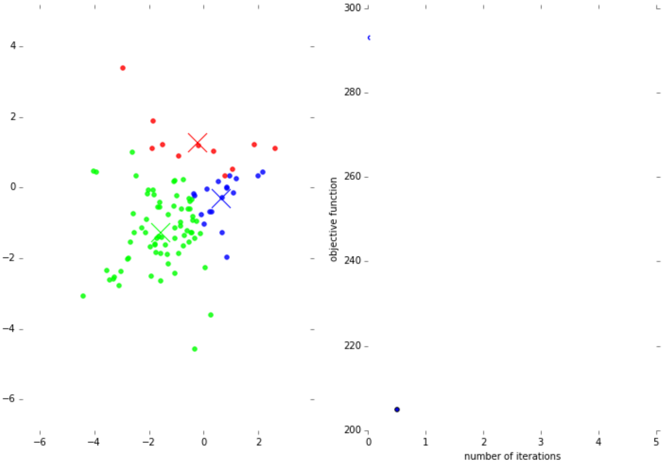
4
2
0
−2
−4
−6
−6 −4 −2 0 2
+2.9306e2
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0 1 2 3 4 5
objective function
number of iterations

K-means demo



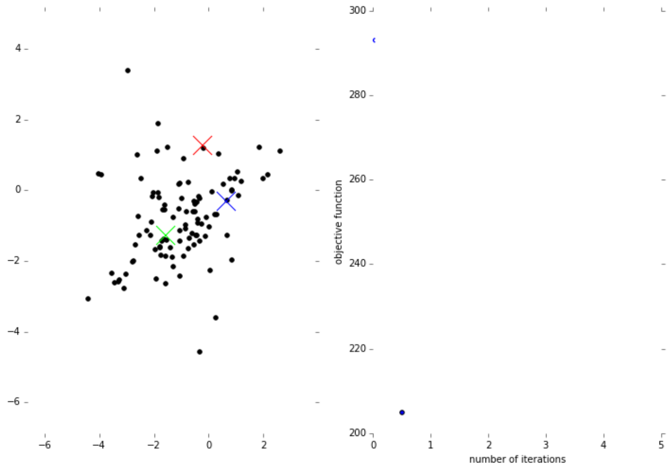
4
2
0
−2
−4
−6
−6
−4
−2
0
2
0.012
+2.9306e2
0.010
0.008
0.006
0.004
0.002
0.000
objective function
0
1
2
3
4
5
number of iterations

K-means demo



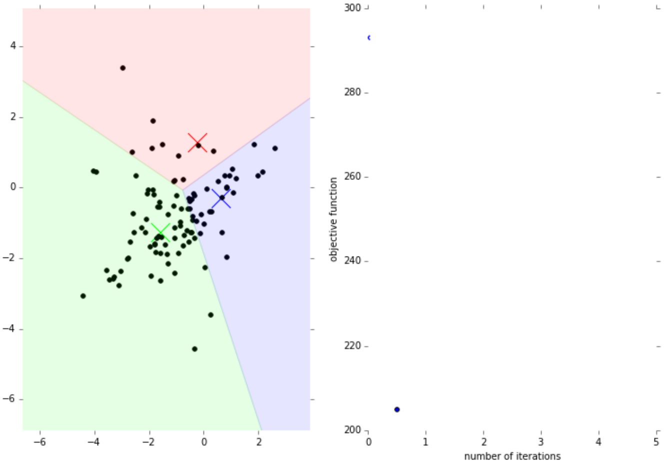
4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
objective function
0
1
2
3
4
5
number of iterations

K-means demo



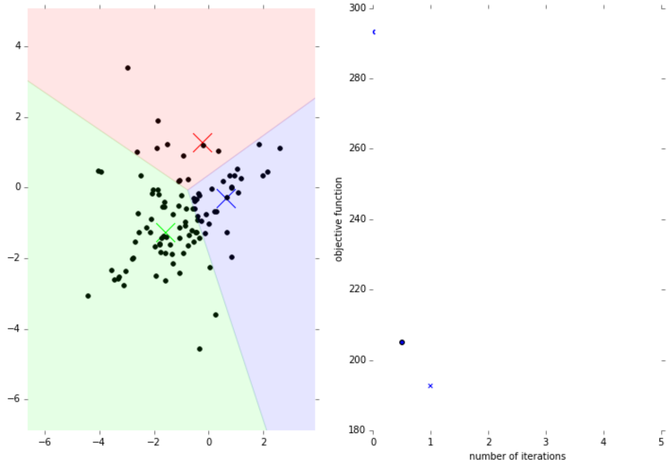
4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
objective function
0
1
2
3
4
5
number of iterations

K-means demo



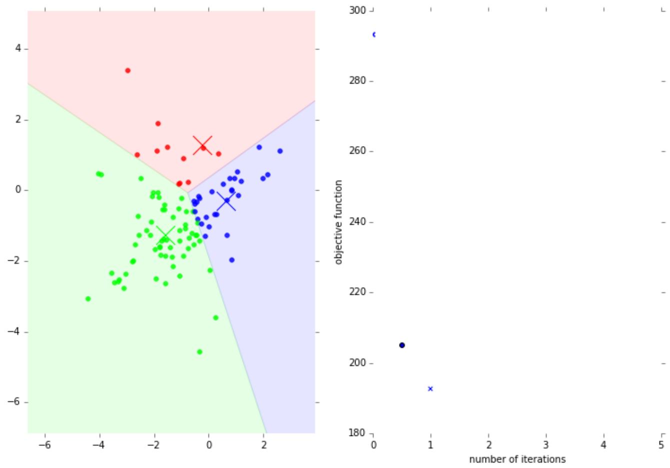
4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
0
1
2
3
4
5
objective function
number of iterations

K-means demo

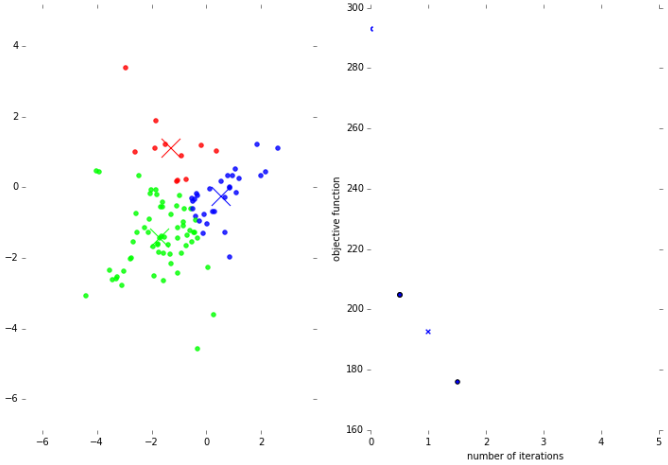


4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
180
objective function
0
1
2
3
4
5
number of iterations

K-means demo

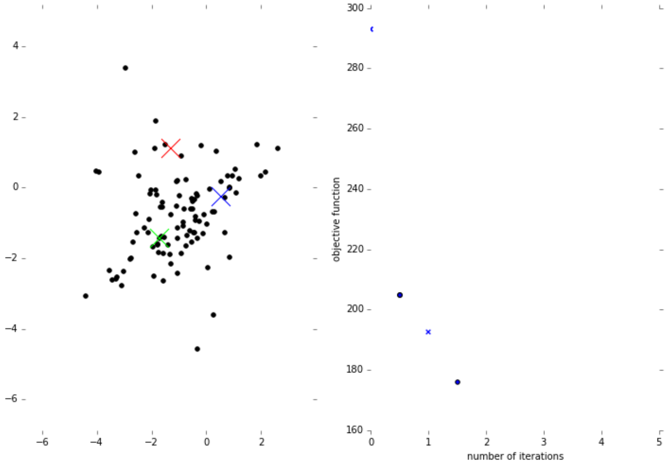


K-means demo

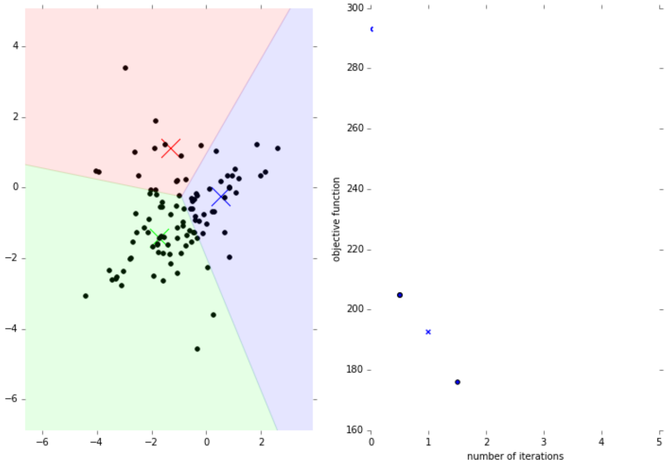


4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
180
160
objective function
0
1
2
3
4
5
number of iterations

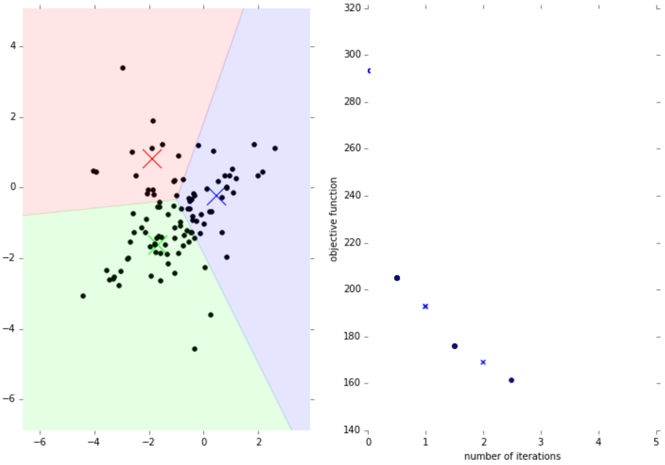
K-means demo



K-means demo

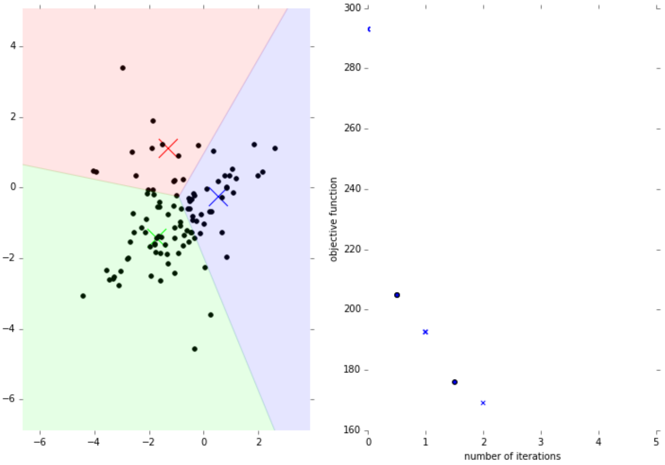


K-means demo



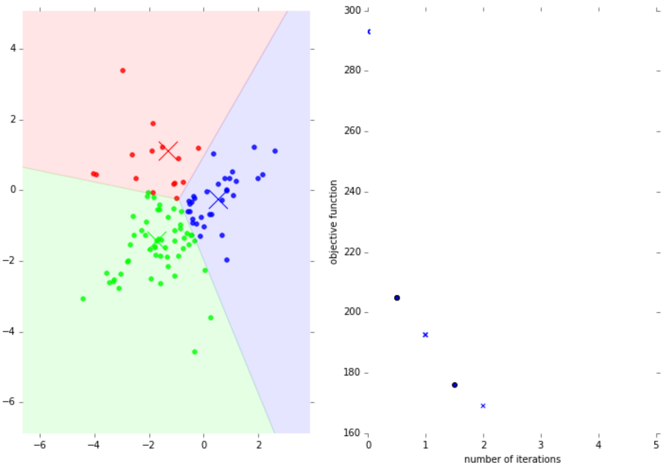
4
2
0
−2
−4
−6
−6
−4
−2
0
2
320
300
280
260
240
220
200
180
160
140
objective function
0
1
2
3
4
5
number of iterations

K-means demo



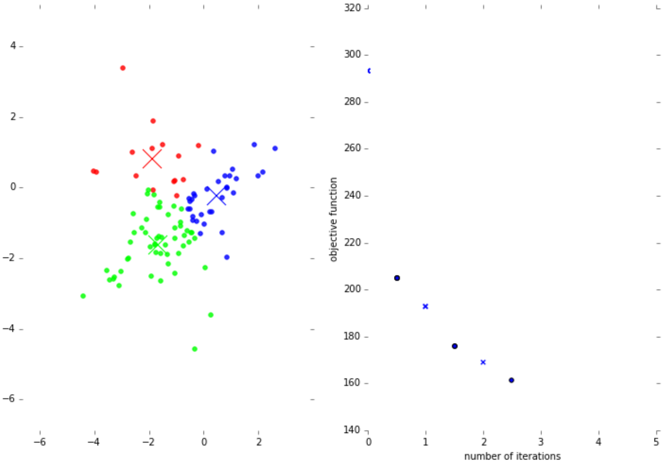
4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
180
160
0
1
2
3
4
5
objective function
number of iterations

K-means demo



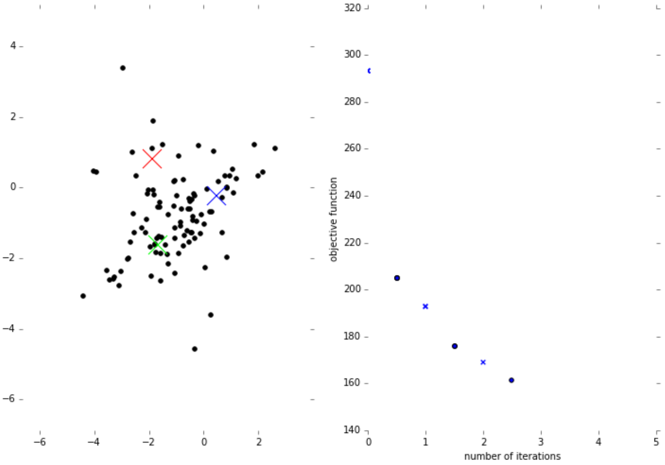
4
2
0
−2
−4
−6
−6
−4
−2
0
2
300
280
260
240
220
200
180
160
0
1
2
3
4
5
objective function
number of iterations

K-means demo



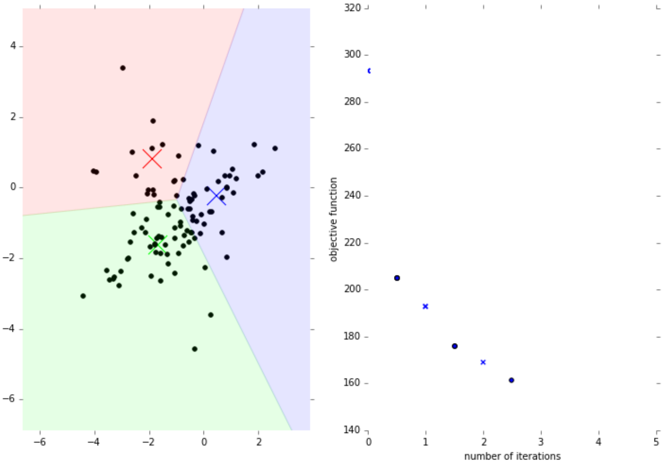
4
2
0
−2
−4
−6
−6
−4
−2
0
2
320
300
280
260
240
220
200
180
160
140
objective function
0
1
2
3
4
5
number of iterations

K-means demo



4
2
0
−2
−4
−6
−6
−4
−2
0
2
320
300
280
260
240
220
200
180
160
140
objective function
0
1
2
3
4
5
number of iterations

K-means demo



4
2
0
−2
−4
−6
−6
−4
−2
0
2
320
300
280
260
240
220
200
180
160
140
0
1
2
3
4
5
objective function
number of iterations

K-means advantages

- Fast to run
- Easy to code:

K-means advantages

- Fast to run
- Easy to code:

```
import numpy as np
from utils import squared_distances

def update_K_means_Z(X, mus):
    d2 = squared_distances(X, mus)
    return (abs((d2.T-np.min(d2, axis=1)).T)==0).astype(int)

def update_K_means_mus(X, Z):
    return np.einsum('nk,nd->kd', Z/(np.sum(Z, axis=0).astype(float)), X)

def K_means_objective(X, Z, mus):
    d2 = squared_distances(X, mus)
    return np.einsum('nk,nk',d2, Z)
```

K-means disadvantages

- Gives no indication of what the clusters are like
- Sensitive to initialization
- Can fail (potential division by zero)
- Can get stuck in poor a local optimum
- Not a generative model

Maximum likelihood (EM) Gaussian Mixture Model

- Generative model: i.e. we specify $p(\text{data}|\text{parameters})$
  - The distribution that generated the data is a weighted sum of K Gaussians
  - Each of the K Gaussians has its own mean and variance: $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
  - the *likelihood* for each data point is:

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

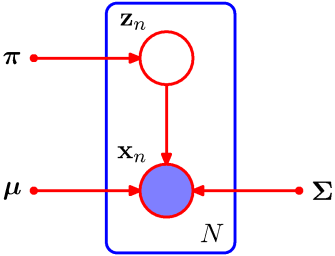
- To generate samples from this model (given the parameters) we could:
  1. Use some sampling method with the full probability distribution $\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
  2. Reformulate the model with an additional variable z determining the class

  Using a latent variable is much easier

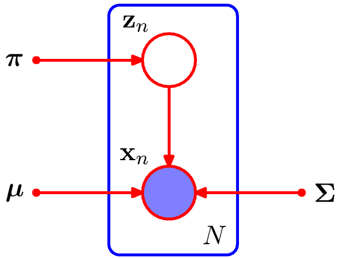
GMM with a latent variable

- $\mathbf{z}$ is a one-of-K variable, so $z_k = 1$ if the class is k, and 0 otherwise
- If $p(z_k = 1) = \pi_k$ then marginalisation of z returns the model

As a graphical model:



GMM with a latent variable



It is now easier to sample:

1. take a sample for $\mathbf{z}$ (using a uniform number generator)
2. take a sample for $p(\mathbf{x}|\mathbf{z})$. This is now a single Gaussian so use e.g. `numpy.random.multivariate_normal`

Example: $K = 3$, and $\boldsymbol{\pi} = (0.4, 0.5, 0.1)$
sample a uniform random variable. Say $u = 0.945$. This falls in class 3, so $\mathbf{z} = (0, 0, 1)$ Now generate sample from
$p(\mathbf{x}|z_3 = 1) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$

Fitting the GMM with EM

- As with K-means:
  - finding the expected values of the z_{nk} is possible, *given* all the parameters
  - if z_{nk} are fixed, it is possible to find the best $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

This results in an alternating algorithm similar to K-means, known as *Expectation Maximization*

Implementation (almost a repeat of a previous lecture)

1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
2. *E-step:* Evaluate responsibilities for every data point $\boldsymbol{x}_i$ using current parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

$$\mathbb{E}(z_{ik}) = r_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3. *M-step:* Re-estimate parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using the current responsibilities r_{ik} (from E-step):

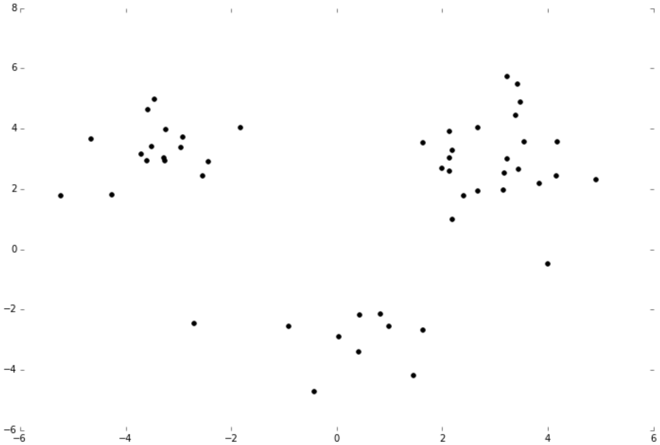
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^N r_{ik} \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^N r_{ik} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

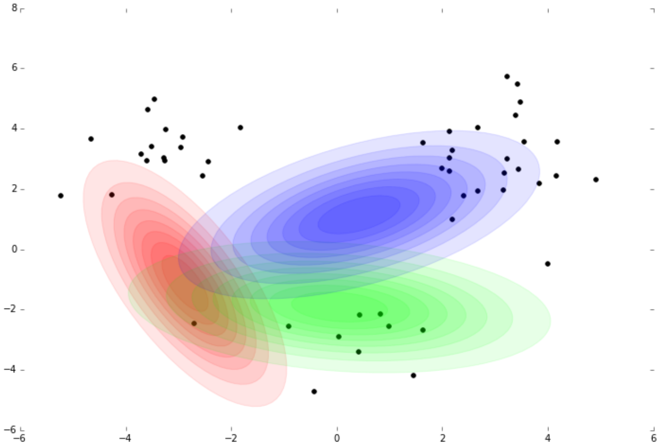
$$\pi_k = \frac{N_k}{N}$$

where $N_k = \sum_{i=1}^N r_{ik}$

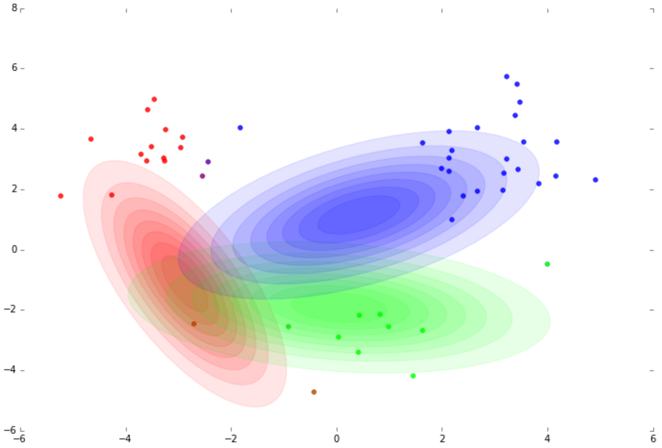
EM demo *data*



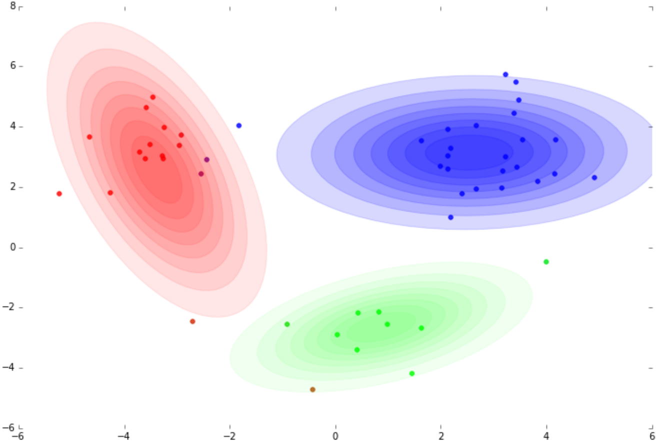
EM demo *initialization*



EM demo *E Step*



EM demo *M Step*



EM demo

- Video 1:
  https://www.youtube.com/watch?v=TLg-fvTfqno
- Video 2:
  https://www.youtube.com/watch?v=uUtpiK5NEAM
- Code:
  https://github.com/hughsalimbeni/variational_inference_demos

Shortcomings of EM GMM

- Sensitive to initialization
- Gives no indication of uncertainty in parameter values
- No easy way of determining the number of clusters
- Can fail due to problematic singularities (if a cluster has fewer points than dimensions the covariance is singular)

[1] though a point estimate (e.g. mode or mean) can be easily obtained if required

Shortcomings of EM GMM

- Sensitive to initialization
- Gives no indication of uncertainty in parameter values
- No easy way of determining the number of clusters
- Can fail due to problematic singularities (if a cluster has fewer points than dimensions the covariance is singular)

The Bayesian approach:

- Less sensitive to initialization
- Provides a *distribution* over parameter values, rather than a point estimate [1]
- Provides the *model evidence* for comparison with other models
- Gives a principled way to determine the number of clusters

[1] though a point estimate (e.g. mode or mean) can be easily obtained if required

Bayesian Gaussian Mixture

- We want the means, covariances and mixture probabilities to be random variables
- For the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, the natural choice is a Normal/Wishart:
- We specify the general shape $\mathbf{W_0}$, a constant that determines the variability of samples ν_0, a center $\mathbf{m}_0$ and a constant b_0 to specify how far the mean should be from m_0 on average.
- $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_0, (\beta_0\boldsymbol{\Sigma}^{-1})^{-1})\mathcal{W}(\boldsymbol{\Sigma}^{-1}|\mathbf{W}_0, \nu_0)$
- We specify a (flat) Dirichlet prior for the mixture probabilities

Visualizing the Normal/Wishart prior

- Video 1:
  https://www.youtube.com/watch?v=-9pyL0WXCsE&feature=youtu.be
- Video 2:
  https://www.youtube.com/watch?v=UO_R8-BaJAU&feature=youtu.be
- Code:
  https://github.com/hughsalimbeni/variational_inference_demos

Bayesian GMM

While the likelihood is the same as before:

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

or

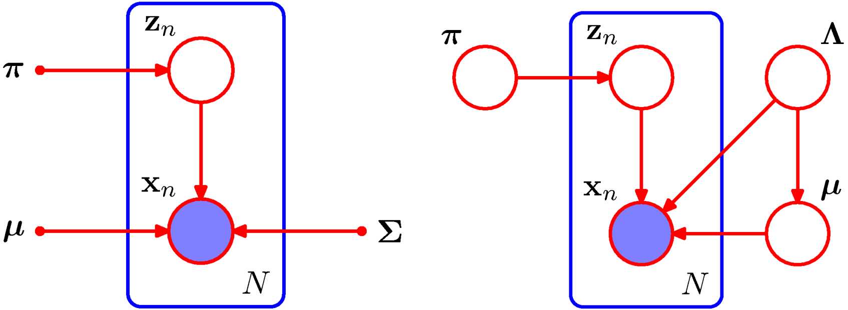
$$p(\boldsymbol{x}_n|\mathsf{Z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{k=1}^{K} N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

We now have a rather more complicated joint distribution:

$$p(\boldsymbol{X}, \mathsf{Z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})p(\boldsymbol{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma})$$

From here we work with $\boldsymbol{\Lambda} = \boldsymbol{\Sigma})^{-1}$

As a graphical model



Maximum likelihood model

Bayesian model

From Bishop PRML 06

Bayesian GMM inference

We need to integrate out all the unobserved variables:

$$p(\boldsymbol{X}) = \iiiint p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda})d\mathbf{Z}d\boldsymbol{\mu}d\boldsymbol{\Lambda}d\boldsymbol{\pi}$$

As the unobserved variables are tangled up in the integrand, unfortunately such integration is analytically intractable.

Variational GMM

- Video 1:
  https://youtu.be/j1LmIB8EoNA
- Video 2:
  https://youtu.be/Fq-oTp2Kpzo
- Code:
  https://github.com/hughsalimbeni/variational_inference_demos

Why we need Bayesian models

- Point estimates can be misleading, and give no indication of uncertainty
- Bayesian methods are much more robust, especially with small data sets
- Bayesian methods incorporate prior beliefs in a principled way

What stops us using Bayesian models?

- Typically intractable in all but the most simple cases

Why we need Bayesian models

- Point estimates can be misleading, and give no indication of uncertainty
- Bayesian methods are much more robust, especially with small data sets
- Bayesian methods incorporate prior beliefs in a principled way

What stops us using Bayesian models?

- Typically intractable in all but the most simple cases
- That's is.

Why we need Bayesian models

- Point estimates can be misleading, and give no indication of uncertainty
- Bayesian methods are much more robust, especially with small data sets
- Bayesian methods incorporate prior beliefs in a principled way

What stops us using Bayesian models?

- Typically intractable in all but the most simple cases
- That's is.

Variational inference is one way of making complex Bayesian models tractable

Problem

We have:

- A generative model: $p(\mathbf{X}|\mathbf{Z})$ and $p(\mathbf{Z})$
- A task:
  - find the model evidence:

$$p(\mathbf{X}) = \int p(\mathbf{X}|\mathbf{Z})\, p(\mathbf{Z})\, d\mathbf{Z}$$

  - find the posterior over the latent variables:

$$p(\mathbf{Z}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{Z})\, p(\mathbf{Z})}{p(\mathbf{X})}$$

We assume:

- Exact inference requires intractable integration

We want:

- To perform exact inference tractably...
- without simplifying the model itself

Two options

1. Approximate the exact model with finite samples

Two options

1. Approximate the exact model with finite samples
   - pros:
     - Asymptotically correct
   - cons:
     - Only finite time available
     - Usually scales poorly with dimension
     - Difficult to determine the quality of approximation
     - Often requires fine tuning to get good results

Two options

1. Approximate the exact model with finite samples
   - pros:
     - Asymptotically correct
   - cons:
     - Only finite time available
     - Usually scales poorly with dimension
     - Difficult to determine the quality of approximation
     - Often requires fine tuning to get good results
2. Use a simpler surrogate model which is as close as possible to the true model

Two options

1. Approximate the exact model with finite samples
   - pros:
     - Asymptotically correct
   - cons:
     - Only finite time available
     - Usually scales poorly with dimension
     - Difficult to determine the quality of approximation
     - Often requires fine tuning to get good results
2. Use a simpler surrogate model which is as close as possible to the true model
   - pros:
     - Can be fast and scalable to high dimension
     - Deterministic
   - cons:
     - Not the true mode
     - Approximation might lose important dependencies
     - May still result in intractable integrals

In summary

Broadly:

- Sampling methods are approximate inference for the exact model
- Variational methods are exact inference for an approximate model

The good news: the 'approximate model' can be guaranteed to be the *best possible* approximation, for a given approximating family

In general:

- High dimensional integration is very hard
- Optimization can be easier

Notation

p	probabilities relating to the exact model
q	probabilities relating to the surrogate model
$\mathbb{E}f(X)$	$= \int f(X)p(X)dX$, assuming the distribution of X is obvious
$\mathbb{E}_{q(Z)}f(X,Z)$	$= \int f(X,Z)q(Z)dZ$, if we need to be careful which distribution we take the expectation over
$\mathcal{L}(X)$	$\log p(X) = \log \int p(X,Z)dZ$ the *log marginal likelihood*

Before we start...

- Easy to work with:
  - $p(X|Z)$. This is just the probability of the data, given the latent variables. If the latent variables are *given* things are easy
  - anything involving q, by design
- Tricky to work with:
  - $p(Z)$, since the true distribution over the unobserved variables is assumed intractable
- Very hard to calculate:
  - $p(X) = \int p(X|Z)\, p(Z)\, dZ$
  - $p(Z|X) = \dfrac{p(X|Z)\, p(Z)}{p(X)}$

Some important things to remember:

- $KL(a(x)||b(x)) = \mathbb{E}_{a(x)} \log \frac{b(x)}{a(x)} dx$
- $KL(a(x)||b(x)) = \mathbb{E}_{a(x)} \log b(x) + H(a)$, $H(.)$ is the entropy
- $KL(a(x)||b(x)) \geq 0$, with equality iff $a \sim b$

The important bit of maths (v1)

- It can be shown that [2] that:
  $\mathcal{L}(X) = \mathbb{E}_{q(Z)} \log \frac{p(X,Z)}{q(Z)} + \mathbb{E}_{q(Z)} \log \frac{q(Z)}{p(Z|X)}$
- The second term is $KL(q(Z) || p(Z|X))$
- We can choose q to make this KL term as close to zero as possible. This is the same as making $q(Z)$ as close as possible to $p(Z|X)$.
- The other term is called the EVidence Lower Bound (ELBO). Minimizing the KL term is the same as maximizing the ELBO

Therefore:

$$\boxed{(\text{max ELBO wrt } q) \iff (q(Z) \text{ is as close as possible to } p(Z|X))}$$

[2] i.e. you will show it in the tutorial

Disclaimer

We have been sloppy with notation

$q(Z)$ depends on X, so it should be written $q(Z|X)$. We are never interested in e.g. $q(X|Z)$, however, so it is safe to drop the dependency

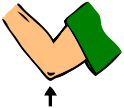
The important bit of maths (v2)

- $\mathcal{L}(X) = \log \mathbb{E}_{q(Z)} \frac{p(X|Z)p(Z)}{q(Z)}$
- Recall importance sampling: $\exp \mathcal{L}(X) \approx \frac{1}{S} \sum \frac{p(X|Z^{(s)})p(Z^{(s)})}{q(Z^{(s)})}$, where $Z^{(s)} \sim q$ and S is the number of samples
- Instead of sampling, use Jensen's inequality [3]. We have:

$$\begin{aligned} \mathcal{L}(X) = \quad & \log \mathbb{E}_{q(Z)} \frac{p(X|Z)p(Z)}{q(Z)} \\ \geq \quad & \mathbb{E}_{q(Z)} \log \left(\frac{p(X|Z)p(Z)}{q(Z)} \right) = \text{ELBO} \end{aligned}$$

[3] $f(\mathbb{E}(Z)) \geq \mathbb{E}(f(Z))$ if f is concave. The logarithm is concave

A closer look at the ELBO



We can write the ELBO in a few different ways

$$
\begin{aligned}
\text{ELBO} &= \mathbb{E}_{q(Z)} \log \frac{p(X|Z)p(Z)}{q(Z)} \\
&= \mathbb{E}_{q(Z)} \log p(X|Z) + \mathbb{E}_{q(Z)} \log \frac{p(Z)}{q(Z)} \\
&= \mathbb{E}_{q(Z)} \log p(X|Z) - KL(q(Z)||p(Z))
\end{aligned}
$$

= reconstructed loglikelihood - a KL penalty (regularizer) term

$$
\begin{aligned}
\text{ELBO} &= \mathbb{E}_{q(Z)} \log \frac{p(X|Z)p(Z)}{q(Z)} \\
&= \mathbb{E}_{q(Z)} \log p(X|Z) + \mathbb{E}_{q(Z)} \log p(Z) + H(q)
\end{aligned}
$$

How to find q?

Clearly the best $q(Z)$ would just be $p(Z|X)$, but that defeats the point...

There are two specific approaches

- Mean field: we assume q factorizes
- Parametric family: we assume q belongs to some tractable family

Today we will cover only the mean field approach

Mean field

- We assume that $q(Z) = q_1(Z_1)q_2(Z_2)...q_M(Z_M)$. Call each factor q_i for convenience
- So we have
$$\begin{aligned} \text{ELBO} = & \ \mathbb{E}_{q(Z)}p(X,Z) - \mathbb{E}_{q(Z)}q(Z) \\ = & \int q_1q_2...q_M \log p(X,Z)dZ_1dZ_2...dZ_M \\ & - \int q_1q_2...q_M \log(q_1q_2...q_M)dZ_1dZ_2...dZ_M \end{aligned}$$
- Using the functional derivative [4] we have [5] $\frac{\delta}{\delta q_1}\text{ELBO} = \int q_2...q_M \log p(X,Z)dZ_2...dZ_M - \log q_1 + \text{const}$.
- Let q_1^* be the optimal q_1 that maximizes the ELBO. Then q_1^* satisfies $\frac{\delta}{\delta q_1}\text{ELBO} = 0$
- This gives $q_1^* \propto \exp \mathbb{E}_{q_2q_3...q_M} \log p(X,Z)$
- Similarly $q_i^* \propto \exp \mathbb{E}_{j\neq i} \log p(X,Z)$, where $\mathbb{E}_{j\neq i}$ means the expectation over all the q_j with $j \neq i$

[4] i.e. $\frac{\delta q(z)}{\delta q(z')} = \delta(z - z')$

[5] this will be an exercise

Mean field Summary

- The optimal factors are given by:

$$q_i^* \propto \exp\left(\mathbb{E}_{j \neq i} \log p(X, Z)\right)$$

- Note we have made no assumption about the form of the q_i, beyond the factorization. This is sometimes called 'free form' optimization for this reason.
- We could find the normalization constant by integrating over Z_i, but in practice we will spot it by inspection

Mean field example 1: 2D Gaussian

Consider a 2D Guassian: $\mathbf{z} \sim N\left(\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \mid \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}^{-1}\right)$

- We assume the variational distribution factorises as $q(\mathbf{z}) = q_1(z_1)q_2(z_2)$. Notice that full distribution doesn't unless $\Lambda_{21} = \Lambda_{12} = 0$
- We know the optimal factor $\log q_1^*(z_1) = \mathbb{E}_{q_2(z_2)} \log p(\mathbf{z}) + \text{const.}$
- Note that this is function of z_1, so we only need consider terms depending on z_1
- For the multivariate normal, the logpdf is just a quadratic form in z_1 (and z_2).
- The details of the derivation are left for the tutorial

Mean field example 1: 2D Gaussian continued

- The final result is:

$$q_1^*(z_1) = N(z_1|\mu_1, \Lambda_{11}^{-1})$$

  and similarly for q_2^*

- Note that we did not specify that the factors should be Gaussian. The Gaussian is the optimal solution over all possible functions, given the factorization we started with

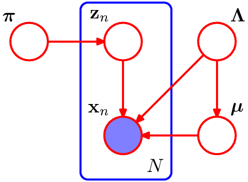
2D Gaussian demo

Video:

https://www.youtube.com/watch?v=aGtWphP2W_Q

Variational Inference for Bayesian GMM

Recall the graphical model:



Or in symbols:

$$p(\boldsymbol{X}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\pi}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda})$$

We choose the form of the variational posterior to be as rich as possible:

$$q(\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\pi}) = q(\boldsymbol{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$$

It turns out that this is all we need to assume to make things tractable

What we need

All we need is two expectations:

$$q^{*}\left(\mathbf{Z}\right)=\exp\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}\left(\log p\left(\mathbf{X},\mathbf{Z},\pi,\boldsymbol{\mu},\boldsymbol{\Lambda}\right)\right)$$

and

$$q^{*}\left(\pi,\boldsymbol{\mu},\boldsymbol{\Lambda}\right)=\exp\mathbb{E}_{\mathbf{Z}}\left(\log p\left(\mathbf{X},\mathbf{Z},\pi,\boldsymbol{\mu},\boldsymbol{\Lambda}\right)\right)$$

The log joint

Recall the full joint:

$$p(\mathbf{X}, \mathbf{Z}, \mu, \mathbf{\Lambda}, \pi) = p(\mathbf{X}|\mathbf{Z}, \mu, \mathbf{\Lambda})p(\mathbf{Z}|\pi)p(\pi)p(\mu|\mathbf{\Lambda})p(\mathbf{\Lambda})$$

Separating out the terms we have:

$$\log p\left(\mathbf{X}, \mathbf{Z}, \pi, \mu, \mathbf{\Lambda}\right) = \sum_{k=1}^{K} \Big[$$

$$\log \prod_n p\left(\mathbf{x}_n | z_{nk}, \boldsymbol{\mu}_k, \Sigma_k\right) +$$

$$\log \prod_n p\left(z_{nk} | \boldsymbol{\pi}_k\right) +$$

$$\log p\left(\boldsymbol{\pi}_k\right) +$$

$$\log p\left(\boldsymbol{\mu}_k | \mathbf{\Lambda}_k\right) +$$

$$\log p\left(\mathbf{\Lambda}_k\right) \Big]$$

In some more detail...

$$
\begin{aligned}
\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \sum_{k=1}^{K} \Big[\\
\log \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{x}_n | \boldsymbol{\mu}_k, {\boldsymbol{\Lambda}_k}^{-1}\right)^{z_k} + \\
\log \prod_{n=1}^{N} {\boldsymbol{\pi}_k}^{z_{nk}} + \\
\log \mathcal{D}\left(\boldsymbol{\pi} | \alpha_0\right) + \\
\log \mathcal{N}\left(\boldsymbol{\mu}_k | m_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}\right) + \\
\log \mathcal{W}\left(\boldsymbol{\Lambda}_k | W_0, v_0\right) \Big]
\end{aligned}
$$

In full glory...

$$\log p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = \sum_{k=1}^{K} \Bigg[$$

$$\sum_{n=1}^{N} z^{nk} \left(-\frac{1}{2}\log|\boldsymbol{\Lambda}_k| - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)\right) +$$

$$\sum_{n=1}^{N} z^{nk} \log \boldsymbol{\pi}_k +$$

$$(\alpha_0 - 1) \log \boldsymbol{\pi}_k +$$

$$\frac{1}{2}\log|\beta_0 \boldsymbol{\Lambda}_k| - \frac{1}{2}(\boldsymbol{\mu}_k - m_0)^T \left(\beta_0 \boldsymbol{\Lambda}_k\right)\left(\boldsymbol{\mu}_k - m_0\right) +$$

$$\left(\frac{N-D-1}{2}\right)\log|\boldsymbol{\Lambda}_k| - \frac{1}{2} tr\left(\mathbf{W}_0{}^{-1}\boldsymbol{\Lambda}\right) \Bigg]$$

where we dropped the constant terms

Start with $\mathbf{Z}$

To compute

$$\log q^*(\mathbf{Z}) = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}\left(\log p\left(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}\right)\right)$$

we need only consider terms that depend on z_{nk}

For Z, terms needed:

$$\log p\left(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}\right)=\sum_{k=1}^{K}\Bigg[$$

$$\sum_{n=1}^{N} z^{nk}\left(-\frac{1}{2}\log|\boldsymbol{\Lambda}_k|-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k\left(\mathbf{x}_n-\boldsymbol{\mu}_k\right)\right)+$$

$$\sum_{n=1}^{N} z^{nk}\log\boldsymbol{\pi}_k+$$

$$(\alpha_0-1)\log\boldsymbol{\pi}_k+$$

$$\frac{1}{2}\log|\beta_0\boldsymbol{\Lambda}_k|-\frac{1}{2}(\boldsymbol{\mu}_k-m_0)^T\left(\beta_0\boldsymbol{\Lambda}_k\right)\left(\boldsymbol{\mu}_k-m_0\right)+$$

$$\left(\frac{N-D-1}{2}\right)\log|\boldsymbol{\Lambda}_k|-\frac{1}{2}tr\left(\mathbf{W}_0{}^{-1}\boldsymbol{\Lambda}\right)\Bigg]$$

Finding $q^*(Z)$

So we have $\log q^*(\mathbf{Z}) = \sum_{nk}$

$$\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}\left(z_{nk}\left(-\frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Lambda_k (\mathbf{x}_n - \boldsymbol{\mu}_k)\right) + z_{nk}\log\boldsymbol{\pi}_k\right) + \text{cst}$$

$+$ constant terms.
Since the expectation is not over z_{nk} we can take the z_{nk} out

$$\log q^*(\mathbf{Z}) = \sum_{nk} z_{nk} \log \rho_{nk}$$

where

$$\log \rho_{nk} = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}\left(-\frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) + \log\boldsymbol{\pi}_k\right)$$

While ρ doesn't look promising, this is actually a nice answer for Z.

The final result for $q^*(Z)$

Taking exponentials we have:

$$q^*(\mathbf{Z}) \propto \prod_n \prod_k {\rho_{nk}}^{z_{nk}}$$

Which is just

$$q^*(\mathbf{Z}) = \prod_n \prod_k {r_{nk}}^{z_{nk}}$$

where r_{nk} is the normalized version of ρ_{nk} , i.e. another categorical random variable with updated probabilities.

- We now know $\mathbb{E}(z_{nk}) = \rho_{nk}$
- Note that we can't calculate the expectations until we know the variational posteriors of the other variables.

The other expectation

Next we consider the second expectation:

$$q^* (\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \exp \mathbb{E}_{\mathbf{Z}} (\log p (\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}))$$

Since Z d-separates $\boldsymbol{\pi}$ from all the other nodes we have $q (\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q (\boldsymbol{\pi}) \, q (\boldsymbol{\mu}, \boldsymbol{\Lambda})$

Note that we didn't have to assume this. It fell out naturally.

For π, terms needed:

$$\log p(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \sum_{k=1}^{K}\Bigg[$$

$$\sum_{n=1}^{N} z^{nk}\left(-\tfrac{1}{2}\log|\boldsymbol{\Lambda}_k| - \tfrac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\mathbf{x}_n-\boldsymbol{\mu}_k)\right)$$

$$\boxed{\sum_{n=1}^{N} z^{nk}\log\boldsymbol{\pi}_k} +$$

$$\boxed{(\alpha_0-1)\log\boldsymbol{\pi}_k} +$$

$$\tfrac{1}{2}\log|\beta_0\boldsymbol{\Lambda}_k| - \tfrac{1}{2}(\boldsymbol{\mu}_k-m_0)^T(\beta_0\boldsymbol{\Lambda}_k)(\boldsymbol{\mu}_k-m_0) +$$

$$\left(\tfrac{N-D-1}{2}\right)\log|\boldsymbol{\Lambda}_k| - \tfrac{1}{2}tr\left(\mathbf{W}_0{}^{-1}\boldsymbol{\Lambda}\right)\Bigg]$$

Note these terms do not depend on μ_k or Λ_k, so we have $q(\boldsymbol{\pi},\boldsymbol{\mu},\Sigma) = q(\boldsymbol{\pi})\,q(\boldsymbol{\mu},\Sigma)$

Terms involving π

So we have have:

$$\log q^*(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \mathbb{E}_{\mathbf{Z}}\left[\sum_{k=1}^{K}\sum_{n=1}^{N} z^{nk}\log \boldsymbol{\pi}_k + (\alpha_0 - 1)\log \boldsymbol{\pi}_k\right]$$

$+$terms not containing $\boldsymbol{\pi}$

So

$$\log q^*(\boldsymbol{\pi}) = \mathbb{E}_Z \sum_{k=1}^{K}\sum_{n=1}^{N} z^{nk}\log \boldsymbol{\pi}_k + (\alpha_0 - 1)\log \boldsymbol{\pi}_k + const$$

Since we know $\mathbb{E}(z_{nk}) = \rho_{nk}$ we have

$$\log q^*(\boldsymbol{\pi}) = \sum_{k=1}^{K}\sum_{n=1}^{N} r^{nk}\log \boldsymbol{\pi}_k + (\alpha_0 - 1)\log \boldsymbol{\pi}_k + const$$

Result for $q^*(\pi)$

Rearranging we have:

$$\log q^*(\boldsymbol{\pi}) = \sum_k \left(\sum_{n=1}^{N} r^{nk} + \alpha_0 - 1 \right) \log \boldsymbol{\pi}_k + const$$

This is exactly the form of another Dirichlet distribution:

$$q^*(\boldsymbol{\pi}) = \mathcal{D}\left(\boldsymbol{\pi} | \alpha_0 + \sum_{n=1}^{N} r^{nk} \right)$$

The remaining $q(\mu, \Lambda)$

Now we can compute $\log q^* (\boldsymbol{\mu},\boldsymbol{\Lambda})$ by looking at all the terms that contain $\boldsymbol{\mu}_k$ or $\boldsymbol{\Lambda}_k$.

It turns out that this is just another Normal/Wishart, but we won't do the details as they are ugly but straightforward (we just need to keep using $\mathbb{E}(z_{nk}) = \rho_{nk}$ and do some heavy duty completing the square)

To conclude

The important point is that all the posteriors can be found analytically, but they all depend on ρ_{nk}, which was defined as

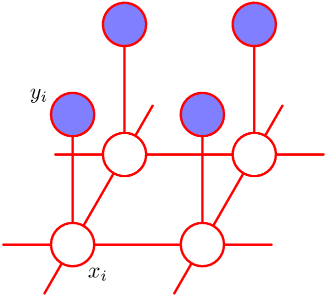
$$\log \rho_{nk} = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}} \left(-\frac{1}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) + \log \boldsymbol{\pi}_k \right)$$

Now we have the variational posteriors over π, μ, Λ we can compute these terms analytically.
We have to proceed iteratively:

- $q^*(\pi)$ and $q^*(\mu, \Lambda)$ depend on $q(Z)$
- $q^*(Z)$ depends on $q(\pi)$ and $q(\mu, \Lambda)$

Questions?

Ising Model



from Bishop PRML 2006

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_i \sum_{j \in \text{nbr}_i} x_i x_j + \sigma \sum_i x_i y_i \right)$$

Where $x_i, y_i \in \{-1, 1\}$ and σ is some constant
Finding $p(\mathbf{x}|\mathbf{y})$ requires a sum over 2^N states

Ising Model 2

- Use a variational posterior $q(\mathbf{x}) = \prod_i q(x_i)$
- For a fully factorized variational posterior we have

$$q_i(x_i) \propto \exp \mathbb{E}_{j \neq i} \left(x_i \sum_{j \in \mathrm{nbr}_i} x_j + \sigma y_i x_i \right)$$

  dropping all terms that do not depend on x_i
- It follows that

$$q_i(x_i) \propto \exp \left(x_i \sum_{j \in \mathrm{nbr}_i} \mu_j + \sigma y_i x_i \right)$$

Where $\mu_j = \mathbb{E}(q_j)$

- q_i depends only on its neighbours
- Closed form updates can be found for μ_i

Ising Model Demo