
(k, P) -Anonymity

KAPRA Algorithm Implementation

-
- Alex Di Stefano (S4638131)
 - Christian Stingone (S4644983)

Contents

- Purpose
 - k and P Anonymization Levels
 - Why two levels
 - KAPRA Algorithm
 - Dataset
 - Analysis
-

Purpose

Time series is one of the most important types of data. It can be produced from Sensors, RFIDS, financial analysis, ...

Such massive data imply vast amount of privacy

We want to anonymize data preserving complex query such as range and pattern matching queries.



So... use k-anonymity? **No, it suffer from Pattern Loss**

Then... (k, P)-anonymity!

k and P Anonymization Levels

k-requirement: Each anonymization envelope appears at least k times.

P-requirement: Consider any k-group G of time series having the identical anonymization envelope, for any time series (r) in G, there are at least P-1 other time series in G having the same QI pattern representation as PR[r]

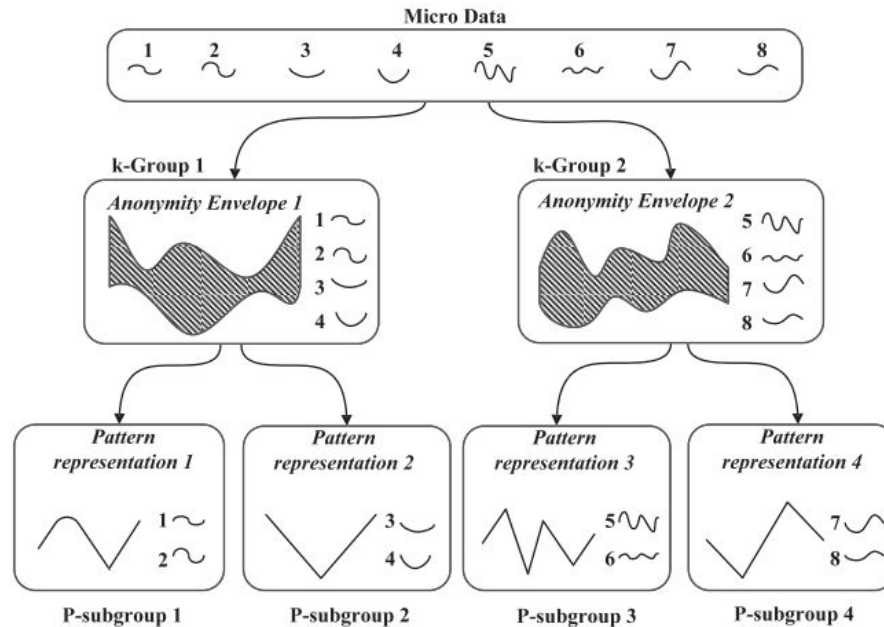
Why two levels?

First level: k-anonymity is required for time series in the entire database. That means the records in the published database can be grouped by the quasi-identifier attribute values, and each group should contain at least k records.

Second level: P-anonymity is required for the pattern representations (PRs) associated with each record in a same group. Specifically, each group can be divided into subgroups, each of which contains at least P records having identical PRs.

Main purpose is to achieve minimal pattern loss

Why two levels?



The k-groups and P-subgroups of (k,P)-Anonymity

KAPPA Algorithm

Supporting Pattern-Preserving Anonymization for Time-Series Data

Lidian Shou, Xuan Shang, Ke Chen, Gang Chen, and Chao Zhang

Abstract—Time series is an important form of data available in numerous applications and often contains vast amount of personal privacy. The need to protect privacy in time-series data while effectively supporting complex queries on them poses nontrivial challenges to the database community. We study the anonymization of time series while trying to support complex queries, such as range and pattern matching queries, on the published data. The conventional k -anonymity model cannot effectively address this problem as it may suffer severe pattern loss. We propose a novel anonymization model called k - P -anonymity for pattern-rich time series. The model addresses both the privacy concern and the patterns of time series to support the former. We demonstrate that our model can prevent leakage through the published data while effectively support a wide variety of queries on the anonymized data. We propose two algorithms to enforce k - P -anonymity on time-series data. Our anonymity model supports customized data publishing, which allows a wider part of the value field of the patterns of the anonymized time series to be published simultaneously. We present evaluation techniques to support query processing on such customized data. The proposed methods are evaluated in a comprehensive experimental study. Our results verify the effectiveness and efficiency of our approach.

Index Terms—Privacy, anonymity, pattern, time series

1 INTRODUCTION

Time series has long been considered one of the most important types of data available in both nature and human society. In recent years, the popularity of sensor networks, RFID, and wireless positioning equipments has further driven the production of time-series data to unprecedented volume and complexity. The publicity of these data on the Internet has nurtured the most creative applications ranging from financial analysis to social community tracking and pattern matching. However, such massive data also imply vast amount of privacy, which, if not appropriately protected, may become exploited as a source for abuse and crimes.

Privacy protection in the publication of time series is a challenging topic mostly due to the complex nature of the data and the way that they are used. In particular, the spectrum of frequently used “complex” queries on time series covers not only range queries on the attribute values at specified time instants but also pattern similarity queries which treat each sequence more globally. Unfortunately, it is no trivial task to support such variety of queries without disclosing the sensitive information of individuals.

Specifically, we consider an essential problem of anonymizing time series while trying to support the queries mentioned above. For example, in a deidentified database of monthly sales of companies, users may issue

1. Range queries which specify value conditions, such as *select * from dataset where sales December < [1 million, 1.2 million]*, or
2. Pattern matching queries which rely on the definition of pattern similarity, such as *Given time series q , select * from data set where similarity(q, q_i) > threshold* for $i \in \{1, 2, \dots, n\}$.

Meanwhile, it is critical to ensure that no identifiers of these companies will be disclosed. However, the time-sensitive attribute values and their patterns can be used as strong quasi-identifiers [20] to break linkage attacks which randomly some of the records (time series). For instance, an adversary can learn from the external source that the monthly sale of the victim happens to be between 1 and 1.2 million, and by checking among the published data to achieve that figure. Then, s/he could issue the above range query on the published data and the link between the victim's ID and the sensitive attribute values is easily established. Similarly, the pattern matching query which retrieves few results from the database could also be used for attacking.

The above example reveals the difficulties encountered when anonymizing time series. On one hand, the instant values and global patterns of time series have to be retained in the published data as much as possible to support various queries. On the other hand, the linkage attacks based on knowledge of values, patterns, or both, have to be prevented.

The conventional solution to prevent linkage attacks is to enforce k -anonymity [21] [14] on the published database, so that each record has its QI attributes identical to at least $k - 1$ other records. Although conventional k -anonymity can be used to resist linkage attacks, it cannot effectively preserve the patterns, which are critical for performing queries on time series. A few previous studies [17], [10] have proposed methods to anonymize sequences or

• The authors are with the College of Computer Science, Zhejiang University Hangzhou 310027, P.R. China.
E-mail: shouliidian@zhu.edu.cn, shangxuan@zhu.edu.cn.
Manuscript received 2 Jan. 2011; revised 27 Nov. 2011; accepted 19 Nov. 2011; published online 1 Dec. 2011.
This paper is an extended version of the short paper titled “Supporting Pattern-Preserving Anonymization for Time-Series Data” published in the Proceedings of the 2011 IEEE International Conference on Data Engineering (ICDE), 2011.
Digital Object Identifier 10.1109/TKDE.2011.201

- Create-tree phase with entire dataset
 - Initialization
 - Node Splitting
- Recycle bad-leaves phase
- Group formation phase
 - Top-Down Preprocessing
 - Group Formation
 - Group Post Processing

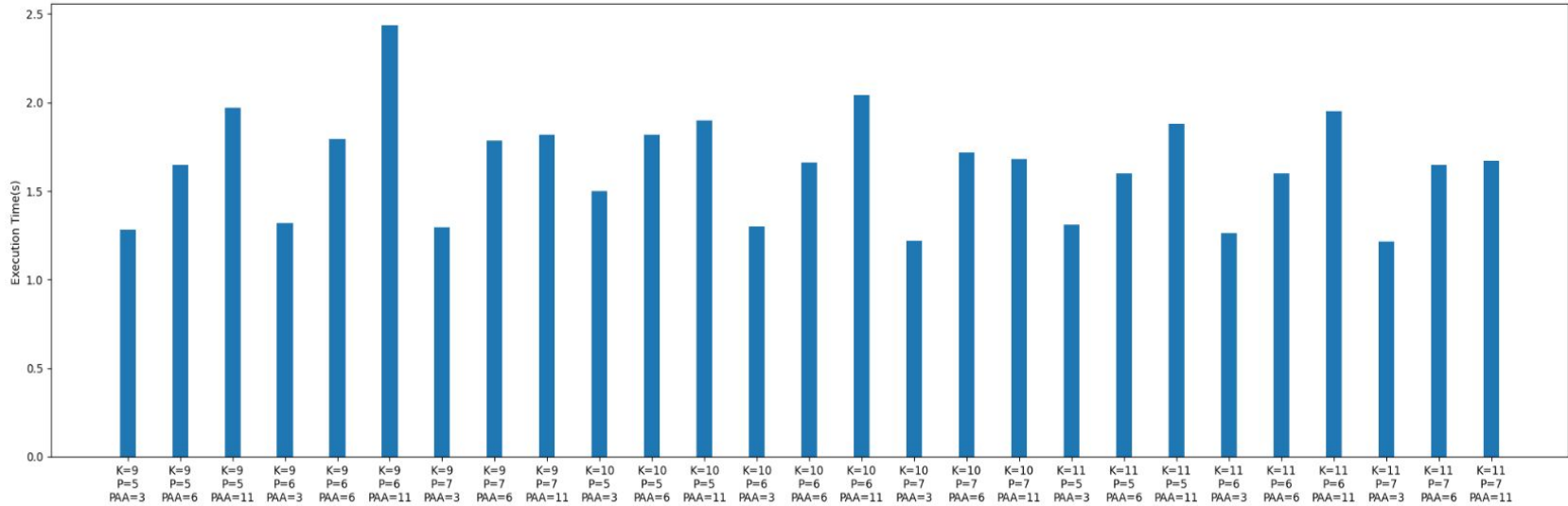
Dataset

StackOverflow Questions Count Time Series: consist of count of various questions of specific libraries for each month. Used for time analysis and Instant Value Loss analysis.

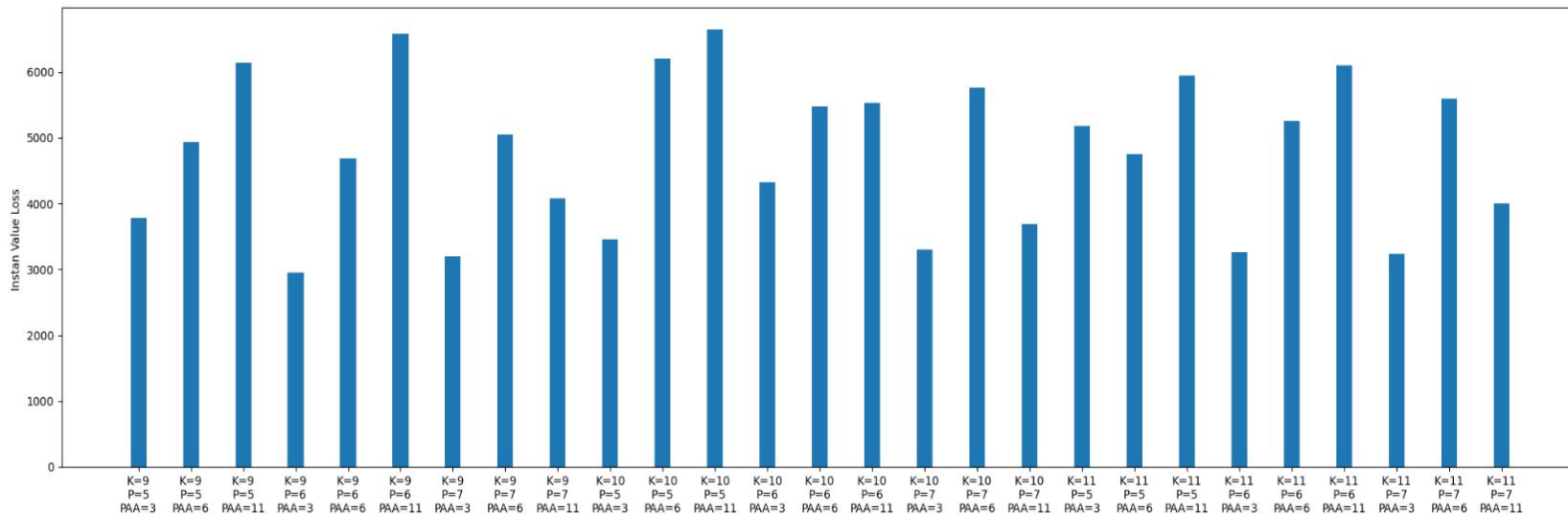
MLTollsStackOverflow:

- 82 columns
- 133 rows

Analysis - Execution Time



Analysis - Instant Value Loss



Results of Analysis

Best solutions:

- Execution Time: **K=11, P=7, PAA=3**
- Instant Value Loss: **K=9, P=6, PAA=3**

As almost always, the best case is about halfway between the two metrics and looking carefully at the two graphs and comparing them we can see that in this case:

K=10, P=7, PAA=3

There is the best compromise for this test run.

Thanks

View full project [here](#).

-
- Alex Di Stefano (S4638131)
 - Christian Stingone (S4644983)