Machine Learning, Lecture 3: Clustering II

S. Nõmm

[1]Department of Software Science, Tallinn University of Technology

21.02.2017

Reminder

The goal is to cluster the data into K clusters, whereas no labeled data are given.

- Case of unsupervised learning.
- K is the hyperparameter.

Probability *versus* Likelihood

- **Data is fixed:** How likely certain set of parameters will result given data set.
- **Parameters are fixed:** What is the probability of drawing given data set with the given set of parameters.

Maximal likelihood estimate

Sometimes referred as maximal likelihood principle.
More formally

- $$\mathcal{L}(\theta \mid x) = P(x \mid \theta)$$
- The goal is to find parameters that maximize the likelihood.
- In many cases natural logarithm of the likelihood function is more easy to deal with. Introduce log-likelihood.

Sufficient statistics

Definition

A statistic $T(X)$ is sufficient for the parameter θ if the conditional probability distribution of the data X, given the statistic $T(x)$ does not depend on the parameter θ

$$P(X = x \mid T(X) = t, \theta) = P(X = x \mid T(X) = t).$$

- A statistic is *sufficient* for a family of probability distributions if the sample from which it was calculated gives no additional information.
- In other words. The value of the *sufficient* statistic (for the parameter) contains all the necessary information to calculate estimate of the parameter.

Gaussian

- One-dimensional
  - Do you remember a bell shaped curve?
  - Parameterized by mean μ and variance σ^2
  - Probability density function (pdf):

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x-\mu)^2}{2\sigma^2}$$

- D-dimensional: Parameterized by mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ.

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}} \mid \Sigma \mid^{1/2} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right]$$

- Derive for the 2- and 3- dimensional cases.

Fitting a Gaussian

Let us suppose, that a sample of n points $\boldsymbol{X} = (x_1, \dots, x_n)^T$ were independently drawn from some Gaussian.
The goal is to find the mean and the variance of the Gaussian.
(Fitting the Gaussian model to the data.)

- Sample mean is used as the estimate of the mean for the Gaussian
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$
- sample variance is used as the estimate of the variance of the Gaussian
$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Why such estimates are correct?

Example

Consider one dimensional Gaussian: Let us suppose that data points in the sample are drawn independently then the probability of data is:

$$P(\boldsymbol{X} \mid \mu, \sigma^2) = \prod_{i=1}^{n} P(x_i \mid \mu, \sigma^2)$$

$$= \ldots = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

As a next step: compute log - likelihood

$$\log P(\boldsymbol{X} \mid \mu, \sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log \sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

Example

$$\log P(\boldsymbol{X} \mid \mu, \sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

The last term

$$\sum_{i=1}^{n}(x_i - \mu)^2 = \sum_{i-1}^{n} x_i^2 - 2\mu\sum_{i=1}^{n} x_i + n\mu^2$$

Likelihood depends on the sample only through $\sum_{i-1}^{n} x_i^2$ and $\sum_{i=1}^{n} x_i$ which are sufficient statistics in this case.

Estimate of the mean μ

Find the partial derivative with respect to μ:

$$\frac{\partial \log P(\boldsymbol{X} \mid \mu\sigma^2)}{\partial \mu} = \frac{1}{\sigma^2}\Big(\sum_{i=1}^{n} x_i - n\mu\Big)$$

Solve the following equation with respect to μ.

$$\frac{1}{\sigma^2}\Big(\sum_{i=1}^{n} x_i - n\mu\Big) = 0 \Rightarrow \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Estimate of the variance σ^2

Find the partial derivative with respect to σ^2:

$$\frac{\partial P(\boldsymbol{X} \mid \mu, \sigma^2)}{\partial \sigma^2} = \frac{1}{2\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2 - \frac{n}{2\sigma^2}$$

Solve the following equation with respect to σ^2

$$\frac{1}{2\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2 - \frac{n}{2\sigma^2} = 0 \Rightarrow \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2.$$

Multivariate case

- Mean estimate

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

- Sample covariance

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T.$$

Latent Variable Models

Latent Variable Models (**LVM**) - models with hidden variables.
An important assumption is that observed variables are correlated because they arise from a hidden common "cause". Let $z_{i,1}, \ldots, z_{i,L}$ are L latent variables, and $x_{i,1}, \ldots, x_{i,D}$ are D visible variables.
The form of the likelihood $\mathcal{L}(x_i \mid z_i)$ and the prior $p(z_i)$ defines the model.

Mixture models

Let $z_i = \{1, \ldots, K\}$, - discrete latent states.

$$\begin{aligned} p(z_i) &= \text{Cat}(\pi) \\ \mathcal{L}(x_i \mid z_i = k) &= p_k(x_i) \end{aligned}$$

Overall model is known as *Mixture model* (we are mixing together K base distributions)

$$p(x_i \mid \theta) = \sum_{k=1}^{K} \pi_k p_k(x_i \mid \theta)$$

where mixed weights π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

Mixture of Gaussians

Mixture of Gaussian (MOG) is the most widely used mixture model. Each base distribution is a multivariate Gaussian with mean μ_k and covariance matrix Σ_k

$$p(x_i \mid \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$$

Mixture of Gaussians

- Latent variables z_i : $\quad z_i = k$ component k generated point x_i.
- $p(z_i = k \mid \pi) = \pi_k$ - probability of being generated by a component.
- $p\big(\boldsymbol{x_i} \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) = \mathcal{N}\big(\boldsymbol{x_i} \mid \boldsymbol{\mu}_k, \sigma_k\big)$ - probability of a given point whereas it is known which component generated it.
- $p\big(\boldsymbol{x_i}, z_i = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) = \pi_k \mathcal{N}\big(\boldsymbol{x_i} \mid \boldsymbol{\mu}_k, \Sigma_k\big)$ - joint probability of generating the component and the point from it.
- $p\big(\boldsymbol{x_i} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x_i} \mid \boldsymbol{\mu}_k, \Sigma_k\big)$ - *marginal probability* of the point.

Parameter estimation for Gaussian Mixture Models

- The goal is to estimate parameters:
  $\boldsymbol{\pi}, \boldsymbol{\mu_k}, \Sigma_k, \quad k = 1, \ldots, K$
- The log-likelihood function of GMM is

$$\log p\big(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) = \sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x_i} \mid \boldsymbol{\mu_k}, \Sigma_k)\Big)$$

- Possible problems:
  - Unidentifiability: K-component mixture has $K!$ possible labeling therefore there is no unique maximal likelihood estimate and in turn no unique maximum a posterior estimate.
  - Summation inside the logarithm ... .

Observe

- The knowledge of component parameters and mixing proportions allows to compute the probability that the component k responsible [1] for the i-th point $p(z_i = k \mid \boldsymbol{x_i}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- The knowledge of the responsibilities allows to compute the estimates for the mixing coefficients π_k.
- The knowledge of responsibilities and mixing coefficients allows to compute the estimates for component means μ_k and variances Σ_k

This leads the idea of two step iterative algorithm:

- **Step E:** Inferring the missing values given the parameters.
- **Step M:** Optimization of the parameters given the "filled data".

[1] Responsibility of the cluster k for point i is the posterior probability that point i belongs to cluster k, $p(z_i = k \mid \boldsymbol{x_i}, \boldsymbol{\theta})$

EM-algorithm

Let us consider K-Means from the probabilistic point of view.

- (E-step) Each data point of the set $\mathcal{D}$ has a probability belonging to cluster j, which is proportional to the scaled and exponentiated Euclidean distance to each representative Y_j . In the k-means algorithm, this is done in a "hard" way, by choosing the smallest Euclidean distance to the representative of Y_j.
- (M-step) The center Y_j is the weighted mean over all the data points where the weight is defined by the probability of assignment to cluster j. The hard version of this is used in k-means, where each data point is either assigned to a cluster or not assigned to a cluster (i.e., 0-1 probabilities).

EM-algorithm

Assumption: the data was generated from a mixture of k distributions with probability distributions $\mathcal{G}_1 \dots \mathcal{G}_k$. Each distribution $\mathcal{G}_i$ represents a cluster and is also referred to as a mixture component.

- (E-Step) Given the current value of the parameters in , estimate the posterior probability $P(\mathcal{G}_i|X_j, \Theta)$ of the component $\mathcal{G}_i$ having been selected in the generative process, given that we have observed data point X_j . The quantity $P(\mathcal{G}_i|X_j, \Theta)$ is also the soft cluster assignment probability that we are trying to estimate. This step is executed for each data point X_j and mixture component G_i.
- (M-Step) Given the current probabilities of assignments of data points to clusters, use the maximum likelihood approach to determine the values of all the parameters in Θ that maximize the log-likelihood fit on the basis of current assignments.

EM-algorithm implementation

In order to avoid confusion let us simplify the notation.

- Intialization
  - Randomly select the data points to use the means
  - Set the covariance matrix for each cluster to be equal to covariance matrix of the full training set.
  - Give each cluster equal prior probabilities φ_j
- Expectation: Calculate the probability that each data point belongs to each cluster. Remind how to compute the probability density function:

$$g_j(x) = \frac{1}{(2\pi)^n |\Sigma_j|} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)}$$

then the probability of a given point to belong to cluster j is given by

$$w_j^{(i)} = \frac{g_j(x)\varphi_j}{\sum_{l=1}^{k} g_l(x)\varphi_l}$$

EM-algorithm implementation

- Maximization: update rules:

$$\varphi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)},$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$