Data Cleaning with Minimal Information Disclosure

Dhruv Gairola

Supervisor : Dr. Fei Chiang

June 22, 2015

Overview

Background

- Most raw datasets contain errors (e.g., misspellings, missing values, etc). Estimated loss of around 600 billion to U.S businesses.

Table: Addresses

	Name	Postal Code
t_1	John Perry	P4M 4K4
t_2	John Perry	P4M 4K4
t_3	John Perry	L4M 5P3

Name $\rightarrow$ Postal Code

Background

- Most raw datasets contain errors (e.g., misspellings, missing values, etc). Estimated loss of around 600 billion to U.S businesses.

Table: Addresses

	Name	Postal Code
t_1	John Perry	P4M 4K4
t_2	John Perry	P4M 4K4
t_3	John Perry	L4M 5P3

Name $\rightarrow$ Postal Code

Data cleaning systems

Table: Target data

	Name	Postal Code
t_1	John Perry	P4M 4K4
t_2	John Perry	P4M 4K4
t_3	John Perry	L4M 5P3

Table: Clean master data

	Name	Postal Code
m_1	John Kerry	Z4M 5P3
m_2	Susie Kerry	Z4M 5P3
m_3	Susie Kerry	Z4M 5P3
m_4	Susie Kerry	Z4M 5P3
m_5	Alice Robertson	B2R 6K6
m_6	Alice Robertson	B2R 6K6

Name $\rightarrow$ Postal Code

Problems

- Existing research does not take privacy considerations into account.

Problems

- Existing research does not take privacy considerations into account.
- Cannot assume that master data is public.

Problems

- Existing research does not take privacy considerations into account.
- Cannot assume that master data is public.
- Different records have different privacy requirements.

Problems

- Existing research does not take privacy considerations into account.
- Cannot assume that master data is public.
- Different records have different privacy requirements.
- Want to minimize the amount of information disclosed from the master data.

Problems

- Existing research does not take privacy considerations into account.
- Cannot assume that master data is public.
- Different records have different privacy requirements.
- Want to minimize the amount of information disclosed from the master data.
- Want the target data to maximally clean its values using information from master data.

Contributions

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.

Contributions

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.

Contributions

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.
3. Define a multi-objective problem based on above two measures. Four optimization functions to model problem.

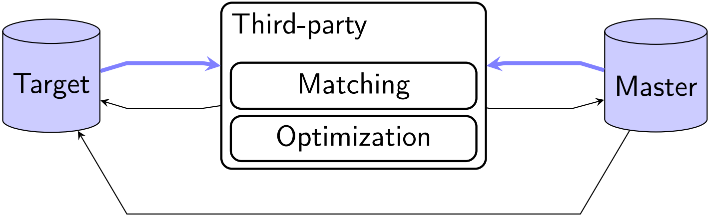
Contributions

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.
3. Define a multi-objective problem based on above two measures. Four optimization functions to model problem.
4. Perform experiments on datasets containing up to 3 million records.

Contributions

Note : Our algorithms work on embedded(obfuscated) records, not actual records. This protects the privacy of individual records.

Flow diagram



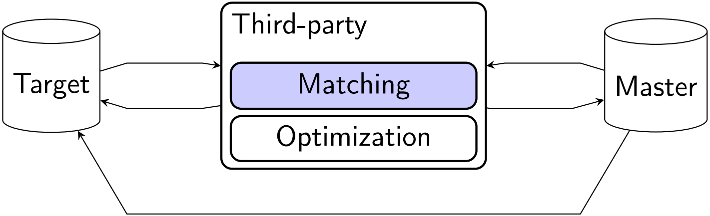
Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Flow diagram



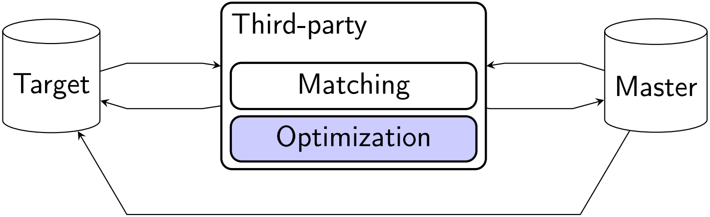
Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Flow diagram



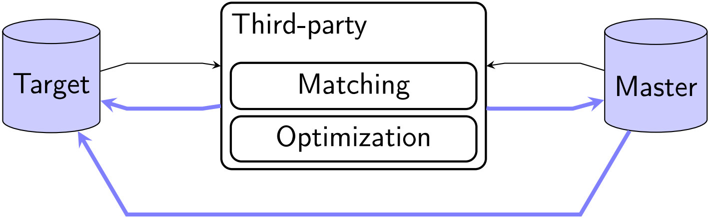
Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Flow diagram



Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Step 1: Embedding

Table: Target data

	Name	Postal Code
t_1	John Perry	P4M 4K4
t_2	John Perry	P4M 4K4
t_3	John Perry	L4M 5P3

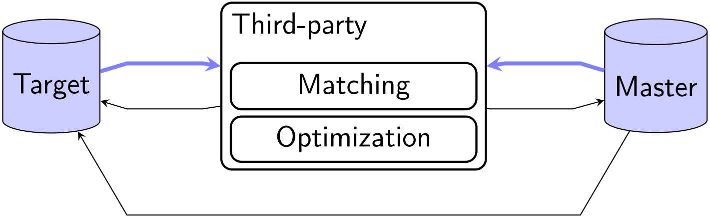
Table: Master data

	Name	Postal Code
m_1	John Kerry	Z4M 5P3
m_2	Susie Kerry	Z4M 5P3
m_3	Susie Kerry	Z4M 5P3
m_4	Susie Kerry	Z4M 5P3
m_5	Alice Robertson	B2R 6K6
m_6	Alice Robertson	B2R 6K6

	Name	Postal Code
t_1	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_2	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_3	[0.51, 0.57, 0.46, 0.46]	[0.5, 0.5, 0.5, 0.5]

	Name	Postal Code
m_1	[0.47, 0.59, 0.47, 0.47]	[0.48, 0.55, 0.48, 0.48]
m_2	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_3	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_4	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_5	[0.57, 0.46, 0.5, 0.46]	[0.5, 0.5, 0.5, 0.5]
m_6	[0.57, 0.46, 0.5, 0.46]	[0.5, 0.5, 0.5, 0.5]

Flow diagram



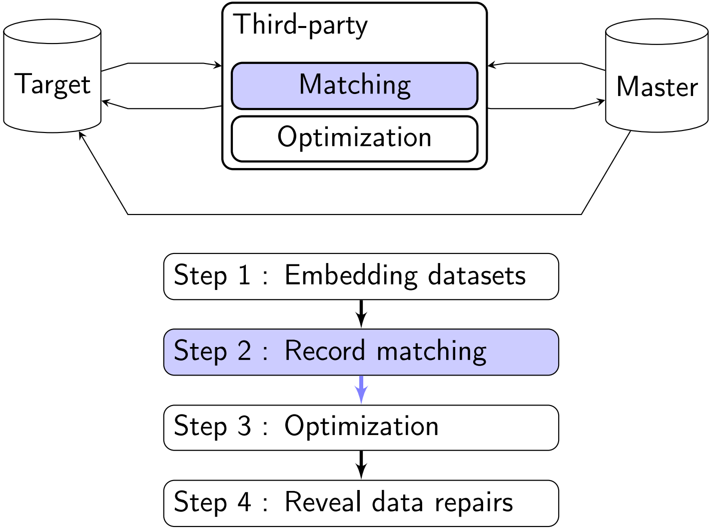
Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Flow diagram



Step 2: Record matching

Table: Embedded target data

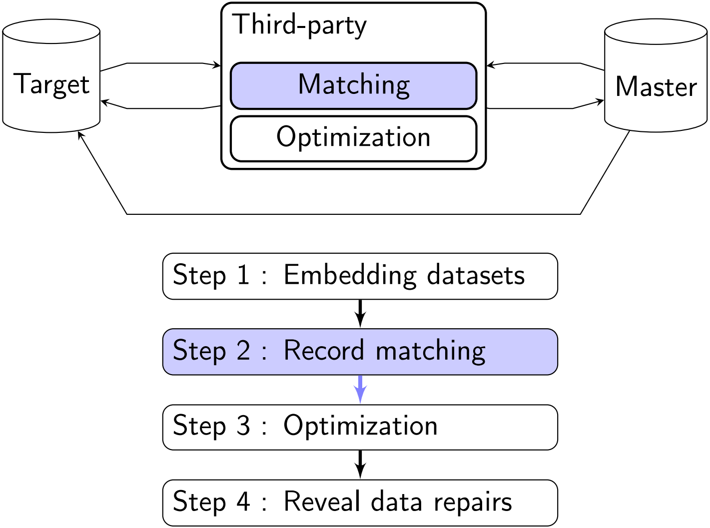
	Name	Postal Code
t_1	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_2	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_3	[0.51, 0.57, 0.46, 0.46]	[0.5, 0.5, 0.5, 0.5]

Table: Embedded reference data

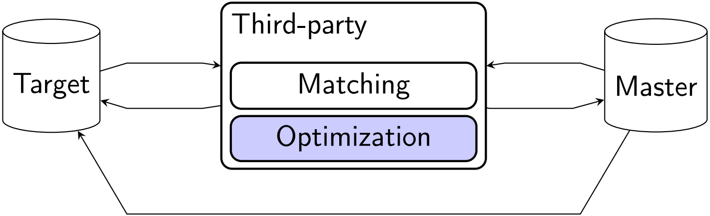
	Name	Postal Code
m_1	[0.47, 0.59, 0.47, 0.47]	[0.48, 0.55, 0.48, 0.48]
m_2	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_3	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_4	[0.49, 0.54, 0.49, 0.49]	[0.48, 0.55, 0.48, 0.48]
m_5	[0.57, 0.46, 0.5, 0.46]	[0.5, 0.5, 0.5, 0.5]
m_6	[0.57, 0.46, 0.5, 0.46]	[0.5, 0.5, 0.5, 0.5]

- t_3 matches m_1. Multiple matches allowed.
- Used normalized euclidean distance for matching.

Flow diagram



Flow diagram



Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Step 3: Optimization

- How much info is disclosed if m_1 is revealed? Does it benefit the target data?

Step 3: Optimization

- How much info is disclosed if m_1 is revealed? Does it benefit the target data?
- Decompose each match to *units*. e.g., (t_3, m_1, Name) and (t_3, m_1, Postal Code). Easy to visualize, easy to design algorithms.

Step 3: Optimization

- How much info is disclosed if m_1 is revealed? Does it benefit the target data?
- Decompose each match to *units*. e.g., (t_3, m_1, Name) and (t_3, m_1, Postal Code). Easy to visualize, easy to design algorithms.
- $\mathcal{U} = \{$ (t_3, m_1, Name), (t_3, m_1, Postal Code) $\}$.

Step 3: Optimization

- Optimization problem : find $C_{opt} \in \mathcal{P}(\mathcal{U})$ where C_{opt} minimizes information disclosure and maximizes data cleaning utility.

Step 3: Optimization, measuring information disclosure

Table: I_1

A	B	C
1		3
1	2	4

Table: I_2

A	B	C
	2	3
1	2	4

$$A \rightarrow B$$

- Grey cell in I_1 contains less info. than grey cell in I_2.

Step 3: Optimization, measuring information disclosure

Table: I_3

A	B	C
1		3
1	2	4

Table: I_4

A	B	C
1		3
1	2	4
1	2	5

$$A \rightarrow B$$

- Grey cell in I_3 contains same amount info. as grey cell in I_4.
- We don't want this. Hence, we introduce frequency information into the privacy measure.

Step 3: Optimization, measuring information disclosure

Table: Clean master data

	Name	Postal Code
m_1	John Kerry	Z4M 5P3
m_2	Susie Kerry	Z4M 5P3
m_3	Susie Kerry	Z4M 5P3
m_4	Susie Kerry	Z4M 5P3
m_5	Alice Robertson	B2R 6K6
m_6	Alice Robertson	B2R 6K6

Table: Info content table

	Name	Postal Code
m_1	0.82	0.99
m_2	0.82	0.46
m_3	0.82	0.46
m_4	0.82	0.46
m_5	0.67	0.67
m_6	0.67	0.67

Name $\rightarrow$ Postal Code

Step 3: Optimization, data cleaning utility

- Use information dependency *ind* to measure data cleaning utility.
- For $C \in \mathcal{P}(\mathcal{U})$, apply C to target table, and then measure *ind*.
- $ind(C) = H(X \cup Y) - H(X)$ for an FD $F : X \to Y$.

Step 3: Optimization, example

- $\mathcal{U} = \{\ (t_3,\ m_1,\ \text{Name}),\ (t_3,\ m_1,\ \text{Postal Code})\ \}$.
- One example of $C \in \mathcal{P}(\mathcal{U})$ is $C = \{(t_3,\ m_1,\ \text{Name})\}$.
- Info disclosure, $pvt(C) = 0.82$.

Table: Info content table

	Name	Postal Code
m_1	0.82	0.99
m_2	0.82	0.46
m_3	0.82	0.46
m_4	0.82	0.46
m_5	0.67	0.67
m_6	0.67	0.67

Step 3: Optimization, example

- Data cleaning utility, $ind(C) = 0$ where C ={(t_3, m_1, Name)}.

	Name	Postal Code
t_1	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_2	[0.51, 0.57, 0.46, 0.46]	[0.48, 0.55, 0.48, 0.48]
t_3	~~[0.51, 0.57, 0.46, 0.46]~~ [0.47, 0.59, 0.47, 0.47]	[0.5, 0.5, 0.5, 0.5]

$ind(C) = 0$, that is the same as

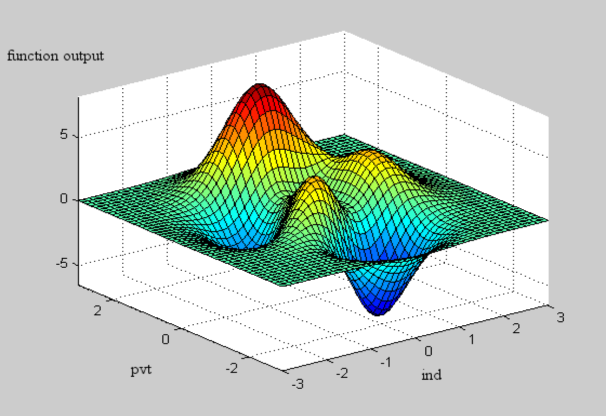
	Name	Postal Code
t_1	John Perry	P4M 4K4
t_2	John Perry	P4M 4K4
t_3	~~John Perry~~ John Kerry	L4M 5P3

$$ind(C) = 0$$

Step 3: Optimization

- We showed how *pvt* and *ind* can be calculated for some $C \in \mathcal{P}(\mathcal{U})$.
- How to find best candidate $C_{opt} \in \mathcal{P}(\mathcal{U})$ s.t. *pvt* is minimized and *ind* is minimized?

Step 3: Optimization



Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
  1. Pick random solution C. Mark it as current solution.

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
  1. Pick random solution C. Mark it as current solution.
  2. Loop k times:

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
  1. Pick random solution C. Mark it as current solution.
  2. Loop k times:
  3. Get a random neighbor of current solution. Denoted C_n.

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
  1. Pick random solution C. Mark it as current solution.
  2. Loop k times:
  3. Get a random neighbor of current solution. Denoted C_n.
  4. If C_n better than C, mark C_n as current soln.

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
  1. Pick random solution C. Mark it as current solution.
  2. Loop k times:
  3. Get a random neighbor of current solution. Denoted C_n.
  4. If C_n better than C, mark C_n as current soln.
  5. Else, mark C_n as current soln with some probability P.

Step 3: Optimization, Simulated annealing search algorithm

- Given an objective function e.g., $fn(C) = 0.5 * pvt(C) + 0.5 * ind(C)$.
    1. Pick random solution C. Mark it as current solution.
    2. Loop k times:
    3. Get a random neighbor of current solution. Denoted C_n.
    4. If C_n better than C, mark C_n as current soln.
    5. Else, mark C_n as current soln with some probability P.
    6. End Loop

Step 3: Optimization, Simulated annealing search algorithm

- Example:

Step 3: Optimization, Simulated annealing search algorithm

- Example:
  1. C ={ (t_3, m_1, Name), (t_3, m_1, Postal Code) }

Step 3: Optimization, Simulated annealing search algorithm

- Example:
  1. C ={ (t_3, m_1, Name), (t_3, m_1, Postal Code) }
  2. Loop 100 times:

Step 3: Optimization, Simulated annealing search algorithm

- Example:
  1. C ={ (t_3, m_1, Name), (t_3, m_1, Postal Code) }
  2. Loop 100 times:
  3. C_n. e.g., $C_n = \{(t_3, m_1, \text{Name})\}$

Step 3: Optimization, Simulated annealing search algorithm

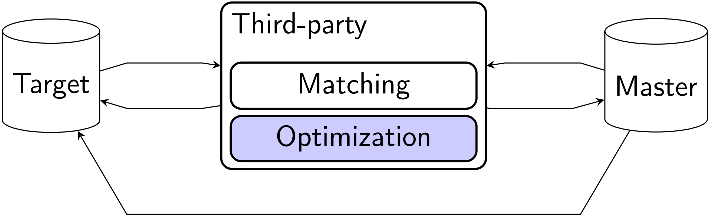
- Example:
  1. $C = \{ (t_3, m_1, \text{Name}), (t_3, m_1, \text{Postal Code}) \}$
  2. Loop 100 times:
  3. C_n. e.g., $C_n = \{(t_3, m_1, \text{Name})\}$
  4. if $fn(C_n) < fn(C)$, current soln is C_n

Step 3: Optimization, Simulated annealing search algorithm

- Example:
  1. $C = \{ (t_3, m_1, \text{Name}), (t_3, m_1, \text{Postal Code}) \}$
  2. Loop 100 times:
  3. C_n. e.g., $C_n = \{(t_3, m_1, \text{Name})\}$
  4. if $fn(C_n) < fn(C)$, current soln is C_n
  5. else, current soln is C_n with some probability P

- Example:
  1. C ={ (t_3, m_1, Name), (t_3, m_1, Postal Code) }
  2. Loop 100 times:
  3. C_n. e.g., $C_n = \{(t_3,\ m_1,$ Name$)\}$
  4. if $fn(C_n) < fn(C)$, current soln is C_n
  5. else, current soln is C_n with some probability P
  6. End Loop

Flow diagram



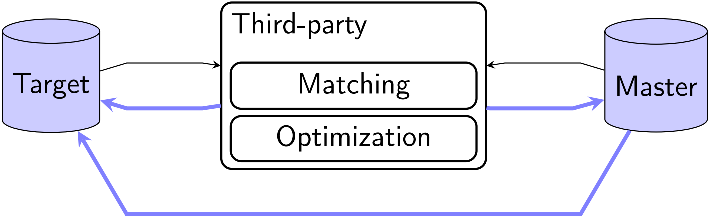
Step 1 : Embedding datasets

Step 2 : Record matching

Step 3 : Optimization

Step 4 : Reveal data repairs

Flow diagram



Step 1 : Embedding datasets

↓

Step 2 : Record matching

↓

Step 3 : Optimization

↓

Step 4 : Reveal data repairs

Step 4: Revealing data repairs

- We have solved the multi-objective problem to get optimal solution C_{opt}. e.g., $\{(t_3,\ m_1,\ \text{Name})\}$

Step 4: Revealing data repairs

- We have solved the multi-objective problem to get optimal solution C_{opt}. e.g., $\{(t_3, m_1, \text{Name})\}$
- Third-party does not know any values inside the solution.

Step 4: Revealing data repairs

- We have solved the multi-objective problem to get optimal solution C_{opt}. e.g., $\{(t_3,\ m_1,\ \text{Name})\}$
- Third-party does not know any values inside the solution.
- Asks master data to directly reveal the data values to the target data. e.g., reveal $m_1[Name]$, which is "John Kerry" to target data for $t_3[Name]$.

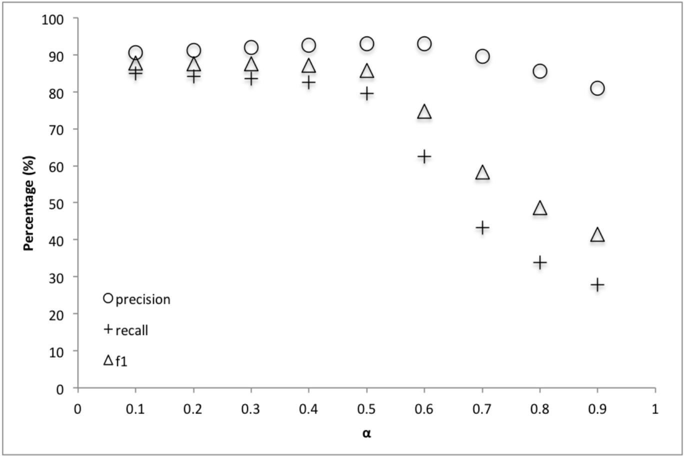
Experiments

- IMDB: 14 attributes; 1.2 million records.
- Books: 12 attributes; 3 million records.
- Java 1.7; 4 virtual CPUs (2.1 GHz each); 32 GB of memory

Experiments

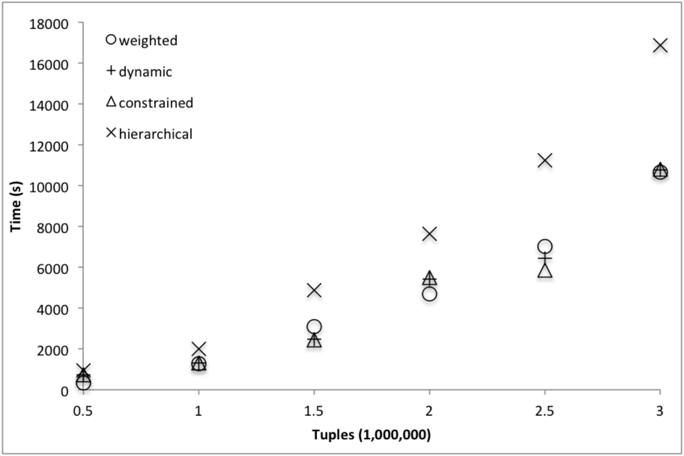
- Accuracy : measure quality of data repairs.
- Performance : running time.
- Comparative.

Experiments : Accuracy



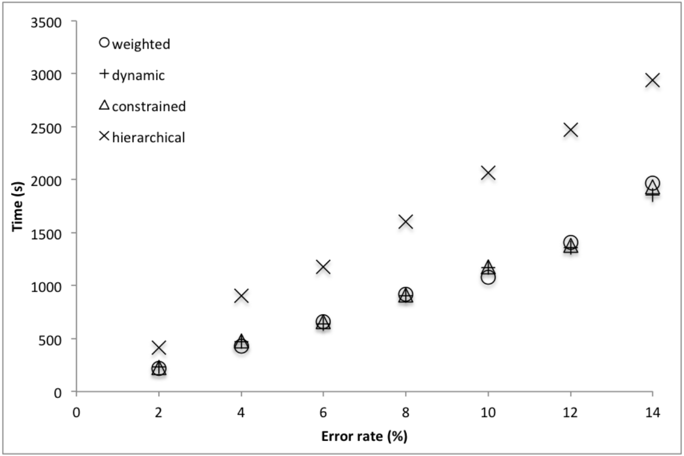
- Inverse correlation between utility and privacy.

Experiments : Performance



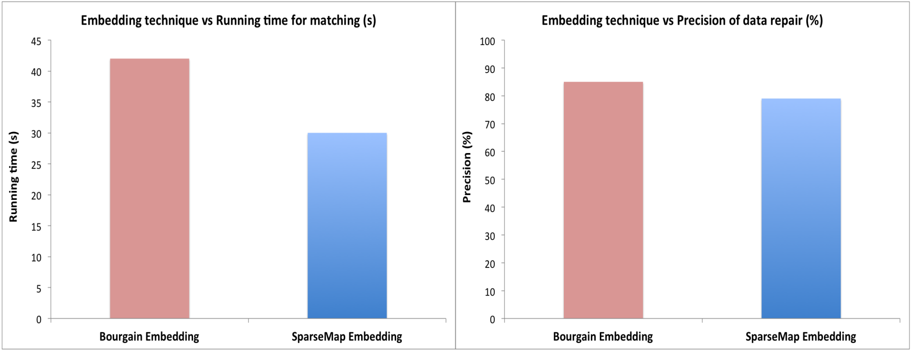
- Slowest function takes 2 hrs on average for 0.5-3 million tuples (with 8% error rate).

Experiments : Performance



- Slowest function takes 30 mins on average for 2-14% error rate (with 0.5 million tuples).

Experiments : Comparative



Conclusion

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.

Conclusion

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.

Conclusion

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.
3. Define a multi-objective problem based on above two measures. Four optimization functions to model problem (not described in this presentation).

Conclusion

1. Data cleaning framework: (i) master data discloses a minimal amount of info. and (ii) target data uses this to maximally clean its values.
2. Propose two measures: (i) information disclosure and (ii) data cleaning utility.
3. Define a multi-objective problem based on above two measures. Four optimization functions to model problem (not described in this presentation).
4. Perform experiments on datasets containing up to 3 million records.

Future work

- Try other models for information disclosure.

Future work

- Try other models for information disclosure.
- Explore other constraints e.g., matching dependencies for the record matching step.

Thank you

Simulated annealing

- If the simulated annealing algorithm is iterated enough, it will find the global optimum with probability approaching 1 (Geman and Geman, 1984).

Simulated annealing

- If the simulated annealing algorithm is iterated enough, it will find the global optimum with probability approaching 1 (Geman and Geman, 1984).
- We are minimizing information disclosure and maximizing data cleaning utility.

Simulated annealing

- If the simulated annealing algorithm is iterated enough, it will find the global optimum with probability approaching 1 (Geman and Geman, 1984).
- We are minimizing information disclosure and maximizing data cleaning utility.
- We have 4 optimization functions to model our problem statement.

Optimization functions

- Weighted method.

$$\min_{C} \quad \alpha * pvt(C) + \beta * ind(C, T') + \gamma * changes(C)$$

- Constrained method.

$$\begin{aligned} \min_{C} \quad & pvt(C) \\ \text{s.t.} \quad & ind(C, T') \leq \varepsilon_i \\ & changes(C) \leq \varepsilon_j \end{aligned}$$

Optimization functions

- Dynamic method.

$$
\begin{aligned}
\min_{C} \quad & pvt(C) \\
\text{s.t.} \quad & ind(C, T') \leq ind(C_0, T') \\
& changes(C) \leq changes(C_0)
\end{aligned}
$$

- Hierarchical method.

$$
\begin{aligned}
\min_{C} \quad & pvt(C) \\
& \min_{R} \quad ind(C, T') \\
& \text{s.t.} \quad pvt(C) \leq \varepsilon_k \\
& \qquad \min_{C} \quad changes(C) \\
& \qquad \text{s.t.} \quad pvt(C) \leq \varepsilon_k \\
& \qquad \qquad ind(C, T') \leq \varepsilon_l
\end{aligned}
$$

Embedding distortion

- (Bourgain) For every n-point metric space there exists an embedding into Euclidean space with distortion O(log n). [Advances in Metric Embedding Theory, Abraham, 2006]
- Experimentally, for a 20-dimensional metric space, we observed 88% precision.

Information disclosure

$$P(a) = \begin{cases} 0 & M_{c \leftarrow a} \not\models F_i \\ \frac{1}{|b|} & otherwise \end{cases}$$

$$b = \{a | M_{c \leftarrow a} \models F_i\}$$
$$inf(c) = H(\mathcal{E}(M, c)) = \sum_{a \in adom(A) \cup v} P(a) \log \frac{1}{P(a)}$$

Information disclosure : our measure

$$P'(a) = freq(a) * P(a)$$
$$einf(c) = \sum_{a \in adom(A) \cup v} P'(a) \log \frac{1}{P'(a)}$$