Privacy in stream mining by Ohad Heller and Eytan Ikan

**Description of the problem context and high-level explanation of the algorithm:**

In data mining, data such as purchased goods, query terms, or individual preferences, is instrumental for some applications e.g association rule mining, query expansion, and predicting user behavior. Still, their publication poses privacy threats. A particular threat is posed by an adversary who has partial knowledge about a certain person’s transaction t, and may use it, along with the published data, to derive additional knowledge about the contents of t, thus identifying previously unknown sensitive items in t. Therefore, there is high importance for privacy preservation in data mining.

There are some algorithms for privacy preserving in static data bases. In our project we will implement an algorithm for data preserving for data streams, meaning a non-static database. We will use a sliding window method. A sliding window is a data structure which has two fields: size and step. The size of the sliding window determines the number of transactions we look at at a certain time and the step determines the number of transactions that are replaced at each step in the stream. We will simulate the sliding window by taking a static database and creating a frame in which we look at a certain number of random transactions from the database and in each step replace some of the transactions with other random transactions from the database.

At first, we will use the algorithm depicted in the article -uncertainty: Inferenceproof transaction anonymization” at each step of the sliding window. The algorithm uses suppression and generalization to maintain -uncertainty”, which is the concept of maintaining the *confidence* of each SAR lower than a threshold ρ. In the suppression method, we will use only global suppression, i.e., a suppressed item is deleted from *all* transactions in the database. In each iteration we have a group SARi (i is the number of iteration) which is the set of SARs that violate ρ-uncertainty, hence *must* be concealed, in the ith iteration. Given an item b, either sensitive or non-sensitive, let C(b, SRi ) be the number of rules in SRi that contain it, and sup(b) the number of transactions in the database that contain it. We define the *payoff ratio* of item b with respect to SRi as payoff(i, b) = C (b,SRi)/sup(b) . Higher payoff is preferable. hence, we will spuress the item with the highest payoff.

In the generalization method, like in the suppression method we will use global generalization, i.e mapping *all* instances of an item b, as well as *all* items of the *same* subtree of hierarchy H, to the *same* level in H. We will generalize only non-sensitive item. In a nutshell, our generalization method is to perform a *top-down particularization* process. It starts out assuming that *all* items in the database are generalized at the top level of hierarchy H, represented by the root node. This state of affairs satisfies ρ-uncertainty but provides no information at all. In order to recover some information, we relax, or *particularize*, the generalization by moving along branches of H in a *greedy* manner.

After we implement this algorithm, we will try to understand the article “Two Privacy-Preserving Approaches for Publishing Transactional Data Streams “ by JINYAN WANG , CHAOJI DENG, AND XIANXIAN LI. The article supplies a more efficient way of updating the association rules after each step of the sliding window.

**Description of the software (language, libraries, etc.):**

For the analysis of the dataset file we will use Python’s libraries: Scipy and Numpy for the analysis of the data, and Pandas for converting the .data file to a more convenient .csv file.

A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record.

After performing the pre analysis on the entire database, we’ll create the first Sliding-Window by randomly choosing a number of entries (we’ll call it window\_size) from the main database, copy those to a new csv file: sliding\_window.csv, and then perform the preprocessing to make sure we maintain the -uncertainty.

**Data set description: What is in the data, and what preprocessing will be done for this project:**

Nursery Data Set- originally developed to rank applications for nursery schools, it contains the following information about the applications for the nursery schools:

**parents**: usual, pretentious, great\_pret

**has\_nurs**: proper, less\_proper, improper, critical, very\_crit

**form**: complete, completed, incomplete, foster

**children**: 1, 2, 3, more

**housing**: convenient, less\_conv, critical

**finance**: convenient, inconv

**social**: non-prob, slightly\_prob, problematic

**health**: recommended, priority, not\_recom

This information directly relates NURSERY to the eight input attributes: parents, has\_nurs, form, children, housing, finance, social, health.

For this project, we’ll pretend a nursery school hired our services for data mining information about nursery students around israel. That company is looking for the best nursery students in the country, and therefore it will need all the information it can get. As a well-known data mining company- we need to maintain our reputation of preserving privacy in the data stream. Therefore, we will preprocess each sliding window we create before sending it to the nursing school, by using the algorithm described above.

**References:** (the paper chosen and some others related to the topic):

J. Cao, P. Karras, C. Raïssi, and K. L. Tan, “-uncertainty: Inference Proof

transaction anonymization,'' in Proc. VLDB, Singapore, 2010,

pp. 10331044.

JINYAN WANG , CHAOJI DENG, AND XIANXIAN LI, “Two Privacy-Preserving Approaches for

Publishing Transactional Data Streams “.