Machine Intelligence 2

4.1 K-means Clustering

Prof. Dr. Klaus Obermayer

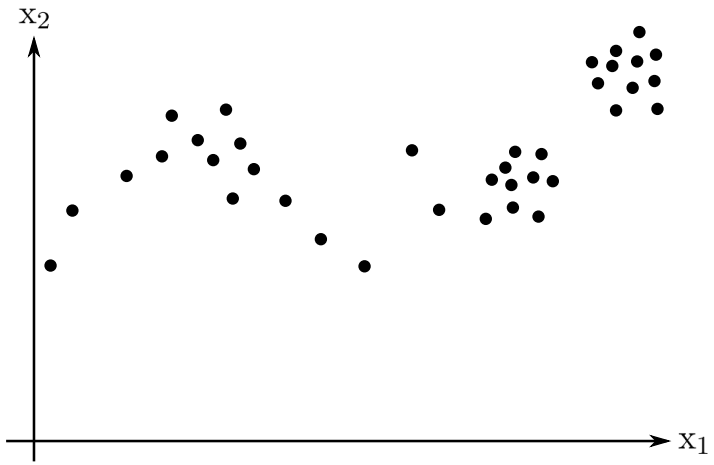
Fachgebiet Neuronale Informationsverarbeitung (NI)

SS 2018

K-means Clustering

Projection methods vs. clustering

observations: $\left\{\underline{\mathbf{x}}^{(\alpha)}\right\}, \alpha = 1, \ldots, p; \quad \underline{\mathbf{x}} \in \mathbb{R}^N$



- $\rightsquigarrow$ high-dimensional
- $\rightsquigarrow$ groups, categories, hidden causes
- $\rightsquigarrow$ interesting directions
- $\rightsquigarrow$ "informative" manifolds

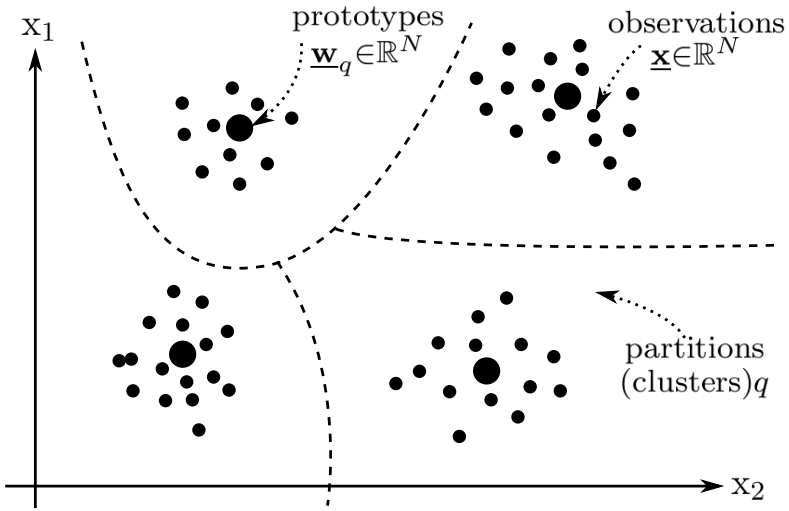
What is the relevant "structure"?

$\Rightarrow$ projection methods: search for "interesting" directions in feature space

$\Rightarrow$ clustering methods: grouping & categorization (and prototypes)

Central clustering

- $\Rightarrow$ unsupervised formation of categories (partitions, clusters) according to predefined criteria
- $\Rightarrow$ description of clusters by prototypes $\leftarrow$ "central" clustering
- $\Rightarrow$ goal: partitioning of observations $\underline{\mathbf{x}}^{(\alpha)}, \quad \alpha = 1, \ldots, p; \quad \underline{\mathbf{x}}^{(\alpha)} \in \mathbb{R}^N$ according to similarity.



Cluster model

$\Rightarrow$ prototypes: $\underline{\mathbf{w}}_q, \quad q = 1, \ldots, M$ (M: number of clusters)

$\Rightarrow$ binary assignment variables $m_q^{(\alpha)}$:

$$m_q^{(\alpha)} = \begin{cases} 1, & \text{if } \underline{\mathbf{x}}^{(\alpha)} \text{ belongs to cluster } q \\ 0, & \text{else} \end{cases}$$

$\Rightarrow$ normalization: $\sum_q m_q^{(\alpha)} = 1$

Cost function

→ average quadratic distance between observations and prototypes

$$E^T_{\left[\left\{m_q^{(\alpha)}\right\},\left\{\underline{\mathbf{w}}_q\right\}\right]} = \frac{1}{p}\sum_{q,\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right)^2$$

→ cost function implicitly quantifies our prior knowledge about the data

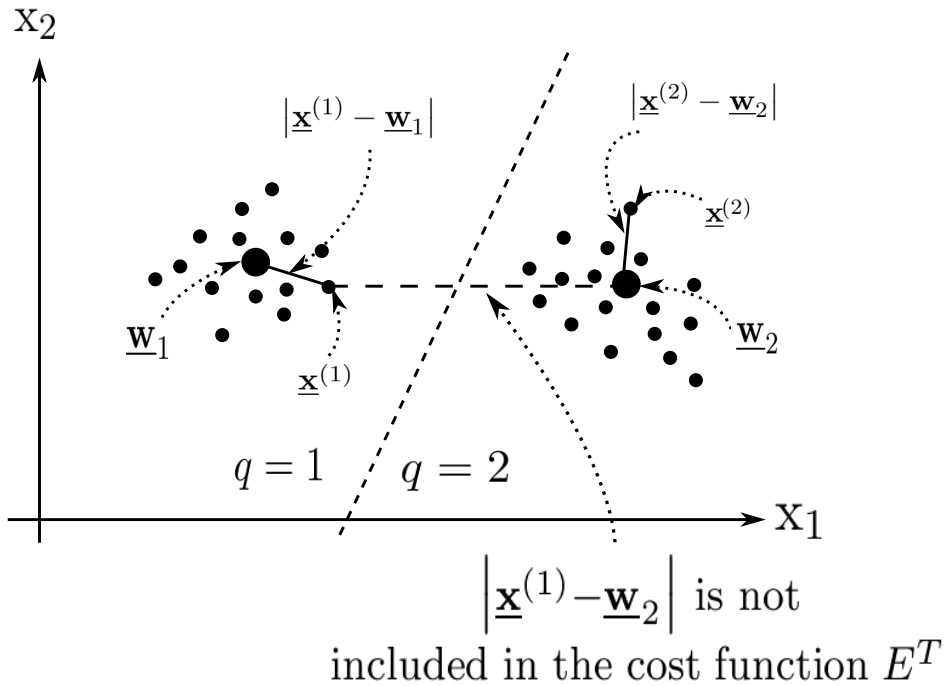
Model selection

Cost function:

$\Rightarrow$ cluster-centers: continuous variables

$\Rightarrow$ cluster-assignment: binary variables.

$\Rightarrow$ dissimilarity measure: squared Euclidean distance.



Batch K-means

Algorithm 1: batch K-means

random initialization of prototypes, e.g. $\underline{\mathbf{w}}_q =< \underline{\mathbf{x}} > +\underline{\eta}_q$, $\underline{\eta}_q$ small random vector

begin loop

(1) choose $m_q^{(\alpha)}$ such that E^T is minimal for the given prototypes

$$m_q^{(\alpha)} = \begin{cases} 1, & \text{if } q = \text{argmin}_\gamma \left| \underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_\gamma \right| \\ 0, & \text{else} \end{cases}$$

$\Rightarrow$ assign every data point to its nearest prototype

(2) choose $\underline{\mathbf{w}}_q$ such that E^T is minimal for the -new- assignments

$$\underline{\mathbf{w}}_q = \frac{\sum_\alpha m_q^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)}}{\sum_\alpha m_q^{(\alpha)}}$$

$\Rightarrow$ set $\underline{\mathbf{w}}_q$ to the center of mass of its assigned data

end

Batch K-means

$$E^T_{\left[\left\{m_q^{(\alpha)}\right\},\left\{\underline{\mathbf{w}}_q\right\}\right]} = \frac{1}{p}\sum_{q,\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right)^2$$

$\Rightarrow$ "condition for extremum"

$$\frac{\partial}{\partial \underline{\mathbf{w}}_q}\left\{\frac{1}{2p}\sum_{q',\alpha} m_{q'}^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{q'}\right)^2\right\} = -\frac{2}{p}\sum_{\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right) \stackrel{!}{=} 0$$

$$\rightsquigarrow \underline{\mathbf{w}}_q = \frac{\sum_{\alpha} m_q^{(\alpha)}\underline{\mathbf{x}}^{(\alpha)}}{\sum_{\alpha} m_q^{(\alpha)}}$$

Batch K-means

$$E^T_{\left[\left\{m_q^{(\alpha)}\right\},\left\{\underline{\mathbf{w}}_q\right\}\right]} = \frac{1}{p}\sum_{q,\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right)^2$$

$\Rightarrow$ condition for minimum:

$$\frac{\partial^2}{\partial \mathrm{w}_{qi}\partial \mathrm{w}_{q'j}}\left\{\frac{1}{p}\sum_{q'',\alpha} m_{q''}^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{q''}\right)^2\right\}$$

$$= \frac{\partial}{\partial \mathrm{w}_{q'j}}\left\{-\frac{2}{p}\sum_{\alpha} m_q^{(\alpha)}\left(\mathrm{x}_i^{(\alpha)} - (\underline{\mathbf{w}})_{qi}\right)\right\} \quad = \left(\frac{2}{p}\sum_{\alpha} m_q^{(\alpha)}\right)\delta_{ij}\delta_{qq'}$$

$\Rightarrow$ diagonal matrix with all positive entries $\rightarrow$ condition for minimum is always satisfied.

$\Rightarrow$ minimizing E^T is not convex optimization problem.

Batch K-means (continued)

$$E^T_{\left[\left\{m_q^{(\alpha)}\right\},\left\{\underline{\mathbf{w}}_q\right\}\right]} = \frac{1}{p}\sum_{q,\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right)^2$$

- $\Rightarrow$ If $\underline{\mathbf{w}}_q$ is center of mass $\Longrightarrow$ $E^T = \text{variance}$.
- $\Rightarrow$ E^T is non-increasing in every step and E^T is bounded from below $\rightarrow$ K-means clustering converges to a (local) optimum of E^T.
- $\Rightarrow$ E^T at the solution can be interpreted as the "size" (variance) of the clusters.

On-line K-means

Algorithm 2: On-line k-Means

random initialization of prototypes, e.g.

$\underline{\mathbf{w}}_q =< \underline{\mathbf{x}} > +\underline{\eta}_q$, $\underline{\eta}_q$ small random vector

select learning step: $0 < \varepsilon << 1$

begin loop

choose a data point $\underline{\mathbf{x}}^{(\alpha)}$

assign data point to its closest prototype q

$$q = \underset{\gamma}{\operatorname{argmin}} \left| \underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_\gamma \right|$$

change corresponding prototype according to

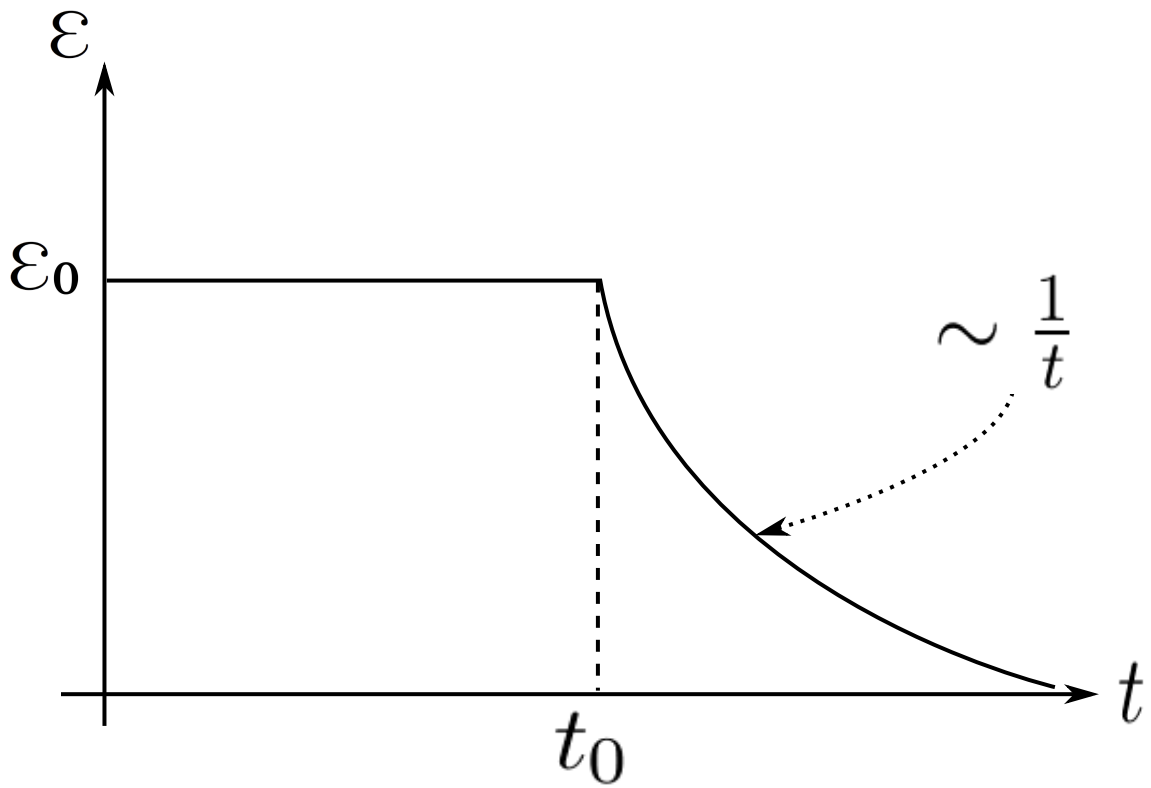
$$\Delta \underline{\mathbf{w}}_q = \varepsilon \left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q \right)$$

change ε

end

On-line K-means

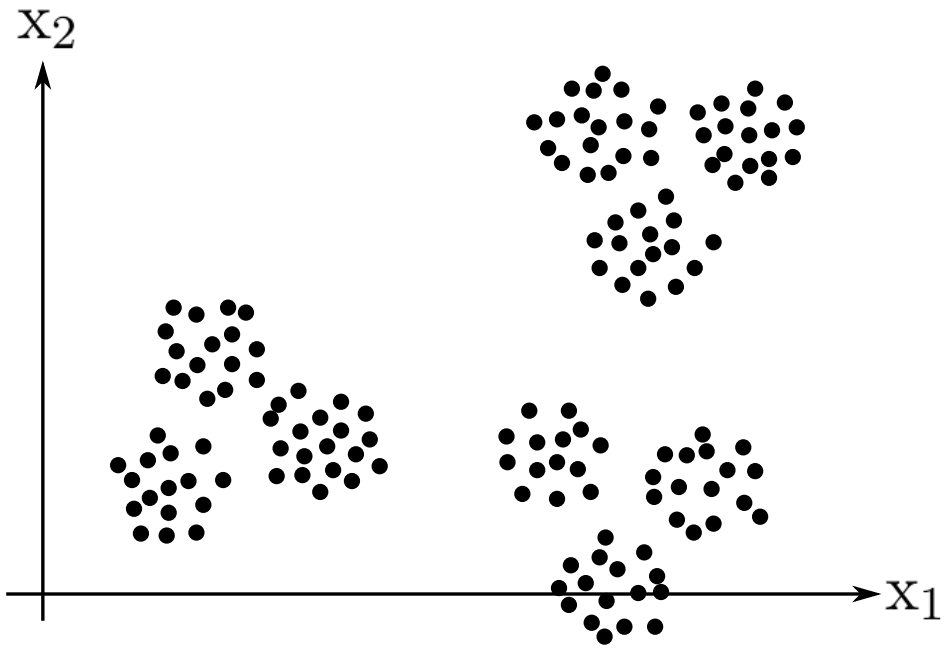
- more robust than batch-learning w.r.t convergence to local minima
- useful for streaming data
- quality of the found solution depends on choosing an appropriate "annealing" schedule for ε: Robbins-Monro conditions



Number of prototypes

- M: hyperparameter

Choice of resolution



1 cluster
3 cluster
9 cluster
many cluster?

$\Rightarrow$ additional assumptions are needed!

Number of prototypes: Choice of resolution

- E^T_{min}: average size of cluster (in terms of variance)
  - $\leadsto$ large for few clusters – small for many clusters
  - $\leadsto$ zero, if number of cluster $\widehat{=}$ number of data points
  - $\leadsto$ E^T_{min} goes down if M increases.

- choice of resolution
  - $\rightarrow$ clusters "smaller" than the variance of noise probably do not capture meaningful structure
  - $\rightarrow$ $E^T_{\text{min}} \geq \sigma^2_{\text{noise}}$ which is a natural boundary on E^T_{min}

Iterative refinement

Algorithm 3: Iterative refinement

initialization: $\underline{\mathbf{w}}_1 = \frac{1}{p}\sum_{\alpha} \underline{\mathbf{x}}^{(\alpha)}$, $\underbrace{\left(E^T_{\min}\right)^*}_{\text{desired minimal variance}}$, $M = 1$

begin loop

  if $E^T_{\min} < \left(E^T_{\min}\right)^*$ **then** STOP

  select partition $q \in \{1, \ldots, M\}$ with largest variance

$$q = \operatorname*{argmax}_{\gamma} \left(\frac{\sum_{\alpha} m_{\gamma}^{(\alpha)} \left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{\gamma}\right)^2}{\sum_{\alpha} m_{\gamma}^{(\alpha)}} \right)$$

  add a new prototype: $\underline{\mathbf{w}}_{M+1} = \underline{\mathbf{w}}_q + \underbrace{\underline{\varepsilon}_q}_{\text{small random vector}}$

  $M \leftarrow M + 1$

  do K-means clustering with these M prototypes

end

Robustness of the clustering solution

- Solution should capture meaningful structure in the data
- Multiple runs with different initializations should yield similar solutions.
- *Caveat:* Permutation of labels does neither change cost nor character of the solution.

$$1, 2, 3, \ldots, M$$
$$9, 1, M, \ldots, 7$$

- $M!$ trivially equivalent optima $\rightarrow$ robustness-criterion has to be adapted.
- Avoid "instability": many structurally different clustering solutions with equal cost

Validation measure

- Model free approaches: Stability based validation
- **Idea:** taking too many or too few clusters leads to unstable partitions
- **Data:** $\mathbf{X}$ is a set containing all the data points *s.t.* $\mathbf{X} = \{\underline{\mathbf{x}}^{(\alpha)}\}, \alpha = 1, \ldots, p; \quad \underline{\mathbf{x}} \in \mathbb{R}^N$
- A solution of the clustering algorithm produces labellings set $\mathbf{Y}$ *s.t.* $\mathbf{Y} = \{y^{(\alpha)}\}$ where $y^{(\alpha)} \in L := \{1, \ldots, M\}$
- Dissimilarity between clustering solutions $\mathbf{Y}_1$ and $\mathbf{Y}_2$:

$$d := \frac{1}{|\mathbf{Y_1}|} \sum_{\alpha} \mathbf{1}\left\{y_1^{(\alpha)} \neq y_2^{(\alpha)}\right\}$$

Lange et. al., 2004, Neural Computation

Validation measure

Algorithm 4: Validation measure

begin for each $M \in \{M_{min}, \ldots, M_{max}\}$

  begin loop for r splits of data

    Split $\mathbf{X}$ into disjoint sets $\mathbf{X}_1$ and $\mathbf{X}_2$ randomly

    Apply clustering algorithm to both $\mathbf{X}_1$ and $\mathbf{X}_2$ to find $\mathbf{Y}_1$ and $\mathbf{Y}_2$

    Compute dissimilarity $d_i := \frac{1}{|\mathbf{Y}_2|} \sum_{\alpha=1}^{p} \mathbf{1} \left\{ \mathbf{Y_2}^{(\alpha)} \neq \phi_1[\mathbf{X}_2^{(\alpha)}] \right\}$

    Where ϕ_1 denotes a classifier trained on $(\mathbf{X_1}, \mathbf{Y}_1)$

  end

  Compute average dissimilarity: $\hat{S}_A = \frac{1}{r} \sum_r d_i$

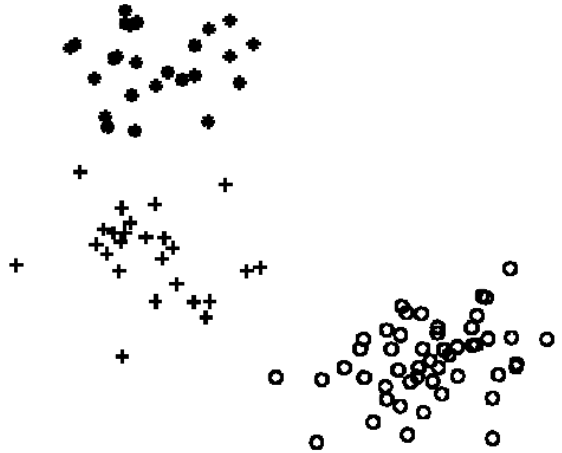
  Compute average dissimilarity $\hat{S}_R$ of an clustering algorithm which assigns clusters randomly using the same data

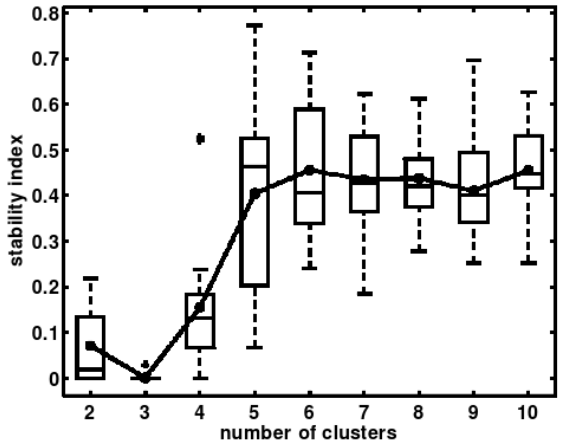
  Calculate stability index: $\bar{S}_M = \frac{\hat{S}_A}{\hat{S}_R}$

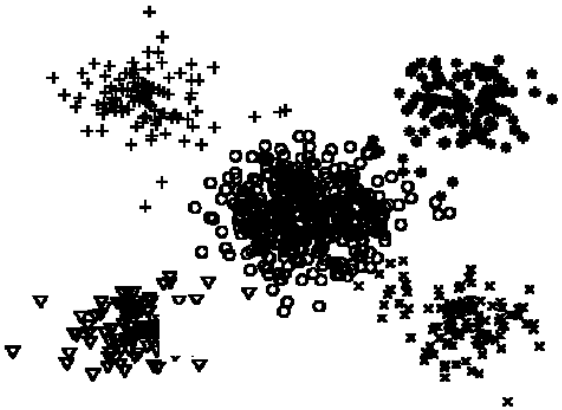
end

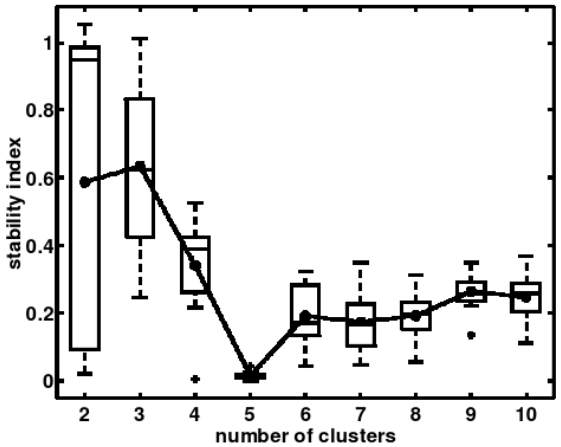
Return $\hat{M} = \text{argmin}_M(\bar{S}_M)$

K-means: Gaussian data









Further remarks

Alternative clustering approaches

- density based models ("model-based" $\Rightarrow$ Gaussian Mixture algorithm)
- hierarchical (connectivity based) clustering
  - single linkage ($\sim$ nearest neighbor)
  - complete linkage
  - average linkage / within group ssq (Ward criterion)
  - agglomerative vs. divisive clustering

Current issues

- "big data": pre-processing (e.g. preselect spatial methods & KD-trees)
- graph-based approaches & spectral clustering

Applications

Image segmentation & compression

- k-means for pixels (e.g. RGB)
- segmentation via cluster-assignment (improvements: context, smoothness ...)
- data: values –> label and residues, MDL principle for cluster number