On Active Learning of Record Matching Packages

Arvind Arasu, Michaela Götz, Raghav Kaushik

Real-World Example: Record Matching

[24] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[25] R. M. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 881–890, 2007.

[26] R. M. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 881–890, Jan. 2007.

[27] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. of the 6th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 169–178, Aug. 2000.

[28] T. M. Mitchell. *Machine Learning*. Mc

Our SIGMOD submission!

Record Matching

- Given two tables R, S

ID	Name	Street	City	Phone
R_1	SWI Alloys Inc	456 Medical Dr	Albany	5181234567
R_2	ABC Rural Telephone	1234 Market St	Fremont	3179876543
R_3	Bank of New York	1 E 31st St	New York	2120001111

S_1	First Tech	156th Ave	Boise	
S_2	ABC Cellular	PO Box 9862	Fremont	3179876543
S_3	SWI Alloys	456 Medical Dr	Albany	

Subjective judgment?

- Find matching pairs of records!

Similarity Space

- Map record pairs into similarity space ψ.
- Learn how to combine similarity measures to predict matches.

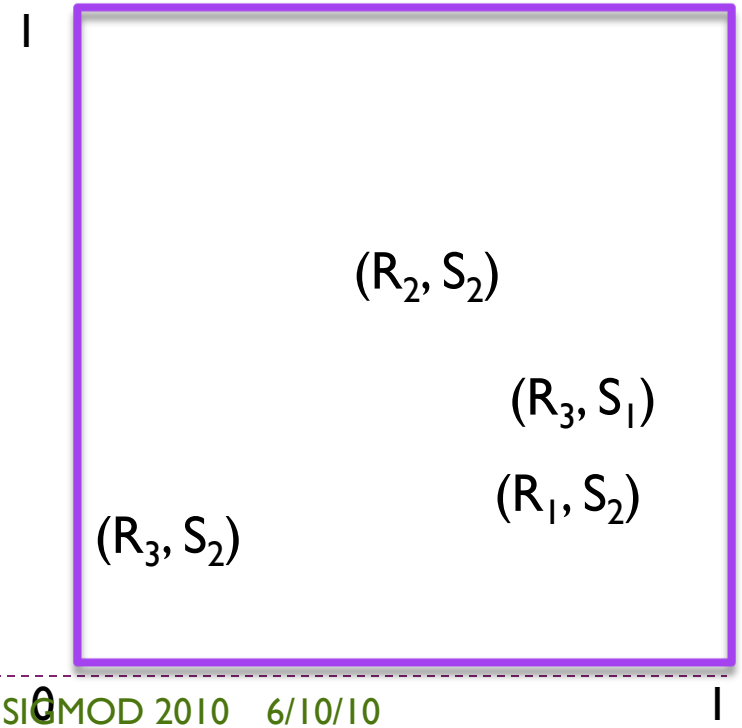
Data

(R_1, S_1)
(R_1, S_2)
(R_2, S_1)
(R_2, S_2)
(R_3, S_1)
(R_3, S_2)

Similarity Features

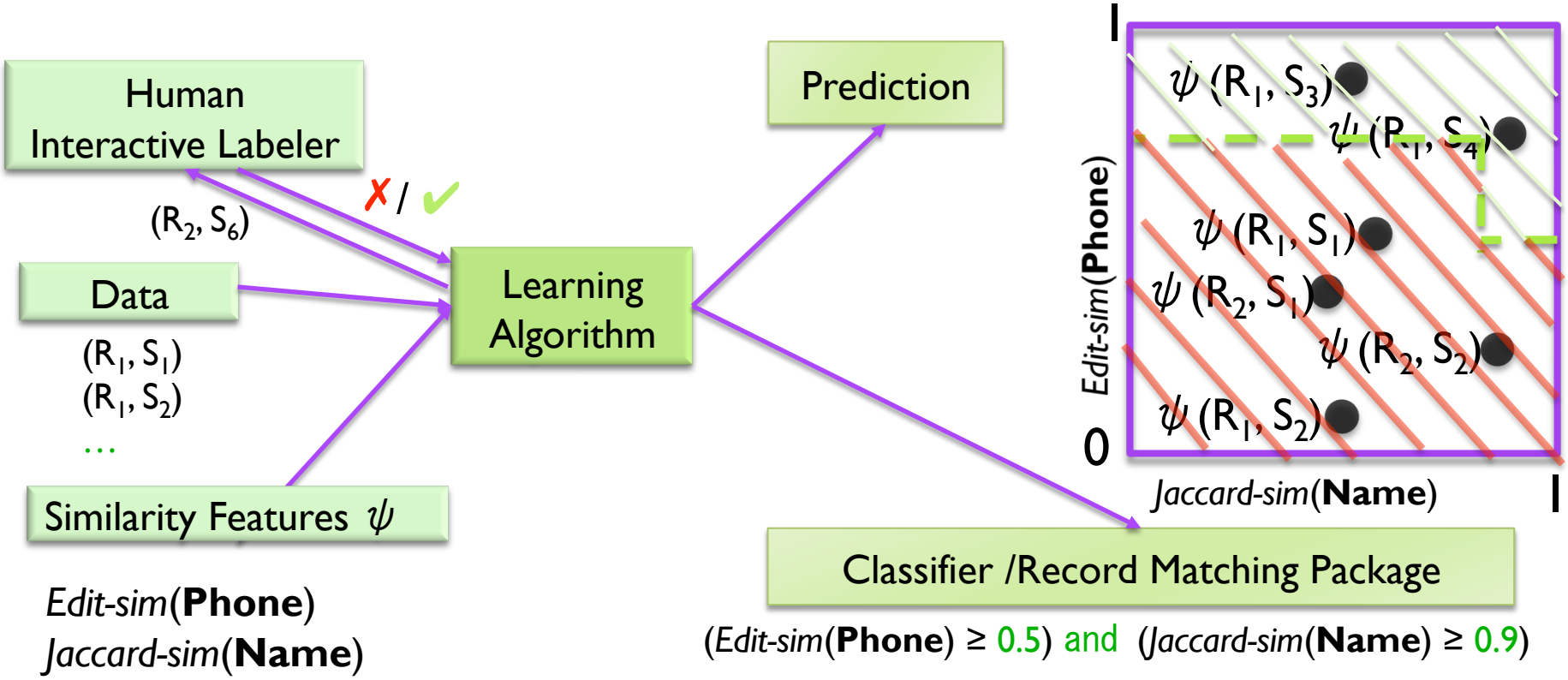
Edit(**Phone**)

Jaccard(**Name**)



Active Learning of Record Matchings

- Learning problem:
  - Given examples of (non-)matching pairs and similarity measures
  - Learn how to combine the similarity measures for prediction



Record Matching - Desiderata

- Complexity
  - Few labeled examples

(R_1, S_5) ✔ (R_2, S_6) ✗
(R_5, S_6) ✔ (R_5, S_7) ✗
(R_6, S_7) ✗

- Quality of classifier
  - high precision & recall
  - low error not good enough

(R_1, S_2) ✗ (R_1, S_1) ✗
(R_1, S_7) ✔ (R_1, S_3) ✗
(R_2, S_6) ✔ (R_1, S_4) ✔
(R_2, S_{10}) ✔ (R_1, S_5) ✔
… …

- Scalability

$R_1, R_2, \ldots, R_{1,000,000}$

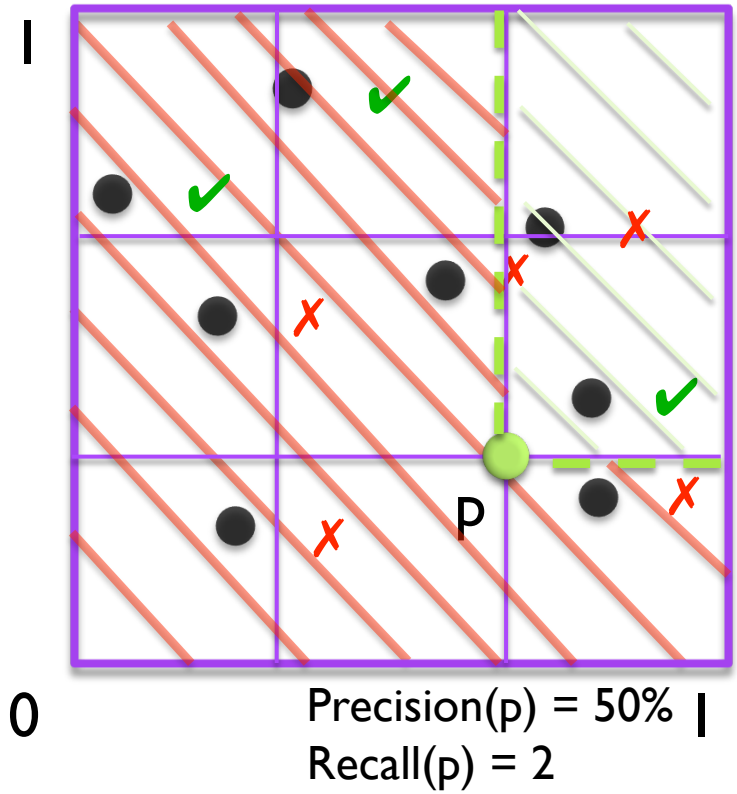
$S_1, S_2, \ldots, S_{1,000,000}$

Problem Statement

Find best classifier:

- Maximize recall

  Subject to precision ≥ τ
- Where:
  - precision: Among pairs classified as matches fraction of true matches
  - recall: proxy: # of pairs classified as matches



Set of classifiers = points in discretized space

Building Block: Precision/Recall Oracles

- Terminology: Say p dominates q if $p_i \geq q_i \ \forall i$ space
- Oracles
  - `Recall`(point p)
    - Proxy: Count number of points dominating p.
  - `Precision`(point p)
    - Compute set of points dominating p
    - Sample a random subset
    - Request labels for these points
    - Estimate precision as fraction of matches

Naïve Algorithm:
For each point: Call precision and recall oracles

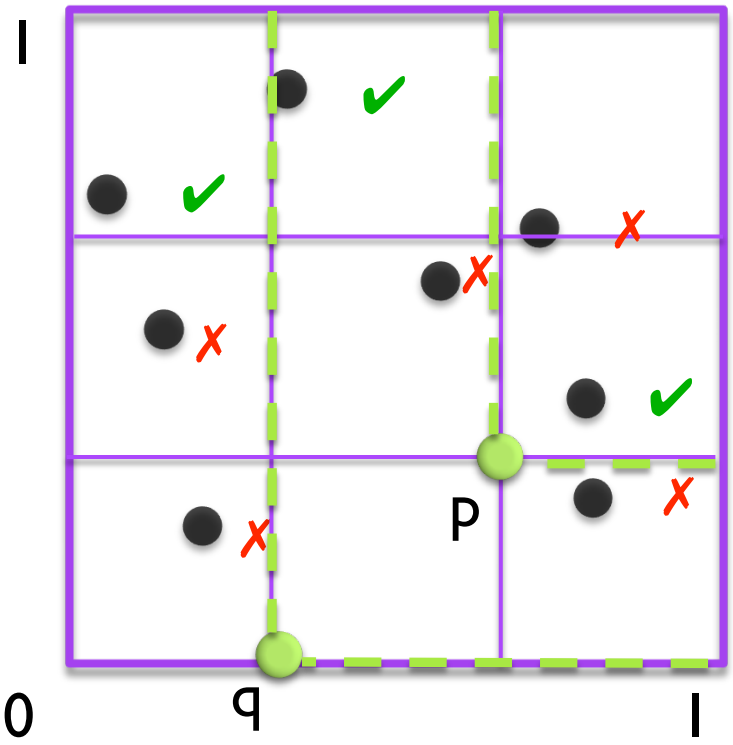
Too many oracle calls
-> high label complexity

Monotonicity Assumption

- Monotonicity Assumption

  If point p dominates point q,

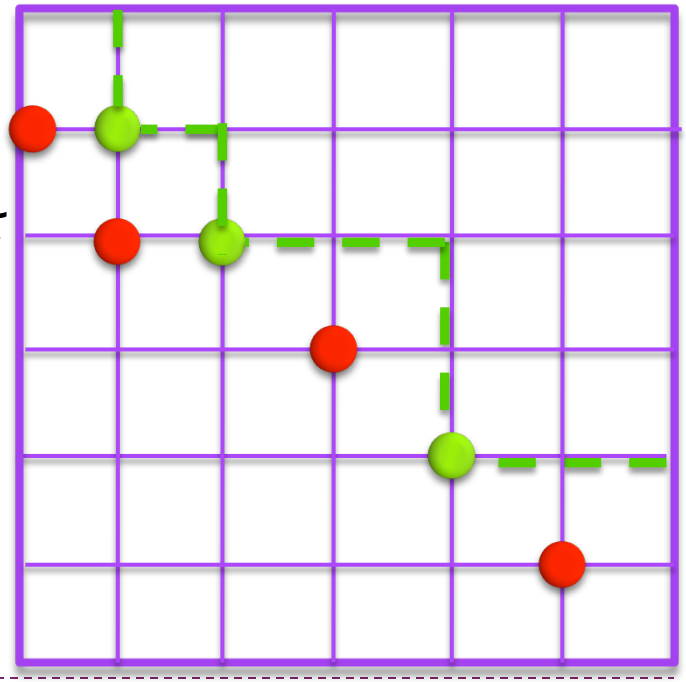
  then Precision(p) ≥ Precision(q)



Algorithm

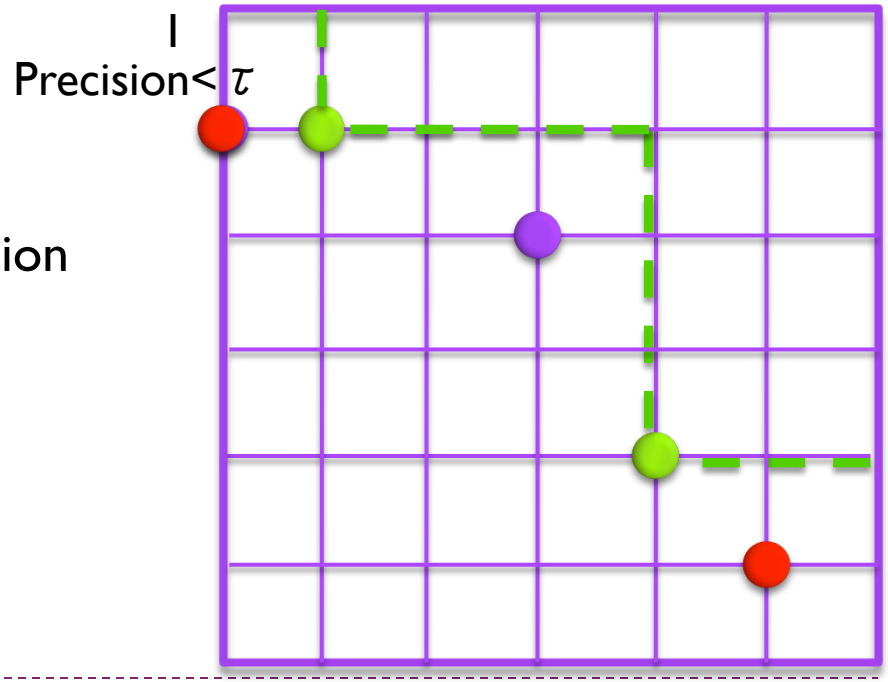
- **Intermediate solution**:
  Green points = minimal points with precision ≥ τ
  Red points = maximal points with precision < τ

- **Incremental computation:** Each round add either red or green point
- Invariant: Candidates = set of points, s.t.
  - If all the candidates had precision < τ
  - Then Green, Red ∪ Candidates is intermediate solution
- **Output** Green point with max recall



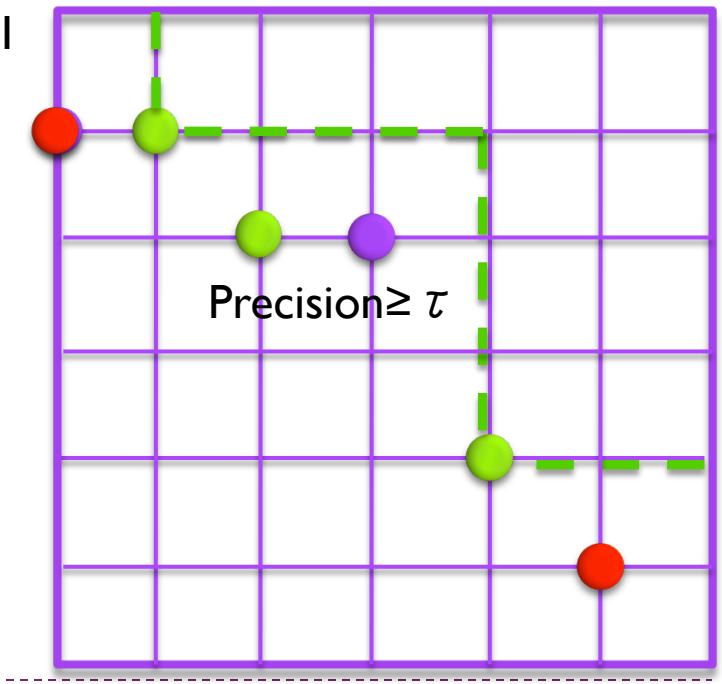
Algorithm

- **Incremental computation**:
  Each round add either red or green point
  - Pop p from Candidates
  - If precision of p < τ
  - Then add p to Red
  - Else
    - Make p smaller in each dimension
      - s.t. p keeps precision ≥ τ
    - Add p to Green
    - Update Candidates



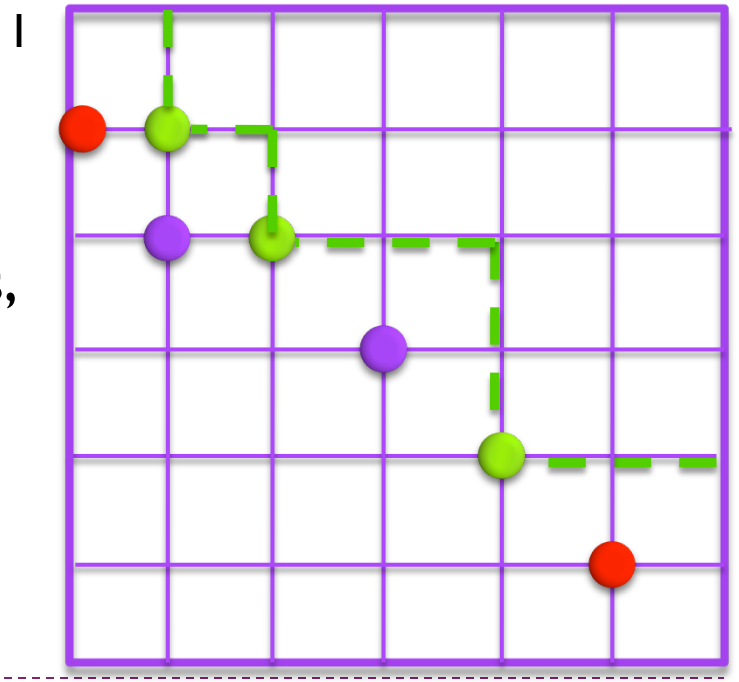
Algorithm

- **Incremental computation**:
  Each round add either red or green point
  - Pop p from Candidates
  - If precision of p < τ
  - Then add p to Red
  - Else
    - Make p smaller in each dimension s.t. its precision remains ≥ τ
    - Add p to Green
    - Update Candidates



Algorithm

- **Incremental computation**:
  Each round add either red or green point
  - Pop p from Candidates
  - If precision p < τ then add p to Red
  - Else
    - Make p smaller in each dimension s.t. p keeps precision ≥ τ
    - Add p to Green
    - Update Candidates
- Invariant: Candidates = set of points, s.t.
  - If all the candidates had precision < τ
  - Then Green, Red ∪ Candidates intermediate solution



Algorithm - Analysis

- Human labeling effort
  - # precision requests: O(#Red points + # Green points * d * log(k)), where d number of similarity measures, k granularity

Desiderata

- Complexity
  - Few labeled examples
  - that are easy to come up with
- Quality of classifier
  - high precision & recall
- Scalability

# of Oracle calls close to instance optimal

Guaranteed quality

Extensions

- Extend set of classifiers
  - s-term DNF
    - Repeatedly find best point and remove points dominating it from further iterations
    - No guarantees about optimality
  - Linear classifiers

Parallels: Finding Maximal Frequent Itemsets

	Record Matching	**Maximal Frequent Itemsets**
Points	$\{0, 1/k, 2/k, \ldots, 1\}^d$ d number of similarity measures, k granularity (of descritizing space)	$\{0, 1\}^d$ d number of items
Goal	Max recall	Max set size
Constraint	Precision $\geq \tau$	frequency $\geq \tau$
# requests	Precision: O(#Red + # Green * d * log(k))	Frequency: O(#Red + # Green * d)

Record Matching as generalization of maximal frequent itemsets enumeration

Experiments:

- Goal:
  - Comparison to previous active learning methods: Sarawagi et al.'02, Tejada et al.'01, our algorithm
- Data set:
  - ORG: $10^6 \times 10^6$ records

Name	Street	City	Zip	Phone

- Similarity dimensions
  - Jaccard similarity / containment, edit similarity
  - Total 8 dimensions

Experiments: Performance of our Algorithm

Recall	
Precision	
Number of labels	
Time per label	

precision threshold $\tau = 0.95$,
learning one point classifier,
averages over 5 random runs

Experiments: Performance of our Algorithm

Recall	29,500
Precision	97%
Number of labels	84
Time per label	0.69 sec

precision threshold $\tau = 0.95$,
learning one point classifier,
averages over 5 random runs

Desiderata

- Complexity
  - Few labeled examples
  - that are easy to come up with
- Quality of classifier
  - high precision & recall
- Scalability

# of Oracle calls close to instance optimal

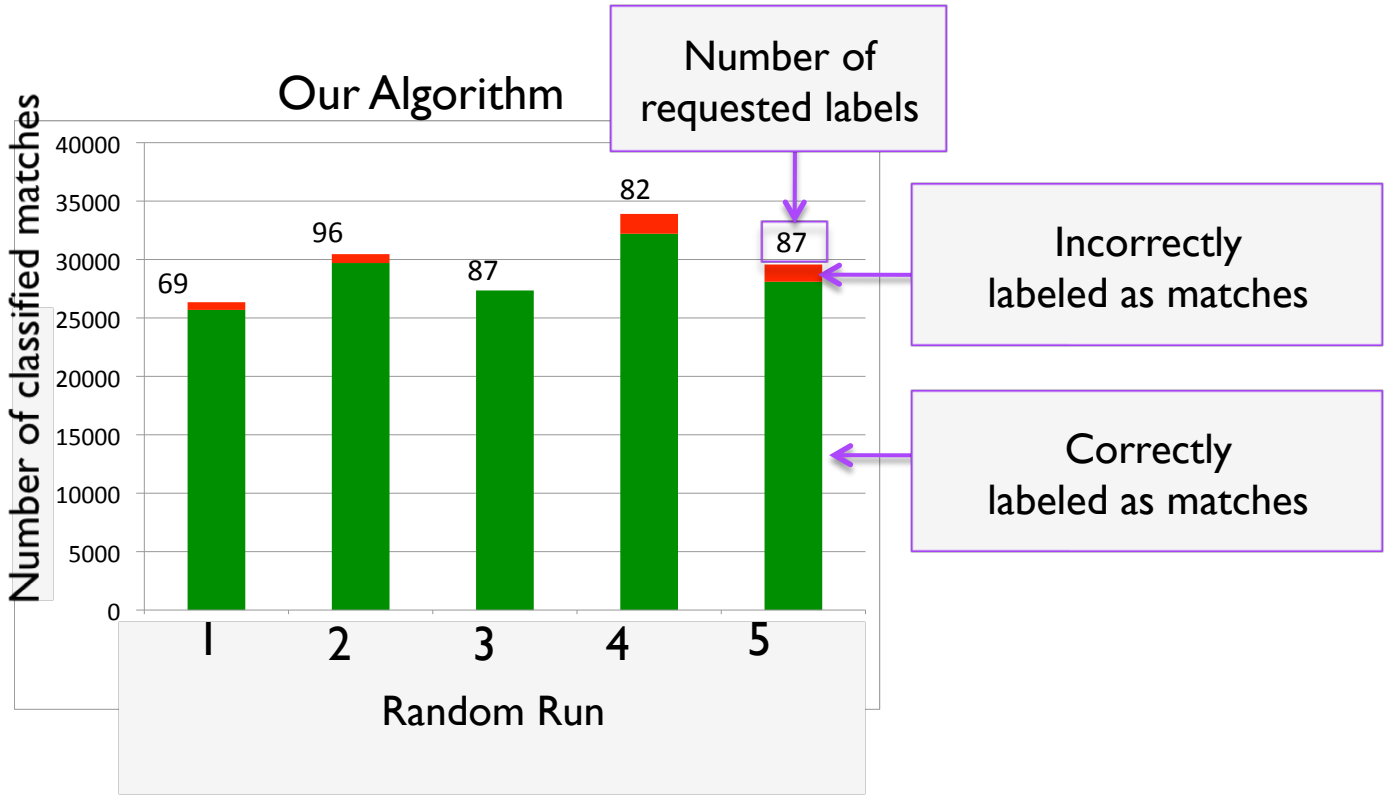
Few labeled examples

Guaranteed quality

High precision as requested

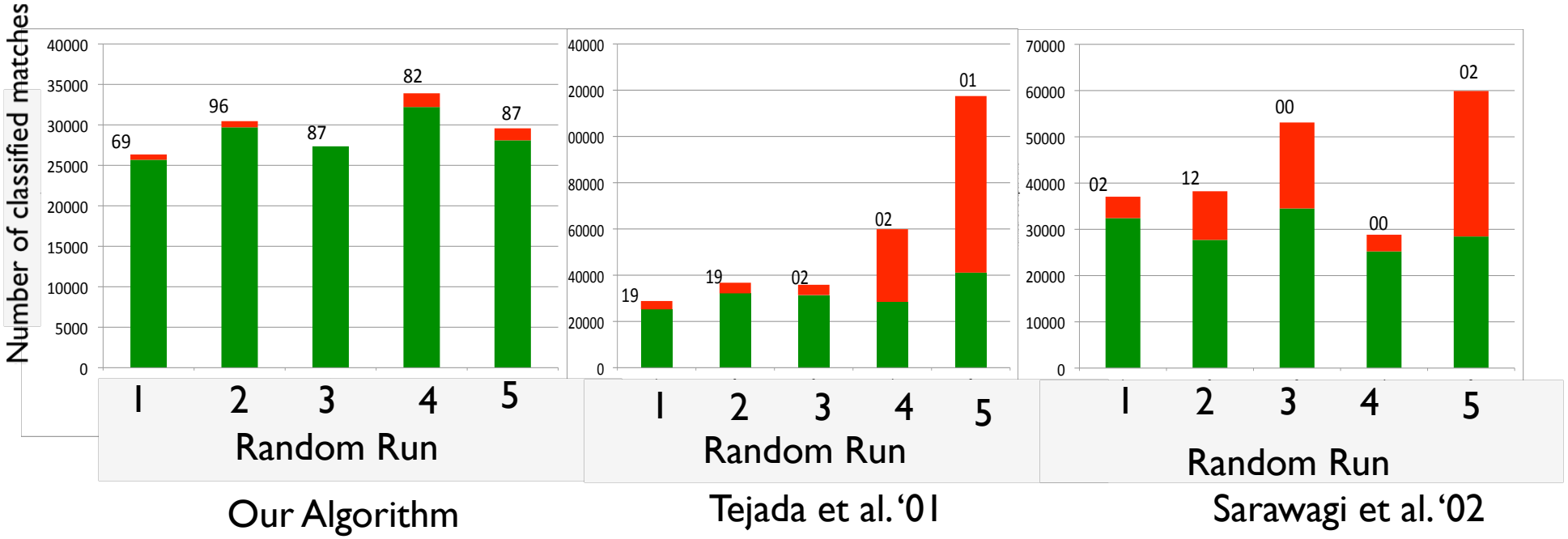
To $10^6 \times 10^6$ record pairs

Experiments: Variations in Quality across Random Runs



- consistently high quality across runs

Experiments: Comparison to Active Learning Algorithms



Compared to other active learning algorithms our algorithm provides more consistent quality

Conclusions - Record Matching Package

- Use active learning
- Exploit monotonicity
- Cleverly navigate through space

- Quality of point-classifier
  - Set precision threshold
    guaranteed highest recall
- Complexity
  - Close to instance-optimal
  - Few labeled examples requested by algorithm
- Scalability to $10^6 \times 10^6$ record pairs

Thanks!
Questions?

Related Work

- Supervised:
  - Examples given upfront:
    - SVM *[Bilenko et al.'03]*
    - CART *[Cochinwalla et al.'01]*
    - Decision Trees *[Chaudhuri et al.'06]*
    - Probabilistic Linkage *[based on Fellegi and Sunter '69]*
  - Active:
    - Decision trees *[Sarawagi et al.'02, Tejada et al.*
- Unsupervised:
  - Probabilistic Linkage & EM *[Winkler '93]*
  - Clustering *[Elfeki et al.'02, Verykios et al.'00]*
  - Combination of distance functions *[Dey et al.'98, Guha et al.'04]*

Difficult to pick good examples

Not targeted at high precision / recall

Thanks!
Questions?

goetz@cs.cornell.edu