Machine Intelligence 2

4.1 K-means Clustering

Prof. Dr. Klaus Obermayer

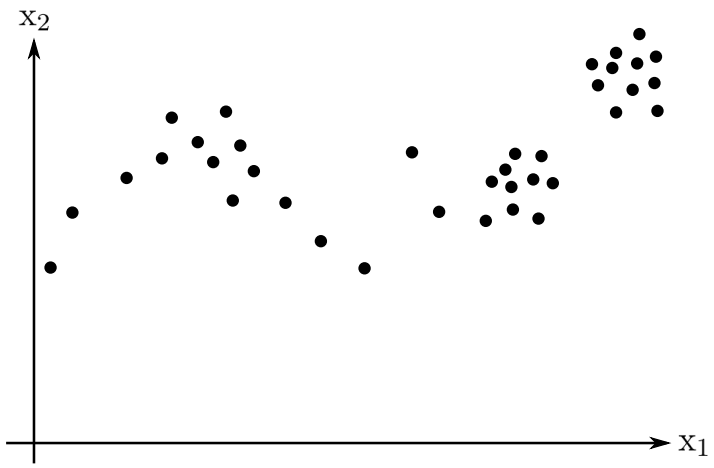
Fachgebiet Neuronale Informationsverarbeitung (NI)

SS 2017

K-means Clustering

Projection methods vs. clustering

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



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

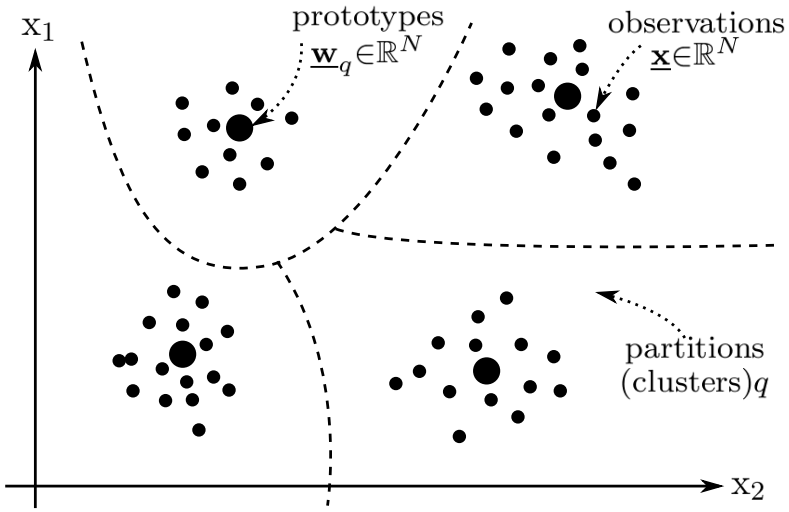
What is the relevant "structure"?

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

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

Central clustering

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



Cluster model

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

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

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

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

Cost function

Quality Criteria:

→ average quadratic distance between observations and prototypes

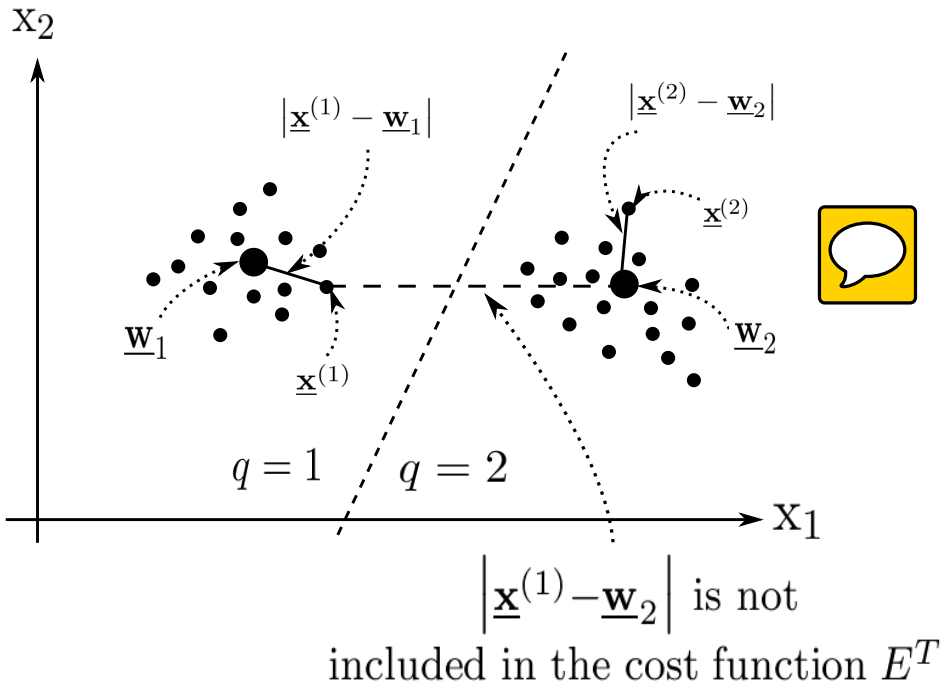
$$E^T_{\left[\left\{m_q^{(\alpha)}\right\},\left\{\underline{\mathbf{w}}_q\right\}\right]} = \frac{1}{p}\sum_{q,\alpha} m_q^{(\alpha)}\left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q\right)^2$$

→ cost function implicitly quantifies our prior knowledge about the data

Model selection

Cost function:

- ⇒ cluster-centers: continuous variables
- ⇒ cluster-assignment: binary variables.
- ⇒ dissimilarity measure: squared Euclidean distance.



Batch K-means

Algorithm 1: batch K-means

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

begin loop

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

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

$\Rightarrow$ 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

Model selection - batch K-means

Cost function:

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

$\Rightarrow$ "Center of mass is optimal" why is it optimal?

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

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

numer of data points in group

Batch K-means

Assumption: All Clusters have same size

Cost function:

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

cost function enspricht variance

$\Rightarrow$ condition for minimum: taking second derivatives of cost function

second derivative

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

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

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

$\Rightarrow$ Note: 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.

$\underline{\mathbf{w}}_q = < \underline{\mathbf{x}} > + \underline{\eta}_q, \quad \underline{\eta}_q$ 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

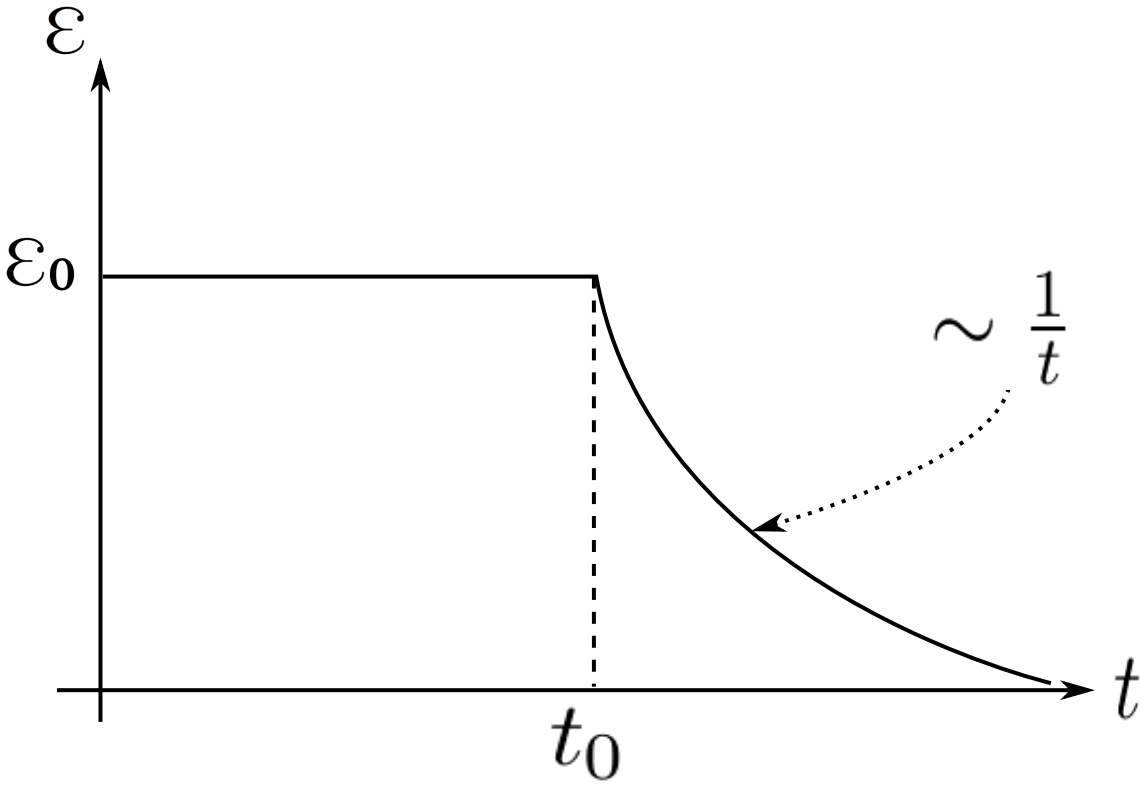
$$\Delta \underline{\mathbf{w}}_q = \varepsilon \left(\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q \right)$$

- change ε

end

On-line K-means

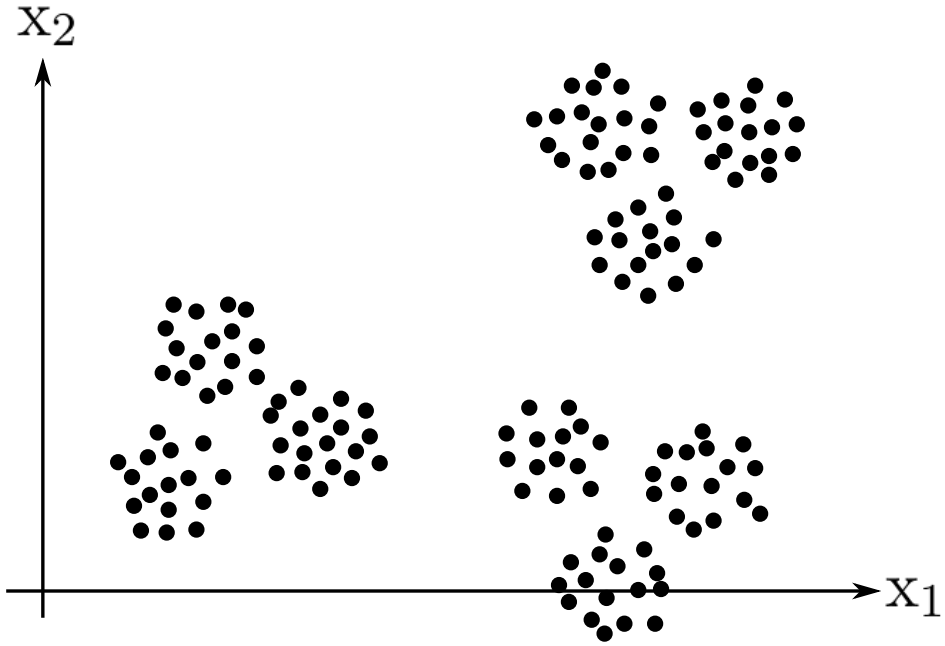
- more robust than batch-learning w.r.t convergence to local minima
- useful for streaming data
- quality of the found solution depends on choosing an appropriate "annealing" schedule for ε: Robbins-Monro conditions



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)
  - $\rightsquigarrow$ large for few clusters – small for many clusters
  - $\rightsquigarrow$ zero, if number of cluster $\widehat{=}$ number of data points
  - $\rightsquigarrow$ $E^T_{\min}$ goes down if M increases.

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

Iterative refinement

Algorithm 3: iterative refinement

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

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 whole k means clustering for every iteration

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.

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

Siehe Aufschrieb onenote

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

Validation measures

- Model free approaches: Stability based validation
- **Idea:** taking to many or too few clusters leads to unstable partitions
- $\mathbf{X} = \left\{\underline{\mathbf{x}}^{(\alpha)}\right\}, \alpha = 1, \ldots, p; \quad \underline{\mathbf{x}} \in \mathbb{R}^N$; A solution of the clustering algorithm is $\mathbf{Y} = (y_1, \ldots, y_p)$ where $y_i \in L := \{1, \ldots, M\}$
- Comparing clustering solutions Y_1 and Y_2:

$$d := \frac{1}{|\mathbf{Y_1}|} \sum_{\alpha} \mathbf{1}\left\{Y_{1,\alpha} \neq Y_{2,\alpha}\right\}$$

Caveats: Permutation, optimal classifier (e.g. KNN, see Lange et. al., 2004)

Validation measure

Algorithm 4: Validation measure

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

begin loop for r splits of data

split data $\mathbf{X}$ into $\mathbf{X}_1^{(i)}$ and $\mathbf{X}_2^{(i)}$ and find corresponding clustering solution $\mathbf{Y}_1^{(i)}$ and $\mathbf{Y}_2^{(i)}$ by applying clustering algorithm

use $(\mathbf{X}_1^{(i)}, \mathbf{Y}_1^{(i)})$ to train a nearest neighbor classifier ϕ and compute $\phi[X_2]$

compute distance $d_i := \frac{1}{|\mathbf{X}_2|} \sum_\alpha \mathbf{1}\left\{\mathbf{Y_2}^{(\alpha)} \neq \phi[\mathbf{X}_2^{(\alpha)}]\right\}$ (but considering the permutations)

Y_1

end

Average distance between partitions: $\hat{S}_{clustering} = \frac{1}{r} \sum_r d_i$

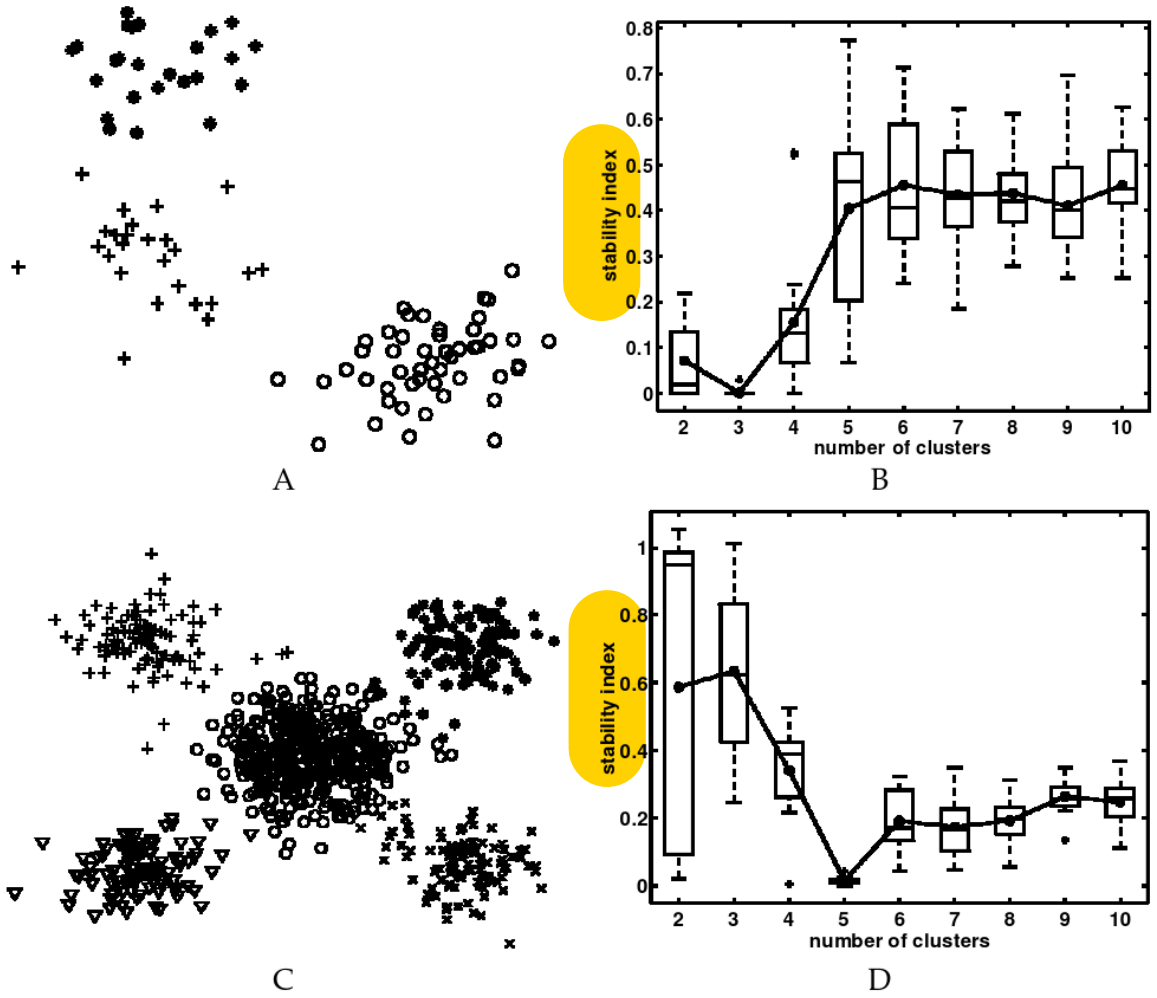
Sample s random clustering assignments and compute empirical average of the distances to estimate $\hat{S}_{random}$

Calculate stability index: $\bar{S}_M = \frac{\hat{S}_{clustering}}{\hat{S}_{clustering}}$

end

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

K-means: Gaussian data



Caveats: Permutation, optimal classifier (e.g. KNN, see Lange et. al., 2004)

Further remarks

Alternative clustering approaches

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

Current issues

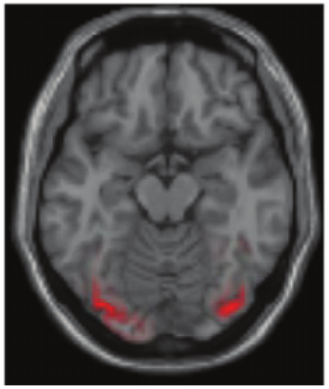
- "big data": pre-processing (e.g. preselect spatial methods & KD-trees)
- Graph-based approaches & spectral clustering

Applications

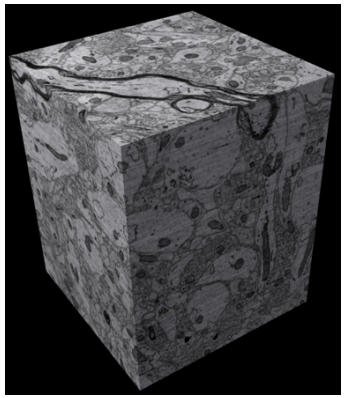
Image segmentation & compression

- k-means for pixels (e.g. RGB)
- segmentation via cluster-assignment (naive: context, smoothness ...)
- compression: $(N_{\text{pix}} \times S(R, G, B) \rightarrow N_{\text{pix}} \times k + \text{codebook})$

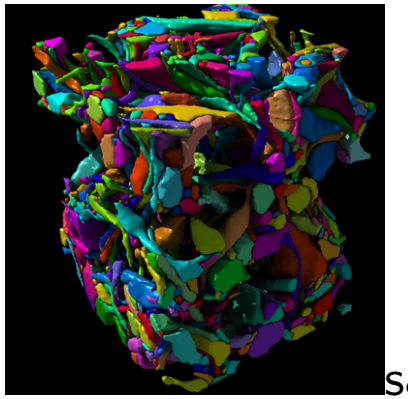
Neuroscience (fMRI, connectomics)



Litvak et.al(2001)



$\Rightarrow$



Seung lab