(k,P)-Anonymity

KAPRA Algorithm Implementation

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

Contents

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.

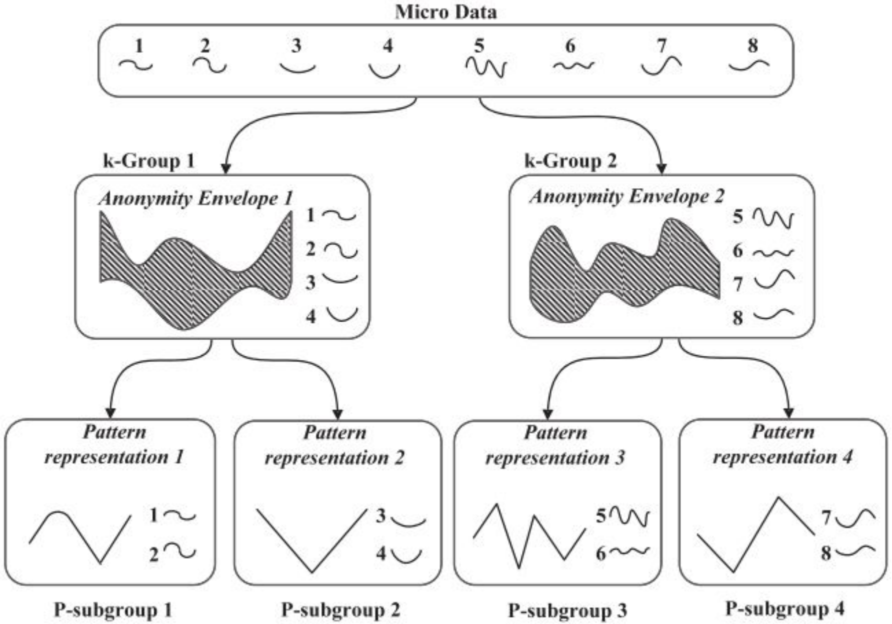
P-requirement: Consider any k-group G of time series having the identical anonymization envelope, for any time series (r) in G, there are at least P-1 other time series in G having the same QI pattern representation as PR[r]

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

KAPRA Algorithm

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 4, APRIL 2013 877

Supporting Pattern-Preserving Anonymization for Time-Series Data

Lidan 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. This model publishes both the attribute values and the patterns of time series in separate data forms. We demonstrate that our model can prevent linkage attacks on 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 certain part of the values but a different part of the pattern of the anonymized time series to be published simultaneously. We present estimation 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, RFIDs, 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 partner matching. However, such massive data also imply vast amount of privacy, which, if not appropriately protected, may become exploited as a source for abuses 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* $\in$ [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 r from data set where* $similarity(r, q) > threshold$ (or $distance(r, q) < \delta$).

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 (QI) to launch linkage attack which reidentify some of the records (time series). For instance, an adversary may learn from the external sources that the monthly sale of the victim happens to be between 1 and 1.2 million in December, among the very few who can achieve that figure. Then, (s)he could issue the above range query on the published table 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], [20] 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: {should, xshang, chenk, cg, chaozhang}@zju.edu.cn.

Manuscript received 7 Mar. 2011; revised 17 Nov. 2011; accepted 19 Nov. 2011; published online 5 Dec. 2011.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-03-0110.
Digital Object Identifier no. 10.1109/TKDE.2011.249.

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

Executions

Dataset 1

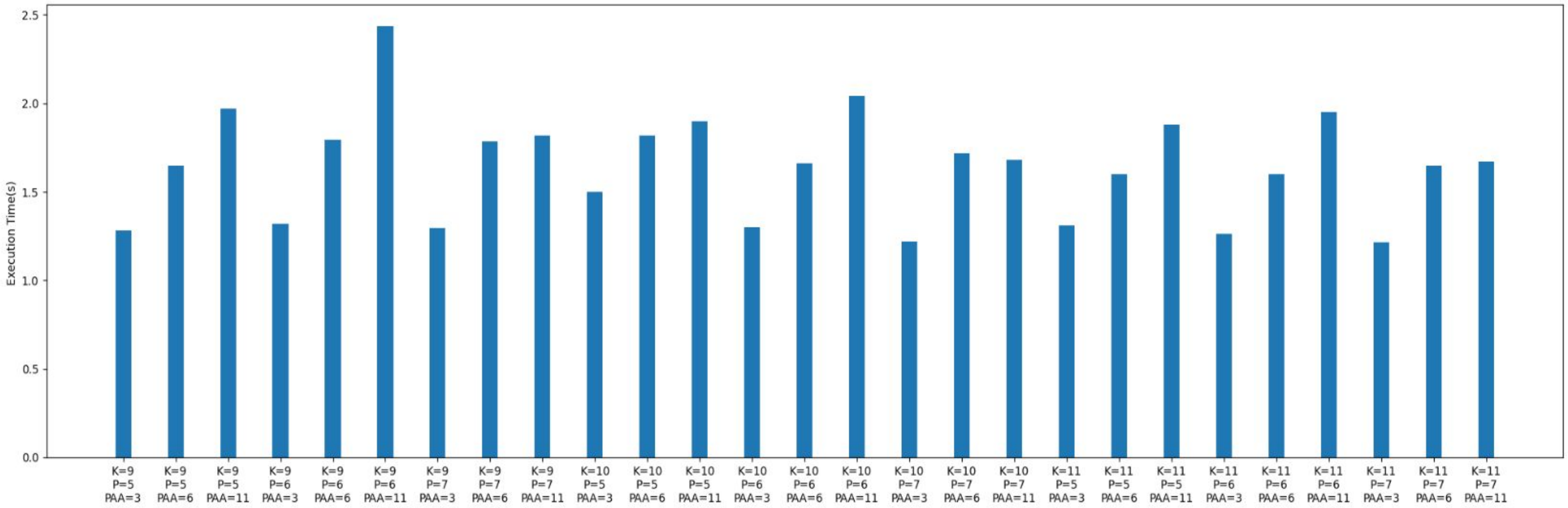
StackOverflow Questions Count Time Series: consist of count of various questions of specific libraries for each month.
Used for time analysis and Instant Value Loss analysis.

MLTollsStackOverflow:

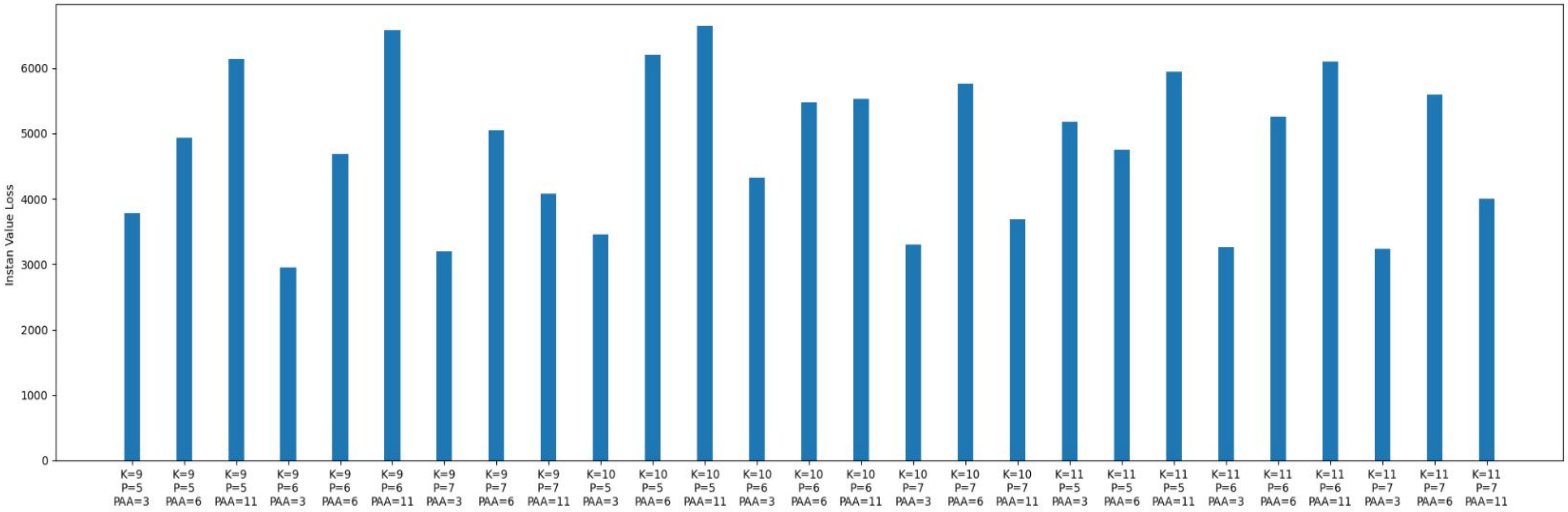
- 82 columns
- 133 rows

https://www.kaggle.com/datasets/aishu200023/stackindex

Analysis (1) - Execution Time



Analysis (1) - Instant Value Loss



Results of Analysis (1)

Best solutions:

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

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.

Dataset 2

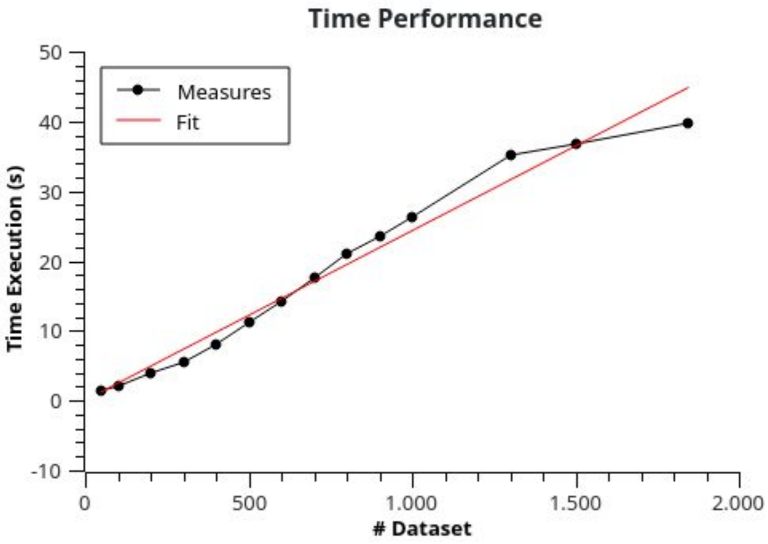
Nifty 50 Index Minute data (2015 to 2022): The dataset contains OHLC (Open, High, Low, and Close) prices of daily data from Jan 2015 to Jan 2022.
Used for time analysis and Instant Value Loss analysis.

NIFTY50-1_day_with_indicators:

- 60 columns
- 1845 rows

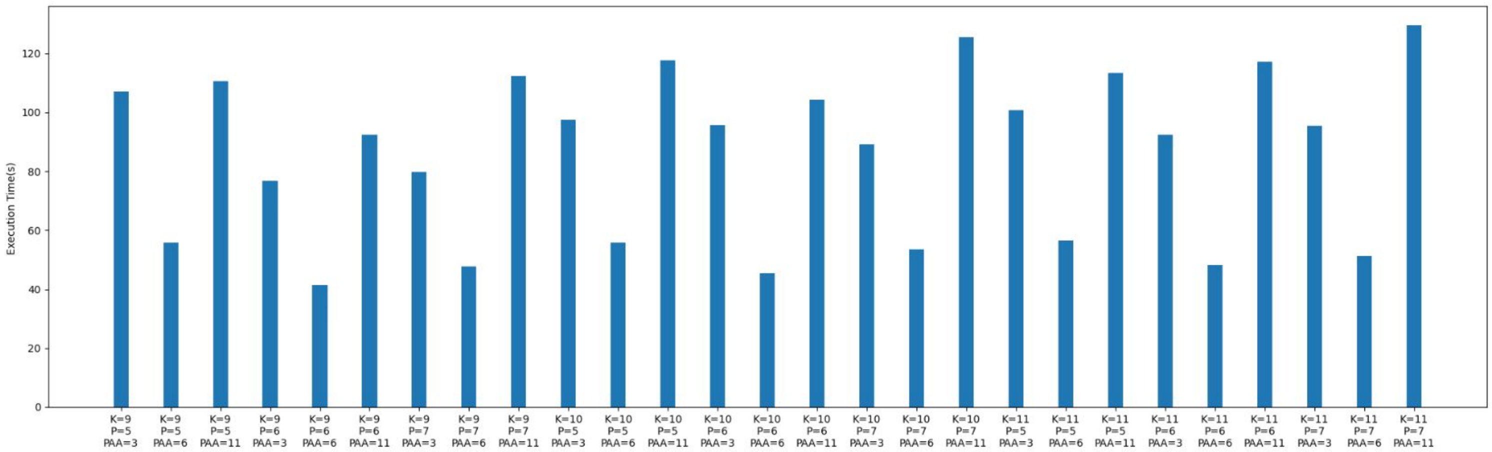
https://www.kaggle.com/datasets/debashis74017/nifty-50-minute-data

Analysis (2) - Time Performance

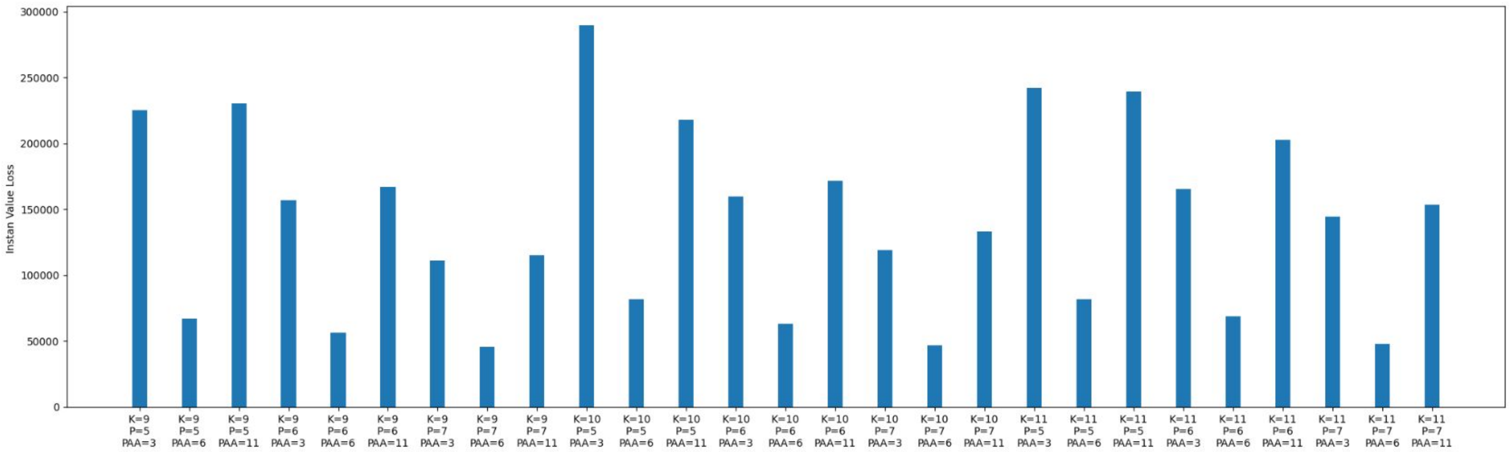


k = 9
P = 7
PAA = 6

Analysis (2) - Execution Time



Analysis (2) - Instant Value Loss



Results of Analysis (2)

Best solutions:

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

The best case still remains one of these two since over several runs these are the values that appear several times as the best solutions.

Statistically these parameters can give a best case for one of the two metrics or both.

Limits in parameter tuning

It can be seen that (k,P)-anonymity is actually a generalization of k-anonymity.

- **If P = 1 and remove all PRs from the dataset**:
  Results will be same as conventional k-anonymity based
- **If P == k**:
  Results will be an enhanced version of conventional k-anonymity based on pattern similarity

In general, P must be no greater than k

Thanks

View full project here.

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