CS8803-MDS

Project Proposal

Lecture 9
09/21/22

Presentation guidelines

Total time: 5min per group (timed)

What to cover

- A short introduction/problem slide
- Your bit-flip slide
- A solution/plan slide

We will have time for at most 1 question after each presentation.

Project Group 1

Tool for Ad-hoc Video Analysis

Ashmita Raju, Gaurav Kakkar, Myna Kalluraya

Problem

- Video database management systems (VDBMSs) are gaining popularity due to the large amount of information which can be extracted from video data.

- Current systems which run queries on VDBMSs require resource expensive pre-training of models.

- They also do not have means to take in users' feedback.

Our Solution

- We propose a novel visual interface to enable compositional queries using off the-shelf computer vision models.

- The ad-hoc task can be broken into a collection of modular subtasks across a sequence of frames.

- User can provide domain-specific hints to accelerate the task.

- Compare accuracy with existing action detection models.

Project Group 2

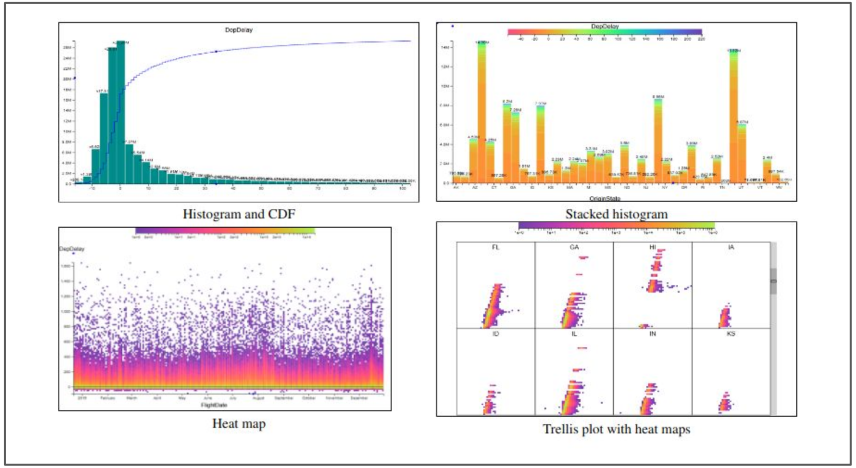
QuickScatter

An Interactive Tool For Finding Correlations in Large Datasets

Eric Martin and Akshay Iyer

Problem To Address (Group 2)

- Lack of interactive visualization tools to explore large datasets.
- Current research focuses on generalized views such as histograms, heat maps, etc...
- Scatter plots ignored due to over plotting.
- Scatter plots valuable for discovering correlations.

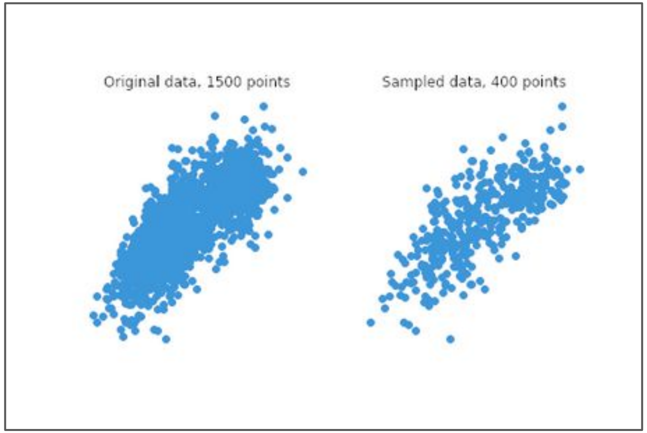


Histogram and CDF

Stacked histogram

Heat map

Trellis plot with heat maps

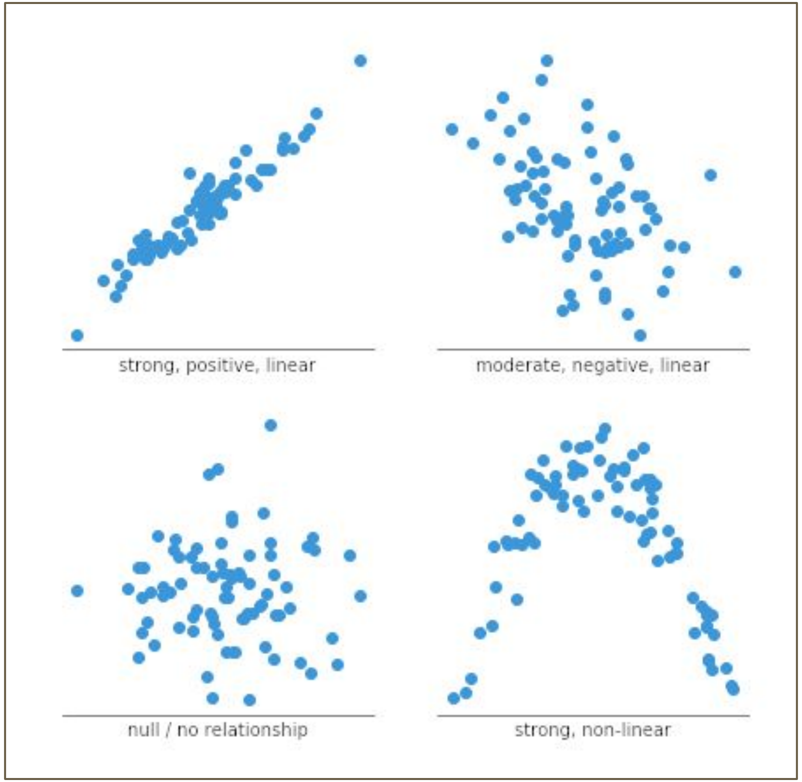


Bit Flip (Group 2)

- Bit:
  - Scatter Plots are not suitable for large datasets due to overplotting. Generalized views should be used instead.
- Flip:
  - Scatter plots can be used with large datasets if sampling methods are applied. Scatter plots provide intuitive views useful for exploratory analysis.

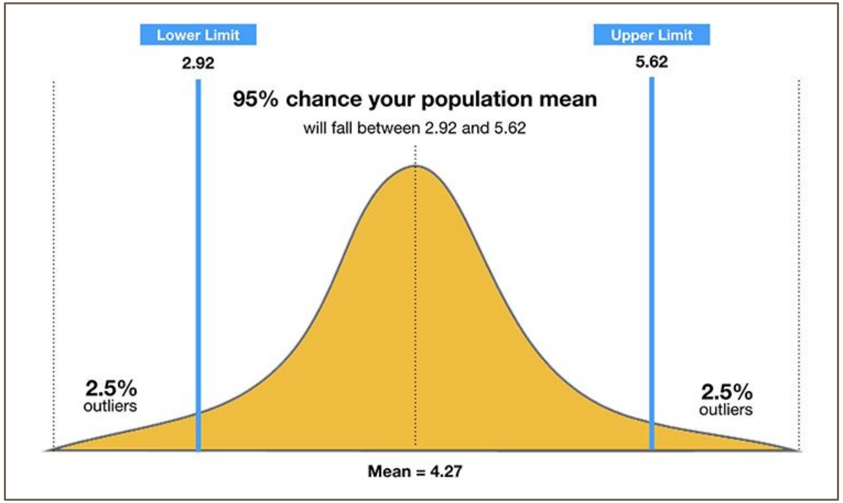
Proposed Solution (Group 2)

- Generate multiple scatter plots with sampled subsets.
- Configure sample sizes appropriate for plot dimensions.
- **Key Assumption**: If there is a correlation, sample views will unveil it.



Plan Of Attack (Group 2)

- Try some different sampling techniques on a smaller video game dataset.
- Observe any found correlations, run statistical tests to determine sample accuracy.
- Apply sampling/preprocessing techniques only larger census dataset, observe results, and display insights back to the user.



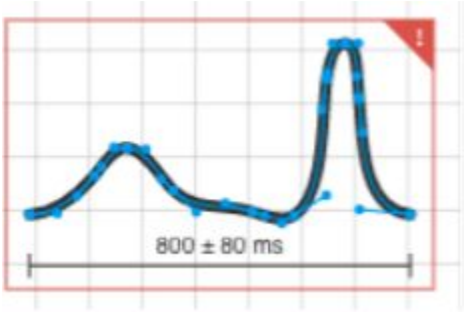
Project Group 3

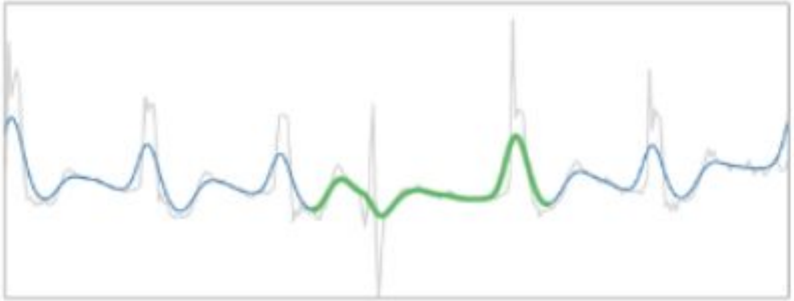
Expressive time series queries using human sketches using complex pattern matching techniques

Vishnu Krishnan, Haotian Sun

Problem

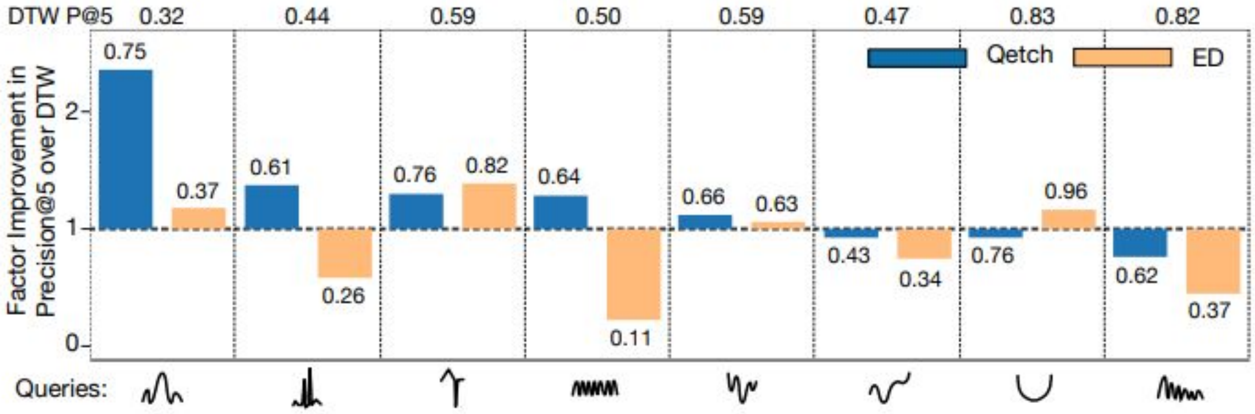
- One of the most promising interactive approaches is matching patterns from the original time series based on users’ hand-drawn sketches.
- Existing implementations utilize only distance metrics for their pattern matching methods such as Dynamic time warping, Euclidean distance and their own original distance metrics.





Bit-Flip

- We propose the use of algorithmic improvements to ensure better performance in time series matching.
- This will be carried out through the usage of more complex matching methods that we will identify and compare it to the existing DTW and ED.



Challenges

- Novel matching algorithms with more advanced distance metrics usually requires additional data information or matching stages.
  - E.g., ML-based matching algorithms need large amount of data and computations for training
  - Stochastic search is sometimes involved, introducing randomness.
- Interfacing the new matching algorithms with the existing querying system.
  - Data preprocessing, data format, etc.

Solution

- First we shall implement complex matching algorithmic techniques
    - New distance metrics for time-series data, learning-based matching algorithms, etc.
- Then we validate the novel algorithms on different time-series datasets
    - 3W datasets, UC Riverside time-series datasets, etc.
- Finally the plan is to evaluate it using statistical metrics
    - Precision, accuracy, recall, F1 score, etc.

Project Group 4

Towards a Non Deep Learning Approach to Explore the Line Charts Similarity Search Problem

Harshal Gajjar, Sahil Ranadive

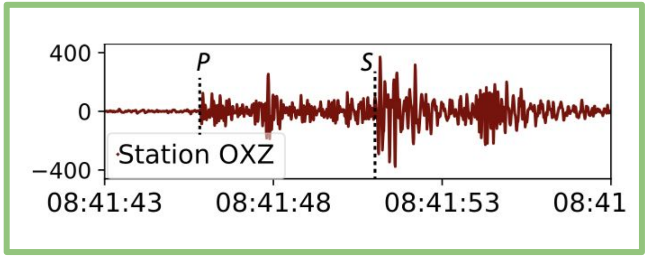
Problem

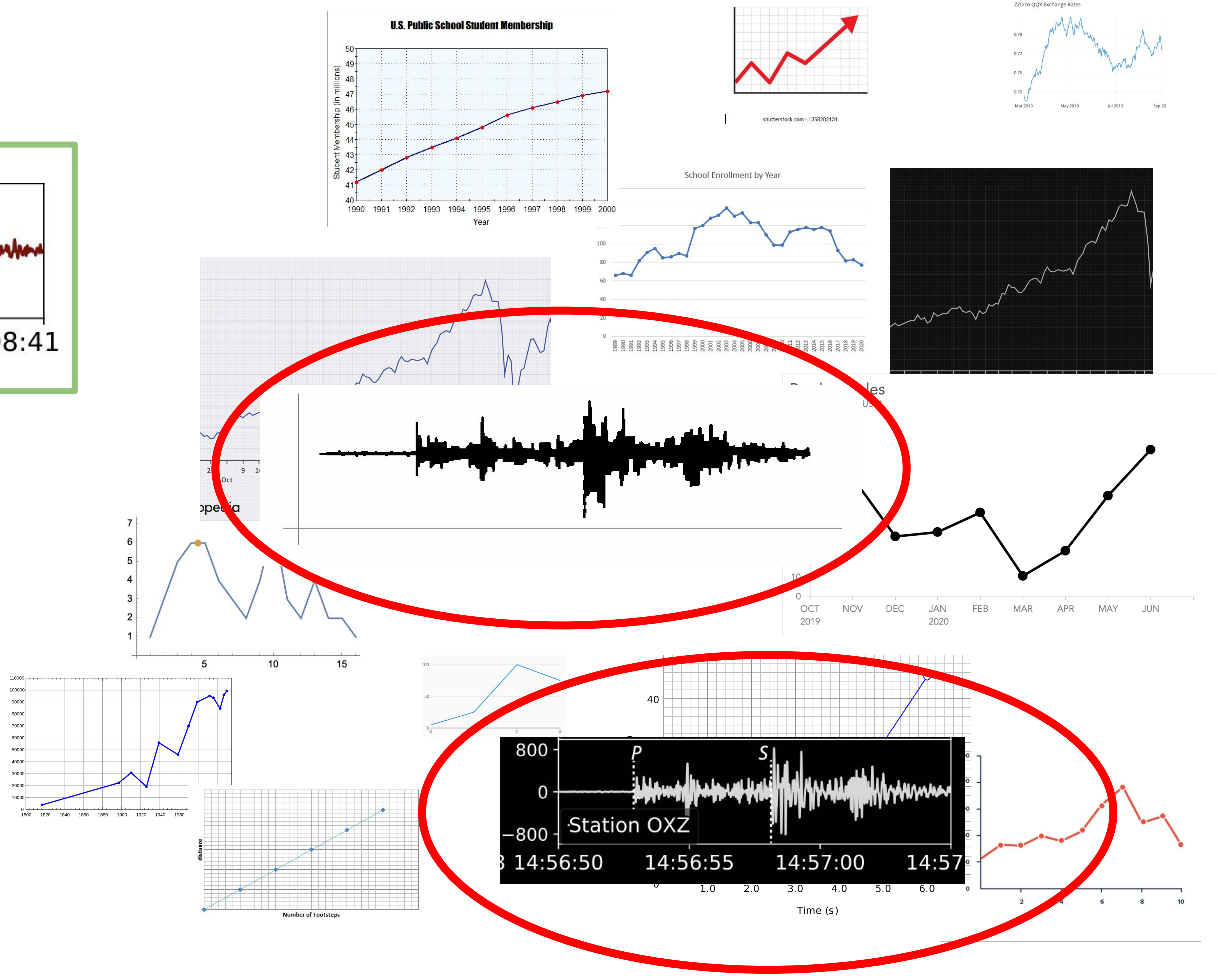
Given a line chart image L and a set S of line chart images, line charts similarity search problem aims to retrieve a subset of S where each line chart is similar to L; where similarity means that the underlying data used to construct the graph is similar (for instance scaled by a factor of 2)

What we want to do

In this project, we introduce a new image-level algorithm (with no access to underlying data) which is robust against changes in visual elements that do not represent change in the underlying data, for instance gridlines, legends, axes, plot curve style, etc.

Summary

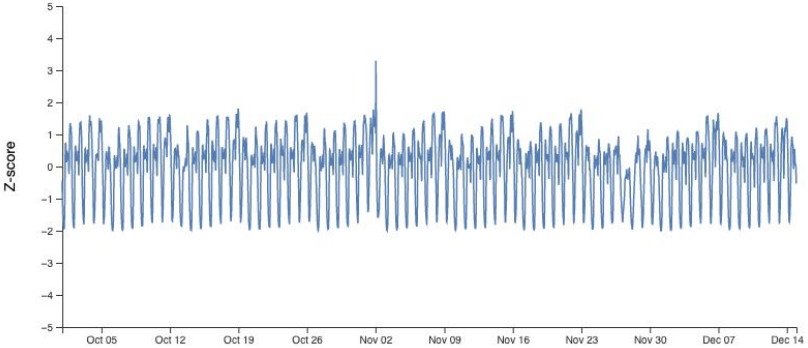

400
0
−400
P
S
Station OXZ
08:41:43
08:41:48
08:41:53
08:41

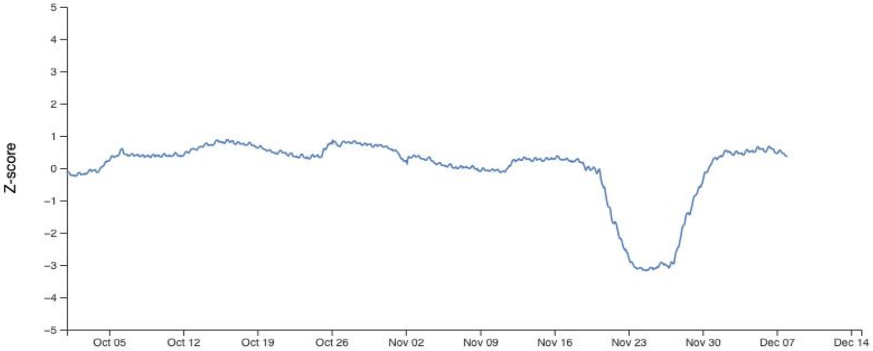

U.S. Public School Student Membership
Student Membership (in millions)
Year
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
shutterstock.com · 1358202131
ZZD to QQY Exchange Rates
School Enrollment by Year
OCT 2019
NOV
DEC
JAN 2020
FEB
MAR
APR
MAY
JUN
800
0
−800
P
S
Station OXZ
14:56:50
14:56:55
14:57:00
14:57
Time (s)
Number of Footsteps
distance

Project Group 5
A Generalized Framework for Time Series Smoothing

Tanya Garg, Siddhi Pandare, Sankalp Sangle

Problem





-> Today’s visualization systems plot noisy raw data, obscuring long-term trends

-> Tough to immediately see that a dip occured at the end

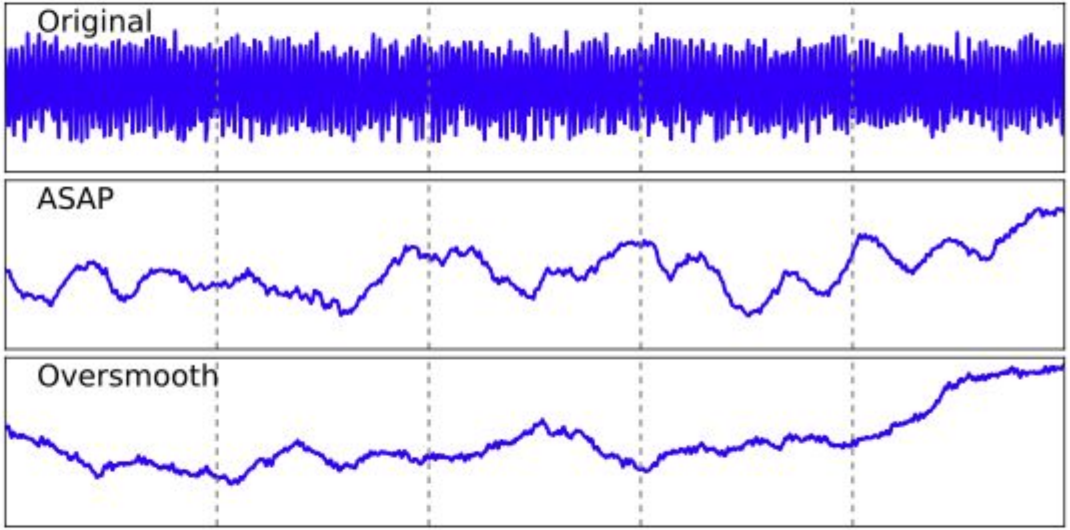
-> Smoothing can help visually highlight these dips or anomalous events

Bit Flip

-> One paper (ASAP) uses a smoothing function (SMA) to remove noise and a statistical measure (Kurtosis) to preserve structure and prevent oversmoothing

-> Temperature dataset user study in ASAP paper showed : no single smoothing function + statistical measure is enough to effectively smooth all types of time series

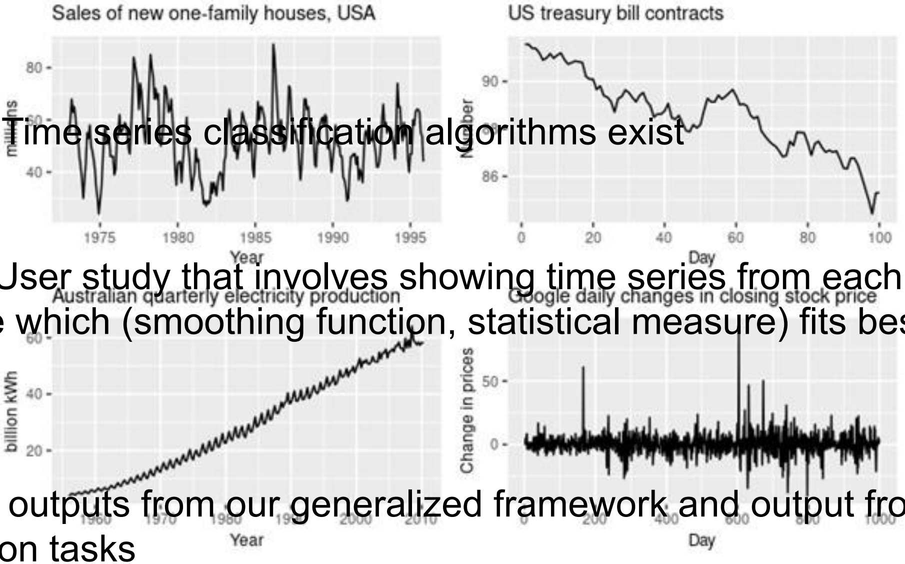
-> Flip: We wis g Time Series data



ASAP: Prioritizing Attention via
Time Series Smoothing
Rong et al.

Solution

- 2-fold problem, given a time series input
  - How to identify time series category
  - Given an identified category, which (smoothing function, statistical measure) works best?



- Solving problem 1 - Time series classification algorithms exist

- Solving problem 2 - User study that involves showing time series from each category to users and making them choose which (smoothing function, statistical measure) fits best

Evaluation: show users outputs from our generalized framework and output from ASAP paper, and compare effectiveness on tasks

Project Group 6

Context-aware SQL Auto-Completion with Reinforcement Learning

Cangdi Li, Yiheng Mao, Ting Yu
Group 6

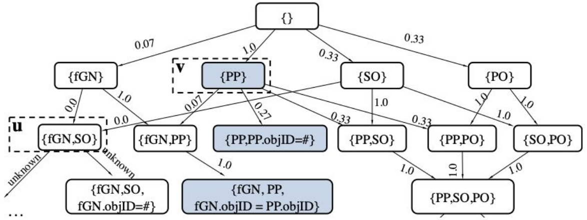
Problem Motivation

- Increasingly more data are stored in DBMS
- Writing effective SQL is difficult for inexperienced analysts
- Even experienced gurus may have a hard time when working on a new database with hundreds of table schemas

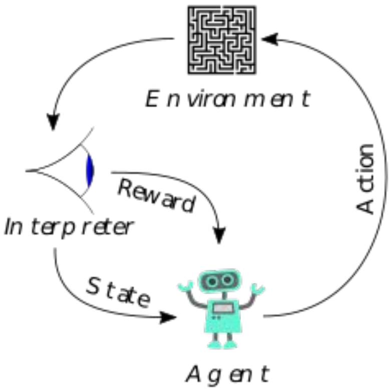
That's where SQL query auto-completion comes in. It runs like a Co-pilot specialized for SQL for data analysts!

Existing Solutions

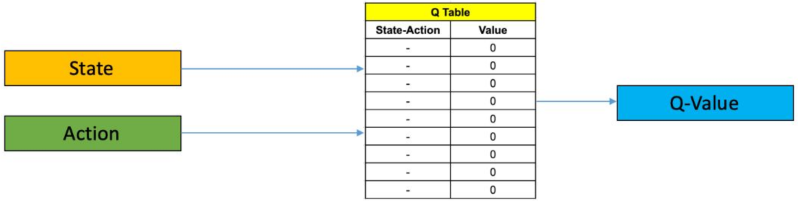
- Existing auto-completion system, SnipSuggest, suggests next query content by using a **Direct Acyclic Graph (DAG)** with conditional probabilities as edge weights built from past queries. But it does **NOT leverage the sequence of user queries** from the current user session.



- On the other hand, ML models that learn from query sequence predict the next query in whole (Meduti et al.), which is a different use case from query auto-completion where the system **constantly responds to a user's interactive typing** (thus the query already has a prefix).



Bit-flip



Q Table	
State-Action	Value
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0

Q Learning

- Given the bit, currently we have **sequential ML models** that can predict the **whole query,** but we lack an **ongoing query auto-completion** intelligence application **with user input prefix**.
- Thus, we propose an idea of using the query prediction result from **Q-Learning** to build a **Directed Acyclic Graph** in a way that each node is a context of the query, like “SELECT”, “WHERE”, attributes, table_names, and the edges are connection that stores the probability.

This would allow the algorithm to provide context prediction throughout the user input duration.

Instantiate Bit-flip solution

- **Why Q-Learning?** We found that a recent evaluation paper demonstrate that Q-Learning is the best performing algorithm on test F1-scores, cumulative train and test latencies and memory consumption in predicting Query.
- **How to Constructing DAG?** Each node is a context, and each edge will be the weights stored in the Q-Learning matrix probability matrix.
- **STEPS:**
    - **Before** the user input, we simply recommend the prediction of the whole query by Q-Learning prediction.
    - **As the user input more context**, if it doesn't match with our entire prediction, we will use the prefix to traverse our DAG and give predictions. This allows providing intermediate predictions when there’s context and also giving recommendations when there’s no context.
    - **After** the completion of the Query, we will use the final query to re-train the Q-Learning model and add such a path in our DAG.

Evaluation

- We evaluate our prediction query against the actual query user executed using cosine similarity between the two queries (one-hot encoded to be vector form).
- Since we change predictions as the user types different results, we may also evaluate query predictions given different lengths of prefix. We may examine how our algorithm performs given different prefixes.

Implications

- Help novice users start **authoring SQL easily**.
- Help any user to **navigate through a complex database** with lighter cognitive overhead.
- If our predictions are good enough, **prefetching and precomputation** could be done, potentially saving query processing time.

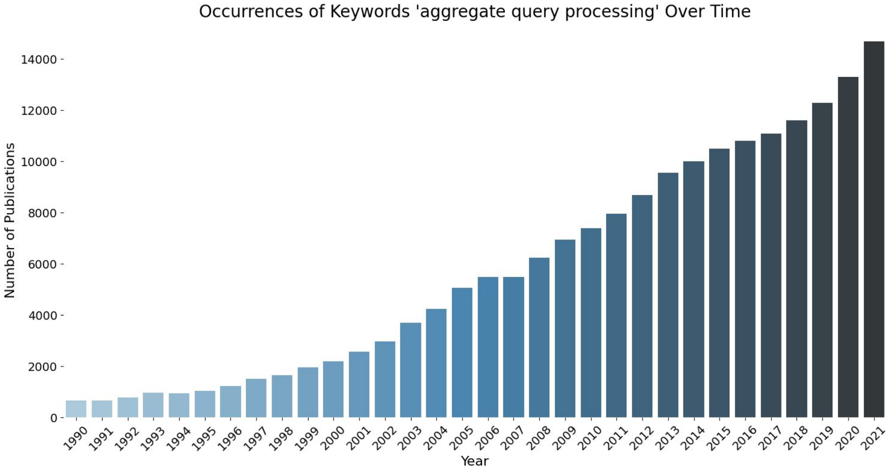
Project Group 7

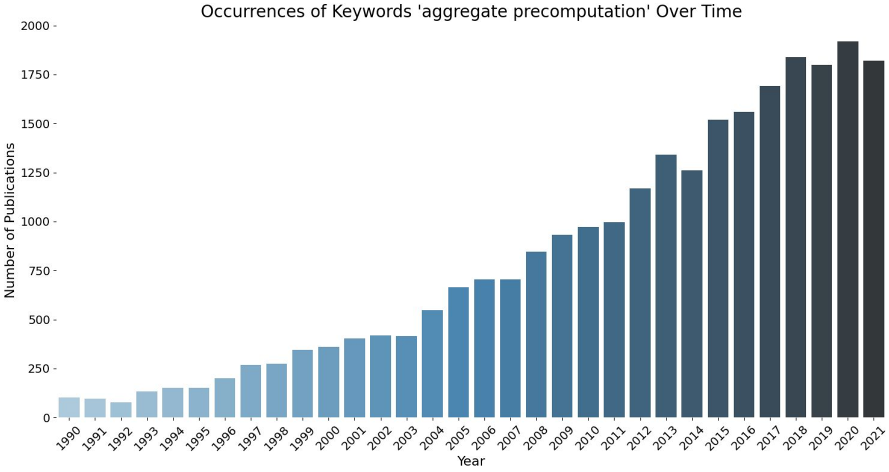
Adaptive Sampling and Aggregation Precomputation on Distributed Databases

Qiandong Tang, Yanhao Wang, Shen En Chen

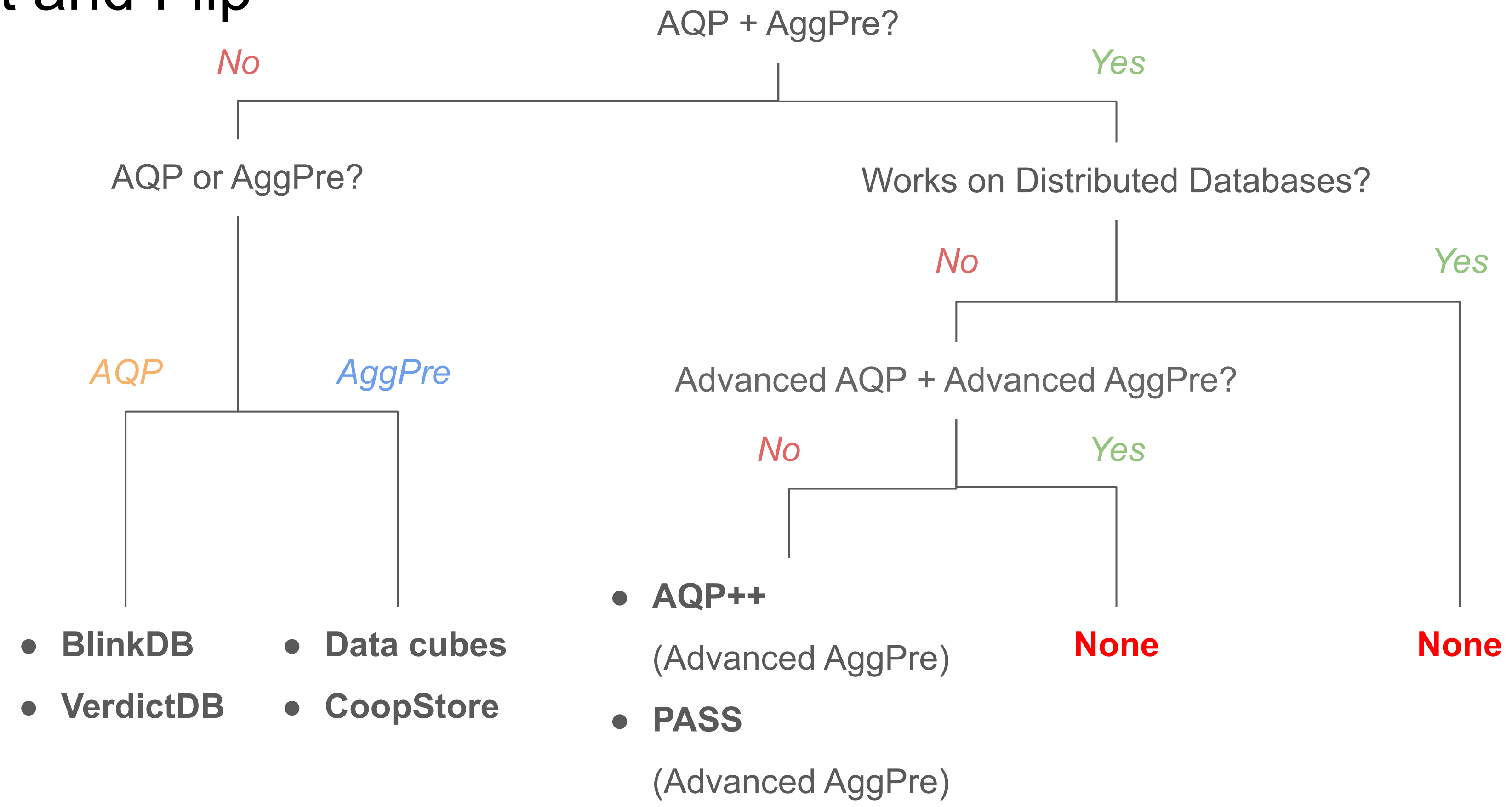
Problem

The exponentially growing volume of data has brought more attention to interactive analysis that can answer aggregation queries within interactive response times.





Bit and Flip



AQP + AggPre?

No

AQP or AggPre?

AQP

- **BlinkDB**
- **VerdictDB**

AggPre

- **Data cubes**
- **CoopStore**

Yes

Works on Distributed Databases?

No

Advanced AQP + Advanced AggPre?

No

- **AQP++**

  (Advanced AggPre)
- **PASS**

  (Advanced AggPre)

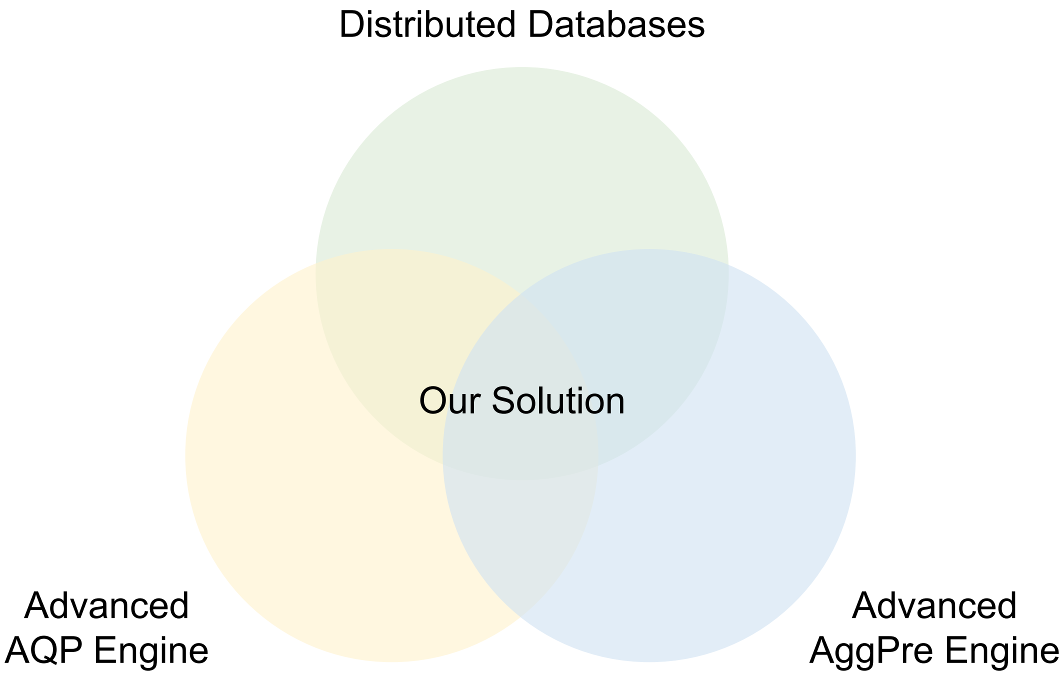
Yes

None

Yes

None

Solution



1. Connect AQP and AggPre in a distributed settings

2. Combine advanced AQP (e.g. multiple multi-dimensional stratified samples with dynamic selection) with advanced AggPre (e.g. partition tree)

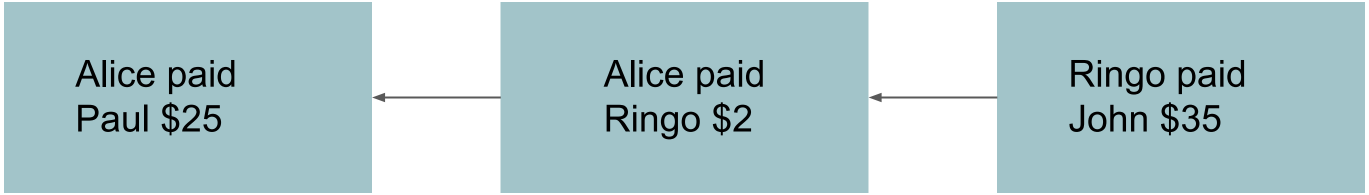
3. Ablation analysis measured by mean response errors and response time. Also analyze on different storage allocation.

Project Group 8

Querying blockchain data

Aniruddha and Abhi

Querying blockchains is hard

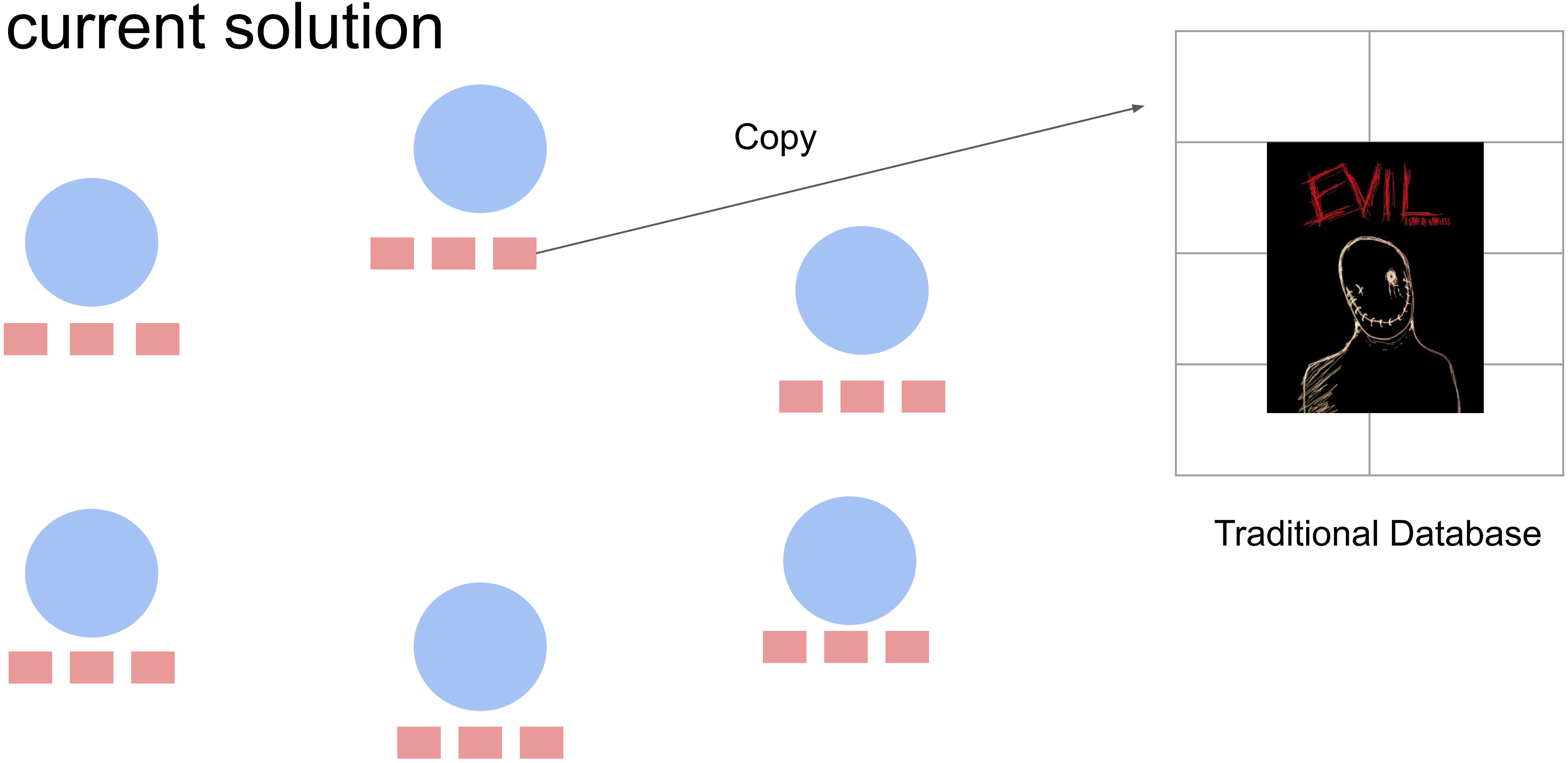


Alice paid Paul $25

Alice paid Ringo $2

Ringo paid John $35

The current solution



Copy

Traditional Database

The bitflip (or two)

1) Do we need to copy the blockchain onto a separate central database? What if we changed the way blocks were structured to make answering queries easier?
2) Do we need to verify if the results of the central database are valid? What if this can be done implicitly?

Project Group 9

A "free" add-on for Q2D (query-to-data)distance estimation from nearest neighbor indexes

Jingfan Meng

Q2D distance estimation·

- Imagine a database that holds one billion 1024-dimensional vectors.
- Given a query point, we want to know whether it is an outlier or not.
- One criterion is to look at the Q2D (query-to-data) distances to 1B points.
- But would it be too expensive to compute them all?

Ad-hoc index vs nearest-neighbor index

- Existing solutions such as HBS (http://proceedings.mlr.press/v97/siminelakis19a) creates an ad-hoc index that samples and computes Q2D distances on a subset of data.
- The flip is: Do we actually need an ad-hoc index?
- We can actually reuse the widely used nearest-neighbor indexes.
- The benefit is that we can grasp Q2D distances (of both near and far distances) and get nearest neighbors at the same time.

Solution and Plan

- Since we use a borrowed index, we cannot control the sampling rate,but fortunately, we discover that the sampling probability from nearest-neighbor indexes has a similar shape to ad-hoc indexes.
- Furthermore, we can calculate the sampling rates by a Monte Carlo simulation.
- We plan to evaluate our solution on both accuracies (Q2D histogram and kernel densities), index size, and computational overheads over existing solutions.

Project Group 10

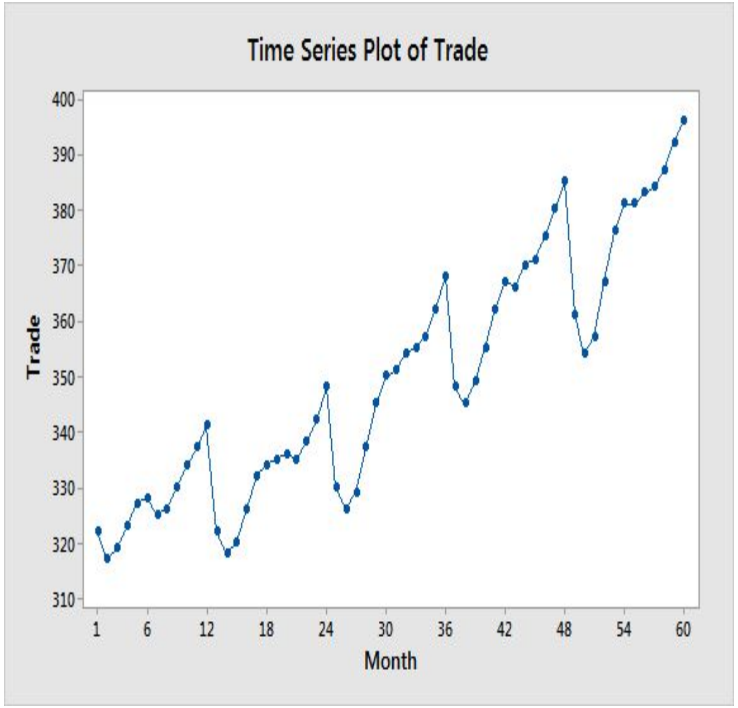
EVALUATION OF DATA MODELS FOR QUERY EFFICIENCY IN TIME SERIES

- SHUBHAM AGARWAL AND HAMSIKA RAMMOHAN

Introduction

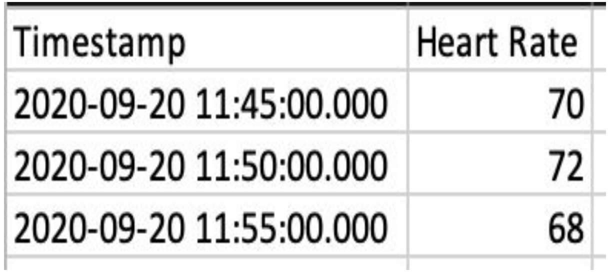
- What is Time Series Database: A software system that is optimized for storing and serving time series through associated time and value pairs.

- Use Cases: Financial Market Trends, Anomaly Detection, Weather Forecasting, Health Data

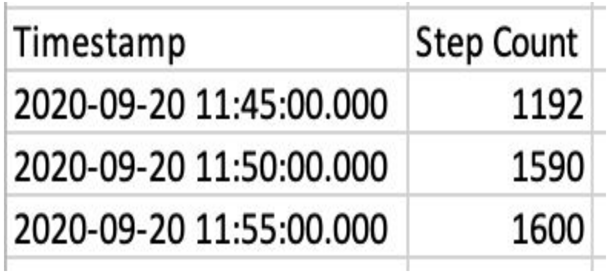


What is missing?

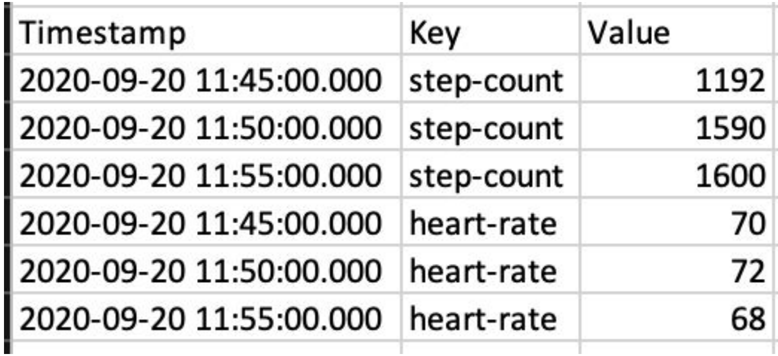
- There hasn’t been much empirical research on the type of data models (schemas) that is most efficient to store the data.



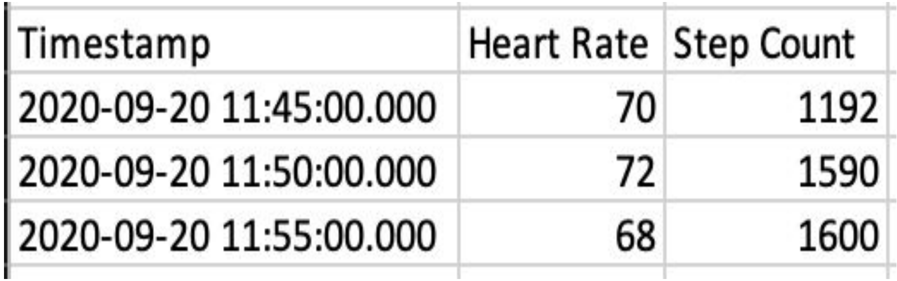
Timestamp	Heart Rate
2020-09-20 11:45:00.000	70
2020-09-20 11:50:00.000	72
2020-09-20 11:55:00.000	68



Timestamp	Step Count
2020-09-20 11:45:00.000	1192
2020-09-20 11:50:00.000	1590
2020-09-20 11:55:00.000	1600



Timestamp	Key	Value
2020-09-20 11:45:00.000	step-count	1192
2020-09-20 11:50:00.000	step-count	1590
2020-09-20 11:55:00.000	step-count	1600
2020-09-20 11:45:00.000	heart-rate	70
2020-09-20 11:50:00.000	heart-rate	72
2020-09-20 11:55:00.000	heart-rate	68



Timestamp	Heart Rate	Step Count
2020-09-20 11:45:00.000	70	1192
2020-09-20 11:50:00.000	72	1590
2020-09-20 11:55:00.000	68	1600

What is missing?(2)

- All organizations have different requirements and resources.

- Time series databases have built-in functions to prepare data for visualizations - uses compute power

- Alternative? - could transfer data to a new system and computation can happen on raw data - more flexibility, high data transfer cost

What we propose?

1. Evaluating existing schemas of the Time Series database to see what works best for Financial Data- **First of a kind!**

2. Compare and evaluate the performance of Java/Python +RDBMS v/s Java/Python + NoSQL v/s time series database for preparing data to present it in the following forms:
    a. Line Chart
    b. Histogram
    c. Pie Chart

Project Group 11

Evaluation of Natural Language Generation Techniques for Augmenting Multi-Class Cyberbullying Datasets

Bojun Yang

Problem

- Cyberbullying removal is important and a novel issue, but no current platform uses effective toxic comment identification
- Very little categorized cyberbullying datasets
- Takes time and money to collect categorized cyberbullying data

Natural Language Generation

- Usually used for tasks like text summarization, translation, dialog response generation, long text generation, and captioning
- RNNs, CNNs, GANs, Transformer based techniques

Project Group 12

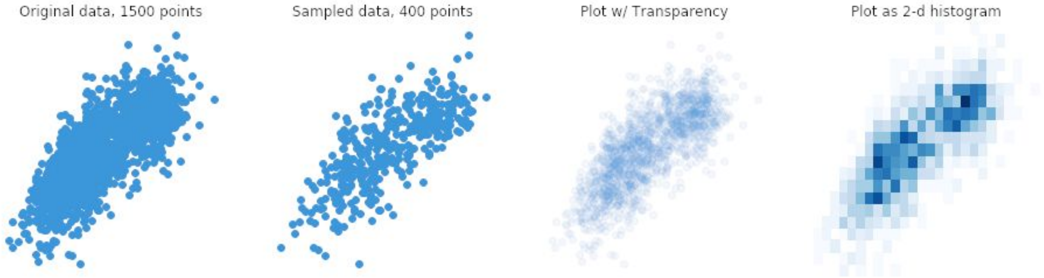
Evaluation of Scatterplot Sampling Techniques for Exploratory Trend Analysis of Massive 2D Datasets

Cuong (Johnny) Nguyen, Andrew Zhao

Problem

- Trend analysis is one of the most important scatterplot EDA tasks.
- For extremely large datasets, scatterplots will take significantly more time and computational resources to render

=> Sampling techniques are necessary to find trends for such cases, but existing literature doesn’t tell us which is most effective from the data scientist perspective



Original data, 1500 points

Sampled data, 400 points

Plot w/ Transparency

Plot as 2-d histogram

Bit

- Previous works have examined the effectiveness of sampling techniques in scatterplot EDA tasks for outlier identification, density detection, shape examination, but not trend analysis.

Flip

- In this paper, we propose a user study in order to test the effectiveness of various sampling techniques in relationship to human performance on the task of scatterplot trend analysis

Our proposed solution

- We will create/find large datasets (2D) in the following categories: Have an obvious interesting trend, have a hard-to-spot trend, don’t have a trend. We will apply SOTA sampling techniques for data exploration onto these datasets with different levels of samples, asking users if they can find a trend.

- We will visualize 2d scatterplots on massive datasets (generated with scikit-learn or some other appropriate library) with a particular trend (or not) and downsampled using our investigated sampling techniques. Then we will ask the users, given the visualization, whether there is a trend in the dataset or not. We will measure user’s task performance on metrics of time and accuracy.