Record Matching with Privacy Constraints

(..in progress..)

Luca Bonomi

Department of Computer Science
Emory University

April 21, 2011
Outline

1. Introduction
   - Problem Definitions
   - Recent Works and Motivations

2. The "Big Plan"
   - Many questions vs. few ideas
   - Embedding strategy
   - Some preliminary results
Problem

Let Alice (A) and Bob (B) be two parties with database $D_A$ and $D_B$ respectively, that are agree on a privacy model. We are interested in deciding if there exists a pair $(x, y)$ of records $x \in D_A$ and $y \in D_B$ which describes the same entity (i.e. match).
What we need?

- **Communication protocol**: how many parties are involved in the matching and how they interact?

- **Privacy model**: the entities involved do not want to share their data. (sensitive data)
  - Cryptographic protocols, the privacy is guaranteed but it too expensive!
  - Data anonymization, k-anonymity, differential privacy, etc...

- **Matching schema**: how we decide if two records describe the same entity?
  - Exact matching
  - Similarity measure and approximate matching.
The definition of the problem is too general and the literature too broad. We better to consider a more specific scenario.

- Record matching between strings. Many similarity measures are used on strings (e.g. Hamming, Ulam). We choose Edit: it is expressive and challenging enough.

- Communication and privacy protocol. In literature, several approaches are consider: 2 or more parties (trusted or semi-honest).
Recent Techniques

- Under the privacy constraints, the parties involved in the matching are not allowed to directly share their data.
  - The matching is done between *embedded* records into a new space.
  - Hybrid techniques: record matching = blocking + differential privacy. [Ali10]
Few words on embedding

Definition

Given two metric spaces $M = (X, d)$ and $M' = (X', d')$, a map $f : X \to X'$ is said to have distortion $\epsilon \geq 1$, or to be an $\epsilon$-embedding, if there exists $\lambda > 0$ such that:

$$\lambda \cdot d(x, y) \leq d'(f(x), f(y)) \leq \epsilon \lambda \cdot d(x, y), \quad \forall x, y \in X.$$

The stress $S$ for a given embedding $f$ on a finite set $X$, is defined as follows:

$$S = \frac{\sum_{x,y \in X} [d'(f(x), f(y)) - d(x, y)]^2}{\sum_{x,y \in X} d(x, y)^2}.$$
Few words on embedding

- Lipschitz Embedding: transforms the original data into a new space by projections on a random base $B$ (e.g. SparseMAP).
- Although the embedding strategy helps from a computational point of view (i.e. the target space is a nice space), it is not well understood which guarantees we have in terms of distortion and privacy on the data.
Motivations

- A possible approach based on SparseMAP: The two parties embed their data according to a shared random based and the matching is done by an external “honest” party. [Mon07]

- Problems:
  - The new space has high dimension: $\Theta(\ln^2 |X|)$, and the base it is large.
  - We need an extra (semi-honest) party (not always suitable).
Random strings are not very informative, why do we not choose strings from our datasets and use them as base?

Can we use differential privacy to solve the query in the new space, so we can avoid the external (honest) party?
Looking for a better embedding

- We want to make strings easy to be distinguished in the new space, and try to reduce the dimensionality.
  - Select a set of candidate strings from the dataset to construct the target space
  - Embedding: via q-grams, and/or SparseMAP.

Luca Bonomi
Record Matching with Privacy Constraints
Select candidates

- Define a hierarchy among the strings in the database:
  - Build the Minimum Spanning Tree on the dataset: each string is a leaf, each edge is weighted with the Edit distance between the strings.
  - Internal nodes represent ancestors. (Longest Common Subsequence)

- Given $k > 0$, we generate a set of candidate strings $G$ by sampling the internal node. (it is naive...we can use some heuristics to select the $k$ "most informative" strings).
Embedding via q-grams

- Given a set of candidate strings $G$, compute the dictionary $D_q(G)$

$$D_q(G) := \{ \text{all blocks of size } q \text{ in } x, \forall x \in G \}.$$  

- So, given $D_q(G) = \{b_0, b_2, \ldots, b_k\}$ and $x \in X$, we define the embedding function $f$ as the characteristic vector $\bar{x}$ of $x$ under $D_q(G)$. Compute $f(x) := \bar{x}$ where,

$$\bar{x}_i := \text{number of occurrences of } b_i \text{ in } x, i = 0, 1, \ldots, k.$$  

- Given $x, y \in X$, their distance in the target space is computed by using the Euclidean distance between their respective vectors $\bar{x}, \bar{y}$.  

Quality of the embedding

Relative Stress reduction with different size of q-gram

- 4 cand. strings
- 6 cand. strings

Distortion vs. q-gram size

- 4 cand. strings
- 6 cand. strings
- upper bound?

Luca Bonomi
Record Matching with Privacy Constraints
Introduction
The "Big Plan"
Summary

Summary

- It seems that a non-random base may help for the embedding.
- Extend/adapt this technique to Lipschitz embedding. (..coming soon..)

Open problems:
- Can we bound the distortion and make the embedding contractive?
- Precision/Recall analysis.
- Privacy analysis.