Machine Learning

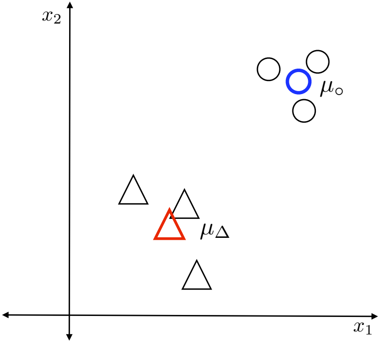
Lecture 10
Clustering

Felix Bießmann

Beuth University & Einstein Center for Digital Future

Clustering

Psychological Models of Categorization: Prototypes



Prototypes $\boldsymbol{\mu}_\Delta$ and $\boldsymbol{\mu}_o$:

$$\boldsymbol{\mu}_\Delta = 1/N_\Delta \sum_n^{N_\Delta} \mathbf{x}_{\Delta,n}$$

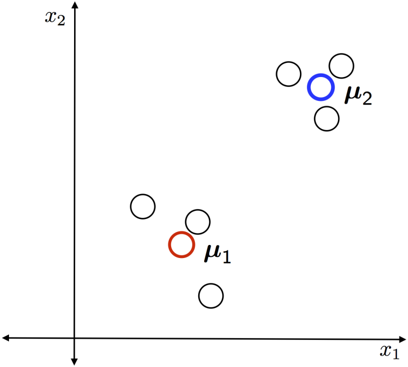
$$\boldsymbol{\mu}_o = 1/N_o \sum_n^{N_o} \mathbf{x}_{o,n}$$

New data points x are assigned to their closest cluster center μ^*

$$\boldsymbol{\mu}^* = \underset{i}{\operatorname{argmin}}(\|\boldsymbol{\mu}_i - \mathbf{x}\|_2) \qquad (1)$$

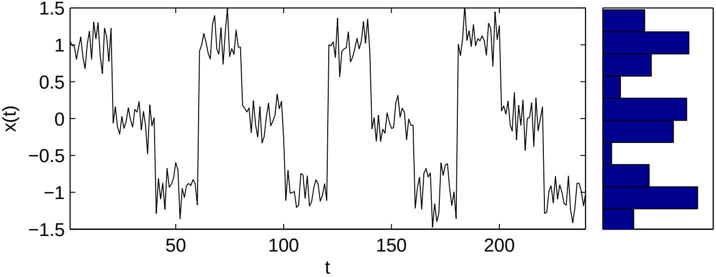
Clustering

Psychological Models of Categorization: Prototypes



The only difference for clustering is:
We do not have labels.

Clustering For Quantization of Analog Signals



Quantization transforms an analog signal into discretized states

This is important for Audio Processing, Compression, . . .

The most popular Clustering Algorithm was proposed for Quantization [Lloyd, 1982]

K-means Clustering

K-Means Algorithm

Re-iterating two steps:

1. Assign each data point $\mathbf{x}_i$ to their closest cluster $\boldsymbol{\mu}_k$
2. Update $\boldsymbol{\mu}_k$ to the mean of the members in that cluster

K-means Clustering

Goal: Given data $\mathbf{x}_1, \ldots, \mathbf{x}_N$ find cluster centers $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_K$ such that the distances of data points to their respective cluster centre are minimized

$$\mathcal{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{c}_{n,k} \|\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{c}_k}\| \quad (2)$$

$$\text{where } \mathbf{c}_{n,k} \begin{cases} 1 & \text{if } \mathbf{x}_n \text{ belongs to cluster } k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

K-means Clustering Algorithm

Algorithm 1 K-means clustering

Require: data $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$, number of clusters k, iterations m.

Choose random data points as initial cluster centres $\boldsymbol{\mu}_1 \leftarrow \mathbf{x}_{i_1}, \dots, \boldsymbol{\mu}_k \leftarrow \mathbf{x}_{i_k}$ where $i_j \neq i_l$ for all $j \neq l$.
$\mathbf{c} \leftarrow \mathbf{0}_N$
$\mathbf{c}^{\text{old}} \leftarrow \mathbf{0}_N$
$i \leftarrow 0$
while $i < m$ **do**
    for $j = 1$ to N **do**
        Find nearest cluster centre $\mathbf{c}_j \leftarrow \operatorname{argmin}_{1 \leq l \leq k} \|\mathbf{x}_j - \boldsymbol{\mu}_l\|_2$
    end for
    for $j \leftarrow 1$ to k **do**
        Compute new cluster centre $\boldsymbol{\mu}_j \leftarrow \frac{1}{|\{l : \mathbf{c}_l = j\}|} \sum_{l : \mathbf{c}_l = j} \mathbf{x}_l$
    end for
    if $\mathbf{c}^{\text{old}} = \mathbf{c}$ **then**
        break
    end if
    $\mathbf{c}^{\text{old}} \leftarrow \mathbf{c}$
    $i \leftarrow i + 1$
end while
return cluster centres $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k \in \mathbb{R}^D$, assignment vector $\mathbf{c}^{\text{old}} \in \mathbb{R}^n$

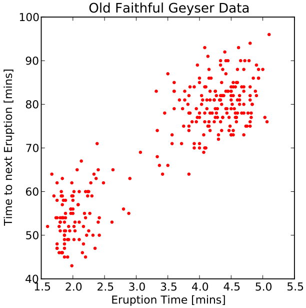
Application Example: Geyser Eruptions

Old Faithful Geyser
Yellowstone National Park, USA

Famous data set for clustering

- Old Faithful Eruptions
- Two dimensions
  1. Time of Eruption [mins]
  2. Time until next Eruption [mins]

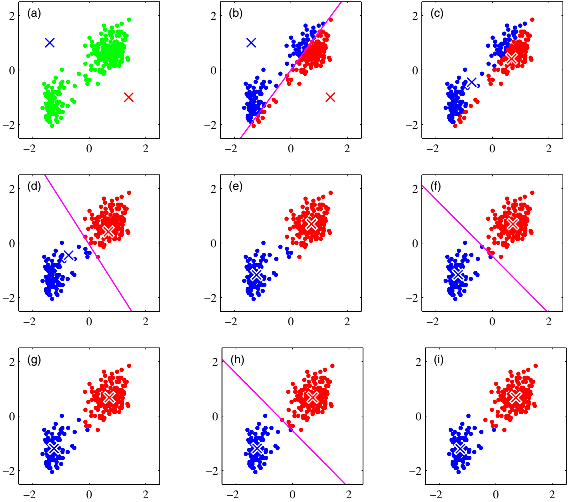
Application Example: Geyser Eruptions



Famous data set for clustering

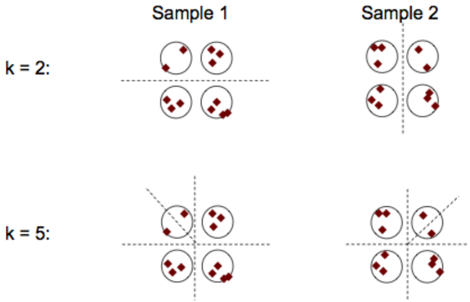
- Old Faithful Eruptions
- Two dimensions
  1. Time of Eruption [mins]
  2. Time until next Eruption [mins]

K-means Clustering Step-by-Step



Clustering Instability

Number of Clusters is a critical parameter



Clusterings are instable if number of clusters is too small or too large

Clustering Instability

- Number of clusters is critical hyper parameter
- In supervised settings we use cross-validation to optimize hyper-parameters for accuracy on test data
- How can we optimize the number of clusters?

→ Choose that k that results in most **stable** clusterings
For a review see e.g. [von Luxburg, 2009]

Clustering Instability Algorithm

Algorithm 2 Clustering Instability

Require: data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, clustering algorithm $\mathcal{A}$, maximal number of clusters K, iterations i.

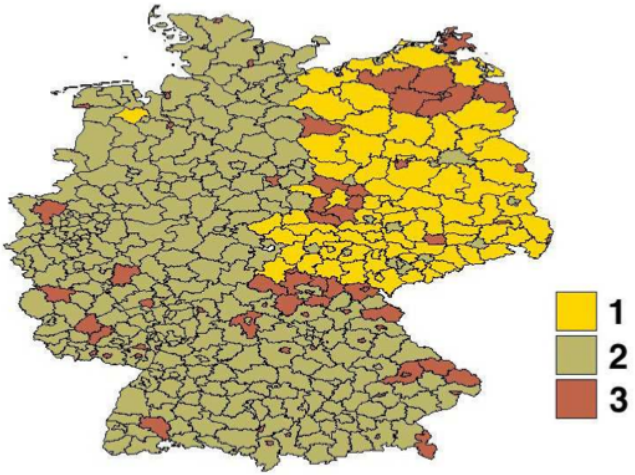
Ensure: optimal number of clusters k^*

```
for k = 2 to K do
    Resample data set (e.g. random draws with replacement)
    for it = 1 to i do
        Cluster data using algorithm A into k clusters
    end for
    Compute minimal (across all label permutations) distance between clusterings
end for
Chose that k that has the minimal instability over resamplings
```

Application: Clustering 'Mentality' of German Population

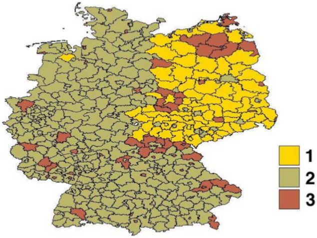
- Until 1989 Germany was divided into
  - a free capitalistic western part
  - a communist eastern part
- Political systems influence mentality of people
- The survey "Perspektive Deutschland" investigated this
  Survey asked questions about:
  - what people desire
  - what people are afraid of

German Regions clustered by 'Mentality' of Population



...based on questionnaires, 'Perspektive Deutschland' poll 2005.

German Regions clustered by 'Mentality' of Population



People in cluster 1 say:
Helping others is important
I am afraid of loosing my job

People in cluster 2 say:
Reaching my own goals is important
I am afraid of loosing my health

Online K-Means

$$
\begin{aligned}
\mathcal{E}(\boldsymbol{\mu}_k) =& \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^2 \\
=& \frac{1}{2}\mathbf{x}^\top \mathbf{x} - \frac{1}{2} 2\mathbf{x}^\top \boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\mu}_k \\
\frac{\partial \mathcal{E}(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k} =& \boldsymbol{\mu}_k - \mathbf{x} \\
\text{Learning rate } \eta =& \frac{1}{n_k} \; (n_k = \text{number of samples in cluster } k) \\
\text{Gradient Step } \boldsymbol{\mu}_k \leftarrow & \boldsymbol{\mu}_k - \eta \frac{\partial \mathcal{E}(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k} = \boldsymbol{\mu}_k - \frac{1}{n_k}(\mathbf{x} - \boldsymbol{\mu}_k)
\end{aligned} \tag{4}
$$

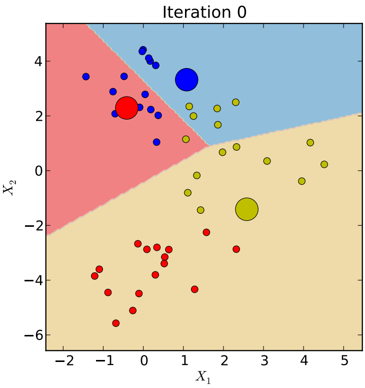
Online K-Means Algorithm

Algorithm 3 Online K-means clustering

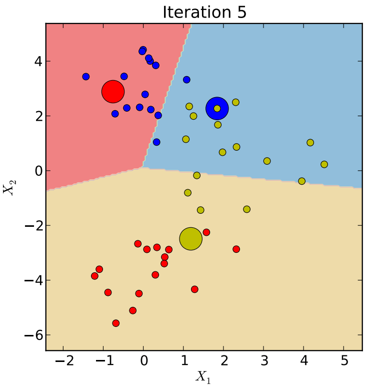
Require: data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, number of clusters k, iterations m.
Ensure: cluster centres $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$

Choose random data points as initial cluster centres
$\boldsymbol{\mu}_k \leftarrow \mathbf{x}_i, \ldots, \boldsymbol{\mu}_k \leftarrow \mathbf{x}_{i_k}$ where $i_j \neq i_l$ for all $j \neq l$.
Initialize cluster assignment counts $n_1, \ldots, n_k \leftarrow 0$
for $i = 1, \ldots, m$ **do**
    Draw a new data point randomly $\mathbf{x}_i$
    Find nearest cluster centre $k^* \leftarrow \mathrm{argmin}_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2$
    Update cluster counts $n_{k^*} \leftarrow n_{k^*} + 1$
    Update cluster centers $\boldsymbol{\mu}_{k^*} \leftarrow \boldsymbol{\mu}_{k^*} + \frac{1}{n_{k^*}}(\mathbf{x}_i - \boldsymbol{\mu}_{k^*})$
end for

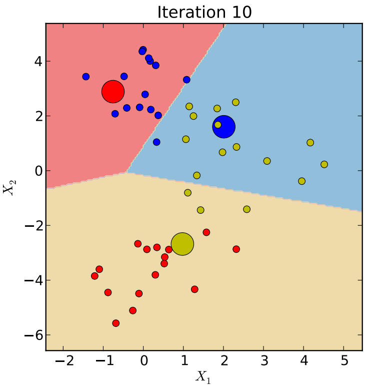
Online K-Means Algorithm - Example



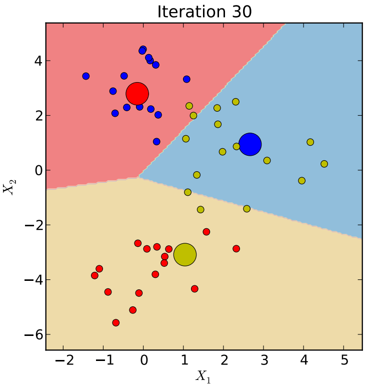
Online K-Means Algorithm - Example



Online K-Means Algorithm - Example



Online K-Means Algorithm - Example



Distance Measures for Real-Valued Data $\mathbf{x} \in \mathbb{R}^D$

Clustering Algorithms need a **distance function** $d(\mathbf{x}_i, \mathbf{x}_j)$

- For real valued data $\mathbf{x} \in \mathbb{R}^D$ we can use the Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \tag{5}$$

- More robust (less sensitive to outliers) is the **city block distance** or $\mathcal{L}_1$ norm

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_1 \tag{6}$$

- Another alternative is the correlation coefficient (also called cosine similarity)

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_1^\top \mathbf{x}_2}{\|\mathbf{x}_1\|_2 \|\mathbf{x}_2\|_2} \tag{7}$$

For standardized data $\sum_i \mathbf{x}_i = 0, \;\; \sum_i \mathbf{x}_i^2 = 1$ maximizing correlation is the same as minimizing euclidean distance.

Distance Measures for Non-Real-Valued Data

- For ordinal variables $\mathbf{x} \in \{\text{low}, \text{medium}, \text{high}\}^D$ we can transform the values into real-valued numbers (for three possible values e.g. $1/3, 2/3, 3/3$) and then apply distance functions for real-valued data
- For categorial variables $\mathbf{x} \in \{\text{red}, \text{green}, \text{blue}\}^D$ we can use a binary coding for the differences

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d}^{D} \mathbf{x}_{id} \neq \mathbf{x}_{jd} \tag{8}$$

This metric is called **Hamming Distance**

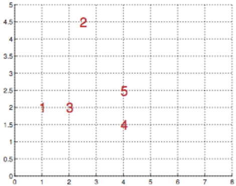
For the sake of simplicity we only consider the euclidean distance

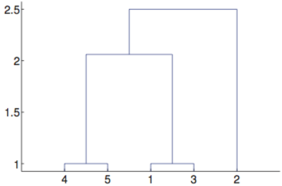
Hierarchical Clustering

- K-Means produces **flat** clusterings
- Often we are interested in a **hierarchy** of clusterings
- Examples:
  - Biological Species
  - Topics in Text Documents

Hierarchical Clustering

- K-Means produces **flat** clusterings
- Often we are interested in a **hierarchy** of clusterings
- Examples:
  - Biological Species
  - Topics in Text Documents





Hierarchical Clustering

- A popular approach to hierarchical clustering is
  1. Start with each data point as one cluster
  2. Successively merge (*agglomerate*) similar clusters

$\rightarrow$ **Agglomerative Clustering**

As most clustering algorithms, these procedures are not defined via objective functions but via algorithms

$\rightarrow$ Difficult to establish convergence criteria

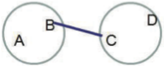
Agglomerative Clustering

Merging requires distance function $d(C_i, C_j)$ for clusters $C_i,\ C_j$

Agglomerative Clustering

Merging requires distance function $d(C_i, C_j)$ for clusters $C_i,\ C_j$

Single Linkage
Distance between two closest points in $C_i,\ C_j$

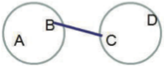


$d(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Agglomerative Clustering

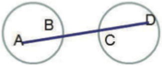
Merging requires distance function $d(C_i, C_j)$ for clusters $C_i,\ C_j$

Single Linkage
Distance between two closest points in $C_i,\ C_j$



$d(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Complete Linkage
Distance between two most distant points

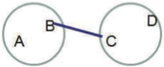


$d(C_i, C_j) = \max_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Agglomerative Clustering

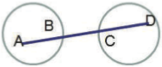
Merging requires distance function $d(C_i, C_j)$ for clusters $C_i,\ C_j$

Single Linkage
Distance between two closest points in $C_i,\ C_j$



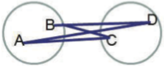
$d(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Complete Linkage
Distance between two most distant points



$d(C_i, C_j) = \max_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Average Linkage
Average distance between all $N_i N_j$ pairs



$d(C_i, C_j) = \frac{1}{N_i N_j} \sum_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

Agglomerative Clustering Algorithm

Algorithm 4 Agglomerative Clustering

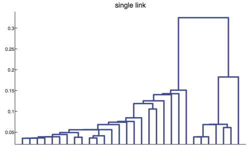
Require: Data points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$, number of clusters k, distance function $d(.,.)$
Ensure: Binary tree of clusters

Initialize each data point as cluster
for $i = 1$ to N **do**
  $C_i \leftarrow i$
end for
Initialize each cluster as available for merging
$\mathcal{S} \leftarrow \{1, \dots, N\}$
while There are clusters to merge **do**
  Pick 2 most similar clusters to merge
  $j, k \leftarrow \mathrm{argmin}_{j,k}\, d(j,k)$
  Merge clusters to new cluster $C_l \leftarrow C_j \cup C_k$
  Mark j, k as unavailable for merging
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{j, k\}$
  if $C_l \notin \mathcal{S}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{l\}$
  end if
end while

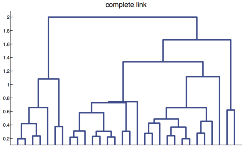
Examples Hierarchical Clustering

Dendrograms (binary clustering trees) of yeast gene expression data

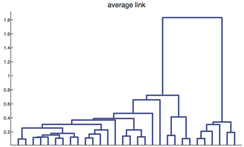
Single Linkage



Complete Linkage



Average Linkage



Taken from [Murphy, 2012]

Summary

- Clustering Algorithms find clusters in data
- Clustering Performance depends on distance function used
- K-Means is one of the most popular clustering algorithms
- For large data sets use Online K-Means
- Wrong number of clusters leads to unstable results
- Hierarchical clustering

References

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer US, 2007.

S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, Mar. 1982. ISSN 0018-9448.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning. The MIT Press, 1 edition, 2012. ISBN 0262018020,9780262018029.

U. von Luxburg. Clustering stability: An overview. *Foundations and Trends in Machine Learning*, 2(3):235–274, 2009.