CS573 Data Privacy and Security

Anonymization methods

Li Xiong

Today

- Recap/Taxonomy of Anonymization
  - Microdata anonymization
- Microaggregation based anonymization

Taxonomy of Anonymization

- Problem Settings/scenarios
- Types of data
- Anonymization techniques
- Information metrics

Problem Settings/Scenarios

- One-time single provider release (base setting)
- Multiple release publishing
- Continuous release publishing
- Collaborative/distributed publishing
  - Slawek's lecture

Types of data

- Relational data (tabular data)
- High dimensional transaction data
  - E.g.Market basket, web queries
- Moving objects data (temporal/spatial data)
  - E.g. Location based services
- Textual data
  - E.g. Medical documents, James’ lecture

Types of Attributes

- Continuous: attribute is numeric and arithmetic operations can be performed on it
- Categorical: attribute takes values over a finite set and standard arithmetic operations don't make sense
  - Ordinal: ordered range of categories
    - ≤, min and max operations are meaningful
  - Nominal: unordered
    - only equality comparison operation is meaningful

Anonymization methods

- Non-perturbative: don't distort the data
  - Generalization
  - Suppression
- Perturbative: distort the data
  - Microaggregation/clustering
  - Additive noise
- Anatomization and permutation
  - De-associate relationship between QID and sensitive attribute

Measuring Privacy/Utility tradeoff

- How to measure two goals?
- k-Anonymity: a dataset satisfies k-anonymity for $k > 1$ if at least k records exist for each combination of quasi-identifier values
- Assuming k-anonymity is enough protection against disclosure risk, one can concentrate on information loss measures

Information Metrics

- General purpose metrics
- Special purpose metrics
- Trade-off metrics

General Purpose Metrics

- General idea: measure "similarity" between the original data and the anonymized data
- Minimal distortion metric (Samarati 2001; Sweeney 2002, Wang and Fung 2006)
  - Charge a penalty to each instance of a value generalized or suppressed (independently of other records)
- ILoss (Xiao and Tao 2006)
  - Charge a penalty when a specific value is generalized

General Purpose Metrics cont.

- Discernibility Metric (DM) (K-OPTIMIZE, Mondrian, l-diversity …)
  - Charge a penalty to each record for being indistinguishable from other records
- Average Equivalence Group size
  - What's the optimal equivalence group size?

Special Purpose Metrics

- Application dependent
- Classification: Classification metric (CM) (Iyengar 2002)
  - Charge a penalty for each record suppressed or generalized to a group in which the record's class is not the majority class
- Query
  - Query error: count queries
  - Query imprecision: overlapped range

Today

- Recap/Taxonomy of Anonymization
- Microaggregation based anonymization

Critique of Generalization/Suppression

- Satisfying k-anonymity using generalization and suppression is NP-hard
- Computational cost of finding the optimal generalization
- How to determine the subset of appropriate generalizations
  - semantics of categories and intended use of data
  - e.g., ZIP code:
    - {08201, 08205} -> 0820* makes sense
    - {08201, 05201} -> 0*201 doesn't

- How to apply a generalization
  - globally
    - may generalize records that don't need it
  - locally
    - difficult to automate and analyze
    - number of generalizations is even larger
- Generalization and suppression on continuous data are unsuitable
  - a numeric attribute becomes categorical and loses its numeric semantics, e.g. age

- How to optimally combine generalization and suppression is unknown
- Use of suppression is not homogenous
  - suppress entire records or only some attributes of some records
  - blank a suppressed value or replace it with a neutral value

Microaggregation/Clustering

- Two steps:
  - Partition original dataset into clusters of similar records containing at least *k records*
  - *For each cluster, compute an aggregation operation and use it to replace the original records*
    - *e.g., mean for continuous data, median for categorical data*

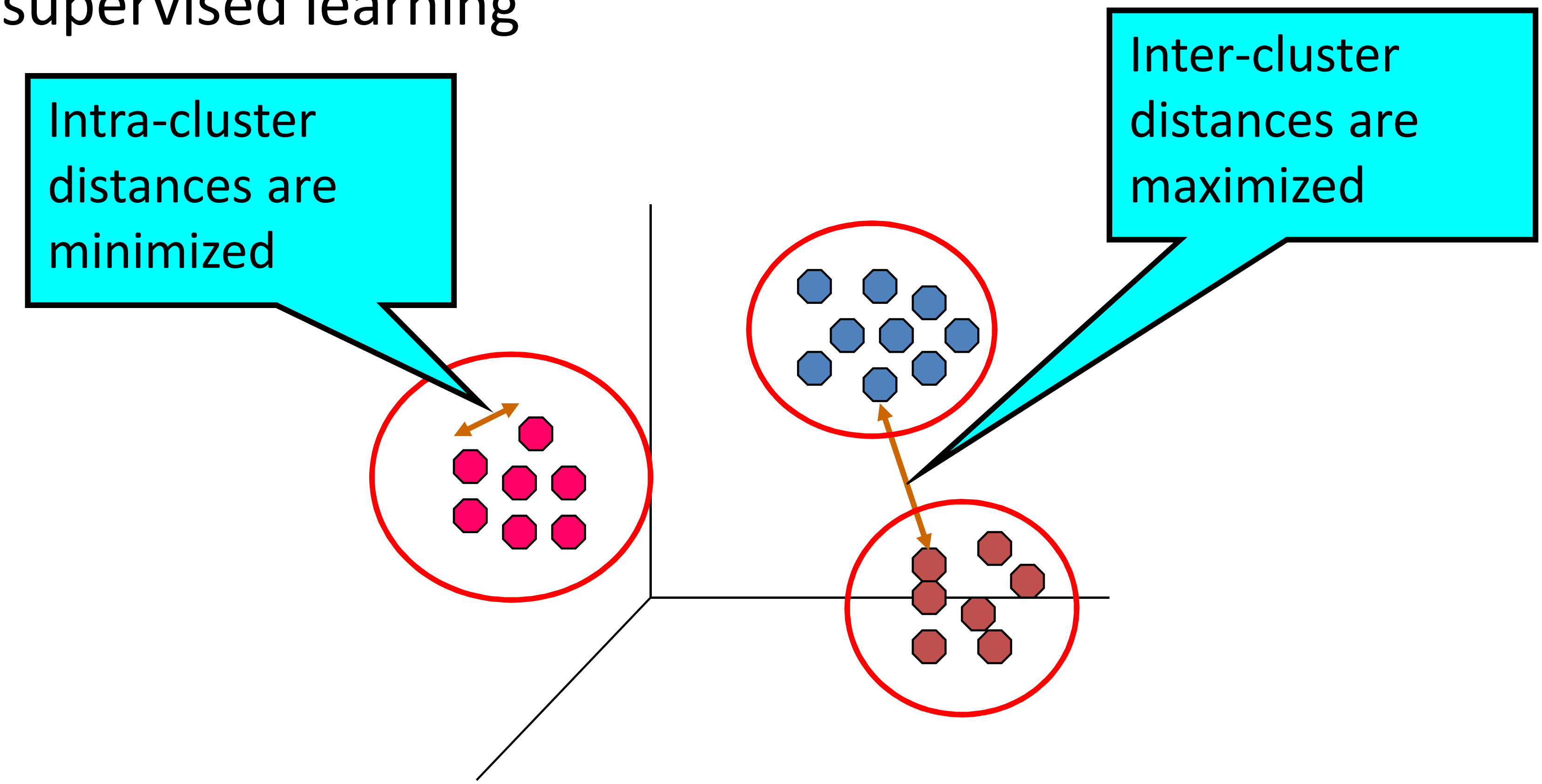
Advantages

- a unified approach, unlike combination of generalization and suppression
- Near-optimal heuristics exist
- Doesn't generate new categories
- Suitable for continuous data without removing their numeric semantics

- Reduces data distortion
  - *K*-anonymity requires an attribute to be generalized or suppressed, even if all but one tuple in the set have the same value.
  - Clustering allows a cluster center to be published instead, “enabling us to release more information.”

What is Clustering?

- Finding groups of objects (clusters)
  - Objects similar to one another in the same group
  - Objects different from the objects in other groups
- Unsupervised learning

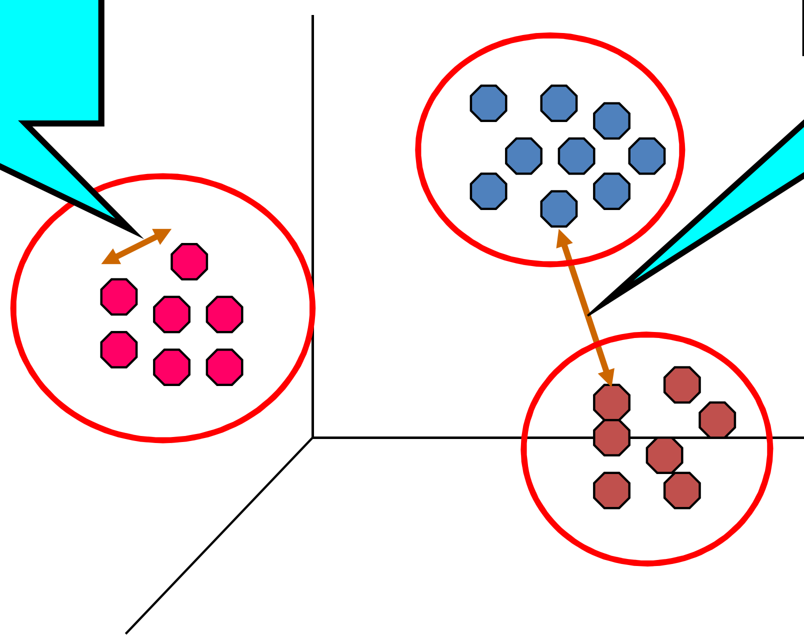


Clustering Applications

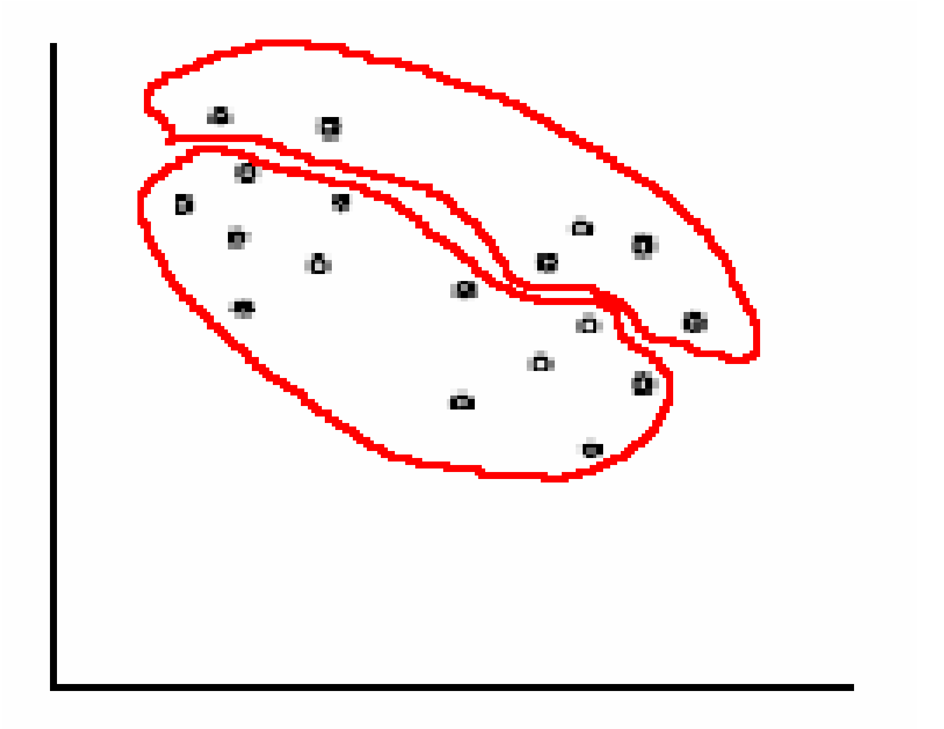
- Marketing research

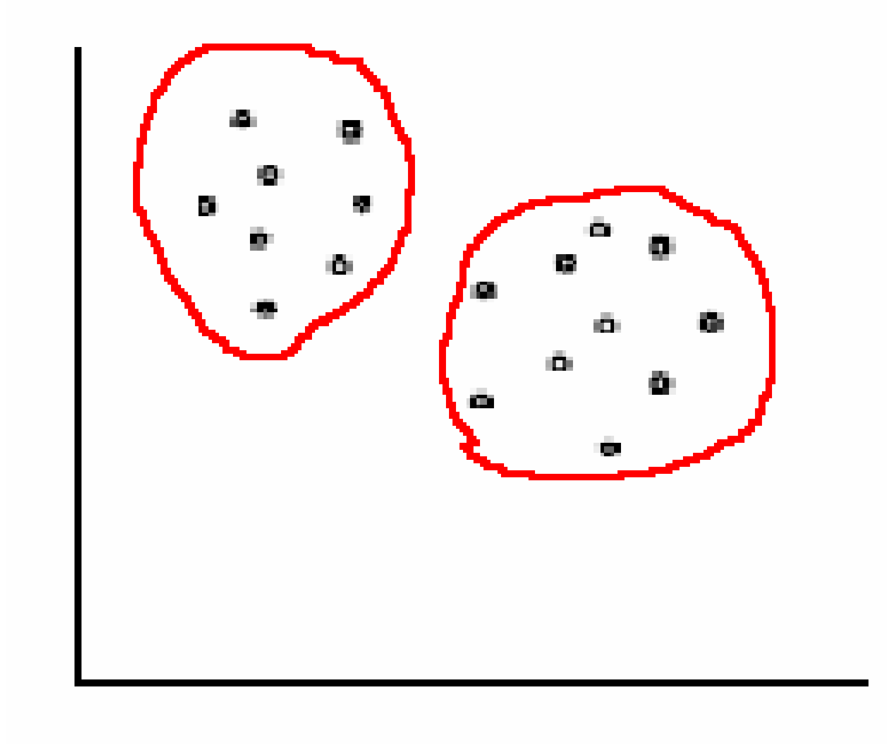
Quality: What Is Good Clustering?

- Agreement with “ground truth”
- A <u>good clustering</u> will produce high quality clusters with
  - Homogeneity - high <u>intra-class</u> similarity
  - Separation - low <u>inter-class</u> similarity



Bad Clustering vs. Good Clustering





Similarity or Dissimilarity between Data Objects

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Euclidean distance

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$$

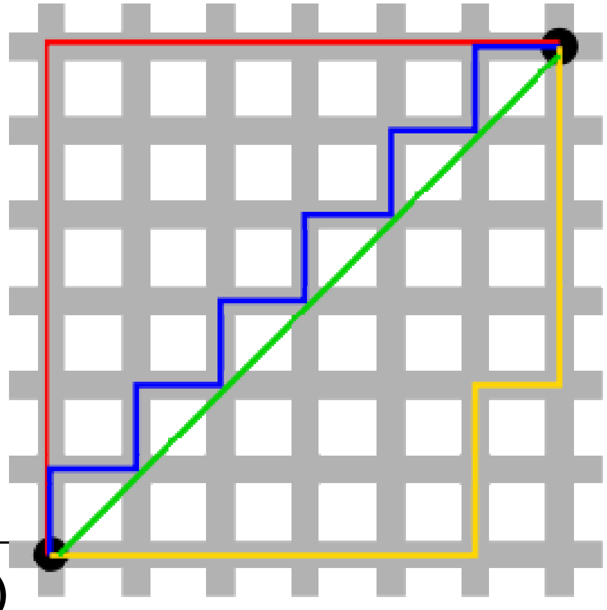
- Manhattan distance

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+\ldots+|x_{i_p}-x_{j_p}|$$

- *Minkowski distance*

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+\ldots+|x_{i_p}-x_{j_p}|^q)}$$

- Weighted



Other Similarity or Dissimilarity Metrics

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Pearson correlation $$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}.$$
- Cosine measure $$\frac{X_i \bullet X_j}{||X_i|| \cdot ||X_j||}$$
- Jaccard coefficient $$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$
- KL divergence, Bregman divergence, …

Different Attribute Types

- To compute $|x_{if} - x_{jf}|$
  - f is numeric (interval or ratio scale)
    - Normalization if necessary
    - Logarithmic transformation for ratio-scaled values

$$x_{if} = Ae^{Bt} \qquad y_{if} = \log(x_{if})$$

  - f is ordinal
    - Mapping by rank $z_{if} = \dfrac{r_{if} - 1}{M_f - 1}$
  - f is nominal
    - Mapping function

    $|x_{if} - x_{jf}| = 0$ if $x_{if} = x_{jf}$, or 1 otherwise
    - Hamming distance (edit distance) for strings

Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Others

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, s.t., the sum of squared distance is minimized

$$\Sigma_{i=1}^{k}\Sigma_{p\in C_i}(p-m_i)^2$$

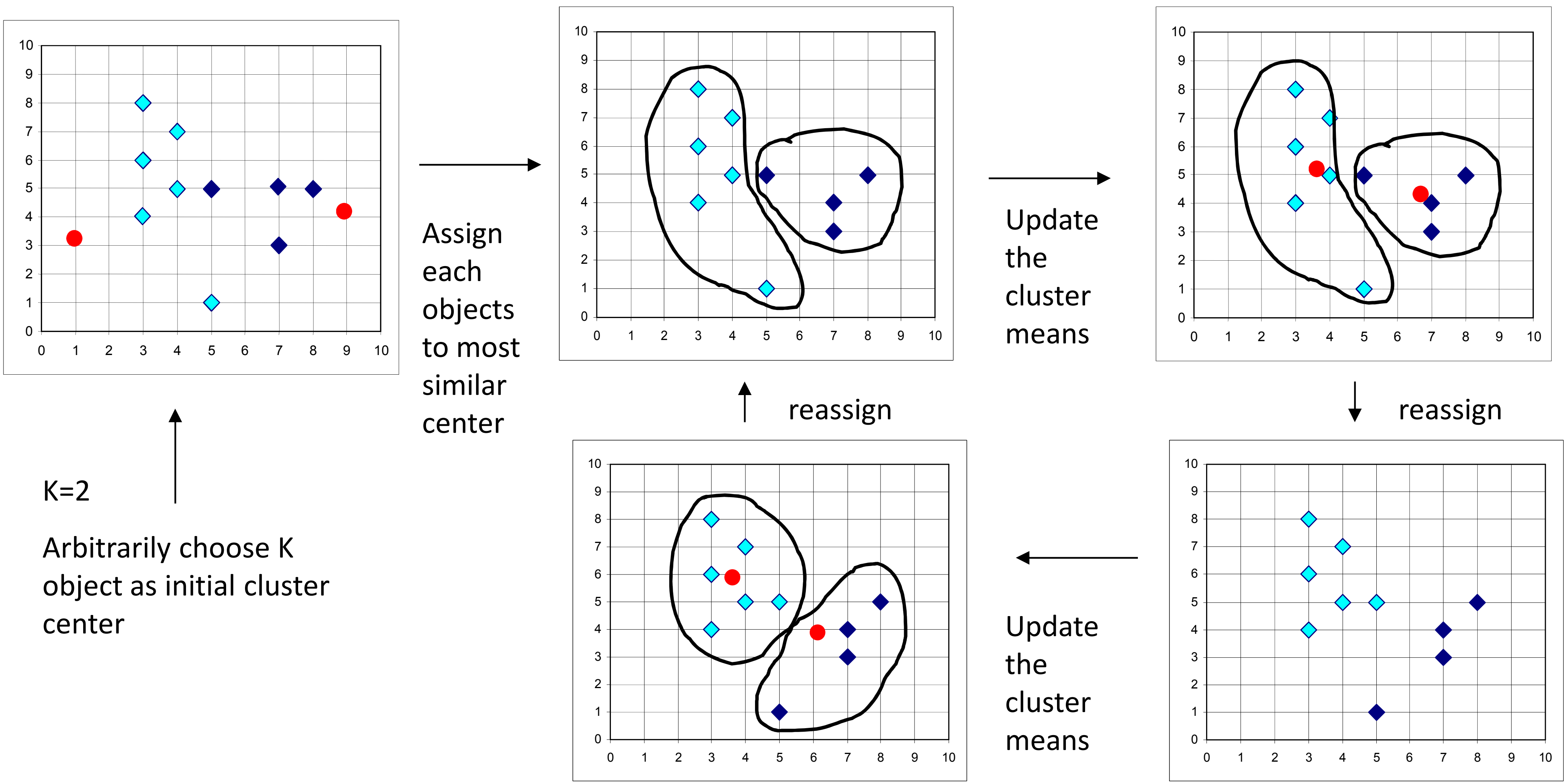
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

K-Means Clustering: Lloyd Algorithm

- Given *k*, and randomly choose *k* initial cluster centers
- Partition objects into *k* nonempty subsets by assigning each object to the cluster with the nearest centroid
- Update centroid, i.e. *mean point* of the cluster
- Go back to Step 2, stop when no more new assignment

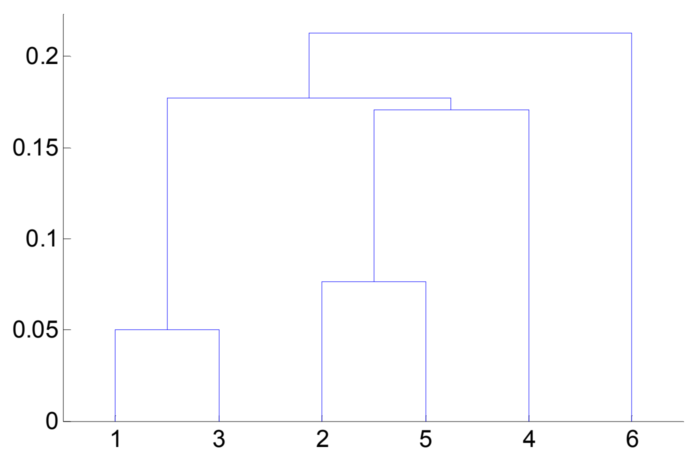
The *K-Means* Clustering Method

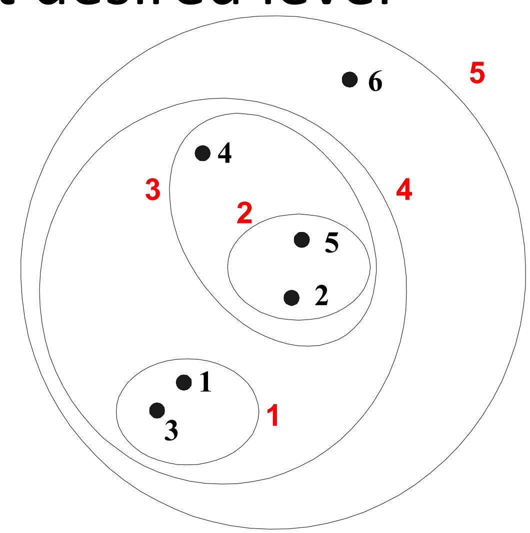
- Example



Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  – A tree like diagram representing a hierarchy of nested clusters
  – Clustering obtained by cutting at desired level





Hierarchical Clustering

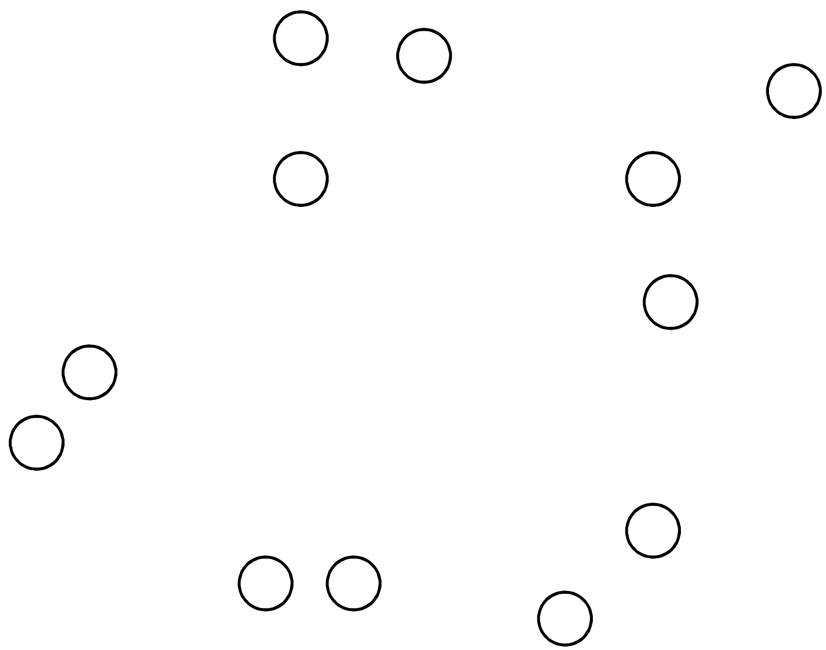
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)

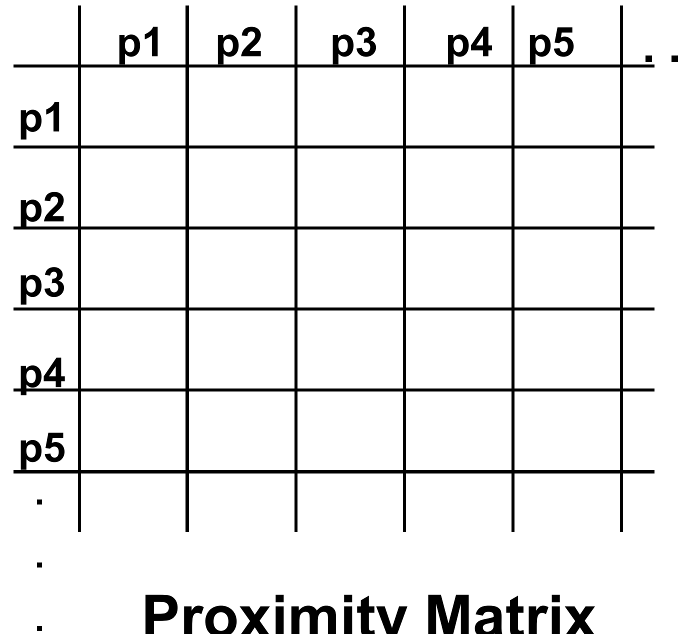
Agglomerative Clustering Algorithm

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

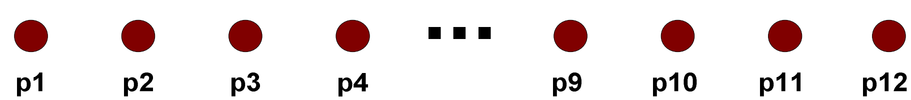
Starting Situation

- Start with clusters of individual points and a proximity matrix

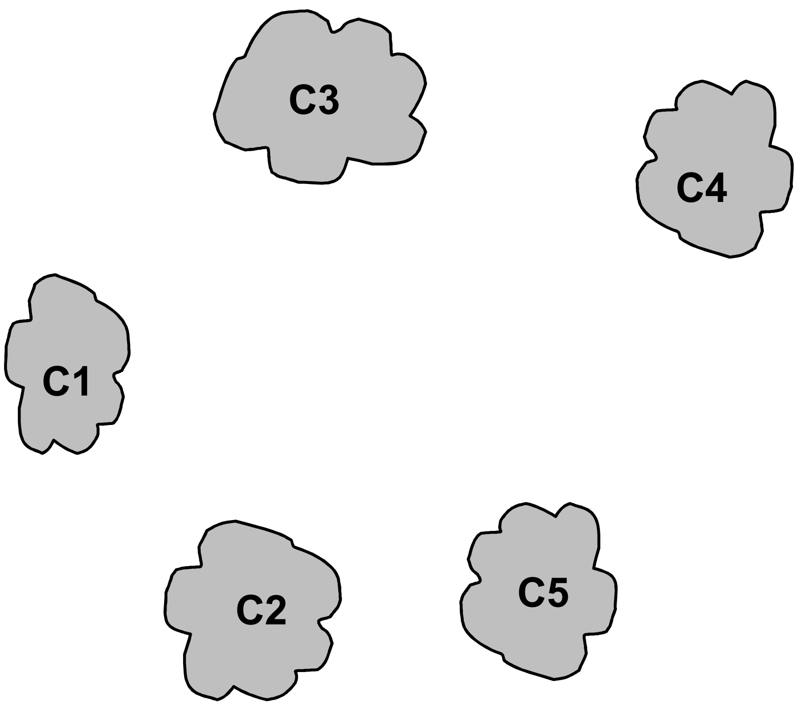




Proximity Matrix

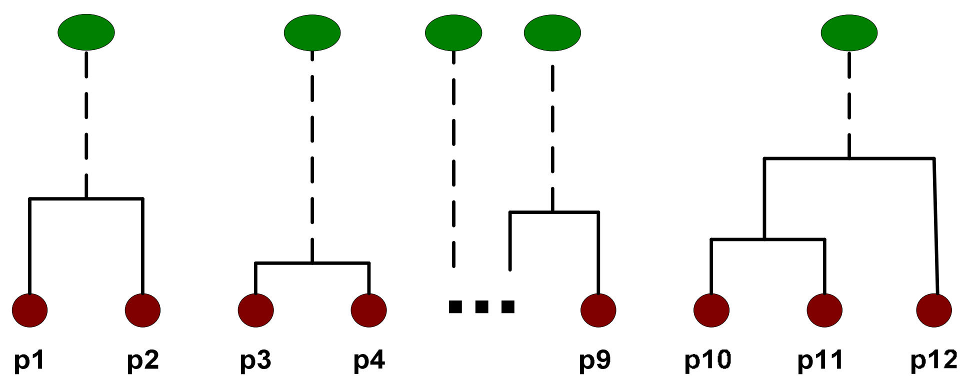


Intermediate Situation

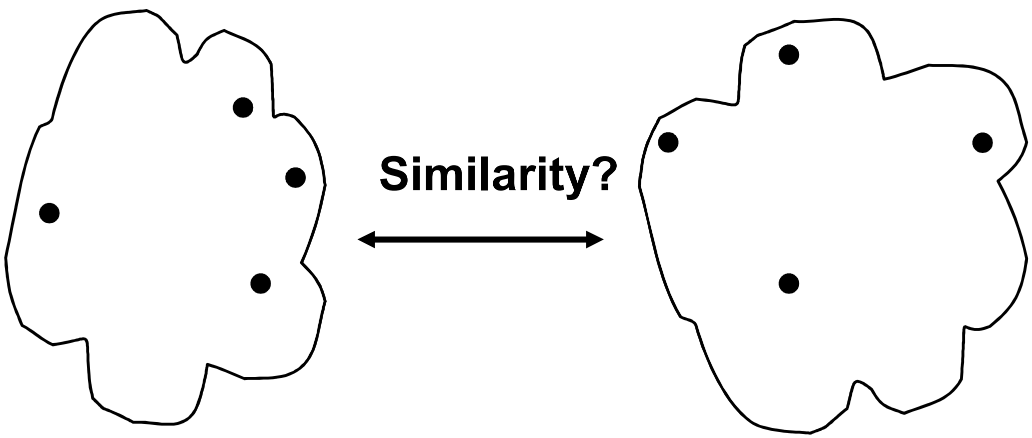


	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



How to Define Inter-Cluster Similarity

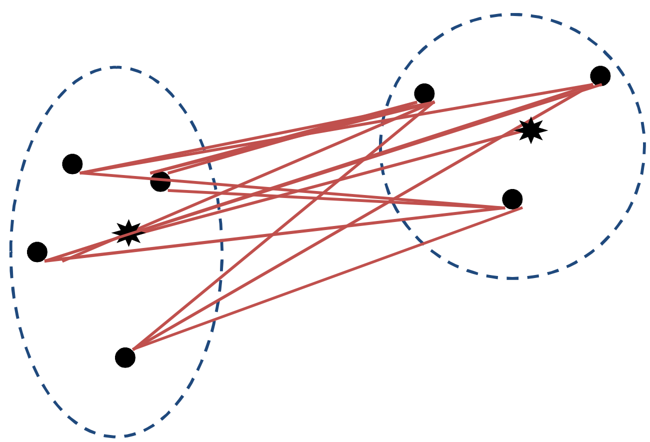


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

Distance Between Clusters

- ***Single Link***: smallest distance between points
- ***Complete Link:*** largest distance between points
- ***Average Link:*** average distance between points
- ***Centroid:*** distance between centroids



Clustering for Anonymization

- Are they directly applicable?
- Which algorithms are directly applicable?
  - K-means; hierarchical

Anonymization And Clustering

- *k*-Member Clustering Problem
  - From a given set of *n* records, find a set of clusters such that
    - Each cluster contains at least *k* records, and
    - The total intra-cluster distance is minimized.
  - The problem is NP-complete

Anonymization using Microaggregation or Clustering

- Practical Data-Oriented Microaggregation for Statistical Disclosure Control, Domingo-Ferrer, TKDE 2002
- Ordinal, Continuous and Heterogeneous k-anonymity through microaggregation, Domingo-Ferrer, DMKD 2005
- Achieving anonymity via clustering, Aggarwal, PODS 2006
- Efficient k-anonymization using clustering techniques, Byun, DASFAA 2007

Multivariate microaggregation algorithm

- MDAV-generic: Generic version of MDAV algorithm (Maximum Distance to Average Vector) from previous papers
- Works with any type of data (continuous, ordinal, nominal), aggregation operator and distance calculation

MDAV-generic(R: dataset, k: integer)

while |R| ≥ 3k

1. compute average record ~*x* of all records in R

2. find most distant record x_r *from ~x*

3. find most distant record x_s *from* x_r

4. form two clusters from k-1 records closest to x_r *and k-1 closest to* x_s

5. Remove the clusters from R and run MDAV-generic on the remaining dataset

end while

if 3k-1 ≤ |R| ≤ 2k

1. compute average record ~x of remaining records in R

2. find the most distant record x_r *from ~x*

3. form a cluster from k-1 records closest to ~x

4. form another cluster containing the remaining records

else (fewer than 2k records in R) form a new cluster from the remaining records

MDAV-generic for continuous attributes

- use arithmetic mean and Euclidean distance
- standardize attributes (subtract mean and divide by standard deviation) to give them equal weight for computing distances
- After MDAV-generic, destandardize attributes

MDAV-generic for categorical attributes

- The distance between two oridinal attributes a and b in an attribute V_i:
  - $d_{ord}(a,b) = (|\{i| \leq i < b\}|) / |D(V_i)|$
    - *i.e., the number of categories separating a and b divided by the number of categories in the attribute*
- The distance between two nominal attributes is defined according to equality: 0 if they're equal, else 1

Empirical Results

- Continuous attributes
  - From the U.S. Current Population Survey (1995)
    - 1080 records described by 13 continuous attributes
    - Computed k-anonymity for k = 3, ..., 9 and quasi-identifiers with 6 and 13 attributes
- Categorical attributes
  - From the U.S. Housing Survey (1993)
    - Three ordinal and eight nominal attributes
    - Computed k-anonymity for k = 2, ..., 9 and quasi-identifiers with 3, 4, 8 and 11 attributes

- IL measures for continuous attributes
  - IL1 = mean variation of individual attributes in original and k-anonymous datasets
  - IL2 = mean variation of attribute means in both datasets
  - IL3 = mean variation of attribute variances
  - IL4 = mean variation of attribute covariances
  - IL5 = mean variation of attribute Pearson's correlations
  - IL6 = 100 times the average of IL1-6

Quasi-identifier length	k	IL_1	IL_2	IL_3	IL_4	IL_5	IL
6	3	0.131	0	0	0.036	0.007	3.48
6	6	0.174	0	0	0.075	0.013	5.24
6	9	0.203	0	0	0.129	0.017	6.98
6	12	0.185	0	0	0.166	0.020	7.42
13	3	0.907	0	0	0.058	0.016	19.62
13	6	1.389	0	0	0.134	0.032	31.10
13	9	1.535	0	0	0.161	0.039	34.70
13	12	1.564	0	0	0.164	0.046	35.48

- MDAV-generic preserves means and variances (IL2 and IL3)
- The impact on the non-preserved statistics grows with the quasi-identifier length, as one would expect
- For a fixed-quasi-identifier length, the impact on the non-preserved statistics grows with *k*

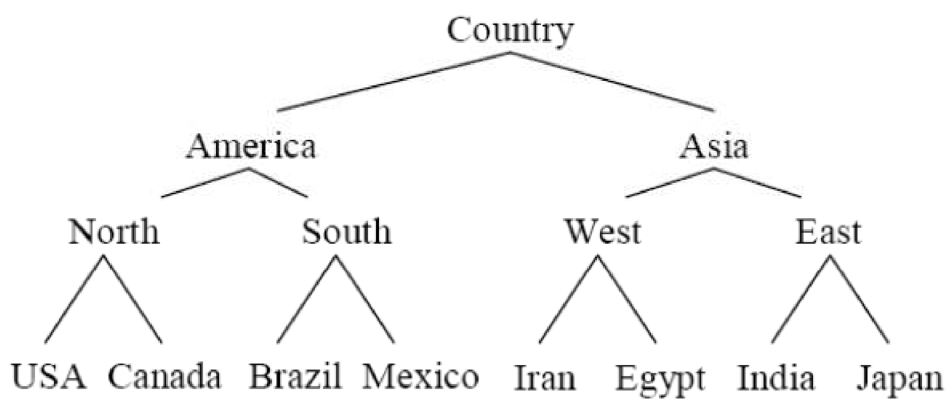
Anonymization using Microaggregation or Clustering

- Practical Data-Oriented Microaggregation for Statistical Disclosure Control, Domingo-Ferrer, TKDE 2002
- Ordinal, Continuous and Heterogeneous k-anonymity through microaggregation, Domingo-Ferrer, DMKD 2005
- Achieving anonymity via clustering, Aggarwal, PODS 2006
- Efficient k-anonymization using clustering techniques, Byun, DASFAA 2007

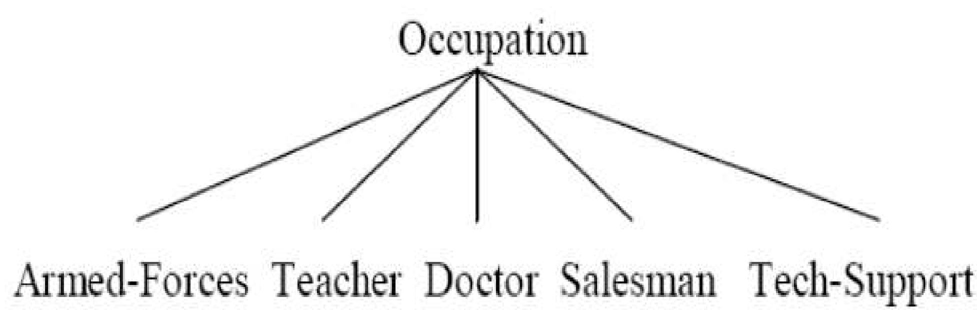
Distance between two categorical values

- Equally different to each other.
  - 0 if they are the same
  - 1 if they are different

- Relationships can be easily captured in a taxonomy tree.



Taxonomy tree of Country

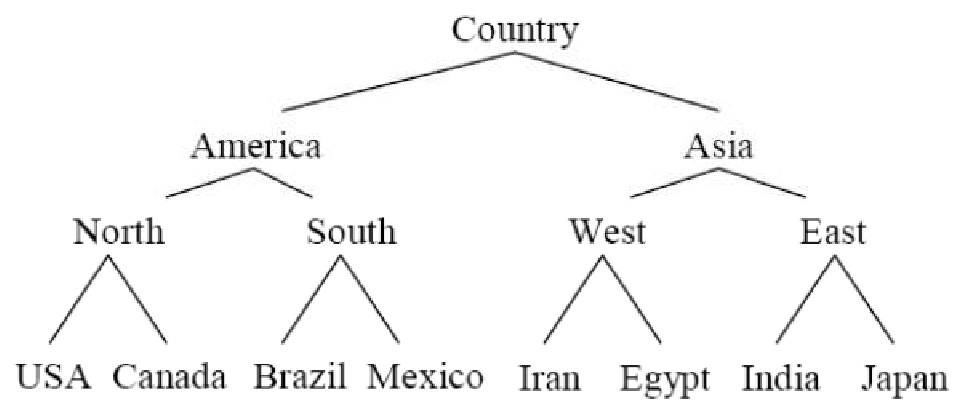


Taxonomy tree of Occupation

Distance between two categorical values

- Definition

  Let D be a categorical domain and TD be a taxonomy tree defined for D. The normalized distance between two values v_i, v_j $\in$ D is defined as:



Taxonomy tree of Country

$$\delta_C(v_1,\ v_2)\ =\ H(\Lambda(v_i,\ v_j))\ /\ \ H(T_{\mathcal{D}})$$

where Λ (x, y) is the subtree rooted at the lowest common ancestor of x and y, and H(T) represents the height of tree T.

Example:

The distance between India and USA is 3/3 = 1.

The distance between India and Iran is 2/3 = 0.66.

Cost Function - Information loss (IL)

- The amount of distortion (i.e., information loss) caused by the generalization process.

  Note: Records in each cluster are generalized to share the same quasi-identifier value that represents every original quasi-identifier value in the cluster.

  – Definition: Let e = {$r_1, \ldots, r_k$} be a cluster (i.e., equivalence class). Then the amount of information loss in e, denoted by IL(e), is defined as:

$$\begin{aligned} \mathrm{IL}(e) &= |e| \cdot \mathrm{D}(e) \\ \mathrm{D}(e) &= \sum_{i=1,\ldots,m} \frac{(MAX_{N_i} - MIN_{N_i})}{|N_i|} \\ &\quad + \sum_{j=1,\ldots,n} \frac{H(\Lambda(\cup_{C_j}))}{H(T_{C_j})} \end{aligned}$$

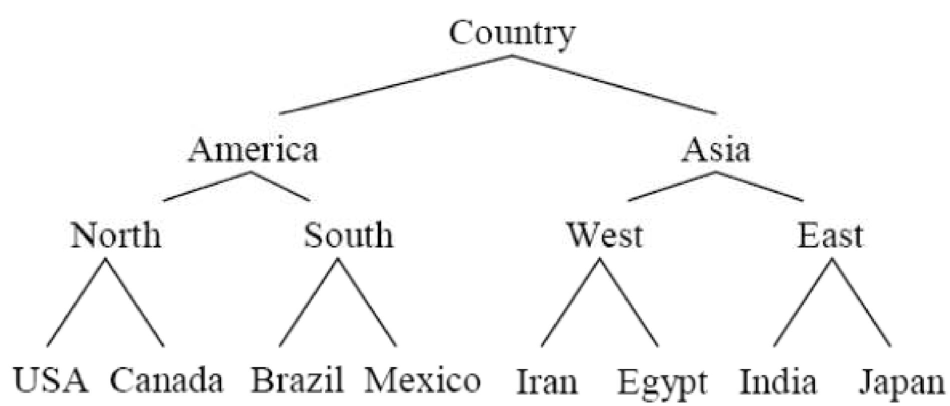
where |e| is the number of records in e, |N| represents the size of numeric domain N, $\Lambda(\cup_{C_j})$ is the subtree rooted at the lowest common ancestor of every value in $\cup_{C_j}$, and H(T) is the height of tree T.

Cost Function - Information loss (IL)

Example

	Age	Country	Occupation	Salary	Diagnosis
r1	41	USA	Armed-Forces	≥50K	Cancer
r2	57	India	Tech-support	<50K	Flu
r3	40	Canada	Teacher	<50K	Obesity
r4	38	Iran	Tech-support	≥50K	Flu
r5	24	Brazil	Doctor	≥50K	Cancer
r6	45	Greece	Salesman	<50K	Fever

Taxonomy tree of Country



Cluster e_1

Age	Country	Occupation	Salary	Diagnosis
41	USA	Armed-Forces	≥50K	Cancer
40	Canada	Teacher	<50K	Obesity
24	Brazil	Doctor	≥50K	Cancer

$IL(e_1) = 3 \cdot D(e_1)$

$D(e_1) = (41-24)/33 + (2/3) + 1 = 2.1818...$

$IL(e_1) = 3 \cdot 2.1818... = 6.5454...$

Cluster e_2

Age	Country	Occupation	Salary	Diagnosis
41	USA	Armed-Forces	≥50K	Cancer
57	India	Tech-support	<50K	Flu
24	Brazil	Doctor	≥50K	Cancer

$IL(e_2) = 3 \cdot D(e_2)$

$D(e_2) = (57-24)/33 + (3/3) + 1 = 3$

$IL(e_2) = 3 \cdot 3 = 9$

Greedy Algorithm

- Find k-member clusters, one cluster at a time
- Assign remaining <k points to the previous clusters

Greedy k-member clustering algorithm

Function ***greedy_k_member_clustering*** **(S, k)**

Input: a set of records S and a threshold value k.

Output: a set of clusters each of which contains at least k records.

```
if( | S | ≤ k )
   return S;
end if;

result = ∅;
r = a randomly picked record from S;
while( | S | ≥ k )
   r = the furthest record from r;
   S = S – {r};
   c = {r};
   while( | c | < k )
      r = find_best_record(S, c);
      S = S – {r};
      c = c ∪ {r};
   end while;
   result = result ∪ {c};
end while;
while( | S | ≠ 0 )
   r = a randomly picked record from S;
   S = S – {r};
   c = find_best_cluster(result, r);
   c = c ∪ {r};
end while;
return result;
```

End;

Function ***find_best_record*** **(S, c)**

Input: a set of records S and a cluster c.

Output: a record $r \in S$ such that $IL(c \cup \{r\})$ is minimal.

```
n = |S|;
min = ∞;
best = null;
for(i = 1,…n)
   r = i-th record in S;
   diff = IL(c ∪ {r}) – IL(c);
   if( diff < min )
      min = diff;
      best = r;
   end if;
end for;
return best;
```

End;

Function ***find_best_cluster*** **(C, r)**

Input: a set of clusters C and a record r.

Output: a cluster $c \in C$ such that $IL(c \cup \{r\})$ is minimal.

```
n = |C|;
min = ∞;
best = null;
for(i = 1,…n)
   c = i-th cluster in C;
   diff = IL(c ∪ {r}) – IL(c);
   if( diff < min )
      min = diff;
      best = c;
   end if;
end for;
return best;
```

End;

classification metric (CM)

– preserve the correlation between quasi-identifier and class labels (non-sensitive values)

$$CM = \sum_{all\ rows} Penalty(row\ r)\ /\ N$$

Where N is the total number of records, and Penalty(row r) = 1 if r is suppressed or the class label of r is different from the class label of the majority in the equivalence group.

Experimentl Results

- Experimental Setup
  - Data: Adult dataset from the UC Irvine Machine Learning Repository
    - 10 attributes (2 numeric, 7 categorical, 1 class)
  - Compare with 2 other algorithms
    - Median partitioning (Mondrian algorithm)
    - *k*-Nearest neighbor

Experimentl Results

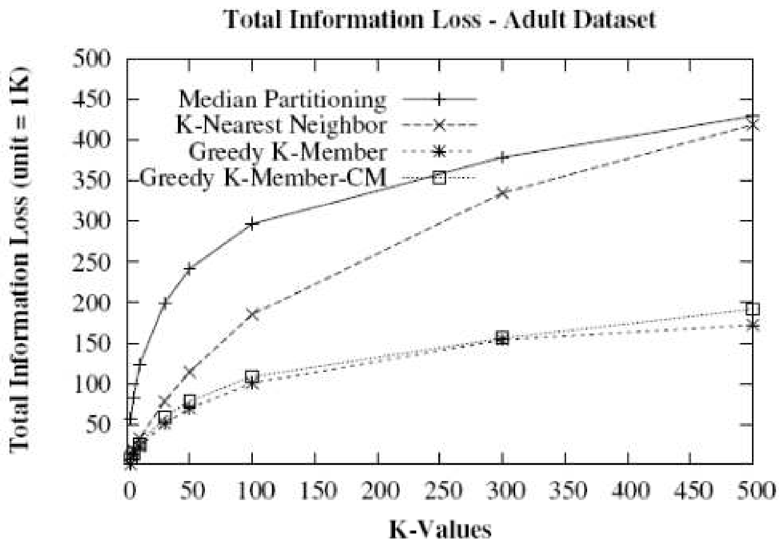


Figure 9: Information Loss Metric

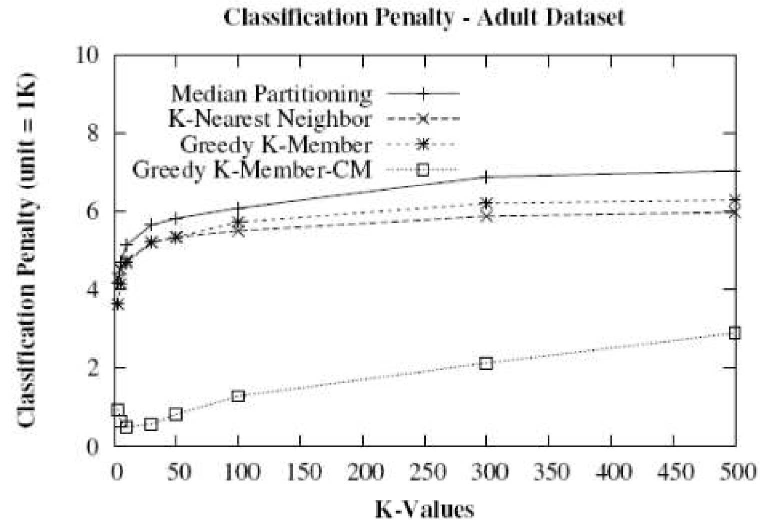


Figure 11: Classification Metric

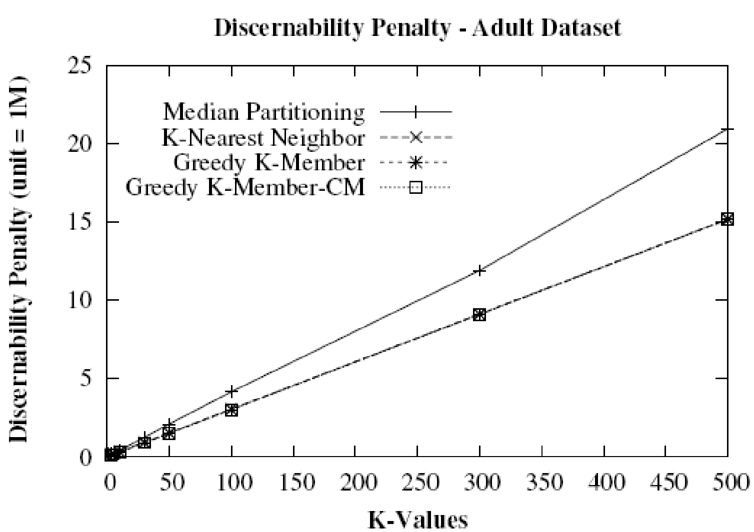


Figure 10: Discernibility Metric

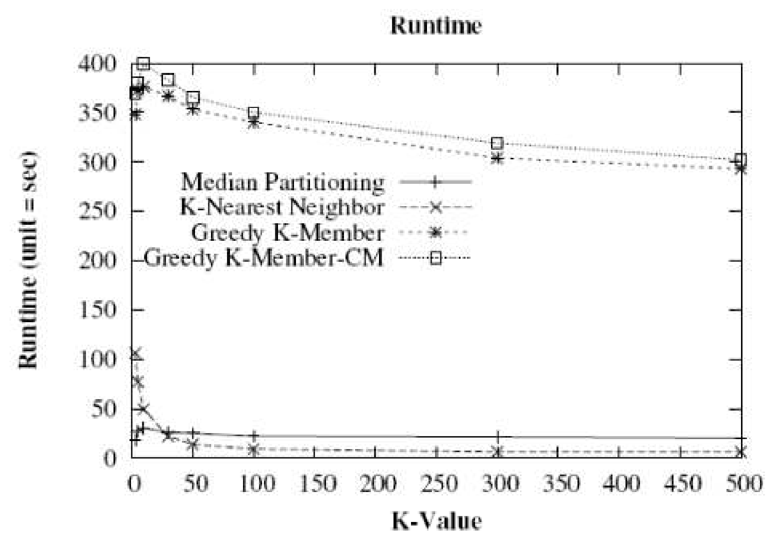


Figure 12: Execution Time

Conclusion

- Transforming the k-anonymity problem to the k-member clustering problem
- Overall the Greedy Algorithm produced better results compared to other algorithms at the cost of efficiency