On Active Learning of Record Matching Packages

Arvind Arasu, Michaela Götz, Raghav Kaushik
Real-World Example: Record Matching


Record Matching

- Given two tables R, S

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Street</th>
<th>City</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁</td>
<td>SWI Alloys Inc</td>
<td>456 Medical Dr</td>
<td>Albany</td>
<td>5181234567</td>
</tr>
<tr>
<td>R₂</td>
<td>ABC Rural Telephone</td>
<td>1234 Market St</td>
<td>Fremont</td>
<td>3179876543</td>
</tr>
<tr>
<td>R₃</td>
<td>Bank of New York</td>
<td>1 E 31st St</td>
<td>New York</td>
<td>2120001111</td>
</tr>
<tr>
<td>S₁</td>
<td>First Tech</td>
<td>156th Ave</td>
<td>Boise</td>
<td></td>
</tr>
<tr>
<td>S₂</td>
<td>ABC Cellular</td>
<td>PO Box 9862</td>
<td>Fremont</td>
<td>3179876543</td>
</tr>
<tr>
<td>S₃</td>
<td>SWI Alloys</td>
<td>456 Medical Dr</td>
<td>Albany</td>
<td></td>
</tr>
</tbody>
</table>

- Find matching pairs of records!

Subjective judgment?
Similarity Space

- Map record pairs into similarity space $\psi$.
- Learn how to combine similarity measures to predict matches.
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

**Human Interactive Labeler**

**Data**
- \((R_1, S_1)\)
- \((R_1, S_2)\)
- \((R_2, S_6)\)

**Learning Algorithm**

**Similarity Features** \(\psi\)

- \(\text{Edit-sim(Phone)}\)
- \(\text{Jaccard-sim(Name)}\)

**Prediction**

**Classifier / Record Matching Package**

- \(\psi(R_1, S_3)\)
- \(\psi(R_1, S_4)\)
- \(\psi(R_1, S_1)\)
- \(\psi(R_2, S_1)\)
- \(\psi(R_2, S_2)\)
- \(\psi(R_1, S_2)\)

\(\text{Edit-sim(Phone)}\)

\(\text{Jaccard-sim(Name)}\)

- \((\text{Edit-sim(Phone)} \geq 0.5)\) and \((\text{Jaccard-sim(Name)} \geq 0.9)\)
- or \((\text{Edit-sim(Phone)} \geq 0.85)\)
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
    - $(R_1, S_2)$ ✗ $(R_1, S_5)$ ✗
    - $(R_1, S_7)$ ✓ $(R_1, S_3)$ ✗
    - $(R_2, S_6)$ ✓ $(R_1, S_4)$ ✓ ✓
    - $(R_2, S_{10})$ ✓ $(R_1, S_5)$ ✓ ✓
  - low error not good enough

- **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 $\geq \tau$

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

If point $p$ dominates point $q$, then $\text{Precision}(p) \geq \text{Precision}(q)$
Algorithm

- **Intermediate solution:**
  - **Green** points = minimal points with precision ≥ \( \tau \)
  - **Red** points = maximal points with precision < \( \tau \)

- **Incremental computation:** Each round add either red or green point

- **Invariant:** \( \text{Candidates} = \text{set of points, s.t.} \)
  - If all the candidates had precision < \( \tau \)
  - Then **Green, Red \cup 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 < \( \tau \)
  - Then add p to Red
  - Else
    - Make p smaller in each dimension
      - s.t. p keeps precision \( \geq \tau \)
    - 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 < \( \tau \)
  - Then add p to Red
  - Else
    - Make p smaller in each dimension s.t. its precision remains \( \geq \tau \)
    - Add p to Green
    - Update Candidates
Algorithm

- **Incremental computation:**
  Each round add either red or green point
  - Pop p from Candidates
  - If precision \( p < \tau \) then add p to Red
  - Else
    - Make p smaller in each dimension s.t. \( p \) keeps precision \( \geq \tau \)
    - Add p to Green
    - Update Candidates

- **Invariant:** Candidates = set of points, s.t.
  - If all the candidates had precision \( < \tau \)
  - Then Green, Red \( \cup \) Candidates
    intermediate solution
Algorithm - Analysis

- Human labeling effort
  - # precision requests: $O(\#\text{Red points} + \# \text{Green points} \times d \times \log(k))$, 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

<table>
<thead>
<tr>
<th></th>
<th>Record Matching</th>
<th>Maximal Frequent Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Points</strong></td>
<td>{0, 1/k, 2/k, ..., 1}^d</td>
<td>{0, 1}^d</td>
</tr>
<tr>
<td></td>
<td>d number of similarity measures,</td>
<td>d number of items</td>
</tr>
<tr>
<td></td>
<td>k granularity (of descritizing space)</td>
<td></td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>Max recall</td>
<td>Max set size</td>
</tr>
<tr>
<td><strong>Constraint</strong></td>
<td>Precision (\geq \tau)</td>
<td>frequency (\geq \tau)</td>
</tr>
<tr>
<td><strong># requests</strong></td>
<td>Precision:</td>
<td>Frequency:</td>
</tr>
<tr>
<td></td>
<td>(O(#\text{Red} + # \text{Green} \ast d \ast \log(k)))</td>
<td>(O(#\text{Red} + # \text{Green} \ast d))</td>
</tr>
</tbody>
</table>

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

- Similarity dimensions
  - Jaccard similarity / containment, edit similarity
  - Total 8 dimensions
Experiments: Performance of our Algorithm

<table>
<thead>
<tr>
<th>Recall</th>
<th>29,500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>97%</td>
</tr>
<tr>
<td>Number of labels</td>
<td></td>
</tr>
<tr>
<td>Time per label</td>
<td></td>
</tr>
</tbody>
</table>

precision threshold $\tau = 0.95$, learning one point classifier, averages over 5 random runs
Experiments: Performance of our Algorithm

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>29,500</td>
</tr>
<tr>
<td>Precision</td>
<td>97%</td>
</tr>
<tr>
<td>Number of labels</td>
<td>84</td>
</tr>
<tr>
<td>Time per label</td>
<td>0.69 sec</td>
</tr>
</tbody>
</table>

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 [Elfekhi 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