Enforcing constraints using Correlation Clustering

CompSci 590.03

Instructor: Ashwin Machanavajjhala

Duke UNIVERSITY

Summary of Hash-based Blocking

- Complex boolean functions can be built to optimize recall using a training set of matches and non-matches

- Locality sensitive hashing functions can strongly distinguish pairs that are close from pairs that are far.

- AND and OR construction help amplify the distinguishing capability of locality sensitive functions.

Duke UNIVERSITY

Outline

- Definition of Blocking
- Hash-based Blocking
  - Boolean functions over attributes
  - minHash: Locality Sensitive Hashing
- Neighborhood-based Blocking
  - Merge/Purge
  - Canopy Clustering

Duke UNIVERSITY

Blocking Algorithms 2

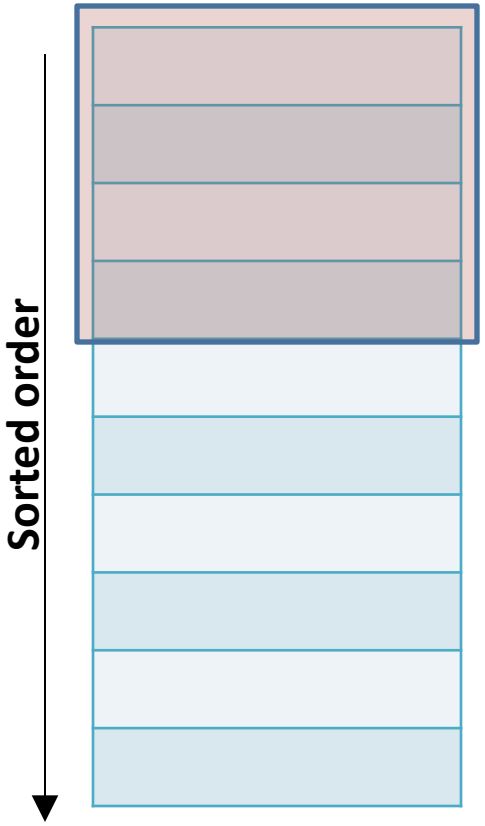
- Pairwise Similarity/Neighborhood based blocking
  - Nearby nodes according to a similarity metric are clustered together
  - Results in non-disjoint canopies.

- Techniques
  - Sorted Neighborhood Approach [Hernandez et al SIGMOD'95]
  - Canopy Clustering [McCallum et al KDD'00]

Duke UNIVERSITY

Sorted Neighborhood [Hernandez et al SIGMOD'95]

- Compute a **Key** for each record.
- **Sort** the records based on the key.
- **Merge**: Check whether a record matches with *(w-1)* previous records.
  - Implementation?
- Perform multiple passes with different keys

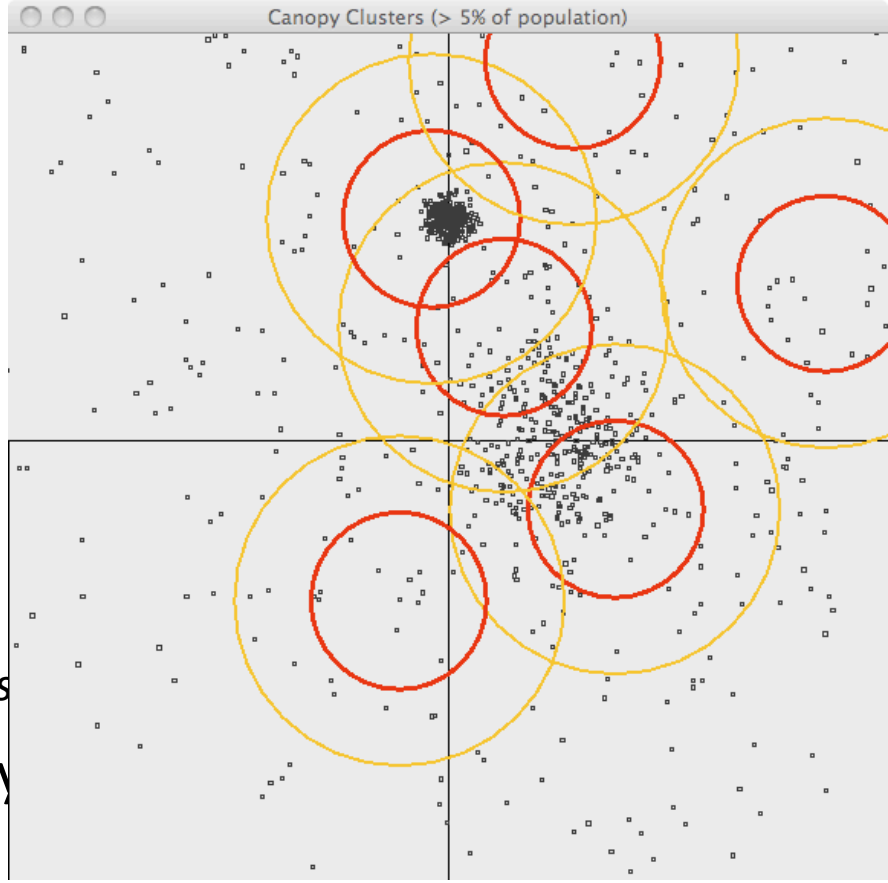


Canopy Clustering [McCallum et al KDD'00]

Input: Mentions M,
$d(x,y)$, a distance metric,
thresholds $T_1 > T_2$

Algorithm:

1. Pick a random element x from M
2. Create new canopy C_x using mentions y s.t. $d(x,y) < T_1$
3. Delete all mentions y from M s.t. $d(x,y) < T_2$ *(from consideration in this*
4. Return to Step 1 if M is not empty



Summary of Blocking

- $O(|R|^2)$ pairwise computations can be prohibitive.
  - Blocking eliminates comparisons on a large fraction of non-matches.

- Hash-based Blocking:
  - Construct (one or more) hash keys from features
  - Records not matching on any key are not compared.

- Neighbohood based Blocking:
  - Form overlapping canopies of records based on similarity.
  - Only compare records within a cluster.

Duke UNIVERSITY

This Class

- Enforcing Constraints in ER
  - Exclusivity: Bipartite Matching
  - Transitivity: Correlation Clustering

Constraints

- **Transitivity**:
  If x and y match, y and z match, then x and z must match
  - Useful in deduplication

- **Exclusivity**:
  If x matches with y, then z cannot match with y
  - Useful in record linkage (matches across two datasets)
  - Each dataset does not have any duplicates.

- **Relational Constraints**:
  If x and y match, then z and w should match
  - If movies are the same, then directors must be the same
  - (We will see in next class)

Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	Transitivity: x=y & y=z => x=	
N	y: x and y must refer to distinct entities Relational: If x,y don't ma then z,w cannot match *If two venues don't mat then their papers don't match*	Soft Exclusivity: x and y are very likely different elements

Note that some of the constraints may be **relational** and require joins

May be **directional** or **bidirectional**

Constraints can be **recursive**, e.g., if two authors have matching co-authors, then they match

Duke UNIVERSITY

Match Dependencies

When matching decisions depend on other matching decisions (in other words, matching decisions are not made independently for each pair), we refer to the approach as ***collective***

Algorithms for Enforcing Constraints

- Record linkage - propagation through exclusivity
  - Weighted k-partite matching

- Deduplication - propagation through transitivity
  - Correlation clustering

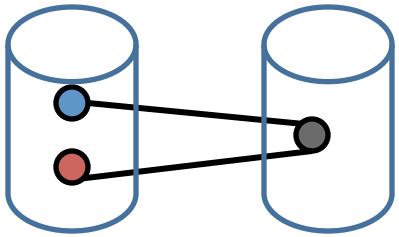
- Collective - propagation through relational constraints
  - Similarity propagation
    - Dependency graphs, Collective Relational Clustering
  - Probabilistic approaches
    - LDA, CRFs, Markov Logic Networks, Probabilistic Relational Models,
  - Hybrid approaches
    - Dedupalog

Duke UNIVERSITY

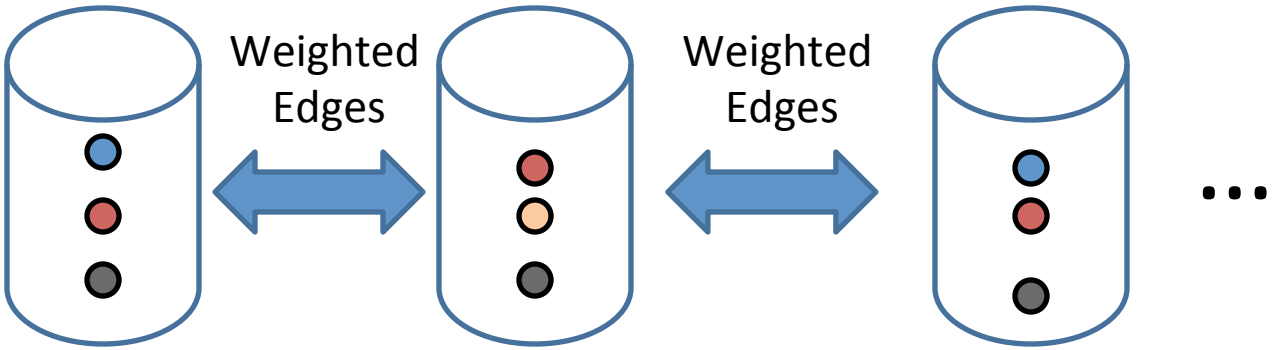
Record Linkage: Exclusivity Constraints

- Matching between (almost) deduplicated databases.
- Each record in one database matches at most one record in another database.

- Pairwise ER may match a record in one database with more than one record in second database

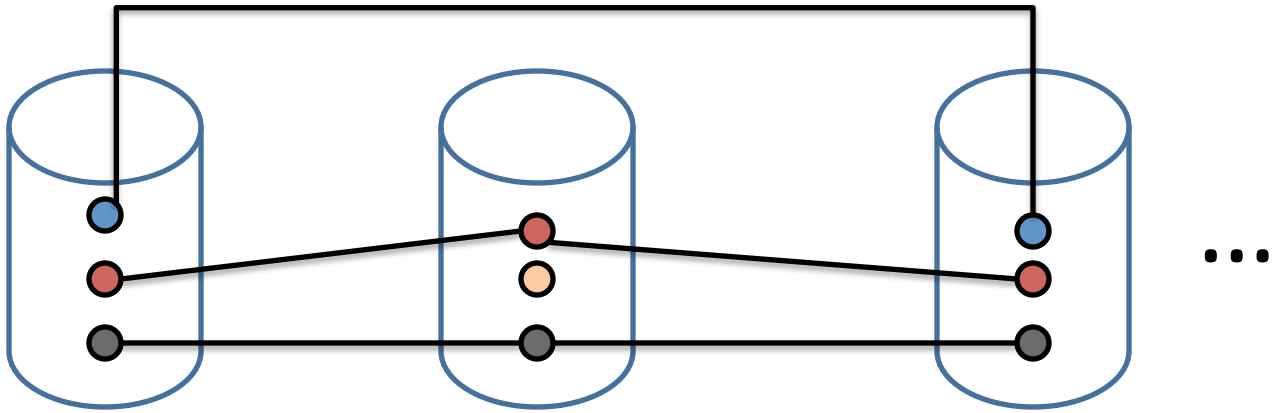


Weighted K-Partite Matching



- Edges between pairs of records from different databases
- Edge weights
  - Pairwise match score
  - Log odds of matching

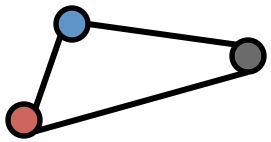
Weighted K-Partite Matching



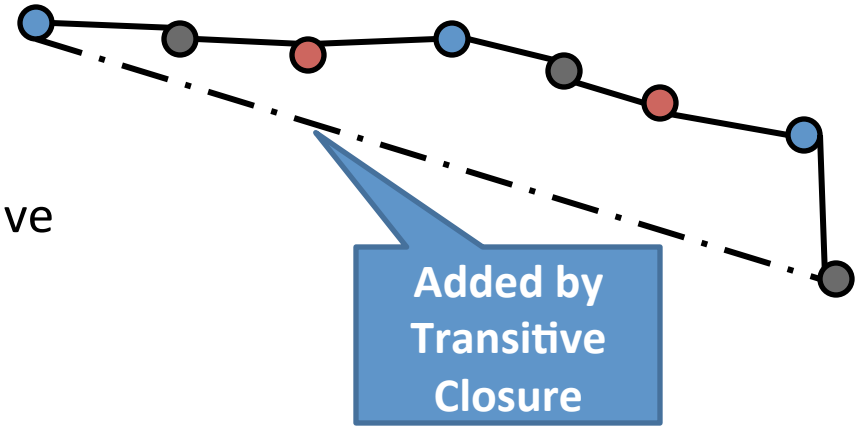
- Find a matching (each record matches at most one other record from other database) that maximize the sum of weights.
- General problem is NP-hard (3D matching)
- Successive bipartite matching is typically used. [Gupta & Sarawagi, VLDB '09]

Deduplication => Transitivity

- Often pairwise ER algorithm output "inconsistent" results
  - $(x, y) \varepsilon M_{pred}$, $(y,z) \varepsilon M_{pred}$, but $(x,z) \not\varepsilon M_{pred}$



- Idea: Correct this by adding additional matches using transitive closure



- In certain cases, this is a bad idea.
  - Graphs resulting from pairwise ER have diameter > 20
    [Rastogi et al ICDE'13]

- Need clustering solutions that deal with this problem directly by reasoning about records jointly.

Clustering-based ER

- Resolution decisions are not made independently for each pair of records

- Based on variety of clustering algorithms, but
  - Number of clusters unknown aprioiri
  - Many, many small (possibly singleton) clusters

- Often take a pair-wise similarity graph as input

Duke UNIVERSITY

Correlation Clustering

- A set of records R
- Pairwise similarities:
  - Cost of placing two records in different clusters (w+)
  - Cost of placing two records in the same cluster (w-)
- Example:
  - If x = y, w+ = 1, w- = 0
  - If x not equal to y, w+ = 0, w- = 1
- Goal: Cluster the records such that sum of w+ for records in different clusters + sum of w- for records within clusters is minimized.

Integer Linear Programming

- $x_{ij} \varepsilon \{0,1\}$, $x_{ij} = 1$ if records *i* and *j* are in the same cluster.
- $w^{+}_{ij} \varepsilon [0,1]$, *cost* of placing *i* and *j* in different clusters
- $w^{-}_{ij} \varepsilon [0,1]$, *cost* of clustering *i* and *j* together

$$\min \sum_{i,j:i<j} x_{ij} w_{ij}^{-} + (1 - x_{ij}) w_{ij}^{+}$$

$$s.t. \forall i,j,k \; x_{ij} + x_{jk} + x_{ik} \neq 2$$

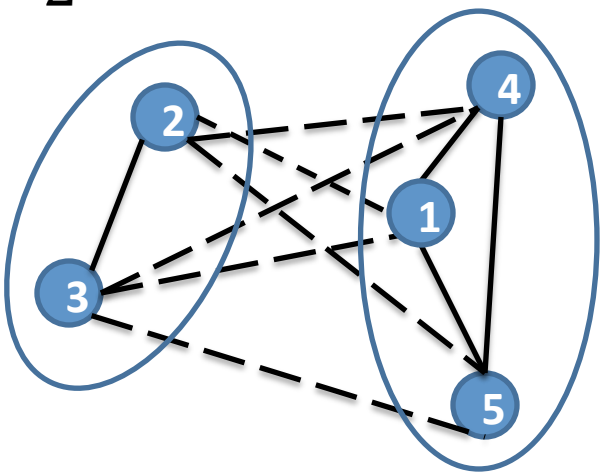
Transitive closure

Correlation Clustering

$$\min \sum_{i,j:i<j} x_{ij} w_{ij}^{-} + (1 - x_{ij}) w_{ij}^{+}$$

$$s.t. \forall i,j,k \; x_{ij} + x_{jk} + x_{ik} \neq 2$$

- Cluster records such that total penalty is minimized
  - Solid edges contribute w^{-}_{xy} to the objective
  - Dashed edges contribute w^{+}_{xy} to the objective



- Cost based on pairwise similarities

$$\{p_{ij} | (i,j) \in R \times R\}$$

  - Additive: $w^{+}_{ij} = p_{ij}$ and $w^{-}_{ij} = (1-p_{ij})$
  - Logarithmic: $w^{+}_{ij} = \log(p_{ij})$ and $w^{-}_{ij} = \log(1-p_{ij})$

Correlations Clustering

- Do not need to specify the number of clusters up front

- Respects the pairwise similarities (from the previous lectures) between objects during the clustering
  - Can encode hard constraints:
    e.g., Terminator 1 and Terminator 2 are different movies
  - Can encode soft constraints:
    Obama and Barak Obama are very likely the same person

Duke UNIVERSITY

Correlation Clustering

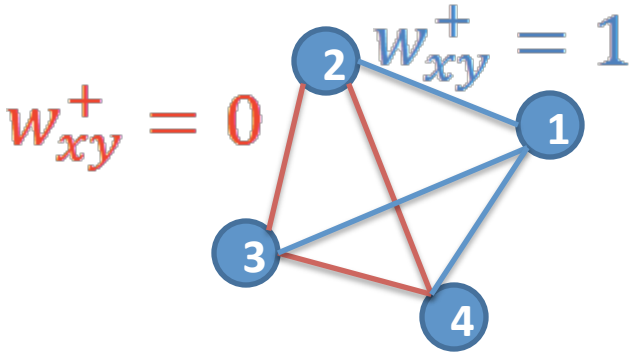
- Solving the ILP is NP-hard [Ailon et al 2008 JACM]

- A number of heuristics [Elsner et al 2009 ILP-NLP]
  - Greedy BEST/FIRST/VOTE algorithms
  - Greedy PIVOT algorithm (3 and 5-approximation)
  - Local Search

PIVOT Algorithm

[Ailon et al 2008 JACM]

- Pick a random *(pivot)* record *p*.
- New cluster = $\{x \mid w^+_{px} > 0\}$



- π = {1,2,3,4} C = {{1,2,3,4}}
- π = {2,4,1,3} C = {{1,2}, {4}, {3}}
- π = {3,2,4,1} C = {{1,3}, {2}, {4}}

When weights are 0/1, E(cost(PIVOT)) < **3** cost(OPT)

For general $w^+_{xy} + w^-_{xy} = 1$, E(cost(PIVOT)) < **5** cost(OPT)

Proof of Approximation

(w+ and w- are in {0,1})

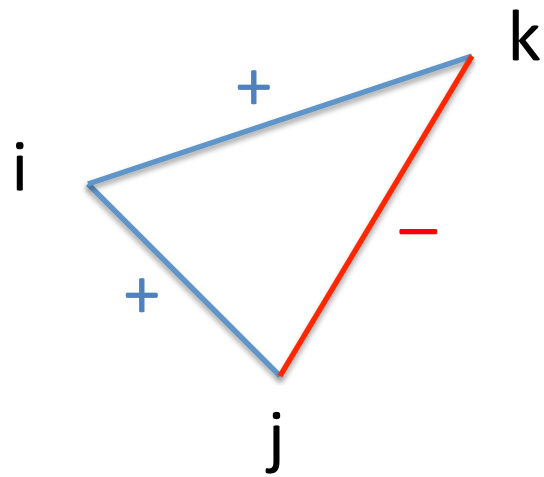
- COPT = cost of optimal solution
- CPIV = cost of the PIVOT algorithm

- + Edge: $w^+_{ij} = 1$, $w^-_{ij} = 0$
  - Incur a cost of 1 if i and j are in different clusters

- - Edge: $w^+_{ij} = 0$, $w^-_{ij} = 1$
  - Incur a cost of 1 if i and j are in the same cluster

Proof of Approximation

- When does PIVOT incur a cost?



- Pick i as PIVOT, - edge kj is in the same cluster
- Pick j as PIVOT, + edge ik is not in the same cluster

Proof of Approximation

- t = (i,j,k) is a bad triangle if it has 2 + edges and 1 – edge.
- Let T be the set of bad triangles

- For every bad triangle, let At be the event:
  "all i, j, k were considered in the same step when the first among them was a pivot"

- Triangle t is charged a unit cost exactly when At occurs

- Triangle t can be charged at most once
  - At's are mutually exclusive.

Duke UNIVERSITY

Proof of Approximation

- Triangle t is charged a unit cost exactly when At occurs

- Triangle t can be charged at most once
  - At's are mutually exclusive.

$$E(C^{PIV}) = \sum_{t \in T} P(A_t) = \sum_{t \in T} p_t$$

Proof of Approximation

- What is the cost of OPT?

- Suppose we had n edge disjoint bad triangles
  - COPT = n

- We can say something similar when bad triangles are not disjoint
  - Suppose ßt is a non-negative weight associated with each bad triangle t such

$$\forall e, \sum_{t:\, e \in t} \beta_t \leq 1$$

  - Then

$$C^{OPT} \geq \sum_{t \in T} \beta_t$$

Proof of Approximation

- We can show

$$\forall e, \sum_{t:\, e \in t} p_t \leq 3$$

$$\Rightarrow C^{OPT} \geq \frac{1}{3} \sum_{t \in T} p_t = \frac{C^{PIV}}{3}$$

Proof of Approximation

- Proof of $\forall e, \sum_{t:\, e \in t} p_t \leq 3$

- Suppose e is an edge shared by a set of bad triangles t1, t2 …
- P(e is charged cost of 1 due to triangle ti) = p_{ti} / 3
- All At's are mutually exclusive

- Thus $P(e\ is\ charged)$

$$= \sum_{t:e \in t} P(e\ is\ charged\ due\ to\ t)$$

$$= \sum_{t:e \in t} \frac{p_t}{3} \leq 1$$

Other Greedy Heuristics

Step 1: Permute the nodes according a random π

Step 2: Assign record *x* to the cluster that maximizes *Quality*

Start a new cluster if *Quality* < 0

Quality:

- BEST: Cluster containing the closest match $max_{y \in C} \; w^{+}_{xy}$
  - [Ng et al 2002 ACL]
- FIRST: Cluster contains the most recent vertex y with $w^{+}_{xy} > 0$
  - [Soon et al 2001 CL]
- VOTE: Assign to cluster that minimizes objective function.
  - [Elsner et al 08 ACL]

Practical Note:

- Run the algorithm for many random permutations , and pick the clustering with best objective value (better than average run)

Local Search

BOEM Algorithm [Gionis et al 2007 TKDD]

- Start with an initial clustering (e.g. output of greedy)
- Remove one random element from a cluster
- Make the Best One Element Move (BOEM)
  - Move it to another cluster or Create a new cluster.

Summary

- Pairwise matching is insufficient for ER when constraints need to be handled
- Transitivity, Exclusivity and Relational constraints are typical.
- Exclusivity
  - Record Linkage: Weighted bipartite matching
- Transitivity
  - Deduplication: Correlation clustering
  - PIVOT greedy algorithm with 3 approximation