Clustering (1)

- Common technique for statistical data analysis to detect structure (machine learning, data mining, pattern recognition, ...)
- Classification of a data set into subsets (clusters)
- Ideally, data in each subset have a similar characteristics (proximity according to distance function)
Clustering (2)

- Needed: Distance function (similarity / dissimilarity), e.g., Euclidian distance
- Clustering quality
  - Inter-clusters distance maximized
  - Intra-clusters distance minimized
- The quality depends on
  - Clustering algorithm
  - Distance function
  - The application (data)
Types of Clustering

- Hierarchical Clustering
  - Agglomerative Clustering (bottom up)
  - Divisive Clustering (top-down)

- Partitional Clustering
  - K-Means Clustering (hard & soft)
  - Gaussian Mixture Models (EM-based)
K-Means Clustering

- Partitions the data into $k$ clusters ($k$ needs to be specified by the user)
- Find $k$ reference vectors $m_j$, $j = 1, \ldots, k$ which best explain the data $X$
- Assign data vectors to nearest (most similar) reference $m_i$

$$\|x' - m_i\| = \min_j \|x' - m_j\|$$

$r$-dimensional data vector in a real-valued space
reference vector (center of cluster = mean)
Reconstruction Error
(K-Means as a Compression Algorithm)

- The total reconstruction error is defined as

\[ E(\{m_i\}_{i=1}^k|X) = \sum_t \sum_i b_i^t \| x^t - m_i \|^2 \]

with

\[ b_i^t = \begin{cases} 
1 & \text{if } \| x^t - m_i \| = \min_j \| x^t - m_j \| \\
0 & \text{otherwise}
\end{cases} \]

- Find reference vectors which minimize the error

- Taking its derivative with respect to \( m_i \) and setting it to 0 leads to

\[ m_i = \frac{\sum_t b_i^t x^t}{\sum_t b_i^t} \]
K-Means Algorithm

Initialize $m_i, i = 1, \ldots, k$, for example, to $k$ random $x^t$
Repeat

For all $x^t \in \mathcal{X}$
$$b_i^t \left\{ \begin{array}{ll}
1 & \text{if } \|x^t - m_i\| = \min_j \|x^t - m_j\| \\
0 & \text{otherwise}
\end{array} \right.$$ 

For all $m_i, i = 1, \ldots, k$
$$m_i \leftarrow \frac{\sum_t b_i^t x^t}{\sum_t b_i^t}$$

Until $m_i$ converge

Recompute the cluster centers $m_i$ using current cluster membership

Assign each $x^t$ to the closest cluster
K-Means Example

Image source: Alpaydin, Introduction to Machine Learning
Strength of K-Means

- Easy to understand and to implement
- Efficient $O(nkt)$
  $n = \#\text{iterations}, k = \#\text{clusters}, t = \#\text{data points}$
- Converges to a local optimum
  (global optimum is hard to find)
- Most popular clustering algorithm
Weaknesses of K-Means

- User needs to specify \#clusters \((k)\)
- Sensitive to initialization
  (strategy: Use different seeds)
- Sensitive to outliers since all data points contribute equally to the mean
  (strategy: Try to eliminate outliers)
Soft Assignments

- So far, each data point was assigned to exactly one cluster
- A variant called soft k-means allows for making fuzzy assignments
- Data points are assigned to clusters with certain probabilities
Soft K-Means Clustering

- Each data point is given a soft assignment to all means
  
  \[ c_{tk} = \frac{\exp(-\beta \|x^t - m_k\|^2)}{\sum_i \exp(-\beta \|x^t - m_i\|^2)}, \quad \sum_k c_{tk} = 1 \]

- \( \beta \) is a “stiffness” parameter and plays a crucial role

- Means are updated
  
  \[ m_k = \frac{\sum_t c_{tk} x^t}{\sum_t c_{tk}} \]

- Repeat assignment and update step until assignments do not change anymore
Soft K-Means Clustering

- Points between clusters get assigned to both of them
- Points near the cluster boundaries play a partial role in several clusters
- Additional parameter $\beta$
- Clusters with varying shapes can be treated in a probabilistic framework (mixtures of Gaussians)
Similarity of Soft K-Means and Expectation Maximization (EM)

- Goal of EM: Find the model parameters that maximize the likelihood of the dataset given the model

- Two sets of random variables
  - Observed data set $d$
  - Hidden variables $c$
    - (assignment of data points to clusters)

- Since the joint likelihood (incl. the hidden variables!) cannot be determined, work with its expectation
Expectation (Expected Value)

- ... is the integral of the random variable with respect to its probability measure
- For discrete random variables this is equivalent to the probability-weighted sum of the possible values

\[ E[x] = E_x[x] = \int_x x p(x) \, dx \]
\[ E_{x,y}[x] = \int_{x,y} x p(x, y) \, dx \, dy \]
\[ E_{x|y}[x] = \int_x x p(x \mid y) \, dx \]
\[ E_{x|y}[g(x)] = \int_x g(x)p(x \mid y) \, dx \]
Expected Data Likelihood

- Observed data \( d = \{d_1, \ldots, d_I\} \)

- Correspondence variables (hidden)
  \( c = \{c_1, \ldots, c_I\} \)

- Joint likelihood of \( d \) and \( c \) given model \( \theta \)
  \[
P(d, c \mid \theta) = \prod_{i=1}^{I} P(d_i, c_i \mid \theta)
  \]
  \[
  \ln P(d, c \mid \theta) = \sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta)
  \]

- Since the values of \( c \) are hidden, optimize the expected value of the log likelihood
  \[
  E_c[\ln P(d, c \mid \theta) \mid \theta, d] = E_c[\sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta) \mid \theta, d]
  \]
Expectation Maximization

- Optimizing the expected log likelihood is usually not easy
- EM iteratively maximizes log likelihood functions
- EM generates a sequence of models $\theta^{[1]}, \theta^{[2]}, \ldots$ of increasing log likelihood
- EM converges to a (local) optimum
Expectation Maximization

- Use so-called Q-function to find the model with the maximum expected data likelihood

- Estimation (E) step
  - Compute expected values for \( c \) given current model \( \theta^{[j]} \)
  - Define the expected data log likelihood as a function of \( \theta \)
    \[
    Q(\theta \mid \theta^{[j]}) = E_c[\ln P(d, c \mid \theta) \mid \theta^{[j]}, d]
    \]

- Maximization (M) step
  - Maximize this expected likelihood
    \[
    \theta^{[j+1]} = \arg\max_{\theta'} Q(\theta' \mid \theta^{[j]})
    \]
Iterating E and M Steps

M-step:
Compute new model components given the expectations

E-step:
Compute the expectations given the current model

random initial model

model
Application: Trajectory Clustering

How to learn typical motion patterns of people from observations?
Application: Trajectory Clustering

- **Input:** Set of trajectories \(d_1, \ldots, d_I\)

\[d_i = \{x_i^1, x_i^2, \ldots, x_i^T\}\]
What we are looking for

- Clustering of similar trajectories into motion patterns $\theta_1, \ldots, \theta_M$
  (Note: From now on $M = \#\text{clusters}$)

- Binary correspondence variables $c_{im}$ indicating which trajectory $s_i$ belongs to which motion pattern $\theta_m$

Problem:

How can we estimate $c_{im}$?
Motion Patterns

- Use $T$ Gaussians with fixed variance to represent each motion pattern.
- If we knew the values of the $c_{im}$, the computation of the motion patterns would be easy.
- But: These values are hidden.
- Use EM to compute:
  - expected values for the $c_{im}$
  - the model $\theta$ (i.e., the set of motion patterns) which has the highest expected data likelihood.
Likelihood of Trajectory $d_i$ given Motion Pattern $\theta_m=\{\theta_1^m, \ldots, \theta^T_m\}$

$$P(d_i \mid \theta_m) = \prod_{t=1}^T P(x_i^t \mid \theta^t_m)$$

$$= \prod_{t=1}^T \exp\left(-\frac{1}{2\sigma^2} \|x_i^t - \mu_m^t\|^2\right)$$

likelihood that the person is at location $x_i^t$ after $t$ observations given it is engaged in motion pattern $\theta_m$
Data Likelihood

- Joint likelihood of a single trajectory and its correspondence vector

\[ P(d_i, c_i \mid \theta) = \prod_{t=1}^{T} \prod_{m=1}^{M} \exp\left( -\frac{1}{2\sigma^2} c_{im} \| x^t_i - \mu^t_m \|^2 \right) \]

- Expected log likelihood w.r.t. \( c \mid \theta, d \)

\[
E_c[\ln P(d, c \mid \theta) \mid \theta, d] = E_c[\ln \sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta) \mid \theta, d]
\]
\[
= E_c\left[ \sum_{i=1}^{I} \ln \prod_{t=1}^{T} \prod_{m=1}^{M} \exp\left( -\frac{1}{2\sigma^2} c_{im} \| x^t_i - \mu^t_m \|^2 \right) \mid \theta, d \right]
\]
\[
= E_c\left[ -\frac{1}{2\sigma^2} \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{m=1}^{M} c_{im} \| x^t_i - \mu^t_m \|^2 \mid \theta, d \right]
\]
Q-Function

(Expectation is a linear operator, move it inside the sum)

\[ Q(\theta' | \theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^{I} \sum_{m=1}^{M} E[c_{im} | \theta, d] \sum_{t=1}^{T} ||x_i^t - \mu_m^t||^2 \]
E-Step: Compute the Expectations Given the Current Model $\theta[j]$

$$E[c_{im} | \theta[j], d] = P(c_{im} | \theta[j], d)$$

$$= P(c_{im} | \theta_m[j], d_i)$$

Bayes’ Theorem

$$\equiv \eta P(d_i | c_{im}, \theta_m[j]) P(c_{im} | \theta_m[j])$$

uniform prior

$$\equiv \eta' P(d_i | \theta_m[j])$$

$$\equiv \eta'' \prod_{t=1}^{T} \exp\left(-\frac{1}{2\sigma^2}\|x_i^t - \mu_m^{t[j]}\|^2\right)$$

normalized likelihood that the $i$-th trajectory belongs to the $m$-th model component
M-Step: Maximize the Expected Likelihood

\[ \theta^{[j+1]} = \underset{\theta'}{\text{argmax}} \ Q(\theta' \mid \theta^{[j]}) \]

\[ = \underset{\theta'}{\text{argmax}} \ \left\{ - \frac{1}{2\sigma^2} \sum_{i=1}^{I} \sum_{m=1}^{M} E[c_{im} \mid \theta^{[j]}, d] \sum_{t=1}^{T} \|x_i^t - \mu_m^t\|^2 \right\} \]

\[ = \underset{\theta'}{\text{argmin}} \ \left\{ \sum_{i=1}^{I} \sum_{m=1}^{M} E[c_{im} \mid \theta^{[j]}, d] \sum_{t=1}^{T} \|x_i^t - \mu_m^t\|^2 \right\} . \]

Compute partial derivative with respect to \( \mu_m^t \).
M-Step: Maximize the Expected Likelihood

\[
\sum_{i=1}^{I} E[c_{im} \mid \theta[j], d] \cdot 2 \cdot (x_i^t - \mu_{m}^{t[j+1]}) \overset{!}{=} 0 \iff
\sum_{i=1}^{I} E[c_{im} \mid \theta[j], d] x_i^t \overset{!}{=} \sum_{i=1}^{I} E[c_{im} \mid \theta[j], d] \mu_{m}^{t[j+1]} \iff
\]

\[
\mu_{m}^{t[j+1]} \overset{!}{=} \frac{\sum_{i=1}^{I} E[c_{im} \mid \theta[j], d] x_i^t}{\sum_{i=1}^{I} E[c_{im} \mid \theta[j], d]}
\]

This is the mean update of soft k-means
EM Application Example: 9 Trajectories of 3 Motion Patterns
EM: Example (step 1)

\[ C_{im}: \]

likelihood that \( s_0 \) belongs to the red motion pattern

\[ \theta^1: \]

\[ A \rightarrow B \quad C \rightarrow A \quad A \rightarrow C \] trajectories
EM: Example (step 2)

\[ C_{im} : \]

\[ \theta^2 : \]
EM: Example (step 3)

$C_{im}$:

$\theta^3$:
EM: Example (step 4)

$C_{im}$:

$\theta^4$:
EM: Example (step 5)

$C_{im}$:

$\theta^5$:
EM: Example (step 6)

\[ C_{im}: \]

\[ \theta^6: \]
EM: Example (step 7)

\[ C_{im} : \]

\[ \theta^7 : \]
EM: Example (step 8)

\[ C_{im}: \]

\[ \theta^8: \]
EM: Example (step 9)

\[ C_{im} : \]

\[ \theta^9 : \]
Estimating the Number of Model Components Greedily

After convergence of the EM check whether the model can be improved

- by introducing a new model component for the trajectory which has the lowest likelihood or
- by eliminating the model component which has the lowest utility.

Select model \( \theta \) which has the **highest evaluation**

\[
E_c[\log P(d, c | \theta) | \theta, d] - \frac{M}{2} \log I
\]

where \( M = \# \text{model components} \), \( I = \# \text{trajectories} \)

Bayesian Information Criterion [Schwarz, `78]
Application Example
Learned Motion Patterns
Summary

- K-Means is the most popular clustering algorithm
- It is efficient and easy to implement
- Converges to a local optimum
- A variant of hard k-means exists allowing soft assignments
- Soft k-means corresponds to the EM algorithm which is a general optimization procedure
Further Reading

E. Alpaydin
Introduction to Machine Learning