Probabilistic clustering and the EM algorithm

Guillaume Obozinski

Ecole des Ponts - ParisTech

INIT/AERFAI Summer school on Machine Learning
Benicàssim, June 26th 2017
Outline

1. The EM algorithm for the Gaussian mixture model

2. More examples of graphical models
K-means

**Key assumption:** Data composed of $K$ “roundish” clusters of similar sizes with centroids $(\mu_1, \cdots, \mu_K)$. 

Problem can be formulated as: 

$$ \min_{\mu_1, \cdots, \mu_K} \frac{1}{n} \sum_{i=1}^{n} \min_k \| x_i - \mu_k \|_2.$$ 

Difficult (NP-hard) nonconvex problem.

**K-means algorithm**

1. Draw centroids at random
2. Assign each point to the closest centroid
   $$ C_k \leftarrow \{ i | \| x_i - \mu_k \|_2 = \min_j \| x_i - \mu_j \|_2 \} $$
3. Recompute centroid as center of mass of the cluster
   $$ \mu_k \leftarrow \frac{1}{|C_k|} \sum_{i \in C_k} x_i $$
4. Go to 2.
K-means

Key assumption: Data composed of $K$ “roundish” clusters of similar sizes with centroids $(\mu_1, \cdots, \mu_K)$.

Problem can be formulated as: $\min_{\mu_1, \cdots, \mu_K} \frac{1}{n} \sum_{i=1}^{n} \min_k \|x_i - \mu_k\|^2$. 
K-means

**Key assumption:** Data composed of $K$ “roundish” clusters of similar sizes with centroids $(\mu_1, \cdots, \mu_K)$.

Problem can be formulated as: $\min_{\mu_1, \cdots, \mu_K} \frac{1}{n} \sum_{i=1}^{n} \min_k \|x_i - \mu_k\|^2$. Difficult (NP-hard) nonconvex problem.
K-means

Key assumption: Data composed of $K$ “roundish” clusters of similar sizes with centroids $(\mu_1, \cdots, \mu_K)$.

Problem can be formulated as: 

$$\min_{\mu_1, \cdots, \mu_K} \frac{1}{n} \sum_{i=1}^{n} \min_k \|x_i - \mu_k\|^2.$$ 

Difficult (NP-hard) nonconvex problem.

K-means algorithm

1. Draw centroids at random
2. Assign each point to the closest centroid
   $$C_k \leftarrow \{i \mid \|x_i - \mu_k\|^2 = \min_j \|x_i - \mu_j\|^2\}$$
3. Recompute centroid as center of mass of the cluster
   $$\mu_k \leftarrow \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$
4. Go to 2
K-means properties

Three remarks:

- K-means is a greedy algorithm

It can be shown that K-means converges in a finite number of steps. The algorithm however typically gets stuck in local minima and in practice it is necessary to try several restarts of the algorithm with a random initialization to have chances to obtain a better solution.

Will fail if the clusters are not round.

A good initialization for K-means is K-means++, (Arthur and Vassilvitskii, 2007), (included in all good libraries).

K-means properties

Three remarks:

- K-means is greedy algorithm
- It can be shown that K-means converges in a finite number of steps.

K-means properties

Three remarks:

- K-means is greedy algorithm
- It can be shown that K-means converges in a finite number of steps.
- The algorithm however typically get stuck in local minima and in practice it is necessary to try several restarts of the algorithm with a random initialization to have chances to obtain a better solution.

K-means properties

Three remarks:

- K-means is greedy algorithm
- It can be shown that K-means converges in a finite number of steps.
- The algorithm however typically get stuck in local minima and in practice it is necessary to try several restarts of the algorithm with a random initialization to have chances to obtain a better solution.
- Will fail if the clusters are not round
- A good initialization for K-means is K-means++, (Arthur and Vassilvitskii, 2007), (included in all good libraries).

Outline

1. The EM algorithm for the Gaussian mixture model
2. More examples of graphical models
The Gaussian mixture model and the EM algorithm
Gaussian mixture model

- $K$ components
- $z$ component indicator
- $z = (z_1, \ldots, z_K)^\top \in \{0, 1\}^K$
- $z \sim M(1, (\pi_1, \ldots, \pi_K))$
- $p(z) = \prod_{k=1}^{K} \pi_{z_k}^{z_k}$
Gaussian mixture model

- $K$ components
- $z$ component indicator
- $z = (z_1, \ldots, z_K)^\top \in \{0, 1\}^K$
- $z \sim \mathcal{M}(1, (\pi_1, \ldots, \pi_K))$
- $p(z) = \prod_{k=1}^{K} \pi_{z_k}^k$
- $p(x|z; (\mu_k, \Sigma_k)_k) = \sum_{k=1}^{K} z_k \mathcal{N}(x; \mu_k, \Sigma_k)$
Gaussian mixture model

- $K$ components
- $z$ component indicator
- $z = (z_1, \ldots, z_K) \top \in \{0, 1\}^K$
- $z \sim \mathcal{M}(1, (\pi_1, \ldots, \pi_K))$

$$p(z) = \prod_{k=1}^{K} \pi_{z_k}^k$$

$$p(x|z; (\mu_k, \Sigma_k)_k) = \sum_{k=1}^{K} z_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$
Gaussian mixture model

- $K$ components
- $z$ component indicator
- $z = (z_1, \ldots, z_K)^\top \in \{0, 1\}^K$
- $z \sim \mathcal{M}(1, (\pi_1, \ldots, \pi_K))$
- $p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$

- $p(x|z; (\mu_k, \Sigma_k)_k) = \sum_{k=1}^{K} z_k \mathcal{N}(x; \mu_k, \Sigma_k)$
- $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$

Estimation: $\arg\max_{\mu_k, \Sigma_k} \log \left[ \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]$
Applying maximum likelihood to the Gaussian mixture

Let $Z = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \}$
Applying maximum likelihood to the Gaussian mixture

Let \( \mathcal{Z} = \{ z \in \{0, 1\}^K | \sum_{k=1}^K z_k = 1 \} \)

\[ p(x) = \]
Applying maximum likelihood to the Gaussian mixture

Let $\mathcal{Z} = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \}$

$p(x) = \sum_{z \in \mathcal{Z}} p(x, z)$
Applying maximum likelihood to the Gaussian mixture

Let \( Z = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \} \)

\[
p(x) = \sum_{z \in Z} p(x, z) = \sum_{z \in Z} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k}
\]
Applying maximum likelihood to the Gaussian mixture

Let \( Z = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \} \)

\[
p(x) = \sum_{z \in Z} p(x, z) = \sum_{z \in Z} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k} = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)
\]
Applying maximum likelihood to the Gaussian mixture

Let \( Z = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \} \)

\[
p(x) = \sum_{z \in Z} p(x, z) = \sum_{z \in Z} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k} = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)
\]

Issue

- The marginal log-likelihood \( \tilde{\ell}(\theta) = \sum_i \log(p(x^{(i)})) \) with \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \) is now complicated
Applying maximum likelihood to the Gaussian mixture

Let \( \mathcal{Z} = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \} \)

\[
p(x) = \sum_{z \in \mathcal{Z}} p(x, z) = \sum_{z \in \mathcal{Z}} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k} = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)
\]

Issue

- The marginal log-likelihood \( \tilde{\ell}(\theta) = \sum_i \log(p(x^{(i)})) \) with \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \) is now complicated
- No hope to find a simple solution to the maximum likelihood problem
Applying maximum likelihood to the Gaussian mixture

Let \( \mathcal{Z} = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1 \} \)

\[
p(x) = \sum_{z \in \mathcal{Z}} p(x, z) = \sum_{z \in \mathcal{Z}} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k} = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)
\]

Issue

- The marginal log-likelihood \( \tilde{\ell}(\theta) = \sum_i \log(p(x^{(i)})) \) with \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \) is now complicated
- No hope to find a simple solution to the maximum likelihood problem
- By contrast the complete log-likelihood has a rather simple form:
Applying maximum likelihood to the Gaussian mixture

Let \( Z = \{ z \in \{0, 1\}^K \mid \sum_{k=1}^K z_k = 1 \} \)

\[
p(x) = \sum_{z \in Z} p(x, z) = \sum_{z \in Z} \prod_{k=1}^K \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right] z_k = \sum_{k=1}^K \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)
\]

**Issue**

- The marginal log-likelihood \( \tilde{\ell}(\theta) = \sum_i \log(p(x^{(i)})) \) with \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \) is now complicated

- No hope to find a simple solution to the maximum likelihood problem

- By contrast the complete log-likelihood has a rather simple form:

\[
\tilde{\ell}(\theta) = \sum_{i=1}^M \log p(x^{(i)}, z^{(i)})
\]
Applying maximum likelihood to the Gaussian mixture

Let $\mathcal{Z} = \{z \in \{0, 1\}^K \mid \sum_{k=1}^{K} z_k = 1\}$

$$p(x) = \sum_{z \in \mathcal{Z}} p(x, z) = \sum_{z \in \mathcal{Z}} \prod_{k=1}^{K} \left[ \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \right]^{z_k} = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

Issue

- The marginal log-likelihood $\tilde{\ell}(\theta) = \sum_i \log(p(x^{(i)}))$ with $\theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K})$ is now complicated
- No hope to find a simple solution to the maximum likelihood problem
- By contrast the complete log-likelihood has a rather simple form:

$$\tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i, k} z_k^{(i)} \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i, k} z_k^{(i)} \log(\pi_k)$$
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \]

\[ \sum_{i=1}^{M} \log p(x(i), z(i)) = \sum_{i,k} z(i)k \log N(x(i); \mu_k, \Sigma_k) + \sum_{i,k} z(i)k \log(\pi_k) \]

If we knew \( z(i) \) we could maximize \( \tilde{\ell}(\theta) \).

If we knew \( \theta = (\pi, (\mu_k, \Sigma_k))_{1 \leq k \leq K} \), we could find the best \( z(i) \) since we could compute the true a posteriori on \( z(i) \) given \( x(i) \):

\[ p(z(i) = 1 | x; \theta) = \frac{\pi_k N(x; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x; \mu_j, \Sigma_j)} \]

\[ \text{\( \rightarrow \)} \text{ seems a chicken and egg problem...} \]

In addition, we want to solve

\[ \max_{\theta} \sum_{i} \log \left( \sum_{z(i)} p(x(i), z(i)) \right) \]

and not

\[ \max_{\theta, z(1), \ldots, z(M)} \sum_{i} \log p(x(i), z(i)) \]

Can we still use the intuitions above to construct an algorithm maximizing the marginal likelihood?
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) \]
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z_{k}^{(i)} \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z_{k}^{(i)} \log(\pi_k), \]
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z^{(i)}_k \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z^{(i)}_k \log(\pi_k), \]

- If we knew \( z^{(i)} \) we could maximize \( \tilde{\ell}(\theta) \).
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log (\pi_k), \]

- If we knew \( z^{(i)} \) we could maximize \( \tilde{\ell}(\theta) \).
- If we knew \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \), we could find the best \( z^{(i)} \) since we could compute the true a posteriori on \( z^{(i)} \) given \( x^{(i)} \):
Applying ML to the multinomial mixture

\[
\tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log(\pi_k),
\]

- If we knew \(z^{(i)}\) we could maximize \(\tilde{\ell}(\theta)\).
- If we knew \(\theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K})\), we could find the best \(z^{(i)}\) since we could compute the true a posteriori on \(z^{(i)}\) given \(x^{(i)}\):

\[
p(z_k^{(i)} = 1 \mid x; \theta) = \frac{\pi_k \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)}; \mu_j, \Sigma_j)}
\]
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log(\pi_k), \]

- If we knew \( z^{(i)} \) we could maximize \( \tilde{\ell}(\theta) \).
- If we knew \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \), we could find the best \( z^{(i)} \) since we could compute the true a posteriori on \( z^{(i)} \) given \( x^{(i)} \):

\[ p(z_k^{(i)} = 1 \mid x; \theta) = \frac{\pi_k \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)}; \mu_j, \Sigma_j)} \]

→ Seems a chicken and egg problem...
Applying ML to the multinomial mixture

$$\tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z^{(i)}_{k} \log \mathcal{N}(x^{(i)}; \mu_{k}, \Sigma_{k}) + \sum_{i,k} z^{(i)}_{k} \log(\pi_{k}),$$

- If we knew $z^{(i)}$, we could maximize $\tilde{\ell}(\theta)$.
- If we knew $\theta = (\pi, (\mu_{k}, \Sigma_{k})_{1 \leq k \leq K})$, we could find the best $z^{(i)}$ since we could compute the true a posteriori on $z^{(i)}$ given $x^{(i)}$:

  $$p(z^{(i)}_{k} = 1 \mid x; \theta) = \frac{\pi_{k} \mathcal{N}(x^{(i)}; \mu_{k}, \Sigma_{k})}{\sum_{j=1}^{K} \pi_{j} \mathcal{N}(x^{(i)}; \mu_{j}, \Sigma_{j})}$$

→ Seems a chicken and egg problem...
- In addition, we want to solve

$$\max_{\theta} \sum_{i} \log \left( \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}) \right)$$
Applying ML to the multinomial mixture

$$\tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z^{(i)}_k \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z^{(i)}_k \log(\pi_k),$$

- If we knew $z^{(i)}$ we could maximize $\tilde{\ell}(\theta)$.
- If we knew $\theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K})$, we could find the best $z^{(i)}$ since we could compute the true a posteriori on $z^{(i)}$ given $x^{(i)}$:

$$p(z^{(i)}_k = 1 | x; \theta) = \frac{\pi_k \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)}; \mu_j, \Sigma_j)}$$

→ Seems a chicken and egg problem...
- In addition, we want to solve

$$\max_{\theta} \sum_{i} \log \left( \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}) \right) \quad \text{and not} \quad \max_{\theta, \{z^{(1)}, \ldots, z^{(M)}\}} \sum_{i} \log p(x^{(i)}, z^{(i)})$$
Applying ML to the multinomial mixture

\[ \tilde{\ell}(\theta) = \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}) = \sum_{i,k} z^{(i)}_k \log \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k) + \sum_{i,k} z^{(i)}_k \log(\pi_k), \]

- If we knew \( z^{(i)} \) we could maximize \( \tilde{\ell}(\theta) \).
- If we knew \( \theta = (\pi, (\mu_k, \Sigma_k)_{1 \leq k \leq K}) \), we could find the best \( z^{(i)} \) since we could compute the true a posteriori on \( z^{(i)} \) given \( x^{(i)} \):

\[
p(z^{(i)}_k = 1 \mid x; \theta) = \frac{\pi_k \mathcal{N}(x^{(i)}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)}; \mu_j, \Sigma_j)}
\]

→ Seems a chicken and egg problem...

- In addition, we want to solve

\[
\max_{\theta} \sum_{i} \log \left( \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}) \right) \quad \text{and not} \quad \max_{\theta, z^{(1)}, \ldots, z^{(M)}} \sum_{i} \log p(x^{(i)}, z^{(i)})
\]

- Can we still use the intuitions above to construct an algorithm maximizing the marginal likelihood?
Principle of the Expectation-Maximization Algorithm

$$\log p(x; \theta) = \log \sum_z z \cdot p(x, z; \theta) \geq \sum_z z \cdot q(z) \log p(x, z; \theta) \cdot q(z) =: L(q, \theta)$$

This shows that $L(q, \theta)$ is typically a concave function. Finally it is possible to show that $L(q, \theta) = \log p(x; \theta) - KL(q || p(\cdot | x; \theta))$. So that if we set $q(z) = p(z | x; \theta(t))$ then $L(q, \theta(t)) = p(x; \theta(t))$. If the complete log-likelihood is a canonical exponential family.
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta)
\]
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}
\]
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}
\]

\[
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)}
\]
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)} \\
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)} \\
= \mathbb{E}_q[\log p(x, z; \theta)] + H(q)
\]
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}
\]

\[
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)}
\]

\[
= \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)
\]
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}
\]

\[
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)}
\]

\[
= \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)
\]

This shows that \( \mathcal{L}(q, \theta) \leq \log p(x; \theta) \)
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)} \\
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)} \\
= \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)
\]

- This shows that \( \mathcal{L}(q, \theta) \leq \log p(x; \theta) \)
- \( \theta \mapsto \mathcal{L}(q, \theta) \) is typically a \textit{concave} function.
Principle of the Expectation-Maximization Algorithm

\[
\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)} \\
\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)} \\
= \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)
\]

- This shows that \(\mathcal{L}(q, \theta) \leq \log p(x; \theta)\)
- \(\theta \mapsto \mathcal{L}(q, \theta)\) is typically a \textbf{concave} function\(^a\).
- Finally it is possible to show that
  \[
  \mathcal{L}(q, \theta) = \log p(x; \theta) - KL(q||p(\cdot|x; \theta))
  \]
Principle of the Expectation-Maximization Algorithm

$$\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}$$

$$\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)} = \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)$$

- This shows that $\mathcal{L}(q, \theta) \leq \log p(x; \theta)$
- $\theta \mapsto \mathcal{L}(q, \theta)$ is typically a concave function.
- Finally it is possible to show that

$$\mathcal{L}(q, \theta) = \log p(x; \theta) - KL(q||p(\cdot|x; \theta))$$

So that if we set $q(z) = p(z \mid x; \theta^{(t)})$ then

$$L(q, \theta^{(t)}) = p(x; \theta^{(t)})$$
Principle of the Expectation-Maximization Algorithm

$$\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z) \frac{p(x, z; \theta)}{q(z)}$$

$$\geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)}$$

$$= \mathbb{E}_q[\log p(x, z; \theta)] + H(q) =: \mathcal{L}(q, \theta)$$

- This shows that $\mathcal{L}(q, \theta) \leq \log p(x; \theta)$
- $\theta \mapsto \mathcal{L}(q, \theta)$ is typically a **concave** function$^a$.
- Finally it is possible to show that
  $$\mathcal{L}(q, \theta) = \log p(x; \theta) - KL(q||p(\cdot|x; \theta))$$

So that if we set $q(z) = p(z | x; \theta^{(t)})$ then
$$L(q, \theta^{(t)}) = p(x; \theta^{(t)})$$.

$^a$If the complete log-likelihood is a canonical exponential family.
A graphical idea of the EM algorithm

\[ L(q, \theta) \]

\[ \ln p(X|\theta) \]
Expectation Maximization algorithm

Expectation step

Maximization step

\[ \theta^{\text{old}} = \theta^{(t-1)} \]

\[ \theta^{\text{new}} = \theta^{(t)} \]
Expectation Maximization algorithm

**Expectation step**

\[ q(z) = p(z \mid x; \theta^{(t-1)}) \]

**Maximization step**

\[ \theta^{\text{old}} = \theta^{(t-1)} \]
\[ \theta^{\text{new}} = \theta^{(t)} \]
Expectation Maximization algorithm

**Expectation step**

1. \( q(z) = p(z | x; \theta^{(t-1)}) \)

2. \( \mathcal{L}(q, \theta) = \mathbb{E}_q \left[ \log p(x, z; \theta) \right] + H(q) \)

**Maximization step**

\[
\begin{align*}
\theta_{\text{old}} &= \theta^{(t-1)} \\
\theta_{\text{new}} &= \theta^{(t)}
\end{align*}
\]
Expectation Maximization algorithm

**Expectation step**

1. $q(z) = p(z \mid x; \theta^{(t-1)})$

2. $\mathcal{L}(q, \theta) = \mathbb{E}_q[\log p(x, z; \theta)] + H(q)$

**Maximization step**

$\theta^{(t)} = \text{argmax} \mathbb{E}_q[\log p(x, z; \theta)]$

\[
\begin{align*}
\theta^{\text{old}} &= \theta^{(t-1)} \\
\theta^{\text{new}} &= \theta^{(t)}
\end{align*}
\]
Expectation Maximization algorithm

Initialize $\theta = \theta_0$

WHILE (Not converged)

**Expectation step**
1. $q(z) = p(z | x; \theta^{(t-1)})$
2. $\mathcal{L}(q, \theta) = \mathbb{E}_q[\log p(x, z; \theta)] + H(q)$

**Maximization step**
1. $\theta^{(t)} = \arg\max_{\theta} \mathbb{E}_q[\log p(x, z; \theta)]$

ENDWHILE
Expected complete log-likelihood

With the notation: $q_{ik}^{(t)} = P_{q_i^{(t)}}(z_k^{(i)} = 1) = \mathbb{E}_{q_i^{(t)}}[z_k^{(i)}]$, we have
Expected complete log-likelihood

With the notation: \( q_{ik}^{(t)} = \mathbb{P}_{q^{(t)}}(z_{k}^{(i)} = 1) = \mathbb{E}_{q^{(t)}}[z_{k}^{(i)}] \), we have

\[
\mathbb{E}_{q^{(t)}}[\tilde{\ell}(\theta)] =
\]
Expected complete log-likelihood

With the notation: $q_{ik}^{(t)} = P_{q(t)}(z_k^{(i)} = 1) = \mathbb{E}_{q(t)}[z_k^{(i)}]$, we have

$$\mathbb{E}_{q(t)}[\tilde{\ell}(\theta)] = \mathbb{E}_{q(t)}[\log p(X, Z; \theta)]$$
Expected complete log-likelihood

With the notation: \( q_{ik}^{(t)} = P_{q_i} (z_k^{(i)} = 1) = \mathbb{E}_{q_i} [z_k^{(i)}] \), we have

\[
\mathbb{E}_{q(t)} [\tilde{\ell} (\theta)] = \mathbb{E}_{q(t)} [\log p(X, Z; \theta)] = \mathbb{E}_{q(t)} \left[ \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}; \theta) \right]
\]
Expected complete log-likelihood

With the notation: \( q_{ik}^{(t)} = P_{q_i}(z_k^{(i)} = 1) = \mathbb{E}_{q_i} [z_k^{(i)}] \), we have

\[
\mathbb{E}_{q(t)} [\tilde{\ell}(\theta)] = \mathbb{E}_{q(t)} [\log p(X, Z; \theta)] \\
= \mathbb{E}_{q(t)} \left[ \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}; \theta) \right] \\
= \mathbb{E}_{q(t)} \left[ \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log (\pi_k) \right]
\]
Expected complete log-likelihood

With the notation: \( q_{ik}^{(t)} = \mathbb{P}_{q_i} (z_k^{(i)} = 1) = \mathbb{E}_{q_i} [z_k^{(i)}] \), we have

\[
\mathbb{E}_{q(t)} [\tilde{\ell}(\theta)] = \mathbb{E}_{q(t)} \left[ \log p(X, Z; \theta) \right]
\]

\[
= \mathbb{E}_{q(t)} \left[ \sum_{i=1}^{M} \log p(x^{(i)}, z^{(i)}; \theta) \right]
\]

\[
= \mathbb{E}_{q(t)} \left[ \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log(\pi_k) \right]
\]

\[
= \sum_{i,k} \mathbb{E}_{q_i^{(t)}} [z_k^{(i)}] \log \mathcal{N}(x^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} \mathbb{E}_{q_i^{(t)}} [z_k^{(i)}] \log(\pi_k)
\]
Expected complete log-likelihood

With the notation: $q_{ik}^{(t)} = P_{q_i}(z_k^i = 1) = E_{q_i}(z_k^i)$, we have

$$E_{q(t)}[\tilde{\ell}(\theta)] = E_{q(t)}[\log p(X, Z; \theta)]$$

$$= E_{q(t)} \left[ \sum_{i=1}^{M} \log p(x_i^{(i)}, z^{(i)}; \theta) \right]$$

$$= E_{q(t)} \left[ \sum_{i,k} z_k^{(i)} \log \mathcal{N}(x_i^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} z_k^{(i)} \log(\pi_k) \right]$$

$$= \sum_{i,k} E_{q_i(t)}[z_k^{(i)}] \log \mathcal{N}(x_i^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} E_{q_i(t)}[z_k^{(i)}] \log(\pi_k)$$

$$= \sum_{i,k} q_{ik}^{(t)} \log \mathcal{N}(x_i^{(i)}, \mu_k, \Sigma_k) + \sum_{i,k} q_{ik}^{(t)} \log(\pi_k)$$
Expectation step for the Gaussian mixture

We computed previously $q_i^{(t)}(z^{(i)})$, which is a multinomial distribution defined by

$$q_i^{(t)}(z^{(i)}) = p(z^{(i)} | x^{(i)}; \theta^{(t-1)})$$
Expectation step for the Gaussian mixture

We computed previously $q_i^{(t)}(z^{(i)})$, which is a multinomial distribution defined by

$$q_i^{(t)}(z^{(i)}) = p(z^{(i)} | x^{(i)}; \theta^{(t-1)})$$

Abusing notation we will denote $(q_{i1}^{(t)}, \ldots, q_{iK}^{(t)})$ the corresponding vector of probabilities defined by

$$q_{ik}^{(t)} = \mathbb{P}_{q_i^{(t)}}(z_k^{(i)} = 1) = \mathbb{E}_{q_i^{(t)}}[z_k^{(i)}]$$
Expectation step for the Gaussian mixture

We computed previously $q_i^{(t)}(z^{(i)})$, which is a multinomial distribution defined by

$$q_i^{(t)}(z^{(i)}) = p(z^{(i)} | x^{(i)}; \theta^{(t-1)})$$

Abusing notation we will denote $(q_{i1}^{(t)}, \ldots, q_{iK}^{(t)})$ the corresponding vector of probabilities defined by

$$q_{ik}^{(t)} = P_{q_i^{(t)}}(z_k^{(i)} = 1) = \mathbb{E}_{q_i^{(t)}}[z_k^{(i)}]$$

$$q_{ik}^{(t)} = p(z_k^{(i)} = 1 | x^{(i)}; \theta^{(t-1)}) = \frac{\pi_k^{(t-1)} \mathcal{N}(x^{(i)}, \mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{j=1}^K \pi_j^{(t-1)} \mathcal{N}(x^{(i)}, \mu_j^{(t-1)}, \Sigma_j^{(t-1)})}$$
Maximization step for the Gaussian mixture

\[
(\pi^t, (\mu_k^t, \Sigma_k^t)_{1 \leq k \leq K}) = \arg\max_\theta \mathbb{E}_{q(t)} [\tilde{\ell}(\theta)]
\]
Maximization step for the Gaussian mixture

\[
(\pi^t, (\mu_k^{(t)}, \Sigma_k^{(t)})_{1 \leq k \leq K}) = \arg\max_{\theta} \mathbb{E}_{q(t)} [\tilde{\ell}(\theta)]
\]

This yields the updates:

\[
\mu_k^{(t)} = \frac{\sum_i x^{(i)} q_{ik}^{(t)}}{\sum_i q_{ik}^{(t)}},
\]

\[
\Sigma_k^{(t)} = \frac{\sum_i (x^{(i)} - \mu_k^{(t)}) (x^{(i)} - \mu_k^{(t)})^\top q_{ik}^{(t)}}{\sum_i q_{ik}^{(t)}}
\]

and

\[
\pi_k^{(t)} = \frac{\sum_i q_{ik}^{(t)}}{\sum_i q_{ik}^{(t)}}
\]
Final EM algorithm for the Multinomial mixture model

Initialize $\theta = \theta_0$

**WHILE** (Not converged)

**Expectation step**

$$q_{ik}^{(t)} \leftarrow \frac{\pi_k^{(t-1)} \mathcal{N}(x^{(i)}, \mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{j=1}^{K} \pi_j^{(t-1)} \mathcal{N}(x^{(i)}, \mu_j^{(t-1)}, \Sigma_j^{(t-1)})}$$

**Maximization step**

$$\mu_k^{(t)} = \frac{\sum_i x^{(i)} q_{ik}^{(t)}}{\sum_i q_{ik}^{(t)}} , \quad \Sigma_k^{(t)} = \frac{\sum_i (x^{(i)} - \mu_k^{(t)}) (x^{(i)} - \mu_k^{(t)})^\top q_{ik}^{(t)}}{\sum_i q_{ik}^{(t)}}$$

and $$\pi_k^{(t)} = \frac{\sum_i q_{ik}^{(t)}}{\sum_{i,k'} q_{ik'}^{(t)}}$$

**ENDWHILE**
EM Algorithm for the Gaussian mixture model III

\[ p(x|z) \quad p(z|x) \]
Outline

1. The EM algorithm for the Gaussian mixture model

2. More examples of graphical models
Factorial Analysis

\[ Z_i \rightarrow \Lambda, \Psi \rightarrow X_i \]

- \( \Lambda \in \mathbb{R}^{d \times k} \) is the matrix of factors or principal directions.

\[ X_i \sim N(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \]

with \( \varepsilon_i \sim N(0, \Psi) \) with \( \Psi \in \mathbb{R}^{d \times d} \), constrained to be diagonal.

The model essentially retrieves Principal Component Analysis for \( \Psi = \sigma^2 I_d \).
Factorial Analysis

- \( \Lambda \in \mathbb{R}^{d \times k} \) is the matrix of factors or principal directions
- \( Z_i \in \mathbb{R}^k \) are the loadings or principal components

\[ Z_i \sim \mathcal{N}(0, I_k) \]
Factorial Analysis

\( \Lambda, \Psi \)

- \( \Lambda \in \mathbb{R}^{d \times k} \) is the matrix of factors or principal directions
- \( Z_i \in \mathbb{R}^k \) are the loadings or principal components
  \[
  Z_i \sim \mathcal{N}(0, I_k)
  \]
- \( X_i \in \mathbb{R}^d \) is the observed data modeled as
  \[
  X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi).
  \]
  with \( \Psi \in \mathbb{R}^{d \times d} \), constrained to be diagonal.
Factorial Analysis

\[ Z_i \xrightarrow{\Lambda, \Psi} X_i \]

- \( \Lambda \in \mathbb{R}^{d \times k} \) is the matrix of factors or principal directions
- \( Z_i \in \mathbb{R}^k \) are the loadings or principal components

\[ Z_i \sim \mathcal{N}(0, I_k) \]

- \( X_i \in \mathbb{R}^d \) is the observed data modeled as

\[ X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

with \( \Psi \in \mathbb{R}^{d \times d} \), constrained to be diagonal.

The model essentially retrieves Principal Component Analysis for \( \Psi = \sigma^2 I_d \).
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

\( \Lambda \) can be learned (up to a rotation on the right) together with \( \Psi \) using an EM algorithm, where \( Z \) is treated as a latent variable.
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

\( \Lambda \) can be learned (up to a rotation on the right) together with \( \Psi \) using an EM algorithm, where \( Z \) is treated as a latent variable.

**Advantages of the probabilistic formulation over vanilla PCA**

- Possible to model non-isotropic noise

- Possible to handle missing entries (treated as latent variables in EM)

- By changing the distributions on \( Z_i \) and \( X_i \), we can design variants of PCA more suitable for different types of data: Multinomial PCA, Poisson PCA, etc.

- Can be inserted in a mixture of Gaussians model to help model Gaussians in high dimension.
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

\( \Lambda \) can be learned (up to a rotation on the right) together with \( \Psi \) using an EM algorithm, where \( Z \) is treated as a latent variable.

**Advantages of the probabilistic formulation over vanilla PCA**

- Possible to model non-isotropic noise
- \( X \) can have missing entries (then treated as latent variables in EM)
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

\( \Lambda \) can be learned (up to a rotation on the right) together with \( \Psi \) using an EM algorithm, where \( Z \) is treated as a latent variable.

**Advantages of the probabilistic formulation over vanilla PCA**

- Possible to model non-isotropic noise
- \( X \) can have missing entries
  (then treated as latent variables in EM)
- By changing the distributions on \( Z_i \) and \( X_i \), we can design variant of PCA more suitable for different type of data: Multinomial PCA, Poisson PCA, etc.
Factorial Analysis

\[ Z_i \sim \mathcal{N}(0, I_k) \]

\[ X_i = \Lambda Z_i + \varepsilon_i \text{ with } \varepsilon_i \sim \mathcal{N}(0, \Psi). \]

\( \Lambda \) can be learned (up to a rotation on the right) together with \( \Psi \) using an EM algorithm, where \( Z \) is treated as a latent variable.

**Advantages of the probabilistic formulation over vanilla PCA**

- Possible to model non-isotropic noise
- \( X \) can have missing entries (then treated as latent variables in EM)
- By changing the distributions on \( Z_i \) and \( X_i \), we can design variant of PCA more suitable for different type of data: Multinominal PCA, Poisson PCA, etc.
- Can be inserted in a mixture of Gaussians model to help model Gaussians in high dimension.
Latent Dirichlet Allocation as Multinomial PCA

Replacing

- the distribution on $Z_i$ by a Dirichlet distribution
- the distribution of $X_i$ by a Multinomial
Latent Dirichlet Allocation as Multinomial PCA

Replacing

- the distribution on $Z_i$ by a Dirichlet distribution
- the distribution of $X_i$ by a Multinomial

Topic proportions for document $i$:

\[ \theta_i \in \mathbb{R}^K \]

\[ \theta_i \sim \text{Dir}(\alpha) \]

Empirical words counts for document $i$:

\[ x_i \in \mathbb{R}^d \]

\[ x_i \sim \mathcal{M}(N_i, B\theta_i) \]
Temporal models

Hidden Markov Model and Kalman Filter

![Diagram showing a sequence of hidden states (z_0, z_1, z_2, ..., z_T) and observations (y_0, y_1, y_2, ..., y_T).]
Temporal models

Hidden Markov Model and Kalman Filter

![Diagram of Hidden Markov Model and Kalman Filter](image)

Conditional Random Field (chain case)

![Diagram of Conditional Random Field](image)

- A structured version of *logistic regression* where the output is a sequence.
More temporal models

Second order auto-regressive model with latent switching state

\[ z_0 \rightarrow z_1 \rightarrow z_2 \rightarrow \ldots \rightarrow z_T \]

\[ y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow \ldots \rightarrow y_T \]
More temporal models

Second order auto-regressive model with latent switching state

Factorial Hidden Markov models (Ghahramani and Jordan, 1996)
Restricted Boltzman Machines (Smolensky, 1986)

\[ P(Y, Z) = \exp \left( \langle Y, \theta \rangle + Z^\top W Y + \langle Z, \eta \rangle - A(\theta, W, \eta) \right) \]

- \( p(Z|Y) = \prod_{i=1}^{d} p(Z_i|Y) \) are independent Bernoulli r.v.
- \( p(Y|Z) = \prod_{i=1}^{d} p(Y_i|Z) \) are independent Bernoulli r.v.

However the model encodes non-trivial dependences between the variables \((Y_1, \ldots, Y_n)\)
Ising model

Reminder: $X = (X_i)_{i \in V}$ is a vector of random variables, taking value in $\{0, 1\}^{|V|}$, whose distribution has the following exponential form:

$$p(x) = e^{-A(\eta)} \prod_{i \in V} e^{\eta_i x_i} \prod_{(i,j) \in E} e^{\eta_{i,j} x_i x_j}$$
Ising model

Reminder: $X = (X_i)_{i \in V}$ is a vector of random variables, taking value in $\{0, 1\}^{|V|}$, whose distribution has the following exponential form:

$$p(x) = e^{-A(\eta)} \prod_{i \in V} e^{\eta_i x_i} \prod_{(i,j) \in E} e^{\eta_{i,j} x_i x_j}$$

The associated log-likelihood is this:

$$\ell(\eta) = \sum_{i \in V} \eta_i x_i + \sum_{(i,j) \in E} \eta_{i,j} x_i x_j - A(\eta)$$
Hidden Markov Random Field
Hidden Markov random Field

\[ p(y|x) = e^{-A(\eta)} \prod_{i \in V} e^{\langle w, x_i \rangle y_i} \prod_{(i,j) \in E} e^{\eta_{i,j} y_i y_j} \]
Hidden Markov random Field

\[ p(y|x) = e^{-A(\eta)} \prod_{i \in V} e^{\langle w, x_i \rangle y_i} \prod_{(i,j) \in E} e^{\eta_{i,j} y_i y_j} \]

The associated log-likelihood is this:

\[ \ell(\eta) = \sum_{i \in V} \langle w, x_i \rangle y_i + \sum_{(i,j) \in E} \eta_{i,j} y_i y_j - A(w) \]
Hidden Markov random Field

\[ p(y|x) = e^{-A(\eta)} \prod_{i \in V} e^{\langle w, x_i \rangle y_i} \prod_{(i,j) \in E} e^{\eta_{i,j} y_i y_j} \]

The associated log-likelihood is this:

\[ \ell(\eta) = \sum_{i \in V} \langle w, x_i \rangle y_i + \sum_{(i,j) \in E} \eta_{i,j} y_i y_j - A(w) \]
References I
