Example-Driven Design of Efficient Record Matching Queries

Venkatesh Ganti

Surajit Chaudhuri

Raghav Kaushik

Microsoft Research

Bee-Chung Chen

University of Wisconsin-Madison

Record Matching

- **Katrina: Given evacuee lists…**

match against enquiries.

First Name	Last Name	Address	Phone	Father	Mother
Holmes	Elois	2723 Third St	938-8374		
Donneaka	Martin		504-974-637	Donald Qautier	
Thomas		2435 Delachise St		Lomax	

Elois	Holmes	Third Street			
Donaka	M		504-974-637	D Oautier	

Record Matching can be Difficult

First Name	Last Name	Address	Phone	Father	Mother
Holmes	Elois	2723 Third St	938-8374		
Donneaka	Martin		504-974-637	Donald Qautier	
Thomas		2435 Delachise St		Lomax	

Elois	Holmes	Third Street			
Donaka	M		504-974-637	D Oautier	

- **Too many options to consider while building a record matching query**
- **Complicated due to errors and representational differences**

Record Matching Queries

Select * from Enquiries R, Evacuees S where

sim1(R.FirstName + R.LastName, S.FirstName + S.LastName) > 0.85

AND sim2(R.Address, S.Address) > 0.83

AND sim3(R.Phone, S.Phone) = 1

OR

sim4(R.FirstName + R.LastName + R.Phone,
S.FirstName + S.LastName + S.Phone) > 0.87

OR

1.5 * sim5(R.FirstName, S.FirstName) – 0.3 * sim6(R.Father + R.Mother, S.Father + S.Mother) > 0.9

Creating RM Queries

- **Challenges**
  - **Which column combinations to compare?**
  - **Which similarity function for each combination?**
    - **Name similarity: soundex or edit distance**
    - **Address similarity: jaccard**
  - **How to determine the thresholds for chosen similarity function-column combination choice?**

Example-Driven Approach

- **Input**
  - **A set of example (r, s) record pairs: matches & non-matches**
  - **A set of candidate operators**
- **Goal**
  - **Construct a query which has the "best quality" when applied to the examples**
- **Quality measure**
  - **Recall: Number of correctly classified matching pairs s.t. the fraction of false positives is less than B**

Previous Work

- **Machine learning (ML) based predicates**
  - **Decision trees**
  - **SVMs – more accurate**
- **However, cannot efficiently implement similarity joins involving ML predicates**
  - **Usually, cross product followed by filter**

SVM Predicates

- Current best method [Bilenko et al.]
- Example SVM predicate
  - 1.5*Jaccard(R.[NAC], S.[NAC]) – 0.3*Edit(R.N, S.N) > 0.9
- May not be efficiently executable
  - Cross product followed by a filter

Our Approach

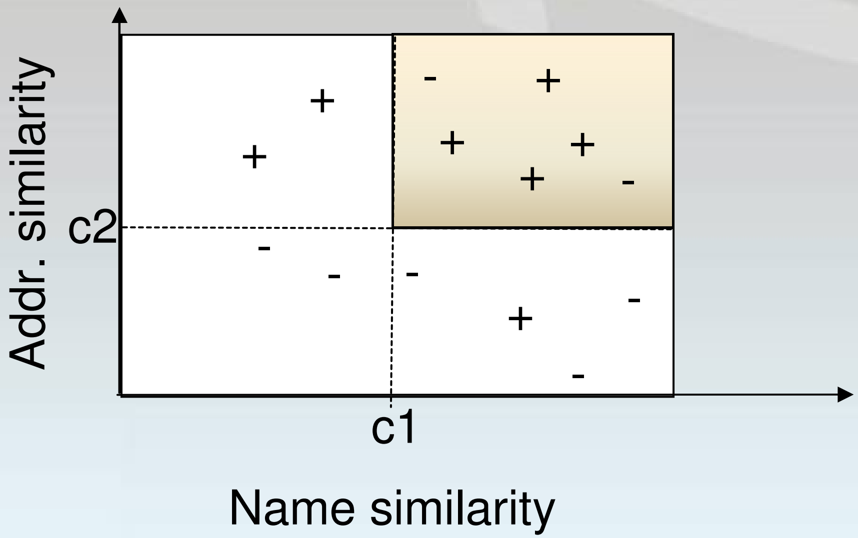
- Constrain class of output queries
  - Efficiently executable
  - Flexible enough to capture a rich set of queries
- Programmers can review & modify
  - If required, add more sophisticated ML predicates to suggested queries

Similarity Space

- **Map examples to +/- points**
  - **D-dimensional: One per similarity fn & column combination**
- **Matches → + and Non-matches → –**

  Name similarity (edit similarity)
  Address similarity (jaccard over ACZ)

Predicate: name similarity > c1 and address similarity > c2

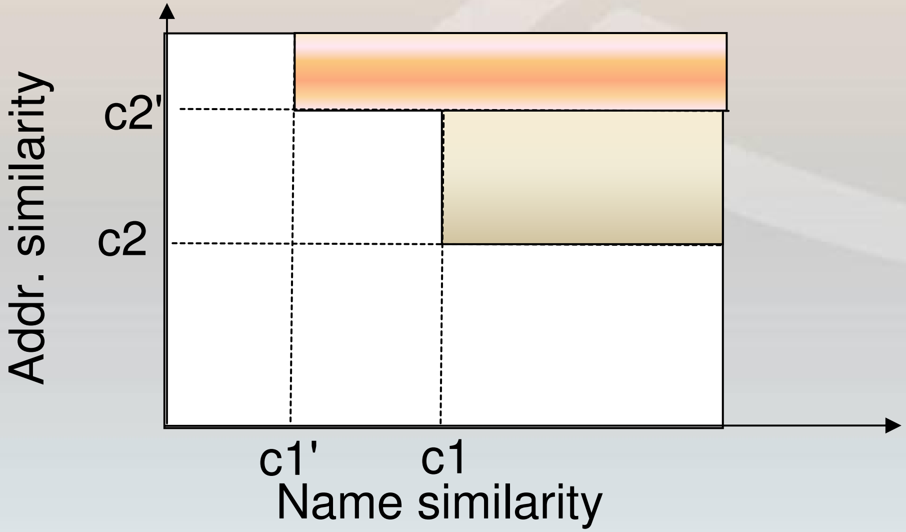


Class of Queries

- Relations R, S (schema [Name, Address, City, Zip])
- *D* similarity functions (and column combinations)
- Class: Union of *top-right rectangular boxes*

Similarity functions

Name similarity (edit similarity)
Address similarity (jaccard over ACZ)



name similarity > c1 and address similarity > c2

OR

name similarity > c1’ and address similarity > c2’

Problem Statement

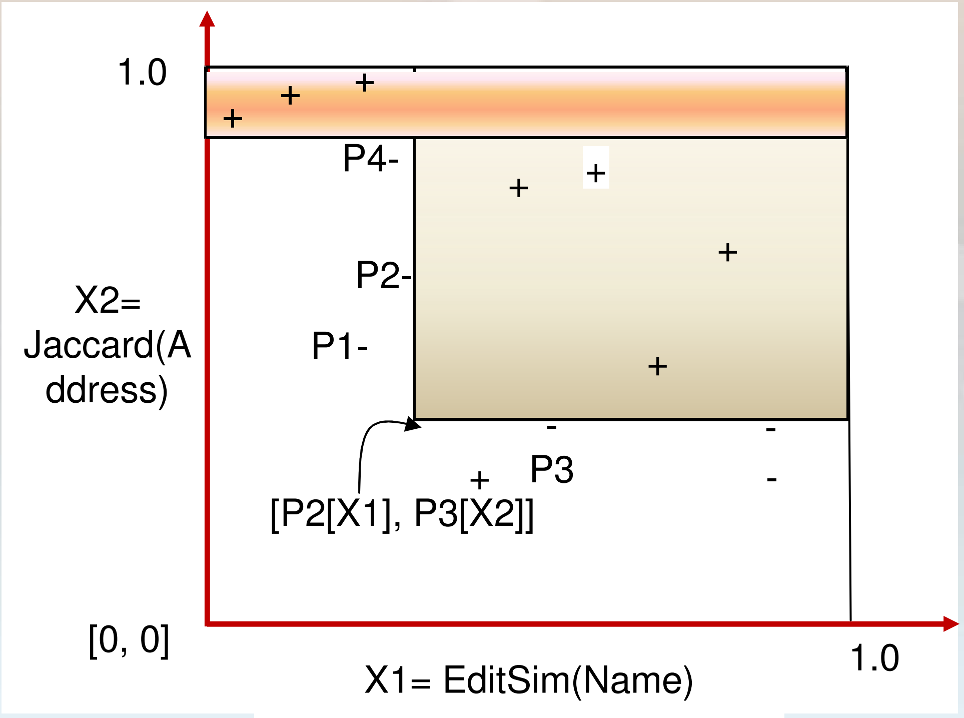
- Given positive and negative points, find K rectangular boxes such that
  - Recall—the number of positive points in them—is maximized
  - Number of negative points they contain is less than *B*
- Sub-space constraints on each rectangular box
  - Not more than d (<= D) dimensional

Algorithm Outline

- Consider B=0
  - No negative points at all in the result
- Extend to B > 0
  - Allow a few negative points in the result

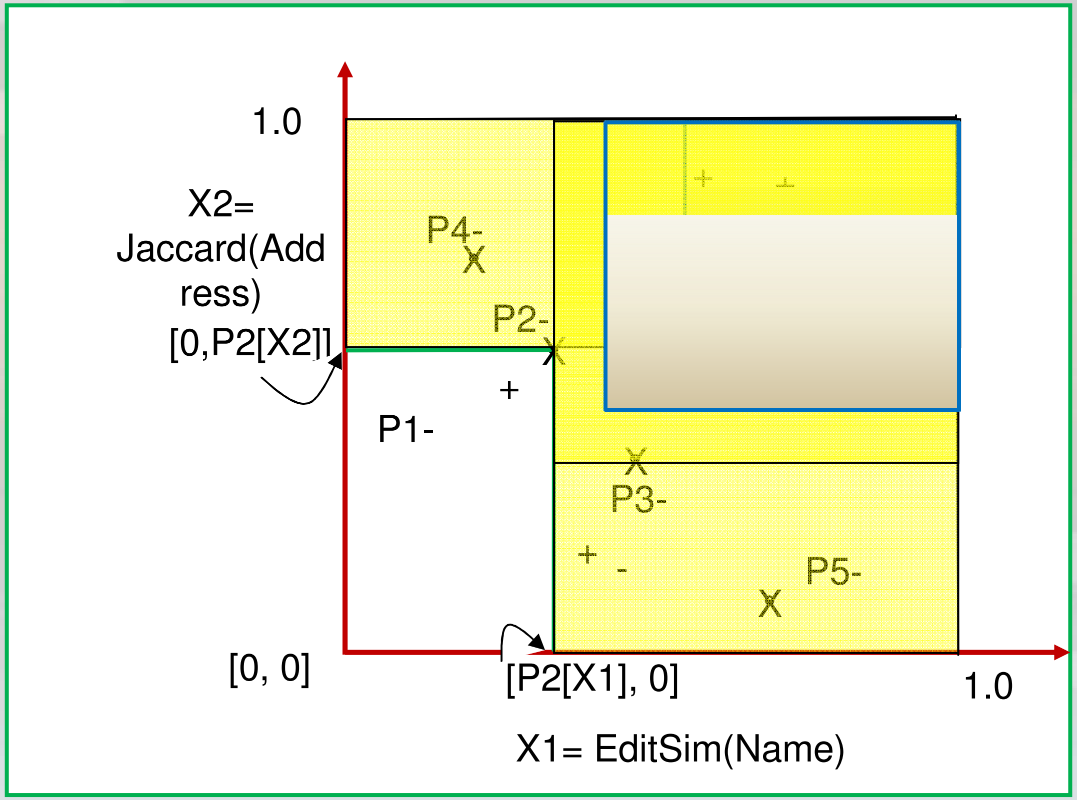
Union of Rectangles

- Find the best valid rectangular box with the maximum number of +'s
- Remove +'s in box and iterate



Best Rectangular Box

- **Recursive search for the best valid rectangular box**



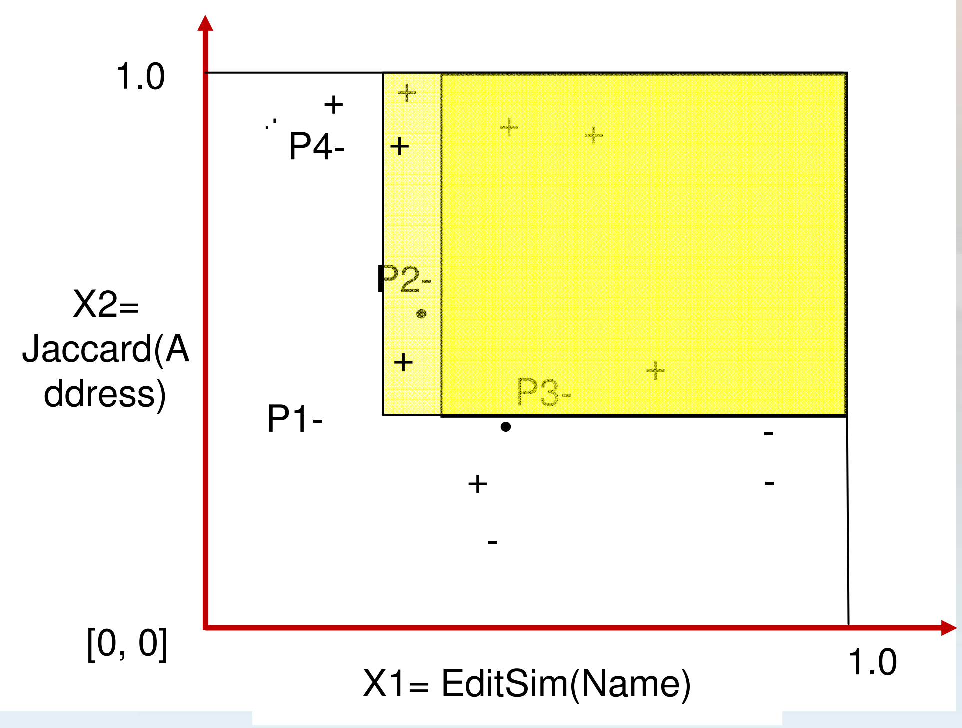
- **Can be applied to D > 2 and for boxes in sub-spaces (i.e., d < D)**

Union of Rectangular Boxes

- Greedy strategy
  - Pick best rectangular box with maximum number of +'s and no –'s
  - Remove +'s contained in box
  - Iterate until
    - All +'s are covered
    - K boxes are picked
- Approximation guarantee
  - Within (1 - 1/e) of the optimal
  - Follows from the greedy solution to the set coverage problem

Allowing Non-matches

- A valid rectangular box may now include a fraction of negative points
- Find the best among all valid boxes
- Recursive algorithm applicable again



Record Transformations

- Consider two records
  - r1: [Matrin Smith, Redmond, WA, 98052]
  - s1: [Martin Smit, NULL, WA, 98052]
- Apply FD zip→ city to s1
  - s1': [Martin Smit, Redmond, WA, 98052]
- For many similarity functions,

  sim(r1, s1) < sim(r1, s1')
- Hence record transformations help identify matches!

Record Transformations (contd)

- Example record transformations
  - FDs to fill in missing values
  - Splitting columns into sub-columns (e.g., address or product names)
- Our framework can be extended to consider such transformations
- Idea: Iteratively add best transformation to the current query

Experimental Evaluation

- Datasets
  - Organization data from an operational data warehouse
  - RIDDLE repository ([Bilenko], UT Austin)
- Techniques compared
  - Addresses: a commercial cleansing tool called Trillium
  - RIDDLE: SVM

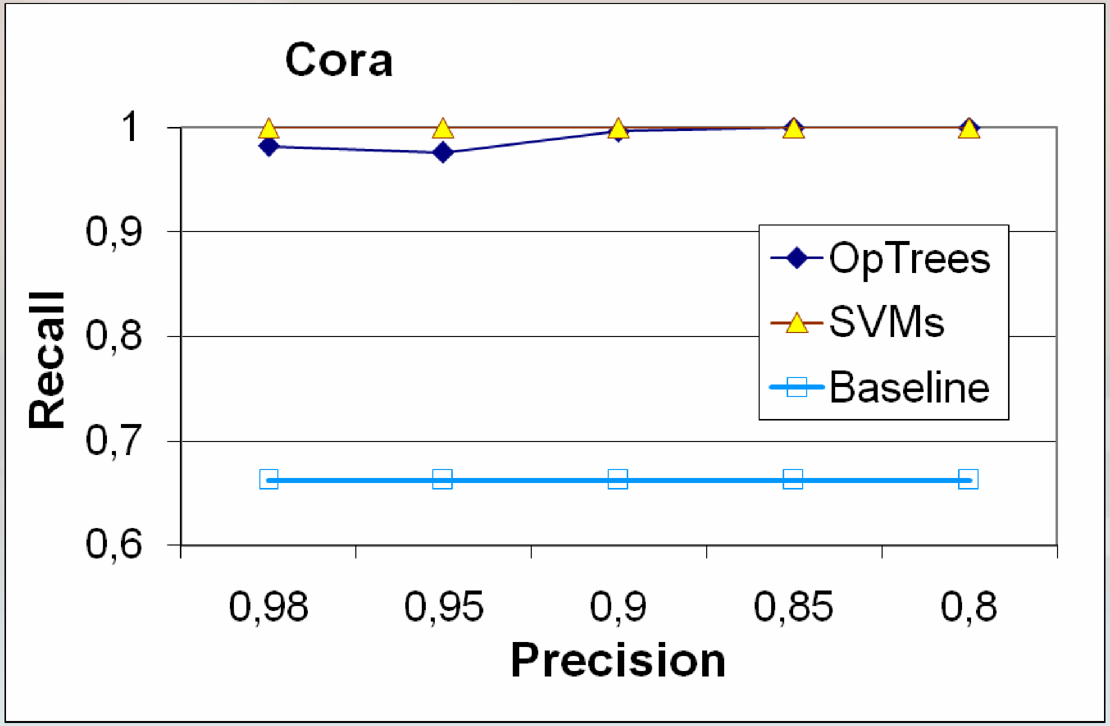
Operator Trees vs. Trillium

- 29 candidate similarity functions
- Zipcode splitter: out-code and in-code
- Out-code → City
- At most 4 similarity functions per box
- Union of at most 4 boxes

	Precision	Recall
Trillium	0.99	144K
Operator Trees	**0.98**	**159K**
Baseline	0.98	80K

Cora Dataset

- **Bibliography data: authors, titles**



Efficiency of Similarity Join

- **Similarity join (jaccard similarity) over 500K record relation with itself**
  - **[VLDB06] SSJoin algorithm**

Threshold	SimJoin
0.9	61 s
0.85	125 s
0.80	285 s

SVM predicate: 10 days
SVM + blocking: 1+ hour

Conclusions

- Example-driven approach to suggest a record matching query
- Considered constrained space of efficiently executable queries
- Empirically demonstrated accuracy

- Web search: “data cleaning project”
  - http://research.microsoft.com/dmx/datacleaning

Questions