Clustering III

Lecture outline

- Soft (model-based) clustering and EM algorithm
- Clustering aggregation [A. Gionis, H. Mannila, P. Tsaparas: Clustering aggregation, ICDE 2004]
- Impossibility theorem for clustering [Jon Kleinberg, An impossibility theorem for clustering, NIPS 2002]

Expectation-maximization algorithm

- Iterative procedure to compute the ***Maximum Likelihood (ML)*** estimate – even in the presence of missing or hidden data
- **EM** consists of two steps:
  - **Expectation step:** the (missing) data are estimated given the observed data and current estimates of model parameters
  - **Maximization step:** The likelihood function is maximized under the assumption that the (missing) data are known

EM algorithm for mixture of Gaussians

- What is a mixture of **K** Gaussians?

$$p(x) = \sum_{k=1}^{K} \pi_k F(x \mid \Theta_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1$$

and **F(x|Θ)** is the Gaussian distribution with parameters **Θ = {μ,Σ}**

EM algorithm for mixture of Gaussians

- If all points **x∊X** are mixtures of **K** Gaussians then

$$p(X) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k)$$

- **Goal:** Find $\pi_1, \ldots, \pi_k$ and $\Theta_1, \ldots, \Theta_k$ such that **P(X)** is maximized
- Or, **ln(P(X))** is maximized:

$$L(\Theta) = \sum_{i=1}^{n} \ln \left\{ \sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k) \right\}$$

Mixtures of Gaussians -- notes

- Every point $\mathbf{x_i}$ is ***probabilistically*** assigned (generated) to (by) the $\mathbf{k}$-th Gaussian

- Probability that point $\mathbf{x_i}$ is generated by the $\mathbf{k}$-th Gaussian is

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

Mixtures of Gaussians -- notes

- Every Gaussian (cluster) **C_k** has an effective number of points assigned to it **N_k**

$$N_k = \sum_{i=1}^{n} w_{ik}$$

- With mean

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} x_i$$

- And variance

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

EM for Gaussian Mixtures

- Initialize the means μ_k, variances Σ_k ($\Theta_k=(\mu_k,\Sigma_k)$) and mixing coefficients π_k, and evaluate the initial value of the loglikelihood

- **Expectation step**: Evaluate weights

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

EM for Gaussian Mixtures

- **Maximization step**: Re-evaluate parameters

$$\mu_k^{new} = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} x_i$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} \left(x_i - \mu_k^{new}\right)\left(x_i - \mu_k^{new}\right)^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

- Evaluate **L(Θ^new)** and stop if converged

Lecture outline

- Soft (model-based) clustering and EM algorithm
- Clustering aggregation [A. Gionis, H. Mannila, P. Tsaparas: Clustering aggregation, ICDE 2004]
- Impossibility theorem for clustering [Jon Kleinberg, An impossibility theorem for clustering, NIPS 2002]

Clustering aggregation

- Many different clusterings for the same dataset!

  - Different objective functions
  - Different algorithms
  - Different number of clusters

- Which clustering is the best?
  - Aggregation: we do not need to decide, but rather find a reconciliation between different outputs

The clustering-aggregation problem

- Input
  - **n** objects $X = \{x_1, x_2, \ldots, x_n\}$
  - **m** clusterings of the objects $C_1, \ldots, C_m$
    - partition: a collection of disjoint sets that cover **X**
- Output
  - a **single partition C**, that is as close as possible to all input partitions
- How do we measure ***closeness of clusterings***?
  - disagreement distance

Disagreement distance

- For object **x** and clustering **C, C(x)** is the index of set in the partition that contains **x**

- For two partitions **C** and **P**, and objects **x,y** in **X** define

$$I_{C,P}(x,y) = \begin{cases} 1 & \text{if } C(x) = C(y) \text{ and } P(x) \neq P(y) \\ & \text{OR} \\ & \text{if } C(x) \neq C(y) \text{ AND } P(x) = P(y) \\ 0 & \text{otherwise} \end{cases}$$

U	*C*	*P*
x_1	1	1
x_2	1	2
x_3	2	1
x_4	3	3
x_5	3	4

- if $I_{P,Q}(x,y) = 1$ we say that **x,y** create a disagreement between partitions **P** and **Q**

- $$D(P,Q) = \sum_{(x,y)} I_{P,Q}(x,y)$$

Metric property for disagreement distance

- For clustering **C**: **$D(C,C) = 0$**
- **$D(C,C')\geq 0$** for every pair of clusterings **C, C'**
- **$D(C,C') = D(C',C)$**
- Triangle inequality?
- It is sufficient to show that for each pair of points **x,y ϵX**: **$I_{x,y}(C_1,C_3)\leq I_{x,y}(C_1,C_2) + I_{x,y}(C_2,C_3)$**
- **$I_{x,y}$** takes values 0/1; triangle inequality can only be violated when
  - **$I_{x,y}(C_1,C_3)=1$** and **$I_{x,y}(C_1,C_2) = 0$** and **$I_{x,y}(C_2,C_3)=0$**
  - Is this possible?

Clustering aggregation

- Given partitions $C_1,\ldots,C_m$ find C such that

$$D(C) = \sum_{i=1}^{m} D(C, C_i)$$

the aggregation cost

is minimized

U	C_1	C_2	C_3	C
x_1	1	1	1	1
x_2	1	2	2	2
x_3	2	1	1	1
x_4	2	2	2	2
x_5	3	3	3	3
x_6	3	4	3	3

Why clustering aggregation?

- Clustering categorical data

U	*City*	*Profession*	*Nationality*
x_1	New York	Doctor	U.S.
x_2	New York	Teacher	Canada
x_3	Boston	Doctor	U.S.
x_4	Boston	Teacher	Canada
x_5	Los Angeles	Lawer	Mexican
x_6	Los Angeles	Actor	Mexican

- The two problems are equivalent

Why clustering aggregation?

- Identify the correct number of clusters
  - the optimization function does not require an explicit number of clusters

- Detect outliers
  - outliers are defined as points for which there is no consensus

Why clustering aggregation?

- Improve the robustness of clustering algorithms
  - different algorithms have different weaknesses.
  - combining them can produce a better result.

Why clustering aggregation?

- Privacy preserving clustering
  - different companies have data for the same users. They can compute an aggregate clustering without sharing the actual data.

Complexity of Clustering Aggregation

- The clustering aggregation problem is NP-hard
  - the median partition problem [Barthelemy and LeClerc 1995].

- Look for heuristics and approximate solutions.

A simple 2-approximation algorithm

- The disagreement distance **D(C,P)** is a metric
- The algorithm **BEST**: Select among the input clusterings the clustering C^* that minimizes **$D(C^*)$.**
  - a 2-approximate solution. Why?

A 3-approximation algorithm

- The **BALLS** algorithm:
  - Select a point **x** and look at the set of points **B** within distance **½** of **x**
  - If the average distance of **x** to **B** is less than ¼ then create the cluster **B**$\cup$**{p}**
  - Otherwise, create a singleton cluster **{p}**
  - Repeat until all points are exhausted

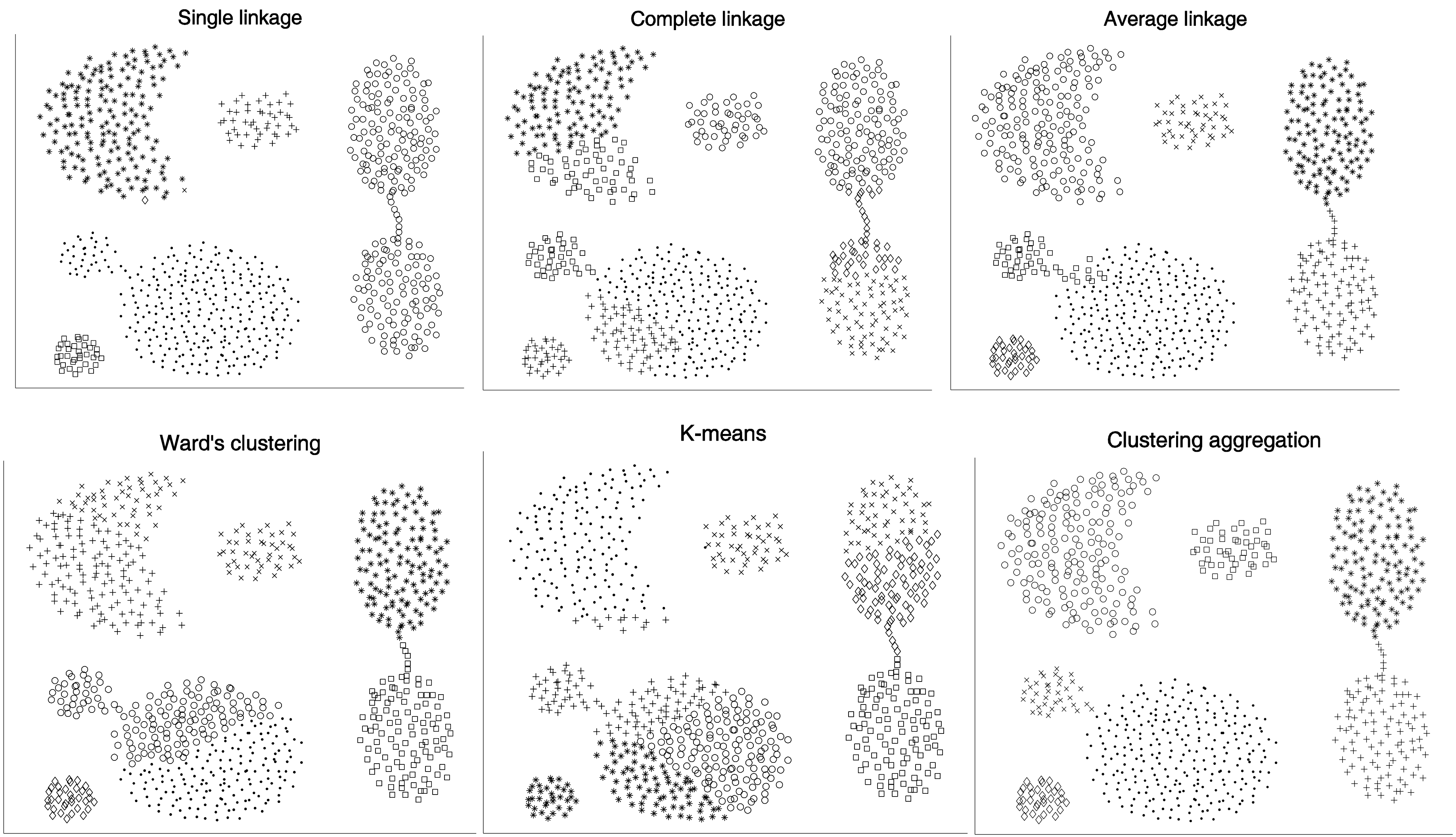
- Theorem: The **BALLS** algorithm has worst-case approximation factor **3**

Other algorithms

- **AGGLO**:
  - Start with all points in singleton clusters
  - Merge the two clusters with the smallest average inter-cluster edge weight
  - Repeat until the average weight is more than ½

- **LOCAL**:
  - Start with a random partition of the points
  - Remove a point from a cluster and try to merge it to another cluster, or create a singleton to improve the cost of aggregation.
  - Repeat until no further improvements are possible

Clustering Robustness



Lecture outline

- Soft (model-based) clustering and EM algorithm
- Clustering aggregation [A. Gionis, H. Mannila, P. Tsaparas: Clustering aggregation, ICDE 2004]
- Impossibility theorem for clustering [Jon Kleinberg, An impossibility theorem for clustering, NIPS 2002]

General form of impossibility results

- Define a set of simple **axioms** (properties) that a computational task should satisfy

- Prove that ***there does not exist an algorithm*** that can simultaneously satisfy all the axioms → **impossibility**

Computational task: clustering

- A ***clustering function*** operates on a set **X** of **n** points. **X = {1,2,…,n}**

- Distance function **d: X** x **X → R** with **d(i,j)≥0**, **d(i,j)=d(j,i)**, and **d(i,j)=0** only if **i=j**

- Clustering function **f**: **f(X,d) = Γ**, where **Γ** is a ***partition*** of **X**

Axiom 1: Scale invariance

- For **a>0**, distance function ***ad*** has values ***(ad)(i,j)=ad(i,j)***
- For any **d** and for any **a>0** we have **f(d) = f(ad)**
- The clustering function should not be sensitive to the changes in the units of distance measurement – should not have a built-in “length scale”

Axiom 2: Richness

- The ***range*** of $\mathbf{f}$ is equal to ***the set of partitions*** of $\mathbf{X}$

- For any $\mathbf{X}$ and any partition $\mathbf{\Gamma}$ of $\mathbf{X}$, there is a distance function on $\mathbf{X}$ such that $\mathbf{f(X,d) = \Gamma}$.

Axiom 3: Consistency

- Let **Γ** be a partition of **X**
- **d, d'** two distance functions on **X**
- **d'** is a **Γ**-transformation of **d**, if
  - For all **i,jϵ X** in the ***same cluster*** of **Γ**, we have **d'(i,j)≤d(i,j)**
  - For all **i,jϵ X** in ***different clusters*** of **Γ**, we have **d'(i,j)≥d(i,j)**
- **Consistency:** if **f(X,d)= Γ** and d' is a **Γ**-transformation of **d**, then **f(X,d')= Γ**.

Axiom 3: Consistency

- **Intuition:** Shrinking distances between points inside a cluster and expanding distances between points in different clusters does not change the result

Examples

- Single-link agglomerative clustering
- Repeatedly merge clusters whose closest points are at minimum distance
- Continue until a stopping criterion is met
  - ***k***-cluster stopping criterion: continue until there are k clusters
  - distance-***r*** stopping criterion: continue until all distances between clusters are larger than r
  - scale-***a*** stopping criterion: let d* be the maximum pairwise distance; continue until all distances are larger than ad*

Examples (cont.)

- Single-link agglomerative clustering with ***k***-cluster stopping criterion does not satisfy richness axiom
- Single-link agglomerative clustering with distance-***r*** stopping criterion does not satisfy scale-invariance property
- Single-link agglomerative clustering with scale-***a*** stopping criterion does not satisfy consistency property

Centroid-based clustering and consistency

- **k**-centroid clustering:
  - **S** subset of **X** for which $\sum_{i \in X} \min_{j \in S}\{d(i,j)\}$ is minimized
  - Partition of **X** is defined by assigning each element of **X** to the centroid that is the ***closest*** to it

- **Theorem:** for every **k≥2** and for **n** sufficiently large relative to **k**, the **k**-centroid clustering function does not satisfy the consistency property

k-centroid clustering and the consistency axiom

- Intuition of the proof
- Let **$k=2$** and **X** be partitioned into parts **Y** and **Z**
- **$d(i,j) \leq r$** for every **$i,j \in Y$**
- **$d(i,j) \leq \varepsilon$**, with **$\varepsilon<r$** for every **$i,j \in Z$**
- **$d(i,j) > r$** for every **$i \in Y$** and **$j \in Z$**

- Split part **Y** into subparts **Y_1** and **Y_2**
- Shrink distances in **Y_1** appropriately
- What is the result of this shrinking?

Impossibility theorem

- For **$n \geq 2$**, there is no clustering function that satisfies all three axioms of scale-invariance, richness and consistency

Impossibility theorem (proof sketch)

- A partition Γ’ is a refinement of partition Γ, if each cluster C’ϵ Γ’ is included in some set Cϵ Γ
- There is a partial order between partitions: Γ’≤ Γ
- Antichain of partitions: a collection of partitions such that no one is a refinement of others
- Theorem: If a clustering function f satisfies scale-invariance and consistency, then, the range of f is an anti-chain

What does an impossibility result really mean

- Suggests a technical underpinning for the difficulty in unifying the initial, informal concept of clustering
- Highlights basic trade-offs that are inherent to the clustering problem
- Distinguishes how clustering methods resolve these tradeoffs (by looking at the methods not only at an operational level)