Computational Intelligence Lab - Probability / K-Means / GMM tutorial

Maysam Haghdan & Matthias Hüser

ETH Zürich

April 26-27, 2018

Sample spaces and probabilities

- A *sample space* Ω is the set of outcomes of a random experiment.

- Subsets $A \subseteq \Omega$ are called *events*.

- For example, consider the experiment of tossing a fair coin twice.
  - Sample space: $\Omega = \{HH, HT, TH, TT\}$
  - Event of at least one "head" occurring: $A = \{HH, HT, TH\}$.

- A *probability distribution* is a function that assigns a real number $\Pr[A]$ to each event $A \subseteq \Omega$.

Random variables

- Usually, we do not deal directly with sample spaces. Instead, we define *random variables* and probability distributions on those.

- A random variable is a function $X : \Omega \to \mathbb{R}$.

- For example, if $X :=$ "the number of heads in two coin tosses", then

$$\begin{aligned} X(HH) &= 2 \\ X(HT) &= 1 \\ X(TH) &= 1 \\ X(TT) &= 0 \end{aligned}$$

Probabilities of random variables

- If we denote by $\mathcal{X}$ the set of values a random variable X can take, we can define probabilities directly on $\mathcal{X}$.

- In the above example, $\mathcal{X} = \{0, 1, 2\}$ and we define

$$\begin{aligned} \Pr[X = 0] &:= \Pr[\{TT\}] \\ \Pr[X = 1] &:= \Pr[\{HT, TH\}] \\ \Pr[X = 2] &:= \Pr[\{HH\}] \end{aligned}$$

- In practice, we often completely forget about the sample space and work only with random variables.

Discrete random variables

- X is called a *discrete random variable* if $\mathcal{X}$ is a finite or countably infinite set.

- Examples:
    - $\mathcal{X} = \{0, 1\}$
    - $\mathcal{X} = \mathbb{N}$
    - $\mathcal{X} = \mathbb{N}^d$

- The corresponding probability distribution

$$P(x) := \Pr[X = x]$$

  is called a *probability mass function*.

- Non-negativity: $P(x) \geq 0, \ \forall x \in \mathcal{X}$

- Normalization: $\sum_{x \in \mathcal{X}} P(x) = 1$

Continuous random variables

- X is called a *continuous random variable* if $\mathcal{X}$ is an uncountably infinite set.

- Examples:
  - $\mathcal{X} = [0, 1]$
  - $\mathcal{X} = \mathbb{R}$
  - $\mathcal{X} = \mathbb{R}^d$

- The corresponding probability distribution $p(x)$ is called a *probability density function*.

- Non-negativity: $p(x) \geq 0, \ \forall x \in \mathcal{X}$

- Normalization: $\int_{\mathcal{X}} p(x) dx = 1$

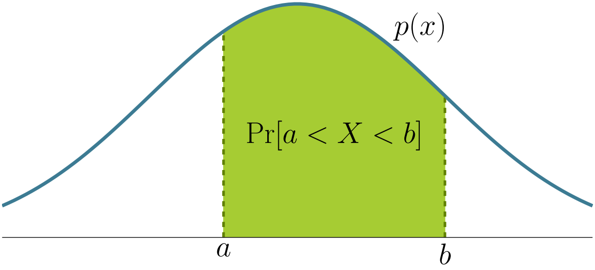
The meaning of density

- *Important:* For continuous random variables

$$p(x) \neq \Pr[X = x] = 0$$

- To acquire a probability, we have to integrate *p* over the desired set

$$\Pr[a < X < b] = \int_a^b p(x)dx$$



Joint distributions

- For two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, their *joint distribution* is defined as

$$P(x, y) := \Pr[X = x, Y = y]$$

- Non-negativity: $P(x, y) \geq 0$

- Normalization: $\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) = 1$

- For example, assume we throw two fair six-sided dice and define $X :=$ “the number on the first die” and $Y :=$ “the number on the second die”.
  - $\mathcal{X} = \mathcal{Y} = \{1, 2, 3, 4, 5, 6\}$
  - $P(6, 6) = \Pr[X = 6, Y = 6] = \frac{1}{36}$

Marginal and conditional distributions

Let $P(x, y)$ be a joint distribution of random variables X and Y.

- The *marginal distribution* of X is defined as

$$P(x) := \Pr[X = x] := \sum_{y \in \mathcal{Y}} P(x, y)$$

- The *conditional distribution* of X given that Y has a known value y is defined as

$$\begin{aligned} P(x|y) &:= \Pr[X = x | Y = y] \\ &:= \frac{P(x, y)}{P(y)} && \text{(defined if } P(y) > 0\text{)} \end{aligned}$$

- Note that for any fixed y, $P(x|y)$ is a distribution over x, i.e.

$$\sum_{x \in \mathcal{X}} P(x|y) = 1, \ \forall y \in \mathcal{Y}$$

Q1: Compute probabilities

A couple has two children, each of them being independently a boy or a girl with 50% probability. Compute the probabilities of the following events.

1. At least one of the children is a girl.
2. Both children are girls.
3. Both children are girls given that the first born is a girl.
4. Both children are girls given that one of them is a girl.
5. Both children are girls given that one of them is a girl named Cassiopeia. *Note: Cassiopeia is an extremely rare name with a frequency of less than 1 in 1,000,000.*

Q1a: At least one girl

Let's denote the i^{th} children by a random variable X_i, for $i \in \{1, 2\}$, taking values in the set $\{girl, boy\}$. We know that X_1 and X_2 are independent, and that for all $i \in \{1, 2\}$, for all $c \in \{girl, boy\}$, $\mathbb{P}(X_i = c) = \frac{1}{2}$.

1. The probability that at least one of them is a girl is given by

$$\mathbb{P}(\{X_1 = girl\} \cup \{X_2 = girl\})$$

$$= \mathbb{P}(\{X_1 = girl\}) + \mathbb{P}(\{X_2 = girl\}) - \mathbb{P}(\{X_1 = girl\} \cap \{X_2 = girl\}).$$

As the events are independent, we have

$$\mathbb{P}(\{X_1 = girl\} \cup \{X_2 = girl\})$$

$$= \mathbb{P}(\{X_1 = girl\}) + \mathbb{P}(\{X_2 = girl\}) - \mathbb{P}(\{X_1 = girl\}) \cdot \mathbb{P}(\{X_2 = girl\}),$$

i.e.

$$\mathbb{P}(\{X_1 = girl\} \cup \{X_2 = girl\}) = \frac{1}{2} + \frac{1}{2} - \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}.$$

The chain rule

- By definition of conditional distributions, we can *always* write a joint distribution of X and Y as a product of conditionals:

$$P(x, y) = P(x|y)P(y)$$

- We can do the same for an arbitrary number of random variables $X_1, \dots, X_n$:

$$P(x_1, \dots, x_n) = P(x_1|x_2, \dots, x_n) \dots P(x_{n-1}|x_n)P(x_n)$$

- Consistency of marginals and conditionals:

$$\begin{aligned}\sum_{y \in \mathcal{Y}} P(x, y) &= \sum_{y \in \mathcal{Y}} P(y|x)P(x) && \text{(chain rule)} \\ &= P(x) \sum_{y \in \mathcal{Y}} P(y|x) \\ &= P(x) && \text{(normalization)}\end{aligned}$$

Bayes' rule

- For two random variables *X* and *Y*, by definition of the conditional distribution of *X* given *Y*:

$$P(x|y) = \frac{P(x,y)}{P(y)}$$

- Also, by the chain rule:

$$P(x,y) = P(y|x)P(x)$$

- Combining the above we get Bayes' rule:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Q2: Diagnostic test Bayes

There is an uncommon disease that has infected 1% of the human population. Assume that we have a test for this disease that is positive on an infected person with probability 99% and negative on a healthy person also with probability 99%.

If my test comes out positive, what is the probability that I am infected?

Diagnostic Test Bayes

Let P (resp. N) denote the event being positive (resp. negative) at the test and I (resp. H) being ill (resp. healthy). We have $\mathbb{P}(P|I) = 0.99$, $\mathbb{P}(I) = 0.01$, $\mathbb{P}(N|H) = 0.99$. We want to find $\mathbb{P}(I|P)$. From Bayes' rule we have

$$\mathbb{P}(I|P) = \frac{\mathbb{P}(P|I)\mathbb{P}(I)}{\mathbb{P}(P)} = \frac{\mathbb{P}(P|I)\mathbb{P}(I)}{\mathbb{P}(P \cap I) + \mathbb{P}(P \cap H)}$$

$$= \frac{\mathbb{P}(P|I)\mathbb{P}(I)}{\mathbb{P}(I)\mathbb{P}(P|I) + \mathbb{P}(H)\mathbb{P}(P|H)},$$

i.e.

$$\mathbb{P}(I|P) = \frac{\mathbb{P}(P|I)\mathbb{P}(I)}{\mathbb{P}(P)}$$

$$= \frac{\mathbb{P}(P|I)\mathbb{P}(I)}{\mathbb{P}(I)\mathbb{P}(P|I) + (1 - \mathbb{P}(I))(1 - \mathbb{P}(N|H))} = \frac{0.99 \cdot 0.01}{0.01 \cdot 0.99 + (1 - 0.01) \cdot (1 - 0.99)} = \frac{1}{2}$$

Independence

- Two random variables X and Y are called *independent*, if knowing the value of X does not give any additional information about the distribution of Y (and vice versa):

$$\begin{aligned} P(x|y) &= P(x) \\ \Leftrightarrow P(y|x) &= P(y) \end{aligned}$$

- Equivalently, X and Y are independent if their joint distribution factorizes:

$$P(x,y) = P(x|y)P(y) = P(x)P(y)$$

IID

- IID $:=$ *I*ndependent and *I*dentically *D*istributed

- Random variables $X_1, ..., X_n$ are called IID if
  - Each of them has the same (marginal) distribution
  - They are mutually independent

- Note that if $X_1, ..., X_n$ are IID, then

$$\begin{aligned} P(x_1, ..., x_n) &= P(x_1)...P(x_n) \\ &= \prod_{i=1}^{n} P(x_i) \end{aligned}$$

Expectation

- The *expectation* of a random variable X is defined as

$$\mu_X := \mathrm{E}[X] := \sum_{x \in \mathcal{X}} x P(x)$$

- Note that the expectation $\mathrm{E}[X]$ is *not* the same as the most likely value $\max_{x \in \mathcal{X}} P(x)$.

- Can also be defined for a function f of X:

$$\mathrm{E}[f(X)] := \sum_{x \in \mathcal{X}} f(x) P(x)$$

Variance

- The *variance* of a random variable X is defined as

$$\mathrm{Var}[X] := \mathrm{E}[(X - \mu_X)^2] := \sum_{x \in \mathcal{X}} (x - \mu_X)^2 P(x)$$

- $\mathrm{Var}[X] \geq 0$

- The *standard deviation* of X is defined as

$$\sigma_X := \sqrt{\mathrm{Var}[X]}$$

Multidimensional moments

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a vector of random variables.

- The expectation of $\boldsymbol{X}$ is defined as

$$\mathrm{E}[\boldsymbol{X}] := (\mathrm{E}[X_1], \ldots, \mathrm{E}[X_n])$$

- The covariance of variables X_i and X_j is defined as

$$\mathrm{Cov}[X_i, X_j] := \mathrm{E}[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]$$

  - $\mathrm{Cov}[X_i, X_i] = \mathrm{Var}[X_i]$
  - X_i, X_j independent $\Rightarrow \mathrm{Cov}[X_i, X_j] = 0$
  - $\mathrm{Cov}[X_i, X_j] > 0$ roughly means that X_i and X_j increase and decrease together.
  - $\mathrm{Cov}[X_i, X_j] < 0$ roughly means that when X_i increases X_j decreases (and vice versa).

Covariance matrix

For a random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ we define its $n \times n$ *covariance matrix* as follows:

$$\Sigma_{\boldsymbol{X}} = \begin{bmatrix} \mathrm{Var}[X_1] & \mathrm{Cov}[X_1, X_2] & \cdots & \mathrm{Cov}[X_1, X_n] \\ \mathrm{Cov}[X_2, X_1] & \mathrm{Var}[X_2] & \cdots & \mathrm{Cov}[X_2, X_n] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_n, X_1] & \mathrm{Cov}[X_n, X_2] & \cdots & \mathrm{Var}[X_n] \end{bmatrix}$$

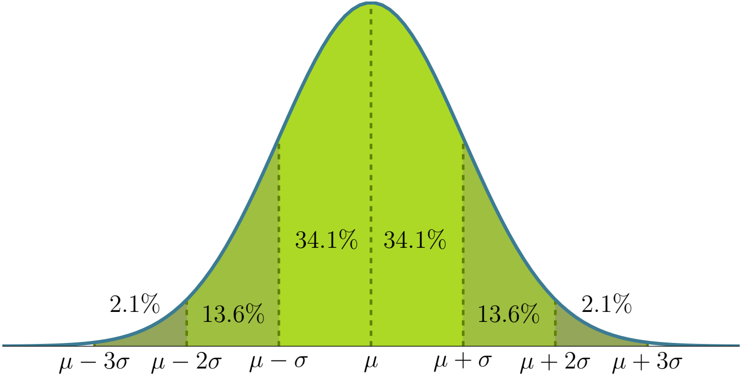
- The diagonal elements are the variances of each random variable $\mathrm{Cov}[X_i, X_i] = \mathrm{Var}[X_i]$.
- $\Sigma_{\boldsymbol{X}}$ is symmetric, because $\mathrm{Cov}[X_i, X_j] = \mathrm{Cov}[X_j, X_i]$.
- $\Sigma_{\boldsymbol{X}}$ is positive semi-definite.
- What does it mean if $\Sigma_{\boldsymbol{X}}$ is diagonal?

Gaussian distribution (1-D)

- Random variable X with $\mathcal{X} = \mathbb{R}$
- Probability density function

$$p(x) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- $\mathrm{E}[X] = \mu, \; \mathrm{Var}[X] = \sigma^2$



Gaussian Distribution (n-D)

- Random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ with $\mathcal{X} = \mathbb{R}^n$
- Probability density function

$$p(\boldsymbol{x}) := \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right)$$

- $\mathrm{E}[\boldsymbol{X}] = \boldsymbol{\mu}$
- Σ is the covariance matrix of $\boldsymbol{X}$ and $|\Sigma|$ is its determinant.

Data vs. distribution

- Be careful to distinguish between *models* (usually smooth parametric distributions) and *data* (sets of points).

- Machine learning:
  - Data = input
  - Distribution = model or assumption

- ML methods usually make some general assumptions about the distribution (e.g. a parametric family), then try to obtain ("infer") the specifics from the data available.

- Example:
  1. Modeling step: Assume a Gaussian distribution as model (parameterized by μ and σ).
  2. Inference step: Estimate parameters μ and σ from data.

The clustering problem

- Consider N data points in a D-dimensional space, i.e. each data point is a D-dimensional vector $\boldsymbol{x}_n$, $n = 1, \ldots, N$.

- Our goal is to partition the data set into K clusters.

- In other words, find K representative vectors (centroids) $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$, one for each cluster.

- Data point $\boldsymbol{x}_n$ belongs to cluster k if the Euclidean distance between $\boldsymbol{x}_n$ and $\boldsymbol{u}_k$ is smaller than the distance to any other centroid.

K-means cost function

Objective

Minimize the following cost function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\boldsymbol{x}_n - \boldsymbol{u}_k\|_2^2.$$

- Data points: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$
- Centroids: $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K \in \mathbb{R}^D$
- Assignments: $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \mathbb{R}^K$ (with $z_{k,n} := (\boldsymbol{z}_n)_k$)

Hard assignment constraints

Each point $\boldsymbol{x}_n$ is assigned to exactly one cluster:

- $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \{0, 1\}^K$
- $\sum_{k=1}^{K} z_{k,n} = 1, \ \forall n \in \{1, \ldots, N\}$

K-means algorithm

1. Initialize centroids $\boldsymbol{u}_1^{(0)}, \ldots, \boldsymbol{u}_K^{(0)}$ and $t \leftarrow 1$.
2. **Cluster assignment.**

$$k^*(\boldsymbol{x}_n) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmin}} \left\{ \|\boldsymbol{x}_n - \boldsymbol{u}_k^{(t-1)}\|_2^2 \right\}, \ \forall n \in \{1, \ldots, N\}$$

$$z_{j,n}^{(t)} = \begin{cases} 1 & \text{, if } j = k^*(\boldsymbol{x}_n) \\ 0 & \text{, otherwise} \end{cases}, \ \forall n \in \{1, \ldots, N\}$$

3. **Centroid update.**

$$\boldsymbol{u}_k^{(t)} = \frac{\sum_{n=1}^N z_{k,n}^{(t)} \boldsymbol{x}_n}{\sum_{n=1}^N z_{k,n}^{(t)}}, \ \forall k \in \{1, \ldots, K\}$$

4. If termination condition (e.g. $\|\boldsymbol{u}_k^{(t)} - \boldsymbol{u}_k^{(t-1)}\|_2^2 < \epsilon, \ \forall k$) is not met, $t \leftarrow t + 1$ and go to step 2.

K-means: Previous Exam Q

We are given a dataset of points $\{-2, 9, 1, -3, 6, 5, 4, 8\}$ in $\mathbb{R}$. Cluster this dataset using the *K*-means algorithm with $K = 2$, initialized at the two random clusters $C_1 = \{9, -2, 5, 8\}$ and $C_2 = \{6, 1, -3, 4\}$. Describe all steps carefully.

[BLACKBOARD]

Q1: Convergence

Show that the K-means algorithm always converges. In particular, consider the following cost function

$$J := \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2,$$

and show that steps 2 and 3 of the K-means algorithm from the lecture minimize this cost function for $\mathbf{z}_n$ and $\mathbf{u}_k$, respectively.

Recall the K-means algorithm

1. Initialize centroids $\boldsymbol{u}_1^{(0)}, \ldots, \boldsymbol{u}_K^{(0)}$ and $t \leftarrow 1$.
2. **Cluster assignment.**

$$k^*(\boldsymbol{x}_n) = \operatorname*{argmin}_{k \in \{1,\ldots,K\}} \left\{ \|\boldsymbol{x}_n - \boldsymbol{u}_k^{(t-1)}\|_2^2 \right\}, \ \forall n \in \{1, \ldots, N\}$$

$$z_{j,n}^{(t)} = \begin{cases} 1 & \text{, if } j = k^*(\boldsymbol{x}_n) \\ 0 & \text{, otherwise} \end{cases}, \ \forall n \in \{1, \ldots, N\}$$

3. **Centroid update.**

$$\boldsymbol{u}_k^{(t)} = \frac{\sum_{n=1}^N z_{k,n}^{(t)} \boldsymbol{x}_n}{\sum_{n=1}^N z_{k,n}^{(t)}}, \ \forall k \in \{1, \ldots, K\}$$

4. If termination condition (e.g. $\|\boldsymbol{u}_k^{(t)} - \boldsymbol{u}_k^{(t-1)}\|_2^2 < \epsilon, \ \forall k$) is not met, $t \leftarrow t + 1$ and go to step 2.

Convergence: Proof strategy

We are showing that the *k*-means algorithm converges, by arguing that each iteration either reduces or keeps the same the value of the objective function *J*, which is

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2 \quad \left(\|\mathbf{x}_n - \mathbf{u}_k\|_2^2 = (x_{1,n} - u_{1,k})^2 + \cdots + (x_{d,n} - u_{d,k})^2\right)$$

with the binary indicator constraint

$$\sum_{k=1}^{K} z_{k,n} = 1 \quad \text{and} \quad z_{k,n} \in \{0, 1\}.$$

Convergence: Cluster assignments

When initializing the algorithm & in each step 2 we set

$$z_{k^*(\mathbf{x}_n),n} = 1 \quad \text{and} \quad z_{k',n} = 0,$$

where

$$k^*(\mathbf{x}_n) = \underset{k}{\operatorname{argmin}} \left\{ \|\mathbf{x}_n - \mathbf{u}_1\|_2^2, \ldots, \|\mathbf{x}_n - \mathbf{u}_k\|_2^2, \ldots, \|\mathbf{x}_n - \mathbf{u}_K\|_2^2 \right\}.$$

This clearly minimizes J for fixed centroids, as we have to assign the value 1 to exactly one $z_{k,n}$, and 0 to all others.

Convergence: Centroid updates

In step 3, the centroid update term:

$$\mathbf{u}_k = \frac{\sum_{n=1}^{N} z_{k,n}\mathbf{x}_n}{\sum_{n=1}^{N} z_{k,n}} \;\forall k,\; k = 1, \ldots, K \tag{1}$$

is equivalent to the condition

$$0 = \sum_{n=1}^{N} z_{k,n}(\mathbf{x}_n - \mathbf{u}_k) \;\forall k,\; k = 1, \ldots, K$$

Convergence: Centroid updates

This is equivalent to setting the derivative of J with respect to $\mathbf{u}_k$ to zero for all k, $k = 1, \ldots, K$, as a particular derivative is given by:

$$\frac{\partial J}{\partial \mathbf{u}_k} = \frac{\partial \sum_{n=1}^{N} z_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2}{\partial \mathbf{u}_k} = \sum_{n=1}^{N} z_{k,n} \begin{bmatrix} \frac{\partial (x_{1,n} - u_{1,k})^2}{\partial u_{1,k}} \\ \vdots \\ \frac{\partial (x_{d,n} - u_{d,k})^2}{\partial u_{d,k}} \end{bmatrix} = -2 \sum_{n=1}^{N} z_{k,n} (\mathbf{x}_n - \mathbf{u}_k)$$

[DETAILS BLACKBOARD]

Considering all the above, it follows that repeating steps 2 and 3 in iterations means that the value of J will converge.

Q2: K-means as matrix factorization

Show that the K-means algorithm solves a matrix factorization problem, in the sense that

$$\arg\min_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{Z}\|_F^2 = \arg\min_{\boldsymbol{Z}} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\boldsymbol{x}_n - \boldsymbol{u}_k\|_2^2,$$

when $\boldsymbol{Z} \in \mathbb{R}^{K \times N}$ is additionally restricted to be an assignment matrix (having exactly a single non-zero entry of 1 in each column). The other matrices are given as follows:

- data matrix $\boldsymbol{X} := [\boldsymbol{x_1} \cdots \boldsymbol{x_N}] \in \mathbb{R}^{D \times N}$,
- centroid matrix $\boldsymbol{U} := [\boldsymbol{u_1} \cdots \boldsymbol{u_K}] \in \mathbb{R}^{D \times K}$,
- assignment matrix $\boldsymbol{Z} := [\boldsymbol{z_1} \cdots \boldsymbol{z_N}] \in \mathbb{R}^{K \times N}$.

K-means as matrix factorization

Notice that

$$\|X - UZ\|_F^2 = \sum_{i=1}^{D} \sum_{j=1}^{N} (x_{i,j} - \sum_{k=1}^{K} u_{i,k} z_{k,j})^2$$

$$= \sum_{i=1}^{D} \sum_{j=1}^{N} \left(\sum_{k=1}^{K} z_{k,j} (x_{i,j} - u_{i,k}) \right)^2 = \sum_{k=1}^{K} \sum_{j=1}^{N} z_{k,j}^2 \sum_{i=1}^{D} (x_{i,j} - u_{i,k})^2,$$

hence

$$\|X - UZ\|_F^2 = \sum_{k=1}^{K} \sum_{j=1}^{N} z_{k,j} \|\mathbf{x}_j - \mathbf{u}_k\|_2^2.$$

The second equality follows from the $\{z_{ij}\}$ are indicator variables with a normalization constraint. Exactly one term in the sums Σ_k is non-zero, so the sums can be freely rewritten, as they both are equal to $x_{ij} - u_{i,k^*}$ where $k*$ is the cluster to which the data point is assigned. **[DETAILS BLACKBOARD]**

Gaussian Mixture Models - Assumption

Assume data is generated from a weighted mixture of K Gaussian distributions:

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

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

Generative probabilistic model

K mixture components with parameters (for $k = 1, \ldots, K$):

- $\boldsymbol{\mu}_k$: mean of the k-th component (similar to centroid $\boldsymbol{u}_k$ in K-means)
- $\boldsymbol{\Sigma}_k$: covariance of the k-th component
- π_k: mixture weight of the k-th component

Gaussian Mixture models - Objective

Same task, different objective

The likelihood of all the data is:

$$p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

Maximize the log-likelihood of the Gaussian mixture model:

$$L(\boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \ln p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} .$$

Which is really hard to optimize with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

K-means vs. mixture models

- *K*-means
  - Hard cluster assignments
  - All clusters are the same (in terms of shape, weight, etc.)
  - Fast runtime (can be used to initialize a mixture model)

- Gaussian mixture models
  - Soft cluster assignments $\leftrightarrow$ probabilities of assigments
  - Each cluster has its own covariance ($\mathbf{\Sigma}_{\boldsymbol{k}}$) and “weight” ($\pi_k$)
  - Slower runtime

Mixture models: Recall for previous Exam Q

Suppose we have a set of N data points in d dimensions, $\mathbf{X} := (x_1, \ldots, x_N)$. We want to model this data using a mixture of K Gaussian distributions $\mathcal{N}(x|\mu_k, \Sigma_k)$. The Gaussian mixture model for one data point is thus defined as,

$$p_\theta(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k),$$

where $\theta = (\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$ are the model parameters.

Briefly explain the differences between K-means and Gaussian mixture model.

[DISCUSSION]

Q1: Log-likelihood

In this exercise we consider the problem of singularities when maximizing the likelihood of a Gaussian mixture model. Assume we are given a data set $\boldsymbol{X}$ consisting of N i.i.d observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and our goal is to cluster these observations using a mixture of K Gaussian distributions.

1. Write down the expression for the log-likelihood of the mixture model given data $\boldsymbol{X}$ (i.e., $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$).

Q1: Log-likelihood

The log-likelihood of the data is given by

$$\ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

Q1: Single point log-likelihood

Now, consider a Gaussian mixture model whose components have covariance matrices given by $\Sigma_k = \sigma_k^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. Write down the expression for the log-likelihood of the mixture model given $\boldsymbol{x}_n$ (i.e., $\ln p(\boldsymbol{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$).

Single point log-likelihood

For the data point x_n we have log-likelihood

$$\ln p(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

Q1: Component likelihood

Now, consider a Gaussian mixture model whose components have covariance matrices given by $\Sigma_k = \sigma_k^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. Compute the likelihood of the j-th mixture component given $\boldsymbol{x}_n$ (i.e. $\mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \Sigma_j)$).

Component likelihood

Computing the likelihood assuming $\boldsymbol{\mu}_j = \mathbf{x}_n$ leads to

$$\begin{aligned} p(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) &= \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{\sigma_j^D} \end{aligned} \tag{2}$$

This follows from plugging in the particular form of the multivariate normal distribution and its isotropic co-variance.

Q1: Degeneracy

Now, consider a Gaussian mixture model whose components have covariance matrices given by $\boldsymbol{\Sigma}_k = \sigma_k^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. What happens to the likelihood of the previous question as $\sigma_j \to 0$? How does this affect the log-likelihood of the mixture model given in question 1?

Degeneracy

As $\sigma_j \to 0$, goes to infinity and so the *likelihood function* will also go to infinity. Thus the maximization of the log likelihood function is not a well posed problem and causes the convergence to be very slow. Note that the other data-points have non-zero likelihood in the other components

[DISCUSSION ON BLACKBOARD]

Q1: Single Gaussian

Now, consider a Gaussian mixture model whose components have covariance matrices given by $\boldsymbol{\Sigma}_k = \sigma_k^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. Can the above situation occur when the mixture model consists of a single Gaussian distribution, i.e. $K = 1$?

Single Gaussian component

The other data-points are forced to be assigned to the de-generat Gaussian (an infinite spike centered at a data-point). The likelihood of these data-points will go to zero exponentially fast, giving an overall likelihood that tends to zero rather than infinity.

[DETAILS ON BLACKBOARD]

Once we have (at least) two components in the mixture, one of the components can have a finite variance and assigns finite probability to the other data-points.

Q1: Heuristic

Now, consider a Gaussian mixture model whose components have covariance matrices given by $\Sigma_k = \sigma_k^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. Can you propose a heuristic to avoid such situations?

Heuristic

We can hope to avoid the singularities by using suitable heuristics, for instance by detecting when a Gaussian component is collapsing and resetting its mean to a randomly chosen value while also resetting its covariance to some large value, and then continuing with the optimization.

Q2: Identifiability

1. Suppose that we have solved a mixture of K Gaussians problem, and have obtained the values of the parameters. How many equivalent solutions are there?

2. This problem is known as *identifiability*. Explain why this is not a problem in the context of data clustering.

Identifiability

1. For any given maximum likelihood solution, a K-component mixture will have a total of $K!$ equivalent solutions corresponding to the $K!$ ways of assigning K sets of parameters to K components.
2. Because any of the equivalent solutions is as good as any other. Using any permutation of these parameters leads to the same clustering with permuted cluster indices.

[DETAILS ON BLACKBOARD]

Further reading

References

- McLachlan, G.J.; Peel, D. (2000). Finite Mixture Models. Wiley.
- Phillips, Steven J. Mount, David M.; Stein, Clifford; Acceleration of K-Means and Related Clustering Algorithms. (2002) Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Dellaert, Frank.; The Expectation Maximization Algorithm. (2002) CiteSeerX 10.1.1.9.9735

Credits

Partially based on slides by Gary Becigneul, Paulina Grnarova, Andrew Bian.

GMM: Introduce Latent Variable

$\{\pi_1, \cdots, \pi_K\}$ can be viewed as the probability of a series of "latent variables" $\boldsymbol{z} = (z_1, \cdots, z_K)^T$ where $z_k \in \{0, 1\}$, $\sum_{k=1}^K z_k = 1$ and

$$p(z_k = 1) = \pi_k \qquad 1 \leq k \leq K$$

The distribution of $\boldsymbol{z}$ is of the form:

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

The conditional distribution of $\boldsymbol{x}$ given a particular value of Z is a Gaussian:

$$p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Then

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

GMM: Latent Variable

Define γ_{nk} as the posterior probability of $z_k = 1$ given $\boldsymbol{x}_n$:

$$\gamma_{nk} = p(z_k = 1 | \boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n | z_k = 1) p(z_k = 1)}{\sum_{q=1}^{K} p(\boldsymbol{x}_n | z_q = 1) p(z_q = 1)}$$

Remember that

$$\pi_k = p(z_k = 1)$$

Then rewrite:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}$$

So we can compute γ if we know $\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.

The EM algorithm - Overview

1. Initialize $\pi_k^{(0)}$, $\boldsymbol{\mu}_k^{(0)}$, $\boldsymbol{\Sigma}_k^{(0)}$ for $k = 1, \dots, K$ and $t \leftarrow 1$.
2. **E-step.** Evaluate responsibilities using current parameters:

$$\gamma_{nk} := \frac{\pi_k^{(t-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{j=1}^{K} \pi_j^{(t-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j^{(t-1)}, \boldsymbol{\Sigma}_j^{(t-1)})}$$

3. **M-step.** Update parameters using new responsibilities:

$$\begin{aligned}
\boldsymbol{\mu}_k^{(t)} &:= \frac{\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} q_{kn}} \\
\boldsymbol{\Sigma}_k^{(t)} &:= \frac{1}{\sum_{n=1}^{N} \gamma_{nk}} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})^T \\
\pi_k^{(t)} &:= \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}
\end{aligned}$$

4. If termination condition is not met, $t := t + 1$ and go to step 2.

EM for GMM

Our goal is to maximize:

$$L(\boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)$$

Take the derivative with respect to $\boldsymbol{\mu}_k$ and set it to zero:

$$-\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

$$-\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_{nk}}$$

So we can compute $\boldsymbol{\mu}$ if we know γ.

EM for GMM

Similarly:

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_{n=1}^{N} \gamma_{nk}} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

So we can compute both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ if we know γ.
What about π?

EM for GMM

Include constraints with a Lagrange multiplier

$$\sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$$

Take the derivative with respect to π_k and set it to zero:

$$\sum_{n=1}^{N} \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} + \lambda = 0$$

Multiply both parts by π_k and sum it up for all k:

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} + \lambda\left(\sum_{k=1}^{K} \pi_k\right) = 0$$

$$N + \lambda = 0$$

EM for GMM

Put $\lambda = -N$

$$\sum_{n=1}^{N} \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} - N = 0$$

$$\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{q=1}^{K} \pi_q \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} = N\pi_k$$

$$\sum_{n=1}^{N} \gamma_{nk} = N\pi_k$$

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_{nk}}{N}$$

So we can compute $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π if we know γ.

EM for GMM

What we have so far:

- We can compute γ if we know $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π
- We can compute $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π if we know γ

Idea:

- Apply coordinate-wise optimization

EM-algorithm:

- Estimate probabilities γ with fixed $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π (E-step)
- Maximize likelihood with respect to $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π for a given γ (M-step)
- Iterate until convergence

EM: Lower Bounding Log-Likelihood

- Expectation Maximization
  - maximize a lower bound on the log-likelihood
  - based on complete data distribution

- Specifically:

$$\log p_\theta(\mathbf{x}) = \log \left[\sum_{k=1}^{K} \pi_k \, p_{\theta_k}(\mathbf{x}) \right] = \log \left[\sum_{k=1}^{K} q_k \frac{\pi_k \, p_{\theta_k}(\mathbf{x})}{q_k} \right]$$

$$\geq \sum_{k=1}^{K} q_k \left[\log p_{\theta_k}(\mathbf{x}) + \log \pi_k - \log q_k \right]$$

  - follows from Jensen's inequality (concavity of logarithm)
  - can be done for the contribution of each data point (additive)

EM: Expectation Step

- Optimize bound with regard to the distribution q
  - formulate Lagrangian (decoupled for each data point)

$$\max_q \left\{ \sum_{k=1}^K q_k \left[\log p_{\theta_k}(\mathbf{x}) + \log \pi_k - \log q_k\right] + \lambda \left(\sum_{k=1}^K q_k - 1 \right) \right\}$$

  - first order optimality condition (setting gradient to zero):

$$\log p_{\theta_k}(\mathbf{x}) + \log \pi_k - \log q_k - 1 + \lambda \overset{!}{=} 0 \iff$$

$$q_k^* = \frac{\pi_k \, p_{\theta_k}(\mathbf{x})}{\sum_{l=1}^K \pi_l \, p_{\theta_l}(\mathbf{x})} = \Pr(z_k = 1 \mid \mathbf{x})$$

  - optimal q–distribution equals posterior (given the parameters)
  - E-step selects the best lower bound on the log-likelihood (making the inequality tight)

EM: Maximization Step

- Maximizing expected complete data log-likelihood with regard to the model parameters θ

- Equivalent to maximizing the lower bound with respect to the model parameters

- Since the model parameters change, the inequality becomes loose again, so the log likelihood ($\log p_\theta(\mathbf{x})$) must increase.

EM in general

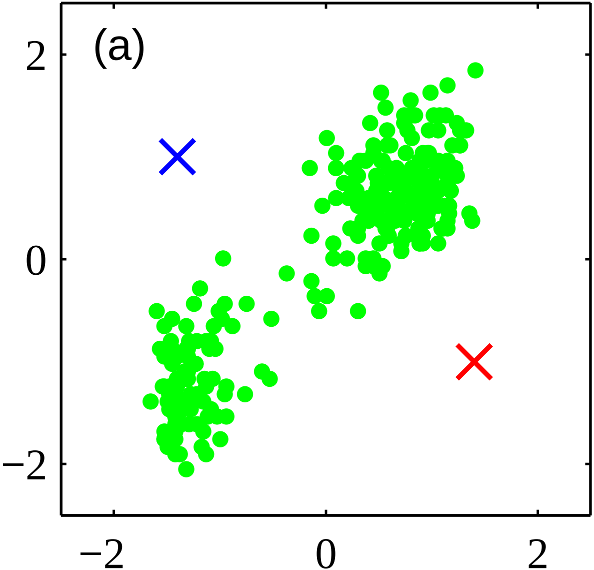
General idea:

- Introduce probabilistic hidden variables such that likelihood is easy to maximize if their (posterior) probabilities are known
- Iterate between estimation of the probabilities and maximization of the complete-data likelihood
- Target function increases at every step until convergence

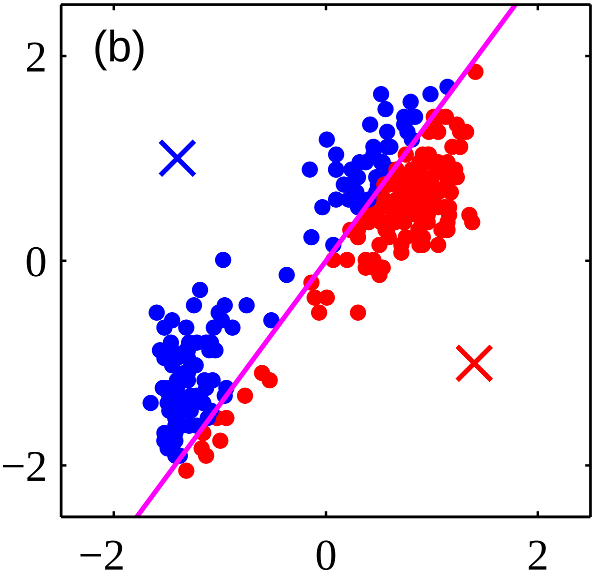
Applications:

- Not only GMM, almost any statistical model

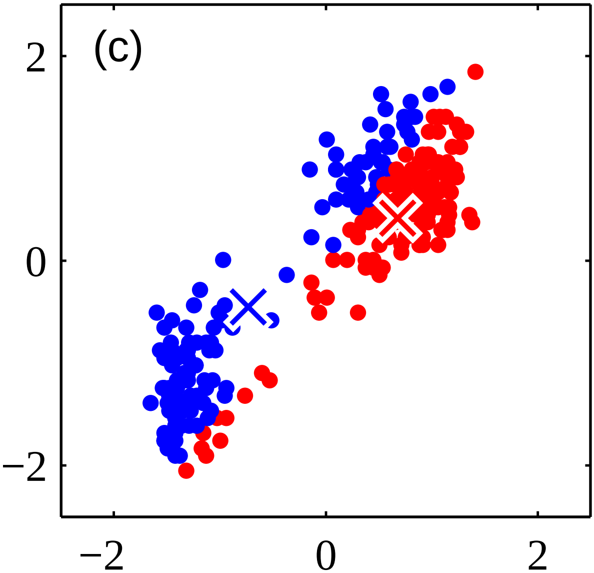
K-Means: Initial configuration



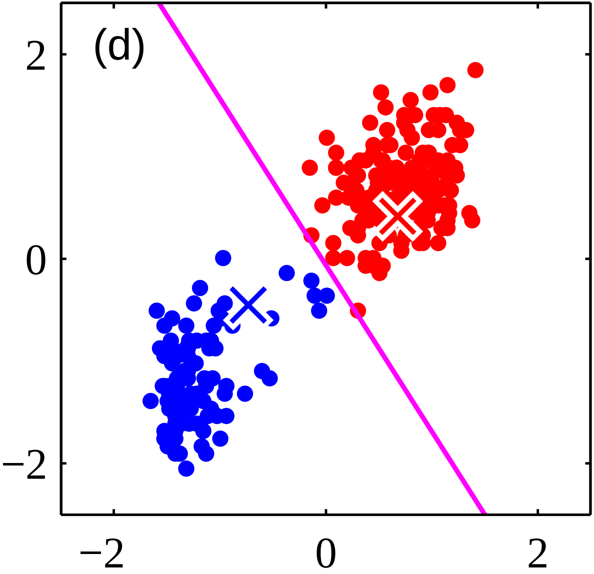
K-Means: First E-step



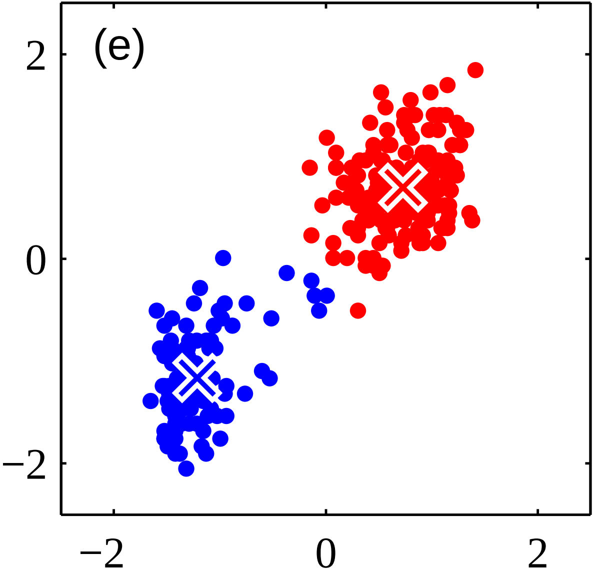
K-Means: First M-Step



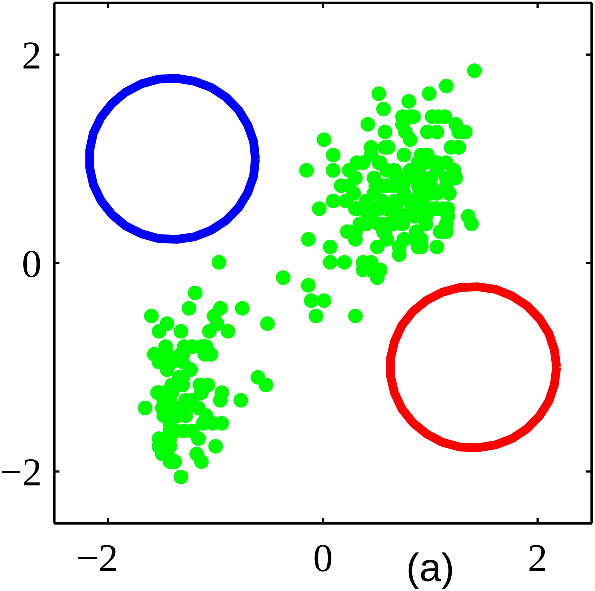
K-Means: Second E-Step



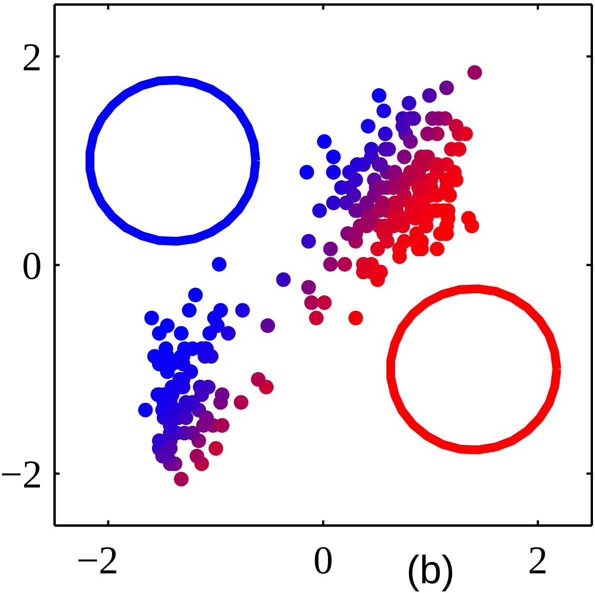
K-Means: Second M-Step



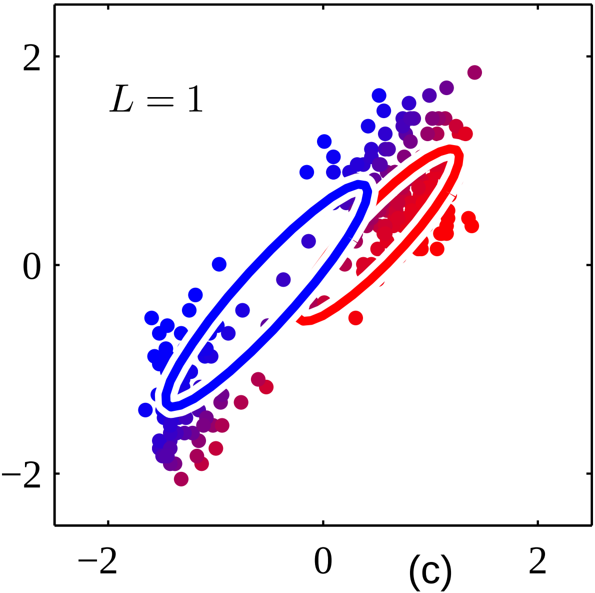
GMM: Initial configuration



GMM: First E-Step

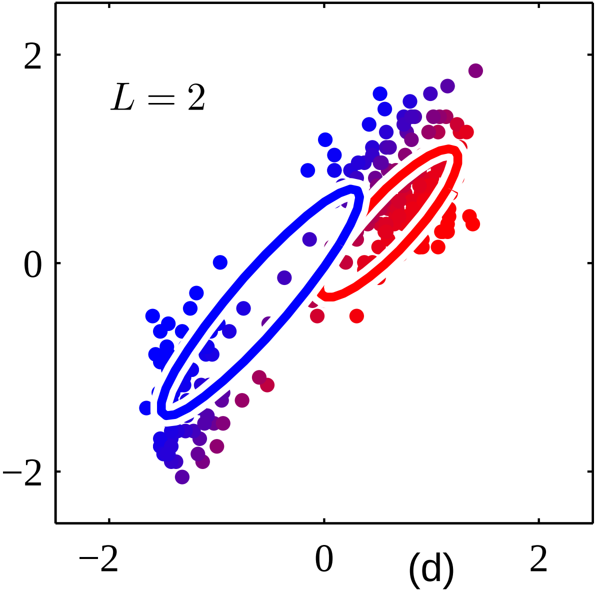


GMM: First M-Step



(c)

GMM: Two EM cycles



GMM: Five EM cycles