Amrita Vishwa Vidyapeetham
Amrita School of Engineering, Coimbatore
Department of Computer Science and Engineering
15CSE401 - Machine Learning & Data Mining
Case Study Review - 1

Team Name: Orion **Group Number: 5**

S.No	Name	Roll Number
1	Abhishek S	CB.EN.U4CSE17003
2	Mahesha Ganapath	CB.EN.U4CSE17038
3	Manojkumar VK	CB.EN.U4CSE17040
4	Sudharshan S	CB.EN.U4CSE17059
5	Vasudevan P	CB.EN.U4CSE17067

Abstract:

Personalized recommendation is crucial to help users find pertinent information. It often relies on a large collection of user data, in particular users' online activity (e.g., tagging/rating/checking-in) on social media, to mine user preference. However, releasing such user activity data makes users vulnerable to inference attacks, as private data (e.g., gender) can often be inferred from the users' activity data. PrivRank is a customizable and continuous privacy-preserving social media data publishing framework protecting users against inference attacks while enabling personalized ranking-based recommendations. Its key idea is to continuously obfuscate user activity data such that the privacy leakage of user-specified private data is minimized under a given data distortion budget, which bounds the ranking loss incurred from the data obfuscation process in order to preserve the utility of the data for enabling recommendations. An empirical evaluation on both synthetic and real-world datasets shows that our framework can efficiently provide effective and continuous protection of user-specified private data, while still preserving the utility of the obfuscated data for personalized ranking-based recommendation.

Introduction:

Developing effective recommendation engines is critical in the era of Big Data in order to provide pertinent information to the users. To deliver high-quality and personalized recommendations, online services such as ecommerce applications typically rely on a large collection of user data, particularly user activity data on social media, such as tagging/rating records, comments, check-ins, or other types of user activity data. In practice, many users are willing to release the data (or data streams) about their online activities on social media to a service provider in exchange for getting high-quality personalized recommendations. Such user activity data are referred to as public data. However, they often consider part of the data from their social media profile as private, such as gender, income level, political view, or social contacts. Those data are referred to as private data. Although users may refuse to release private data, the inherent correlation between public and private data often causes serious privacy leakage.

Existing Work:

To protect user privacy when publishing user data, the current practice mainly relies on policies or user agreements, e.g., on the use and storage of the published data. However, this approach cannot guarantee that the users' sensitive information is actually protected from a malicious attacker. Therefore, to provide effective privacy protection when releasing user data, privacy-preserving data publishing has been widely studied. Its key idea is to obfuscate user data such that published data remains useful for some application scenarios while the individual's privacy is preserved. According to the attacks considered, existing work can be classified into two categories. The first category is based on heuristic techniques to protect ad-hoc defined user privacy. The second category is theory based and focuses on the fact that published data should provide attackers with as little additional information as possible beyond background knowledge.

Disadvantage:

To release private data, the inherent correlation between public and private data often causes serious privacy leakage. It is thus crucial to protect user private data when releasing public data to recommendation engines.
More distortion of public data leads to better privacy protection, as it makes it harder for adversaries to infer private data.

Proposed Work:

PrivRank is a customizable and continuous privacy-preserving social media data publishing framework. It continuously protects user-specified data against inference attacks by releasing obfuscated user activity data, while still ensuring the utility of the released data to power personalized ranking-based recommendations. To provide customized protection, the optimal data obfuscation. In order to reduce the privacy leakage, we obfuscate X to obtain $\overline{X}$ based on a probabilistic obfuscation function $p(\overline{X}|X)$, which encodes the conditional probability of releasing $\overline{X}$ when observing X. Intuitively, $p(\overline{X}|X)$ should be designed such that any inference attack on Y should be rendered weak. Meanwhile, it also keeps some utility of $\overline{X}$ by limiting the distortion budget in the obfuscation process, which can be modeled by a constraint ΔX as follows:

$$E_{X,\overline{X}}(\mathrm{dist}(X, \overline{X})) \leq \Delta X$$

where dist(X, $\overline{X}$) is a certain distance metric that measures the difference between $\overline{X}$ and X. ΔX limits the expected distortion w.r.t. the probabilistic obfuscation function p($\overline{X}$|X). The data distortion budget can ensure the utility of the released data. Considering the data utility for enabling personalized ranking-based recommendation, the data distortion budget is measured and bounded using ranking distance. In summary, the key idea of our solution is to learn p($\overline{X}$|X) that minimizes ΔC under a given distortion budget ΔX.

Dataset Description:

We intend to use IBMGenerator. It is a Synthetic Data Generator for Itemsets and Sequences.

Itemset Datasets:

These datasets mimic the transactions in a retailing environment, where people tend to buy sets of items together, the so called potential maximal frequent set. The size of the maximal elements is clustered around a mean with a few long itemsets. A transaction may contain one or more of such frequent sets. The transaction size is also clustered around a mean, but a few of them may contain many items. Let D denote the number of transactions, T the average transaction size, I the size of a maximal potentially frequent itemset, L the number of maximal potentially frequent itemsets, and N the number of items. The data is generated using the following procedure. We first generate L maximal itemsets of average size I by choosing from the N items. We next generate D transactions of average size T by choosing from the L maximal itemsets.

Link : https://github.com/zakimjz/IBMGenerator