CS8803-MDS

Final Presentation

12/05/22

Group 7

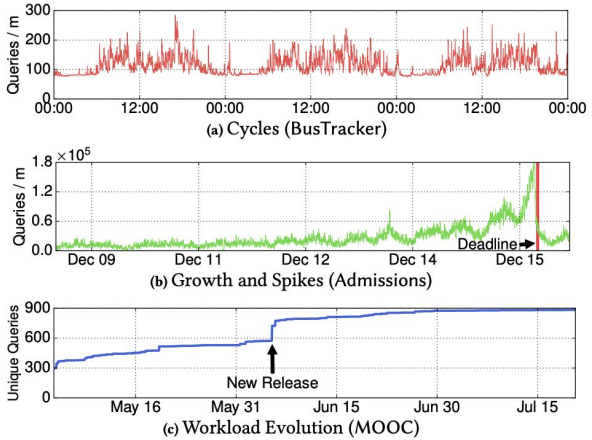
Workload-Aware Adaptive Sample Update for AQP

Qiandong Tang, Yanhao Wang, Shen En Chen

Problem

Query workload for AQP can change over time:

1. **Interactive and exploratory data analysis** where users only pay attention to the broad trends or anomalies
2. **Visualizations** that only require granularities up to screen resolution and human perceivable details
3. **Major events in production/workflow**:
   a. Admission deadlines
   b. New feature release



[1] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J. Gordon. 2018. Query-based Workload Forecasting for Self-Driving Database Management Systems. In Proceedings of the 2018 International Conference on Management of Data, ACM, Houston TX USA, 631–645. DOI:https://doi.org/10.1145/3183713.3196908

Bit: Prior Work

Workload-Aware Databases

- **Continuous On-Line Tuning (COLT)**
  monitors and analyzes the workload of a database system by collecting statistics from a DBMS [1]
- **QueryBot 5000**
  predicts the expected arrival rate of queries in the future based on historical workload [2]

AQP Robust to Workload Shifts

- **ML-AQP**
  estimates the result of new queries in efficiently and inexpensively by training ML models on vectorized SQL queries and adapt to workload shifts through re-training [3]
- **PASS**
  combines AQP and AggPre and considers the workload shift edge-case scenarios during evaluation [4]

[1] Karl Schnaitter, Serge Abiteboul, Tova Milo, and Neoklis Polyzotis. 2007. On-Line Index Selection for Shifting Workloads. In 2007 IEEE 23rd International Conference on Data Engineering Workshop, 459–468. DOI:https://doi.org/10.1109/ICDEW.2007.4401029

[2] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J. Gordon. 2018. Query-based Workload Forecasting for Self-Driving Database Management Systems. In Proceedings of the 2018 International Conference on Management of Data, ACM, Houston TX USA, 631–645. DOI:https://doi.org/10.1145/3183713.3196908

[3] Fotis Savva, Christos Anagnostopoulos, and Peter Triantafillou. 2020. ML-AQP: Query-Driven Approximate Query Processing based on Machine Learning. DOI:https://doi.org/10.48550/arXiv.2003.06613

[4] Xi Liang, Stavros Sintos, Zechao Shang, and Sanjay Krishnan. 2021. Combining Aggregation and Sampling (Nearly) Optimally for Approximate Query Processing. DOI:https://doi.org/10.48550/arXiv.2103.15994

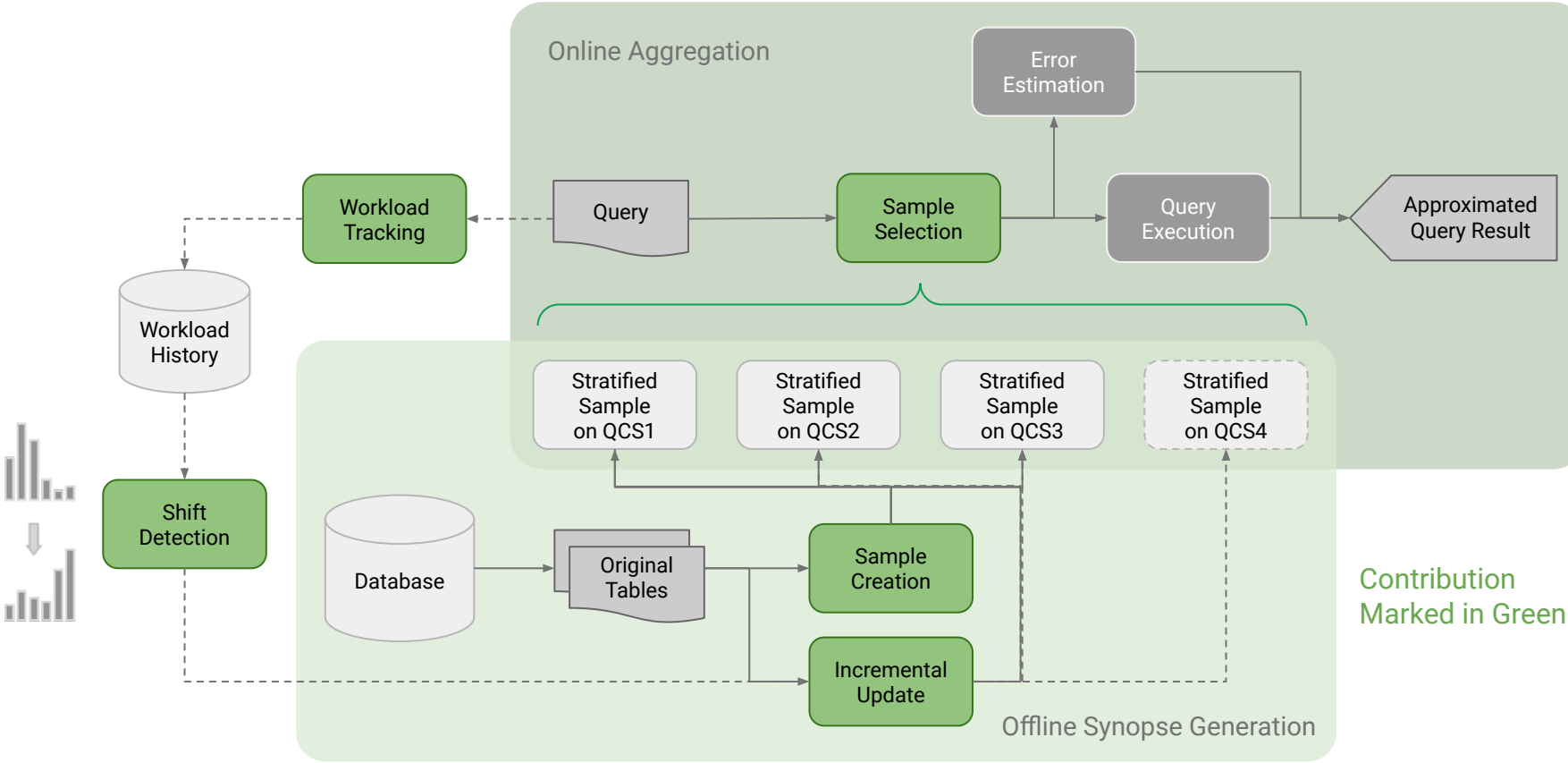
Our Flip: Workload-Aware Adaptive Sample Update

More flexible sample creation, update, and selection:

1. **Flexible sizes of stratified samples** on QCS's using linear programming
2. **Sample selection** based on coverage and frequency
3. **Tracking workload** by QCS frequencies
4. **Detecting workload shifts** by weighted Jaccard similarity
5. **Incremental sample update** with sample key indexing

And a more systematic **workload shift benchmark**.

System Overview



Sample Creation

Maximize the coverage of our samples against incoming queries, with the following constraints:

- Query frequency in the workload
- Sample QCS coverage against incoming queries
- Storage costs

Sample Creation

Maximize the coverage of our samples against incoming queries:

$$G = \sum_j p_j \cdot \max_i c_{ij} \cdot \sqrt{\alpha_i \cdot z_i}$$

constrained to

$$\sum_{i=1}^{m} S(\phi_i) \cdot \alpha_i \cdot z_i \leq \mathbb{C}$$

- i indexes over all m possible sample QCS ϕ_i
- j indexes over all incoming queries q_j
- G = Maximization objective
- p_j = Frequency of column set of query q_j
- C_{ij} = Coverage of QCS ϕ_i against query q_j
- $S(\phi_i)$ = Unit storage cost of QCS ϕ_i
- α = Size of minimum group for ϕ_i
- $\mathbb{C}$ = Total storage budget
- z_i = Sampling ratio of ϕ_i

→ **Solve with linear programming**

Sample Selection

At runtime, we **rank the samples** by two factors:

1. Sampling ratio
2. QCS coverage for the query

$$i = \arg\max_i r_{ij} = \arg\max_i \left[z_i \cdot c_{ij} \right]$$

- i indexes over all m possible sample QCS ϕ_i
- j indexes over all incoming queries q_j
- C_{ij} = Coverage of QCS ϕ_i against query q_j
- z_i = Sampling ratio of ϕ_i

Sample Update

We **parse the query with ANTLR 4** (ANother Tool for Language Recognition) to obtain non-alias column names in the following parts of the query

- WHERE
- GROUPBY
- HAVING

Given a QCS ϕ_i , instead of dropping the corresponding sample and resampling a sample of different size, we **append or drop only the minimum number of records** needed. This requires indexing of the records in the tuples by their keys.

Implementation and experiments in progress.

Workload Tracking and Shift Detection

We define the **similarity between workloads** of different time using **weighted Jaccard similarity**:

$$S_w = \frac{\sum_{k=1}^{m} \min(V_k, T_k)}{\sum_{k=1}^{m} \max(V_k, T_k)} \in [0, 1]$$

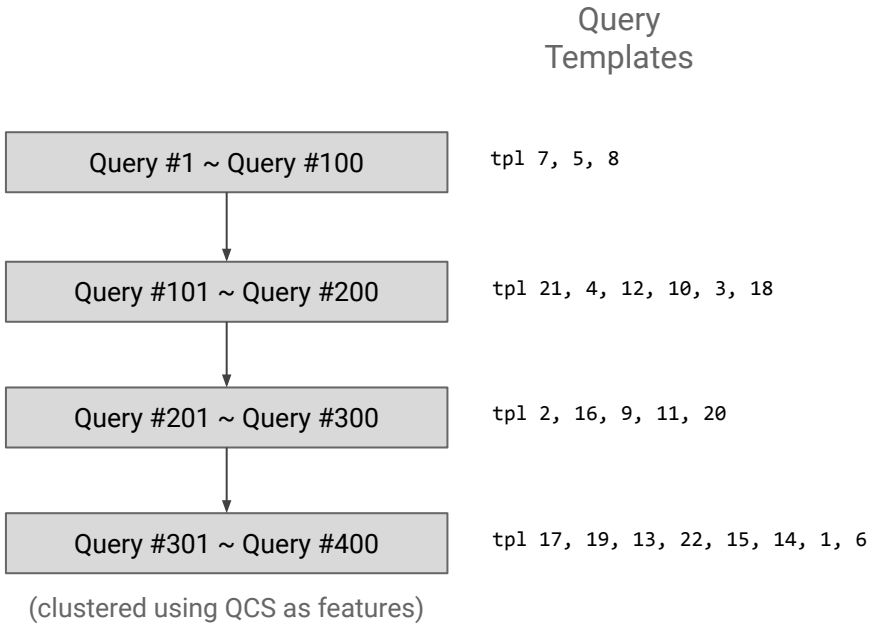
where $V, T \in \mathbb{R}^m$ are the sparse vector representation of the count of each unique column set. For example, V_k represents the count of the k-th unique possible column set of the data ordered in an arbitrary order.

We plan to identify an optimal similarity threshold a “workload shift” via experimentation. The system will update periodically or upon detecting the threshold being met.

Implementation and experiments in progress.

Workload Shift Design

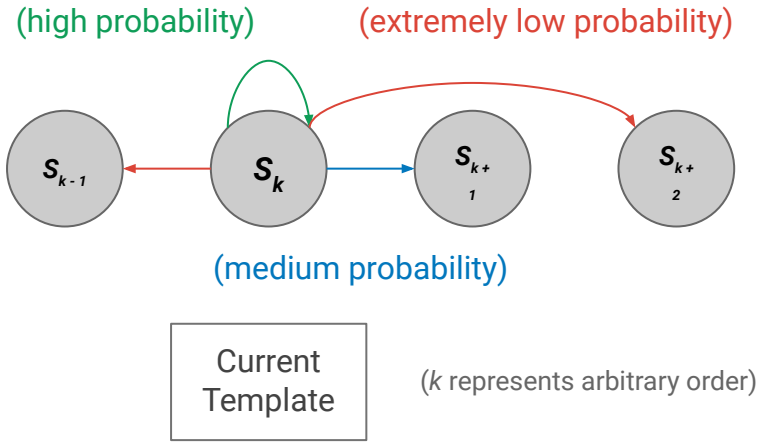
Hard Shift



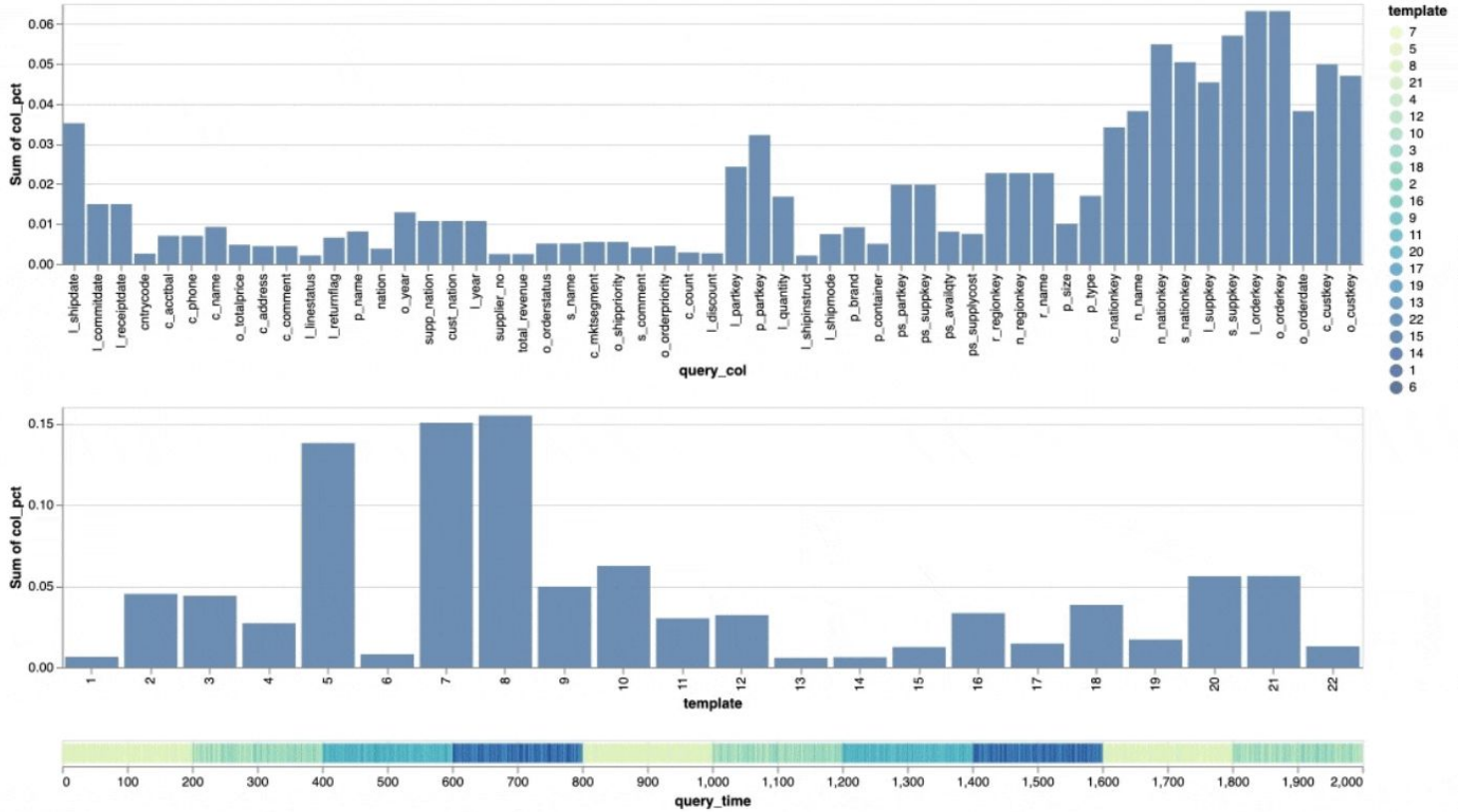
(clustered using QCS as features)

Gradual Shift

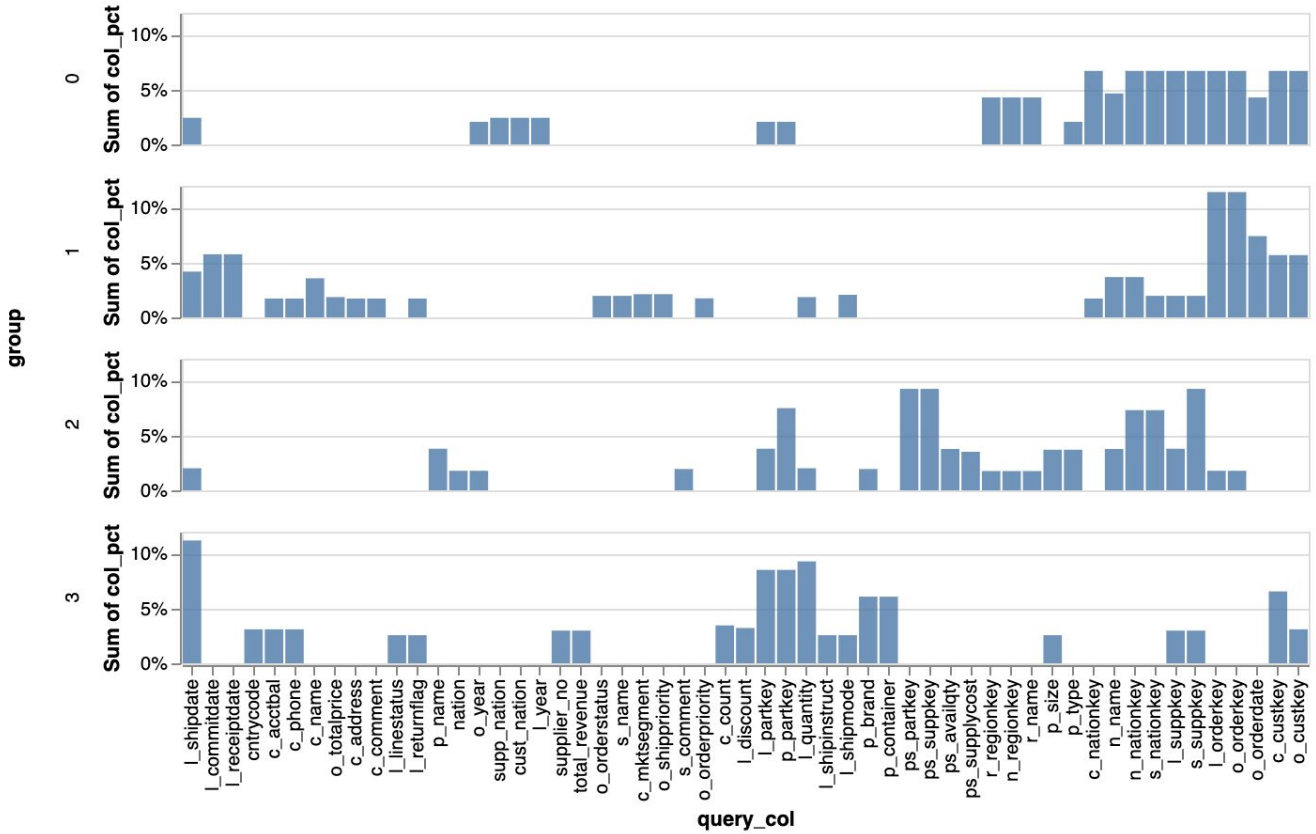
We model **query templates as states** and transitions between query templates as a **Markov Decision Process** (MDP) when generating queries:



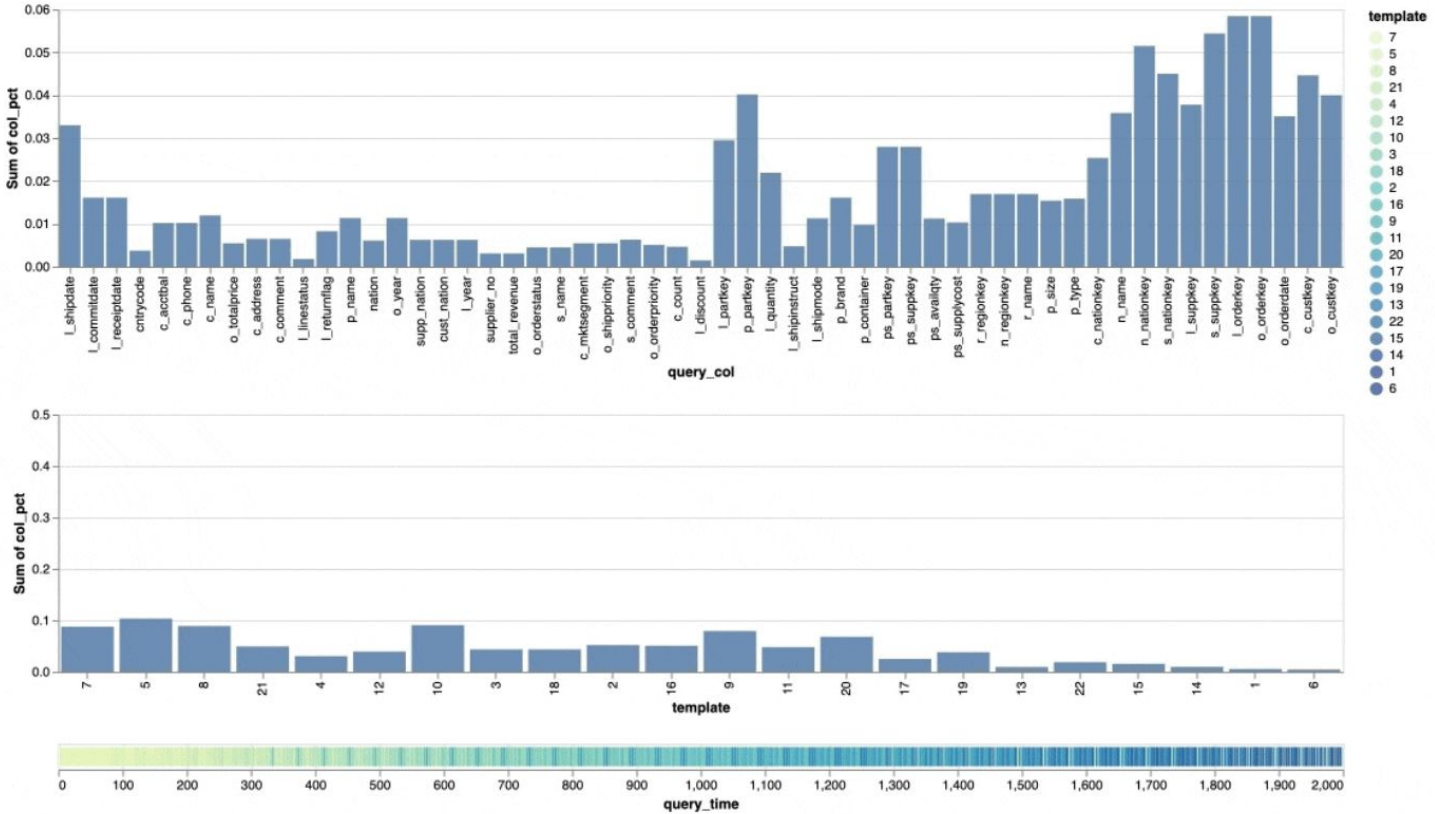
Workload Shift Design - Hard Shift



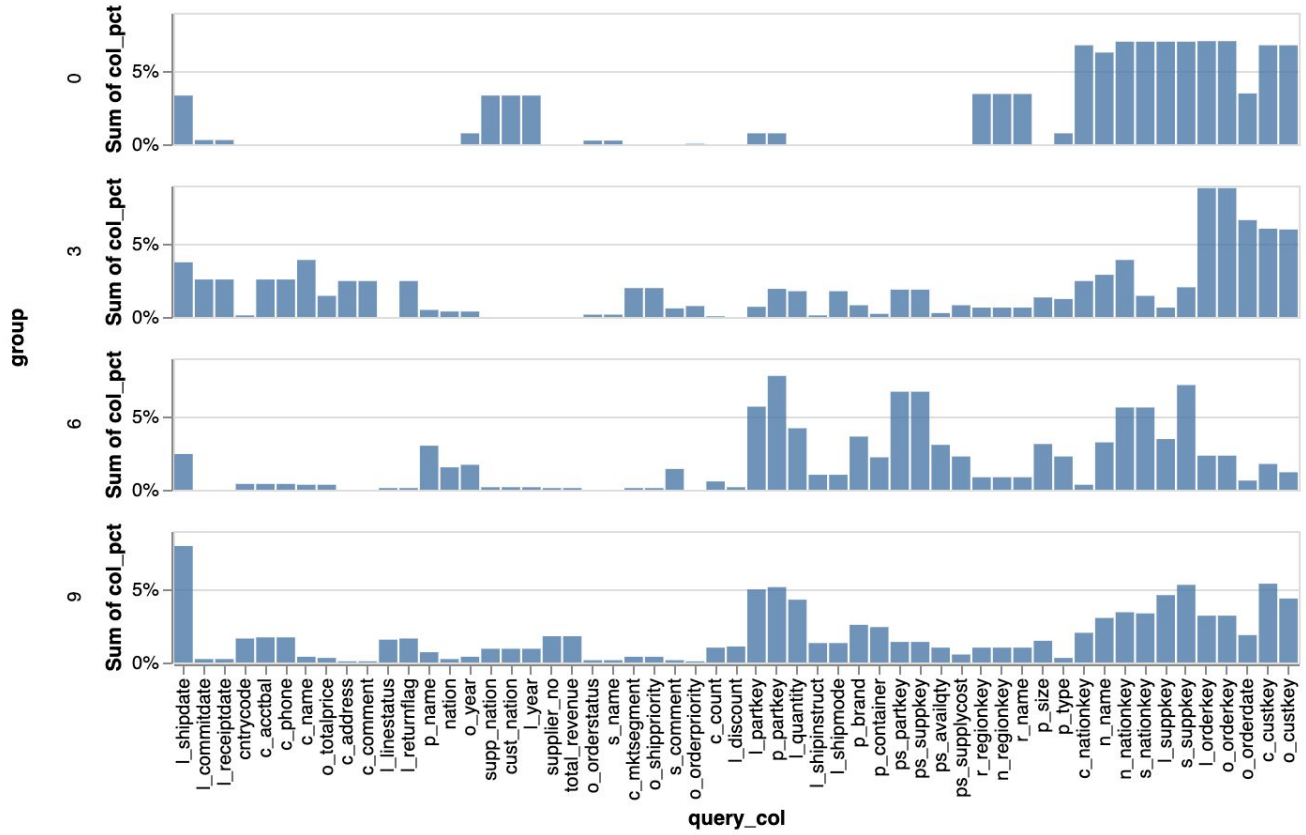
Workload Shift Design - Hard Shift



Workload Shift Design - Gradual Shift



Workload Shift Design - Gradual Shift



Technical Implementations

Extending the SIGMOD ’18 implementation of VerdictDB, we incorporate:

1. Custom **query template parsing** using ANTLR 4
2. **Linear programming** calculation for sample creation
3. A custom **query optimizer**

In addition, we:

1. **Modify the TPC-H query templates** to focus only on the `lineitem` table.
2. Design and **generate query workloads** with varying degrees of shifts.

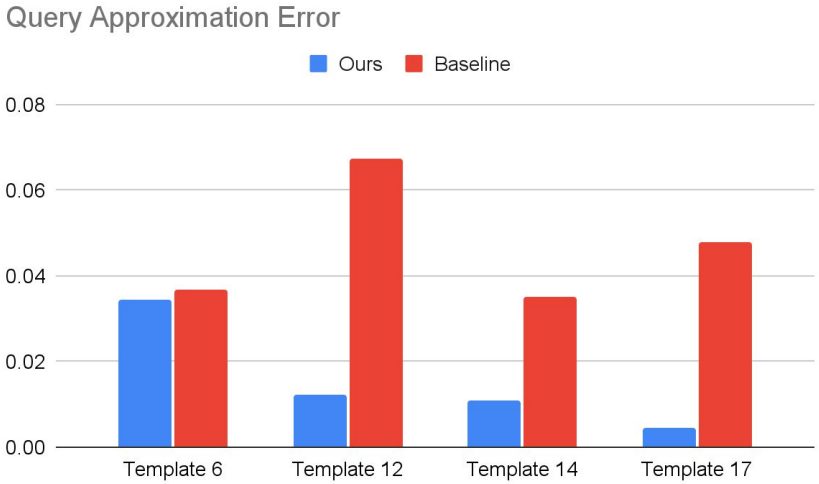
Experiments

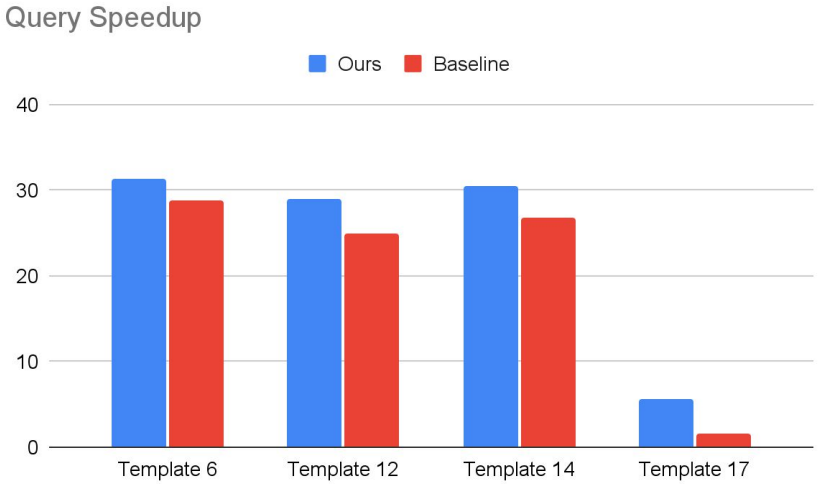
We used a 10GB TPC-H dataset and compare the **approximation error** and **response time** of 4 queries mainly on the `lineitem` table, averaged over 5 runs **with fixed storage budget**:

- **No Sampling**: PostgreSQL with no sampling
- **Baseline VerdictDB**: uniformly sampled 3% of `lineitem` table
- **Ours**: stratified sampled selected QCS of `lineitem` table

  (the QCS is selected by our linear programming results)

Current Results





Discussion and Conclusion

1. We uncovered many discrepancies between existing work and their implementation and many were extremely outdated, making it **difficult to extend upon or even re-implement the systems**.
2. Our approach is able to **yield up to 10x lower approximation error** compared to uniform sampling.
3. On average, our approach achieves **24x speedup**, while the baseline uniform sampling is 20x.
4. However, online experiments with **workload shifts are needed** to prove the validity of our hypothesis.

Future Work

1. Compare our implementation against the heuristic sampling strategy of VerdictDB.
2. Complete implementation of the workload shift detection and incremental sample update.
3. Evaluate our system with (1) the simulated workload and (2) real-world workload with data shift in an online setting.
4. Optimize storage layouts for the samples.
5. Extend our evaluation to multi-table schemas.
6. Scale our evaluation to big data tables.

Q&A

Group 10 - Constraint SAITS

- Shubham Agarwal & Hamsika Rammohan

What is Time Series

- Time Series Database are usually composed of timestamp and associated data
- They are relatively large and uniform as compared to other datasets.

timestamp	cluster	hostname	cpu	iops
2015-04-28T17:50:00Z	Cluster-A	host-a	10	10
2015-04-28T17:50:10Z	Cluster-A	host-b	20	30
2015-04-28T17:50:20Z	Cluster-A	host-a	5	8

Timestamp | Subject dimension | Metrics and measurements

Background and Problem

- Time series data is applicable to almost every domain
- Time series data is widely used in data analysis and machine learning
- Due to system failures or human errors, there may be missing timesteps in the time series
- Many algorithms require data to be of consistent length and simply joining two points around the missing point is enough, hence, data imputation is needed

Previous Work

- BRITS: Uses recurrent dynamics to effectively impute the missing values in multivariate time series.The missing data is part of a RNN graph that is involved in the backpropagation process
- SAITS [2]: Learns missing values by a joint-optimization training approach of imputation and reconstruction of self-attention models to perform missing value imputation for multivariate time series

[1] Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018). Brits: Bidirectional recurrent imputation for time series. Advances in neural information processing systems, 31.

[2] Du, W., Côté, D., & Liu, Y. (2022). SAITS: Self-Attention-based Imputation for Time Series. arXiv preprint arXiv:2202.08516.

Bit and Flip

- The two approaches work well for data imputation but they don’t leverage domain knowledge to aid the imputation process
- PINNs (Physics Informed Neural Networks) [3] show that domain knowledge can help improve predictions
- Hence, we apply create and apply domain specific functions on SAITS and evaluate the results

[3] Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2022). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 1-12.

Data

- We use SPX USD stock prices obtained from the open source financial data set
- 5 features: Open (0), High (1), Low (2), Close (3) and Price (4) with date and time.
- Settings to test our Model: We mask 5%, 10%, 15%, 20% and 25% of data points in two setups:
    - One feature for x% of time steps
    - All features for x% of time steps

Our Proposed Model - CSAITS

- Constrained SAITS (CSAITS), it is domain informed SAITS, where the predictions made at each epoch are constrained by certain domain specific values.
- SAITS is based on self-attention. It learns missing values from a weighted combination of two diagonally masked self-attention (DMSA) modules
- SAITS calculates loss function based on two learning tasks:
    - Masked Imputation Task (MIT)
    - Observed Reconstruction Task (ORT)

Our Proposed Model

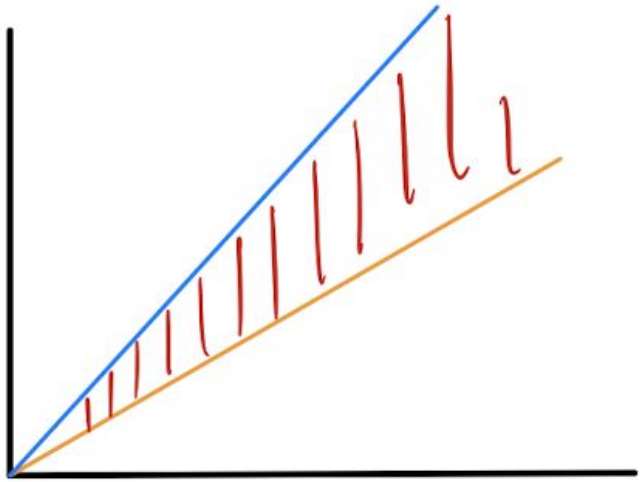
- Add an auxiliary loss (L_{aux})to penalize any predictions that violate constraints set by domain specific functions.

$$L = L_{ORT} + L_{MIT} + L_{aux}$$

Our Proposed Model: What is L_{aux}

- $CSAITS_7$: In this loss function we use Simple Moving Average (SMA) for 7 days as a one sided bound

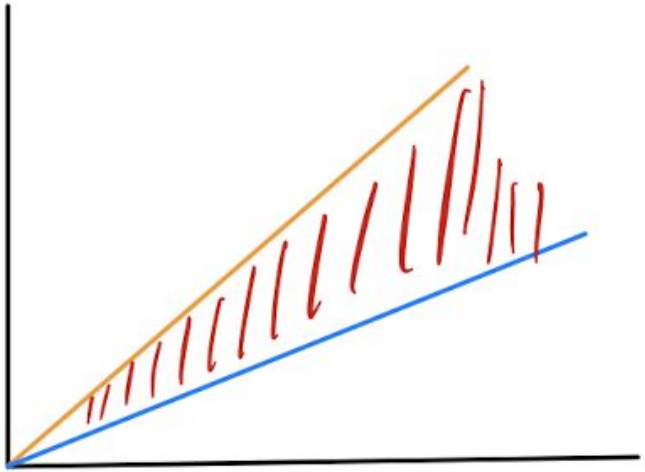
$$L_{aux} = \frac{|\tilde{X}_3 - SMA_7|}{M}$$



Our Proposed Model: What is L_{aux}

- $CSAITS_{28}$: In this loss function we use Simple Moving Average (SMA) for 28 days as a one sided bound

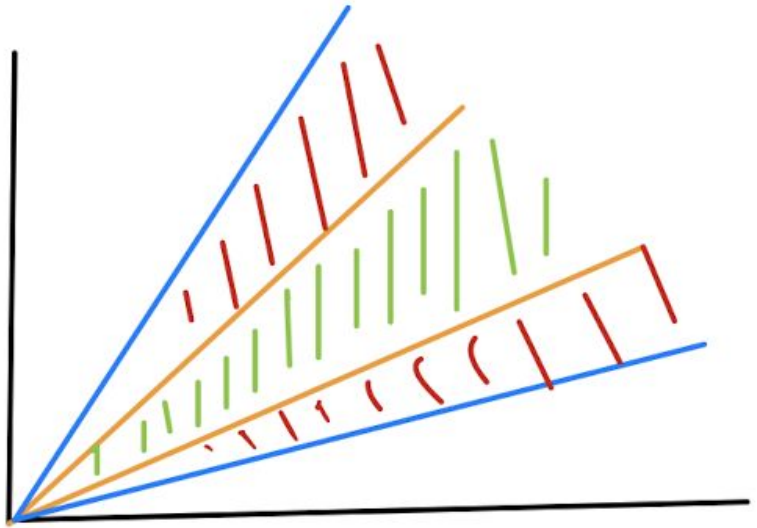
$$L_{aux} = \frac{|\tilde{X}_3 - SMA_{28}|}{M}$$



Our Proposed Model: What is L_{aux}

- CSAITS$_{7\text{-}28}$: Uses range between SMA_7 and SMA_{28} and penalizes only if the predicted value is outside the range.

$$L_{aux} = \begin{cases} |\tilde{X}_3 - max(SMA_{28}, SMA_7)|, & X_3 \geq max(SMA_{28}, SMA_7) \\ |\tilde{X}_3 - min(SMA_{28}, SMA_7)|, & X_3 \leq min(SMA_{28}, SMA_7) \\ 0, & otherwise \end{cases}$$



Results

Model	5%			10%			15%			20%			25%		
	MAE	RMSE	MRE	MAE	RMSE	MRE	MAE	RMSE	MRE	MAE	RMSE	MRE	MAE	RMSE	MRE
Mean	53.99	196.65	0.03	44.42	244.42	0.02	45.27	272.28	0.03	42.97	266.43	0.02	43.81	270.05	0.02
BRITS	1,227	1,259	0.67	1,296	1,330	0.70	1274	1307	0.69	1229	1263	0.67	1237	1271	0.67
SAITS	96	108	0.05	98	112	0.05	103	118	0.06	102	118	0.06	89.78	103.69	0.05
$CSAITS_7$	59.71	68.26	0.03	**39.49**	**46.28**	0.02	52.08	59.38	0.02	50.62	62.55	0.02	46.69	55.28	0.02
$CSAITS_{28}$	**45.37**	**51.44**	0.02	42.29	50.77	0.02	44.83	**51.74**	0.02	51.29	59.78	0.02	**39.48**	**45.44**	0.02
$CSAITS_{7-28}$	47.57	55.75	0.02	53.63	60.77	0.03	**44.27**	51.92	0.02	**42.93**	**49.98**	0.02	48.18	56.88	0.02

Table 1: MAE, RMSE and MRE Comparison for Data Imputation Task for masked time series for 'LOW'. The errors were averaged over 100 runs. Best values for MAE and RMSE have been highlighted for each column. Results for other individual features had the same trends.

Results

Model	5%		10%		15%		20%		25%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BRITS	1784	1831	1832	1882	1831	1882	1818	1869	1814	1862
SAITS	1189	1261	1190	1264	1237	1311	1201	1285	1263	1341
$CSAITS_7$	1235	1308	919	1009	805	907	848	950	888	980
$CSAITS_{28}$	1100	1171	816	914	699	812	750	857	758	863
$CSAITS_{7-28}$	1131	1204	881	972	741	853	767	870	835	934

Table 2: MAE and RMSE Comparison amongst Machine learning models BRITS, SAITS and CSAITS when all features are masked for x% of time steps, x takes up values of 5, 10, 15, 20 and 25. We excluded MRE because it did not provide any additional information in table 1. The errors were averaged over 100 runs. Best values for MAE and RMSE have been highlighted for each column.

Results

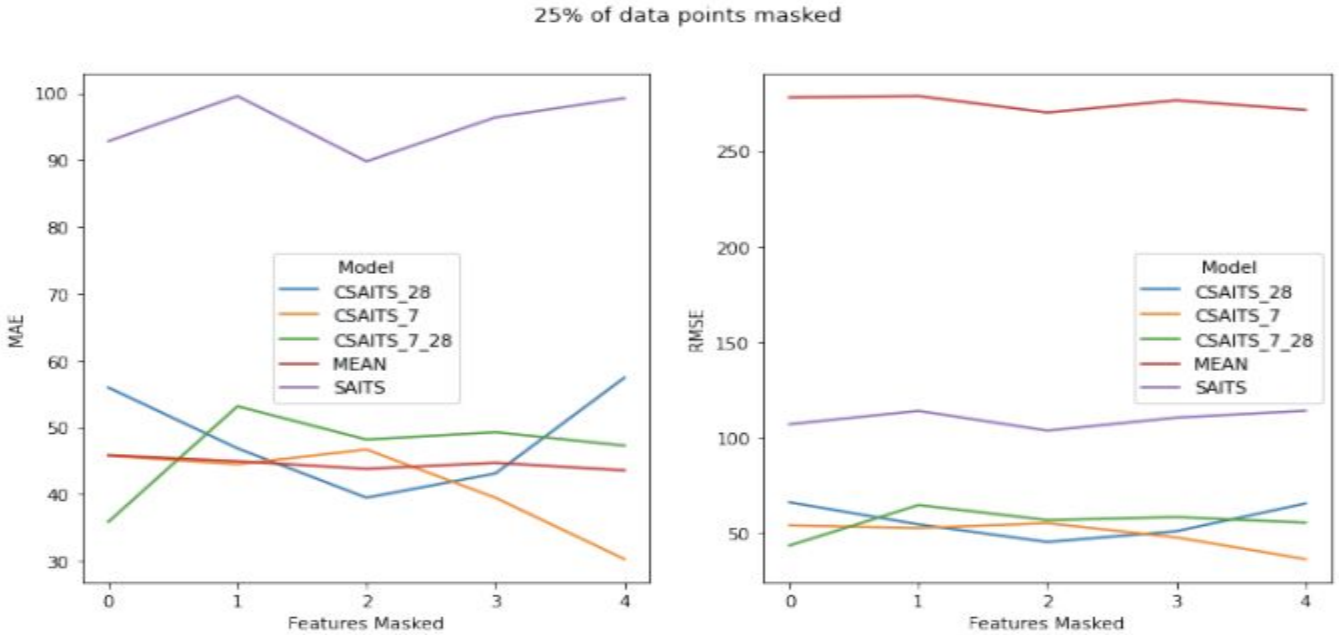


Figure 6: MAE for 25% of data points with the feature id masked

Conclusion

- Domain specific constraints help models to recover data more accurately
- CSAITS performs better than Statistical Methods (Mean) that are heavily dependent on nearby points
- Reduction in loss by ~50% over SAITS

Group 8

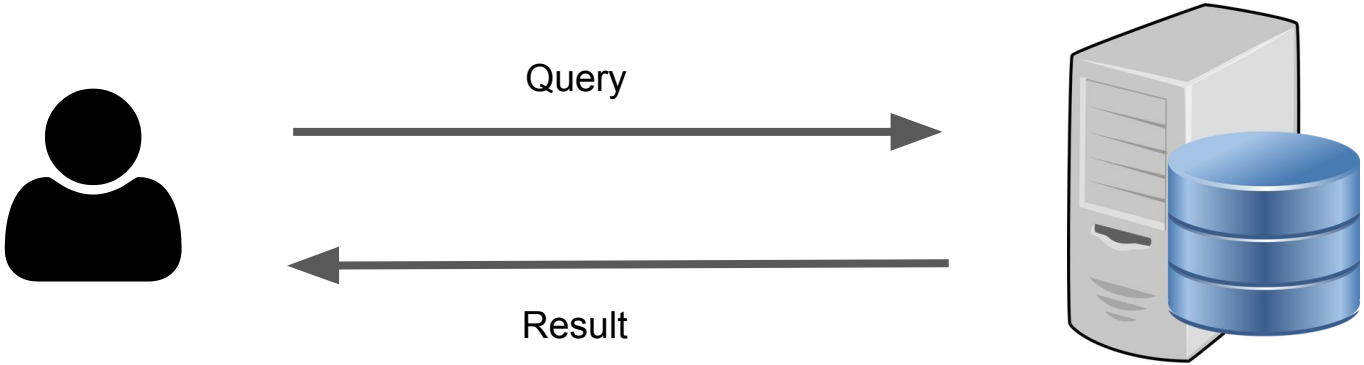
VChainsaw: Parametrizing trust in verifiable boolean range queries

Abhinav Hampiholi and Aniruddha Mysore

The problem setting - unverified results

Dune

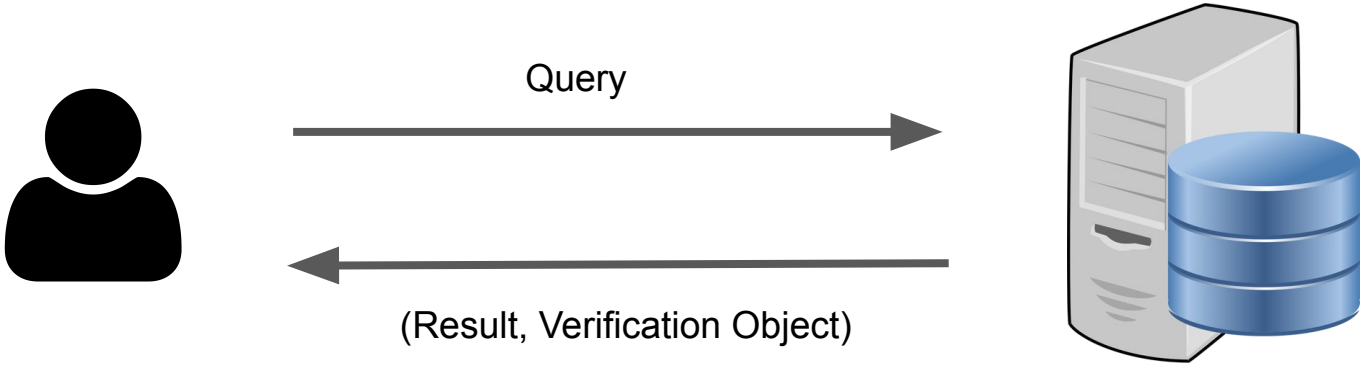
- A user node wants to query a blockchain database



- Advantages: fast, easy to implement
- Disadvantage: The service provider may be malicious and provide incorrect results

The problem setting - verifiable results (vChain+)

- A user node wants to query a blockchain database



- Advantages: The user can verify the result set. Service provider cannot be malicious
- Disadvantage: Very slow

The Bit

Current blockchain query systems are **all or nothing**. Either the user is completely in the dark and has no guarantees about the integrity of results or has complete assurance that the results are ‘correct’ and pays a large latency price.

The Flip

We propose that trust does not need to be binary and design a middle ground. We allow the user to define how ‘strict’ of a correctness proof they require. A less strict proof means lower latency but weaker guarantees on the result set and vice versa.

A user may be satisfied with a less strict proof for a number of reasons

- The service provider is not completely untrustworthy
- The application that the user is interested in does not have very strict requirements

Methodology

- What are boolean range queries?
- What makes a query correct?
- How does the user ‘verify’ a query?

Boolean Range queries

We model our blockchain as a sequence of objects

$$o_i = \{block\ number,\ V_i,\ W_i\}$$

An example object:

$$o1 = \{1,\ [300000,\ 2000],\ "Atlanta"\}$$

Boolean Range Queries

A boolean range query is of the form

$$q = < [b_s, b_e], [\alpha, \beta], \delta >$$

$b_s = start\ block$

$b_e = end\ block$

$\alpha = range\ for\ first\ num\ attr$

$\beta = range\ for\ second\ num\ attr$

$\delta = Boolean\ function\ on\ the\ set\ valued\ attribute$

An example Boolean Range Query

```
let query_data: Value = json!({
    "start_blk": 1,
    "end_blk": 300,
    "range": [(200000,300000), (2000,3000)],
    "keyword_exp": {
        "or": [
            { "input": "'Atlanta'" },
            { "input": "'Augusta'"}
        ]
    },
});
```

What makes a query correct?

- **Soundness**. None of the objects returned as results have been tampered with and all of them satisfy the query conditions.
- **Completeness**. No valid result is missing from the result set.
- **N-completeness**: At least an n-fraction of all objects that satisfy the query are part of the result set.
- N is the trust parameter
- For example: If a result set is 0.8-complete, the user can conclude that at least 80% of all valid objects are present in the result set.

How does a user ‘verify’ a result set?


```
let query_data: Value = json!({
    "start_blk": 1,
    "end_blk": 300,
    "range": [(200000,300000), (2000,3000)],
    "keyword_exp": {
        "or": [
            { "input": "'Atlanta'" },
            { "input": "'Augusta'"}
        ]
    },
});
```

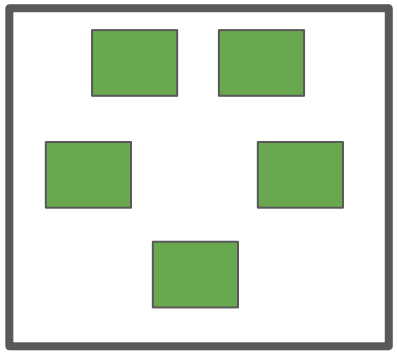
Total number of objects = 10

Result set = {1, 2, 3, 4, 5}

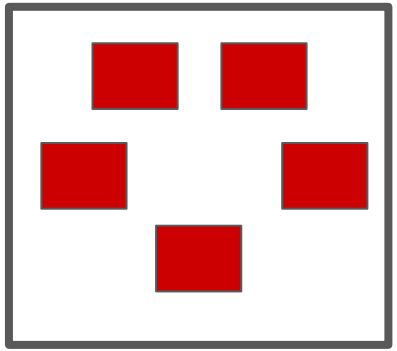
Verification Object = {5 match proofs, 5 mismatch proofs}

Proving N-completeness

Say the user is okay with 0.625 completeness in the previous example



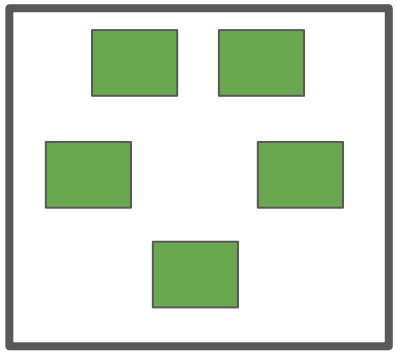
Match Proofs



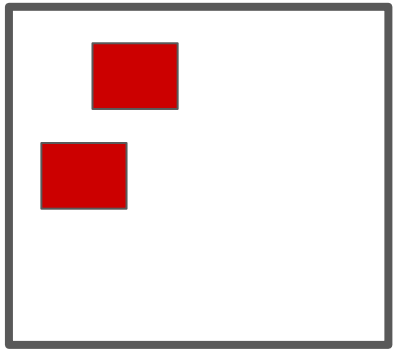
Mismatch Proofs

Proving N-completeness

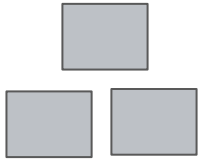
Say the user is okay with 0.625 completeness in the previous example



Match Proofs



Mismatch Proofs



Unknown objects

In general,

N-completeness allows the service provider to generate proofs that are just ‘good-enough’. This implies smaller proofs and faster queries!

Suppose there are t objects in a chain
Let $f \cdot t$ of these objects satisfy a query q
Suppose the user requests an N-complete result set
An honest service provider needs to return
$f \cdot t$ match proofs and
$(1 - \frac{f}{N}) \cdot t$ mismatch proofs if $N > f$
0 mismatch proofs if $N \leq f$

High level implementation

Result set = {All matches!}

Verification Object = {All match proofs, “Some” mismatch proofs}

Number of mismatch proofs is given by previous equation

Service Provider

Verifies each of the proofs that are provided.

Assumes that all the “skipped” proofs are in fact matches. This is the worst case.

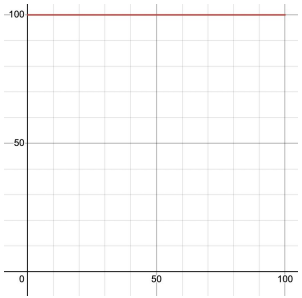
Calculates the lower bound on fraction of matches = (matches/(matches+unknown))

Accepts if this is >=N

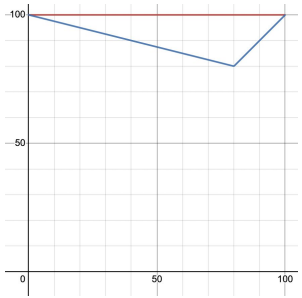
The User

How the number of proofs in VO vary with N

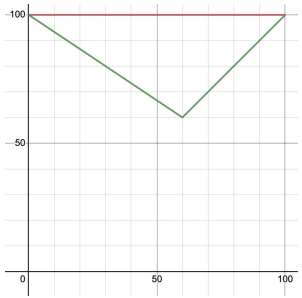
Assuming total number of objects = 100



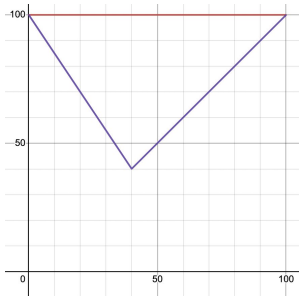
N = 1



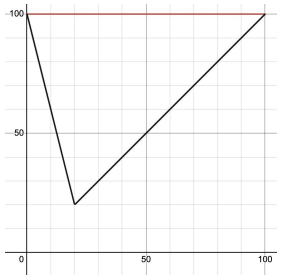
N = 0.8



N = 0.6



N = 0.4



N = 0.2

x = Number of matches [0,100]

y=Total number of proofs needed (match + mismatch)

Evaluation Method

Dataset - Foursquare (limited by hardware to 300/20000 blocks)

Workload - Boolean Range queries

```
1 [40733596, 74003139] {'US', 'JazzClub'}
1 [40758102, 73975734] {'US', 'Gym'}
1 [40732456, 74003755] {'US', 'IndianRestaurant'}
1 [42345907, 71087001] {'US', 'IndianRestaurant'}
1 [39933178, 75159262] {'US', 'SandwichPlace'}
1 [40652766, 74003092] {'US', 'BowlingAlley'}
1 [40726961, 73980039] {'US', 'DiveBar'}
1 [40756353, 73967676] {'US', 'Bar'}
1 [37779837, 122494471] {'US', 'SeafoodRestaurant'}
1 [34092793, 118281469] {'US', 'Bar'}
1 [40591334, 73960725] {'US', 'Nightclub'}
1 [40733630, 74002288] {'US', 'JazzClub'}
1 [41941562, 87664011] {'US', 'Pub'}
1 [34075314, 118253499] {'US', 'Bar'}
1 [40724822, 73981456] {'US', 'DiveBar'}
1 [40739685, 74006020] {'US', 'FrenchRestaurant'}
1 [40756119, 73972532] {'US', 'HotelBar'}
1 [42346127, 71080363] {'US', 'FrenchRestaurant'}
1 [40718363, 73990817] {'US', 'Bar'}
1 [40722206, 73981720] {'US', 'Theater'}
1 [40722842, 73994116] {'US', 'CubanRestaurant'}
```

```
let queryi_param_data = json!({
        "start_blk": 1,
        "end_blk": 4000,
        "range": [(1,79950024), (2,122494474)],
        "keyword_exp": {
            "or": [
                { "input": "'US'" },
                { "input": "'Gym'"}
            ]
        },
    });
```

Results

Query Performance	10 blocks (seconds)	20	30	40	50	60	70	**300**
Baseline Q1	12.28	17.61	23.52	28.73	39.48	45.23	51.59	210.30
Baseline Q2	6.220	8.710	11.66	14.12	19.27	22.73	25.86	105.47
Baseline Q3	7.46	11.2	15.98	18.18	25.77	29.32	35.18	**148.44**
No proof Q3 (bplus tree)	7.11	10.86	14.37	17.75	24.26	29.17	33.36	**122.23**

Conclusion

1. Implemented changes to the state-of-the-art algorithm to test our hypothesis, “relaxing completeness constraint improves query performance time”

2. We demonstrate **17%** speedup by reducing proof computation on b-tree lookups

3. Next step: Further studies with varying ***N***

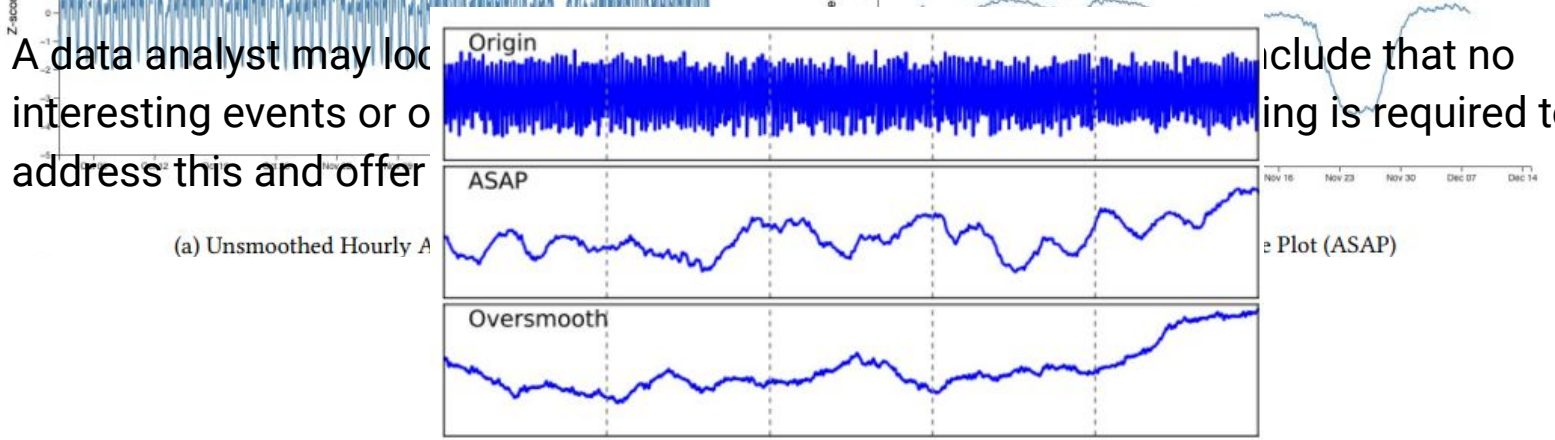
Group 5

Smooth Operator: Step Towards a Generalized Smoothing Framework

Sankalp Sangle, Siddhi Pandare, Tanya Garg

Background and Problem

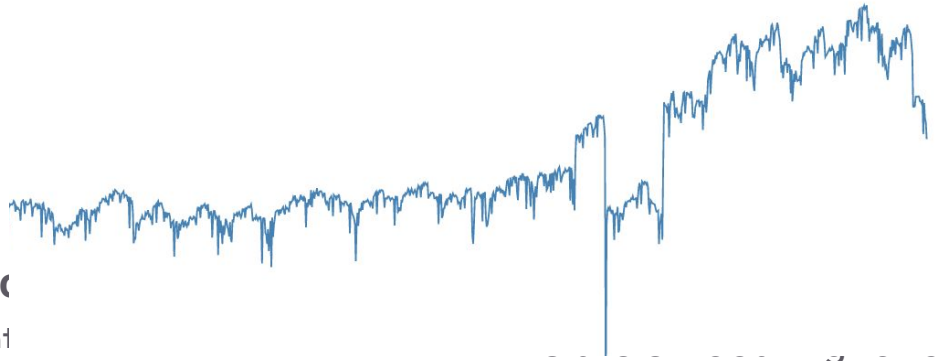
- ➔ Present systems present noisy plots when plotting raw data; this makes it hard to get meaningful insights from them easily => Smoothing it helps!
- ➔ Previous efforts (which will be discussed later on) do not generalize well and at times the user wants to see a different result.
- ➔ A data analyst may loc … clude that no interesting events or o … ing is required to address this and offer



(a) Unsmoothed Hourly A … e Plot (ASAP)

Kexin Rong and Peter Bailis. 2017. ASAP: prioritizing attention via time series smoothing. Proc. VLDB Endow. 10, 11 (August 2017), 1358–1369. https://doi.org/10.14778/3137628.3137645

The Bit



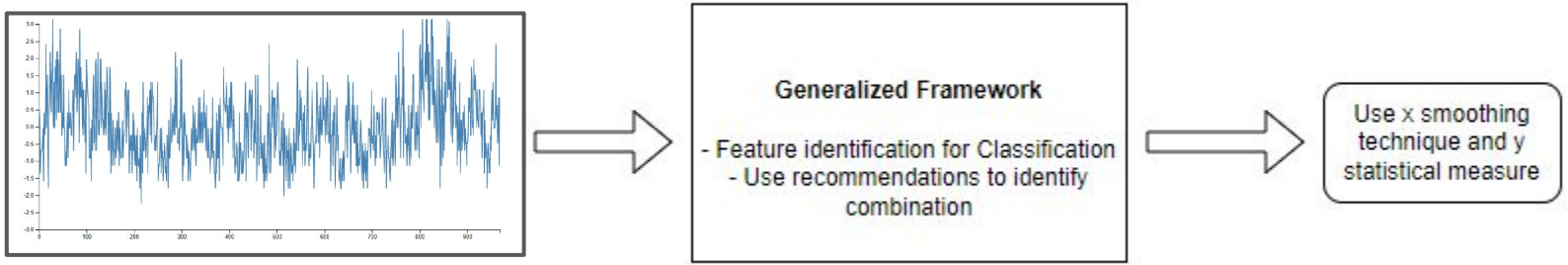
- ➔ Smoothing
  - ◆ **LineSmoo**
    varied dat
    the
  - ◆ Hov
    the
- ➔ Prevent
  - ◆ **AS**
    sm
    me
  - ◆ Hov

sian, mean, etc on a
ions according to
sive
e fea
rese
fun
o all

1. P. Rosen and G. J. Quadri, "LineSmooth: An Analytical Framework for Evaluating the Effectiveness of Smoothing Techniques on Line Charts," in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pp. 1536-1546, Feb. 2021, doi: 10.1109/TVCG.2020.3030421.
2. Kexin Rong and Peter Bailis. 2017. ASAP: prioritizing attention via time series smoothing. Proc. VLDB Endow. 10, 11 (August 2017), 1358–1369. https://doi.org/10.14778/3137628.3137645

The Flip

We propose a generalised framework that can suggest the best combination of a smoothing technique and statistical measure.



In this paper, we take the first step towards this approach by identifying what combination of smoothing function and statistical measure best fits a category or type of time series and make relevant recommendations.

Methodology and Technical Details

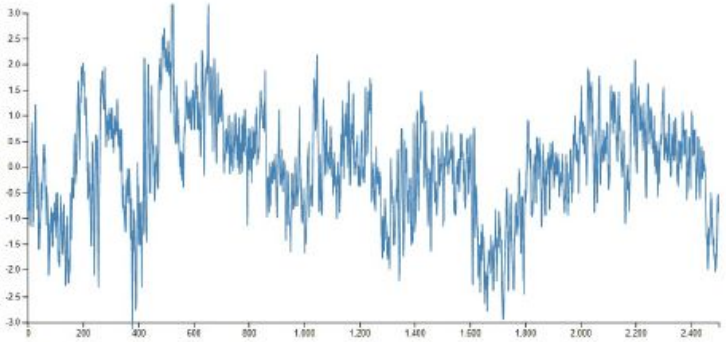
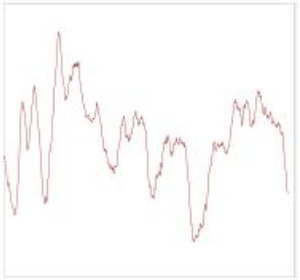
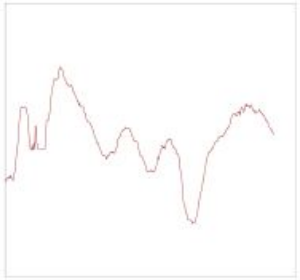
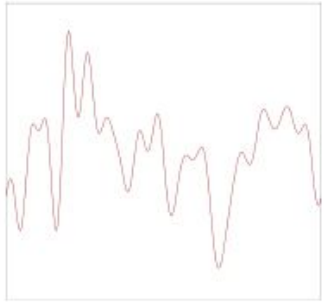
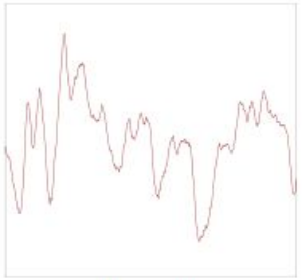
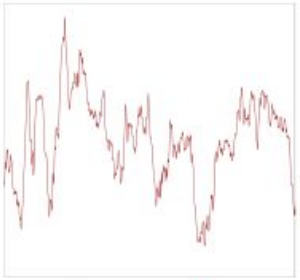
- Collected datasets that broadly re … series.
- 
- 
- 
- 
- 
- 

Figure 3: Unsmoothed EEG data

(a) Mean

(b) Median

(e) Low pass fileter

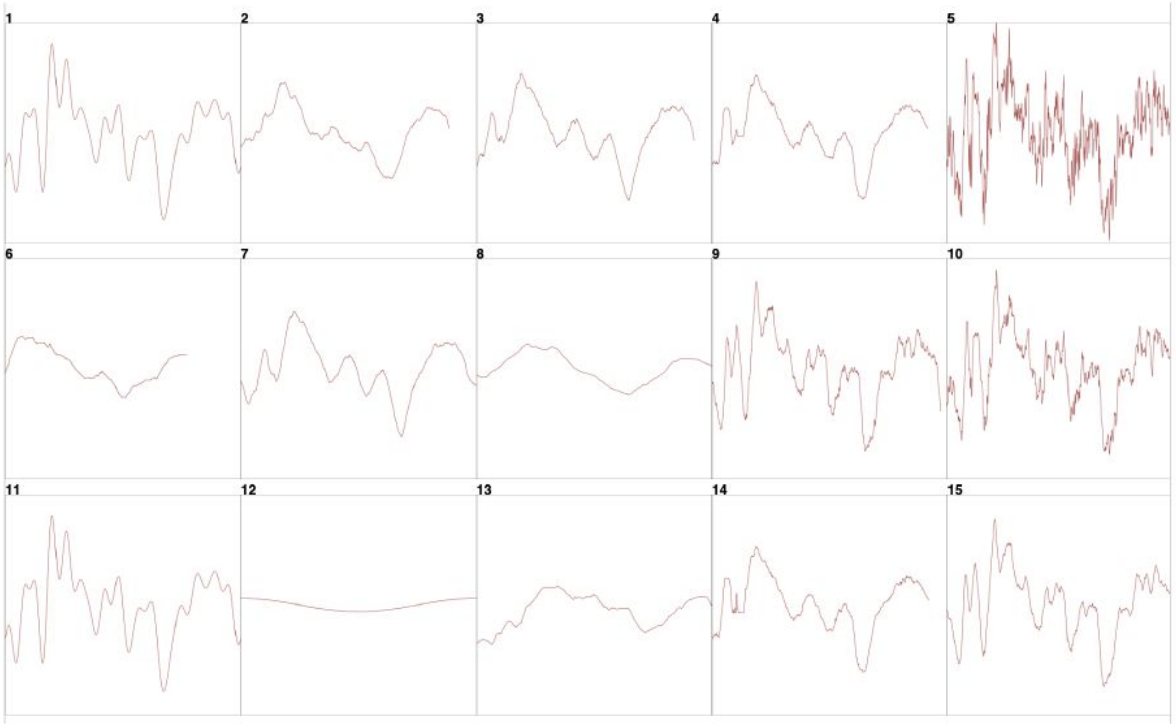
(c) Gaussian

(d) Exponential

… ıness and the statistical measure of the smooth

$$score = \alpha * \sigma + (1 - \alpha) * SM$$

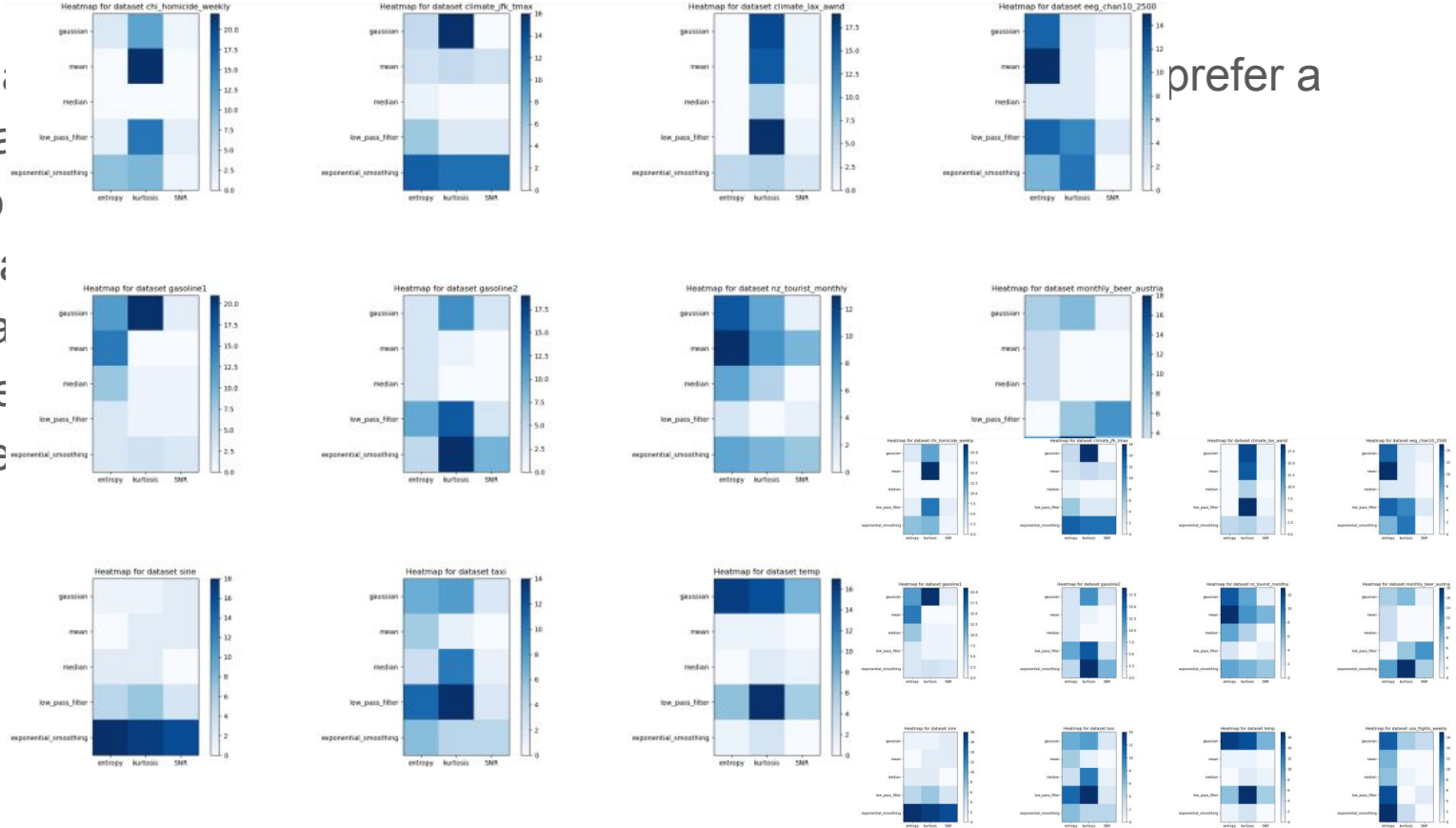
Methodology and Technical Details (Contd.)



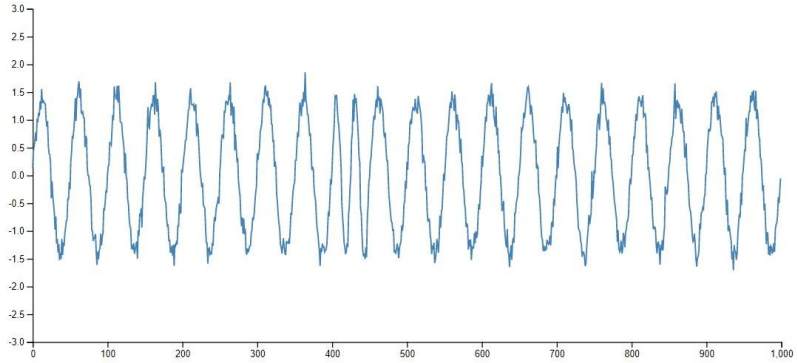
- ➔ We di ns of smoo sets.
- ➔ We pr udy. For each data s df of 3 (smoo

Results - Qualitative

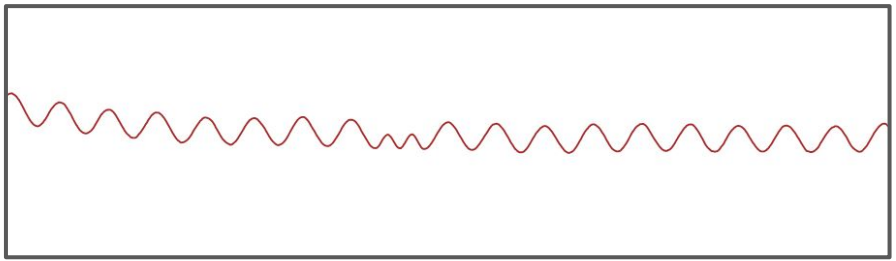
- ➔ Some da… prefer a particula…
- ➔ SNR do…
- ➔ Seasona…
- ➔ Gaussia…
- ➔ Mixed fe…
- ➔ The que…



Results - Qualitative



Expected

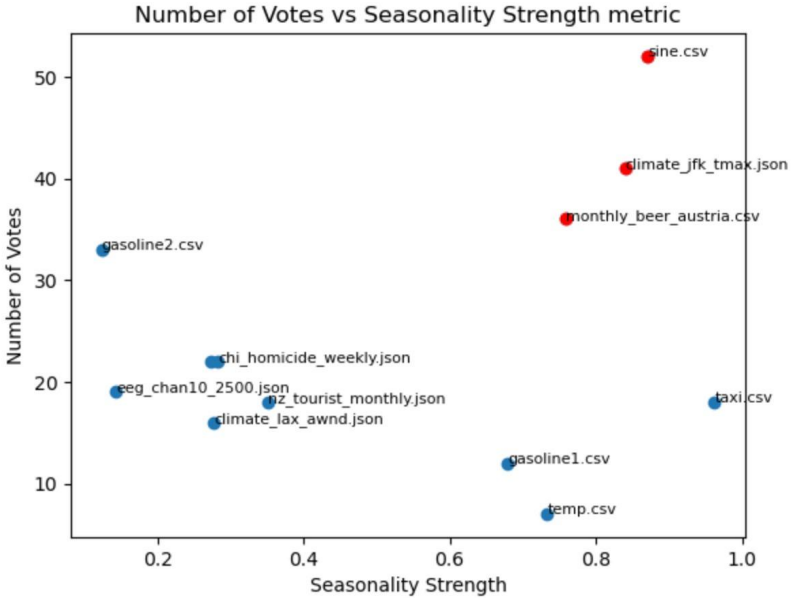


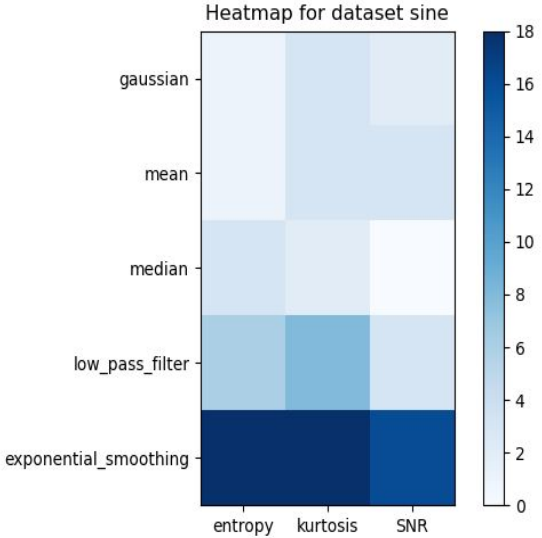
Actual

Results - Quantitative

➔ Strength of Seasonality:

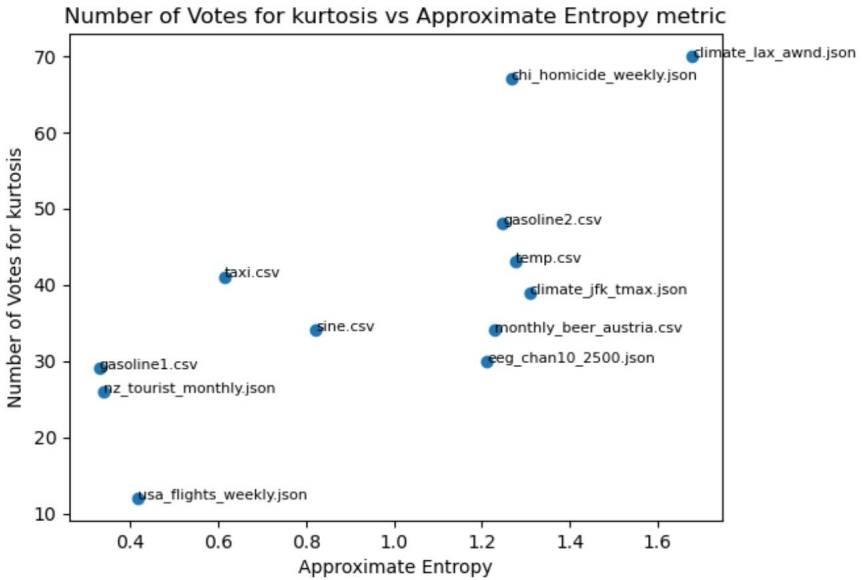
$$F_S = \max\left(0, 1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(S_t + R_t)}\right)$$

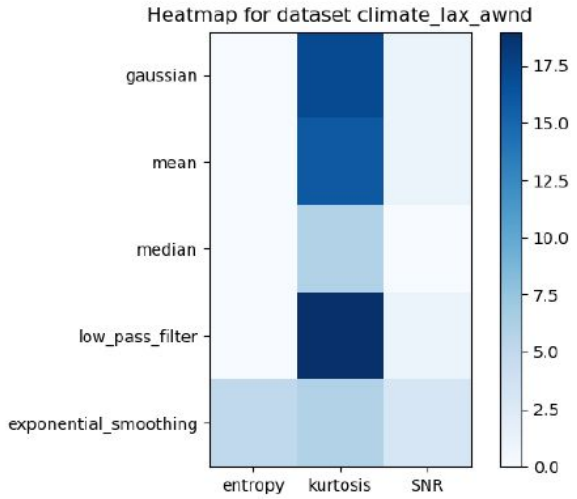




Results - Quantitative (Contd.)

➔ Approximate Entropy: Quantifies the amount of the unpredictability of fluctuations.





Key Takeaways

- ➔ We presented the idea of a **generalized smoothing framework** and divided the main problem into two important subsections.
- ➔ Here, we addressed the second task of understanding how the ideal **combination of a smoothing technique and statistical measure depend on the time series** in question.
- ➔ Using our threshold algorithm and user study, we **prioritise the users choice** and get quantitative and qualitative insights to make **recommendations for particular features** of a time series dataset. This can be incorporated in the final framework.

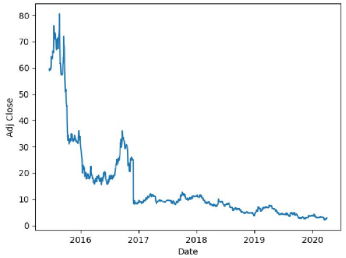
Group 4

Line Chart Similarity

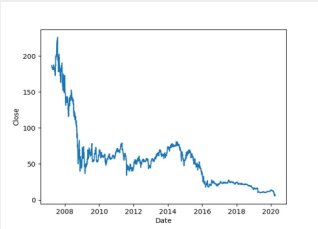
Authors: Sahil and ~~Sahil~~ Harshal

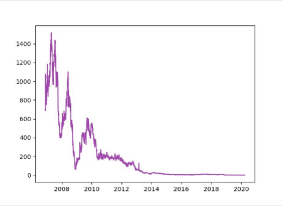
What's the problem, briefly, and why does it matter?

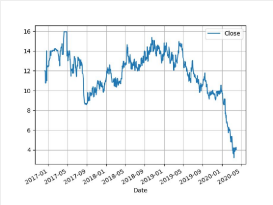
Query Image:

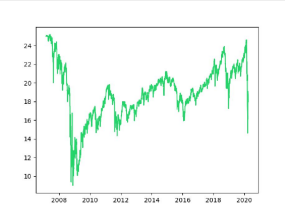


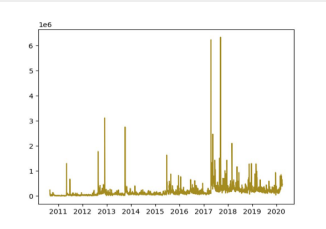
Images in dataset:

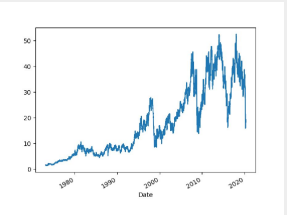










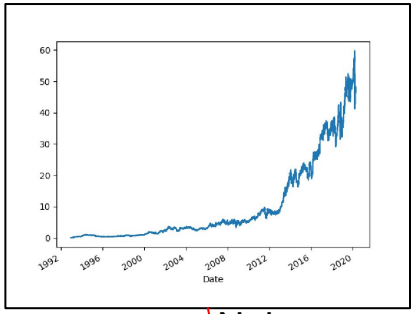


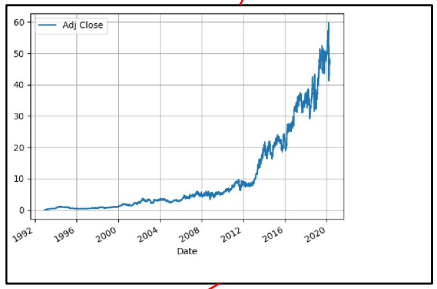
More similar underlying data

Less similar underlying data

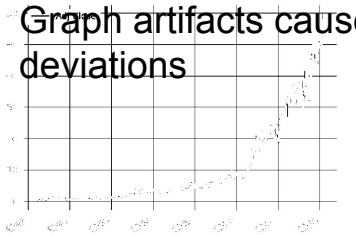
Why hasn't prior work been able to address the problem?

NN don't find the curve





Graph artifacts cause deviations



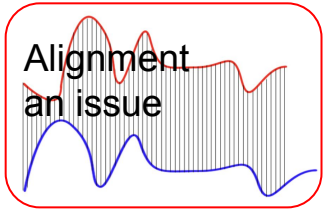
Noisy conversion

{1., 1.5, 1.5, 1.5, 2, 2, ...}

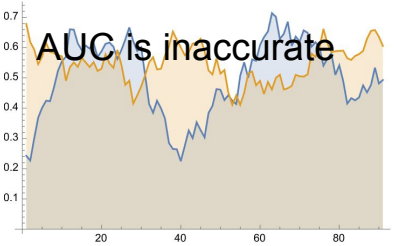
{1.13, 2.36, 2.10, 1.50, 2.79, 2.68, ...}

Data unavailable

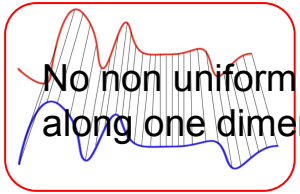
Alignment an issue



AUC is inaccurate



No non uniform scaling along one dimension/axis

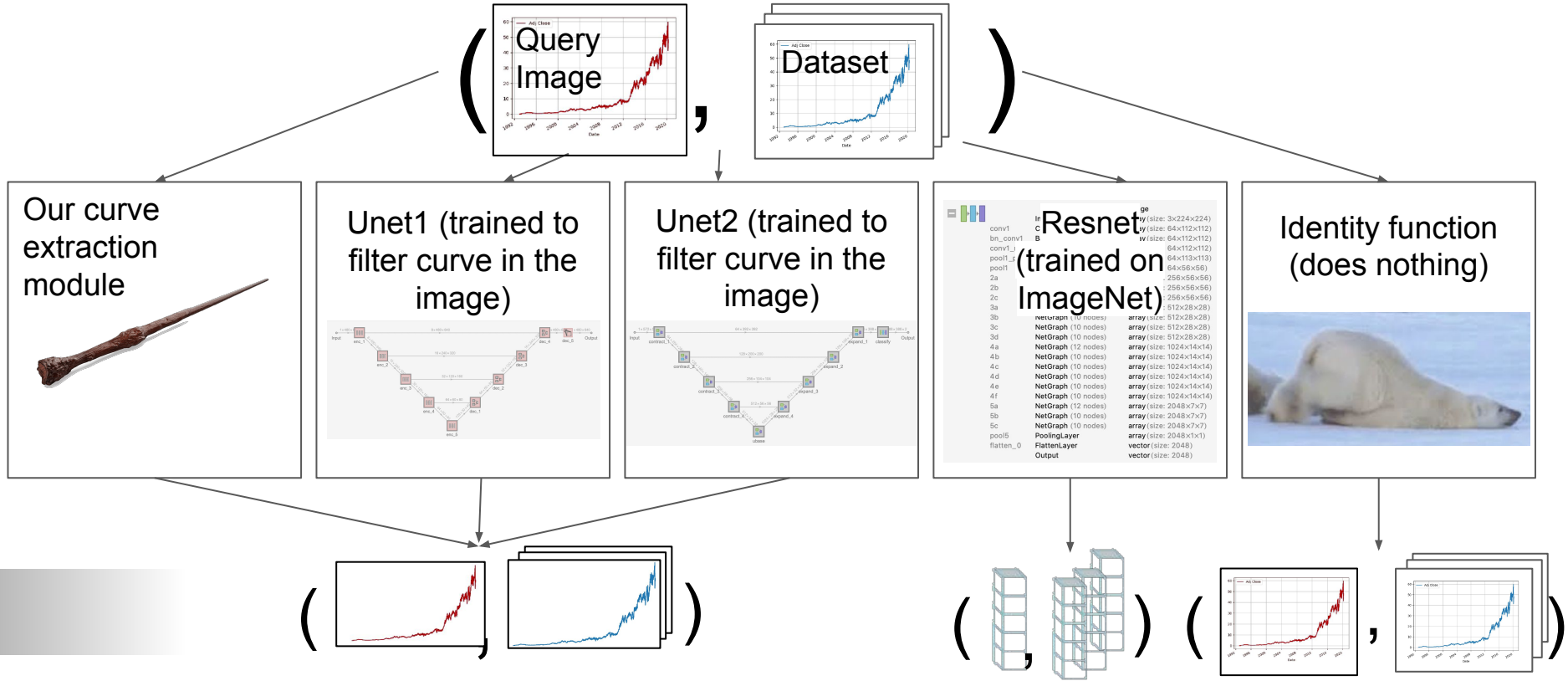


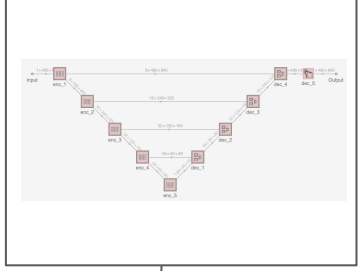
NetChain[uninitialized
Input port: array
Output port: vector (size: 2)]

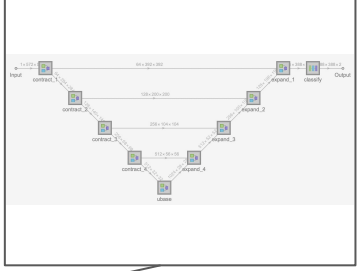
What’s your big idea?

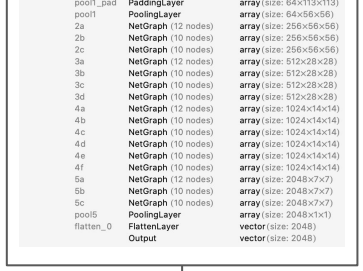
In this project, we introduce a new image-level algorithm (with no access to underlying data) which is robust against changes in visual elements that do not represent change in the underlying data, for instance gridlines, etc.

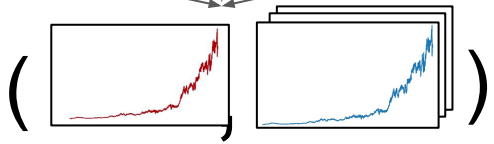
Explain enough technical detail for the listener to understand what you did, at a high level.

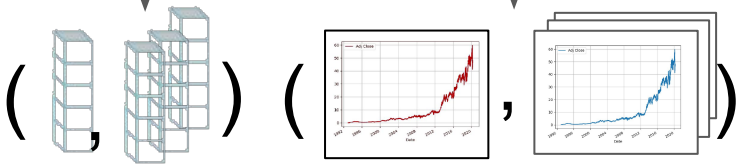










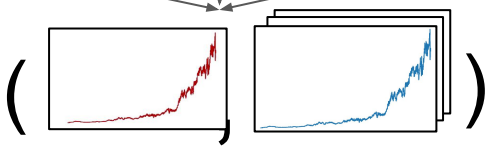


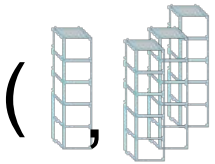
Affine transformation for alignment with respect to the query image (i.e. finding Homography Matrix)

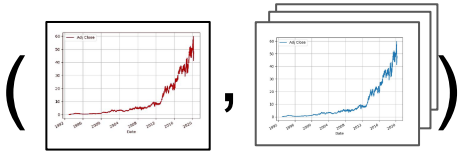
Curve Overlap Ratio

Euclidean Distance

Cosine Distance







Affine transformation for alignment with respect to the query image (i.e. finding Homography Matrix)

Curve Overlap Ratio

Euclidean Distance

Cosine Distance

matplotlib_560_285.png_OurEuclideanDistance.csv

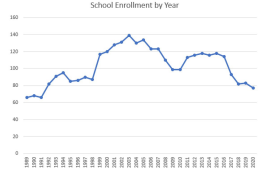
Users > harshal > OneDrive - Harshal Gajjar > Semesters > sem11 > RE-ProfKong > Graphs > Results2_50

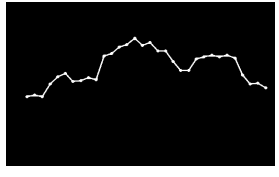
```
1  "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AEE.csv/matplotlib_560_285.png"
2  "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AEE.csv/matplotlib.png"
3  "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AEE.csv/gridlines.png"
4  "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AEE.csv/df.png"
5  "/home/sahil
6  "/home/sahil
7  "/home/sahil                                                    .png"
8  "/home/sahil
9  "/home/sahil                                                    .png"
10 "/home/sahil                                                    5.png"
11 "/home/sahil/Desktop/MDS/Graphs/FinalDataset/BIF.csv/df_326_449.png"
12 "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AAON.csv/gridlines_467_459.png"
13 "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AAON.csv/sns_353_248.png"
14 "/home/sahil/Desktop/MDS/Graphs/FinalDataset/AAON.csv/sns.png"
```

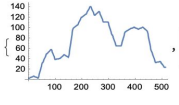
Ranking

Explain enough technical detail for the listener to understand what you did, at a high level.

```
giveFinalCurve[imagePath_] := Module[{},
  graph = Import[imagePath];
  graph = RemoveAlphaChannel@ImageCompose[ImageResize[Image[{{1}}], ImageDimensions@graph], graph];

  hFilter = Closing[graph, BoxMatrix[{100, 0}]];
  vFilter = Closing[graph, BoxMatrix[{0, 100}]];
  grid = ImageApply[Min, {hFilter, vFilter}];

  cleanGraph = DeleteSmallComponents[Erosion[Dilation[Binarize[ImageDifference[graph, grid]], 2], 2]];

  largestComponent = Sort[ComponentMeasurements[cleanGraph, "OuterPerimeterCount"], #1[[2]] > #2[[2]] &][[1, 1]];
  finalCurve = ComponentMeasurements[cleanGraph, "Image"][[largestComponent]][[2]];

  finalCurve2 = finalCurve;
  While[True,
   If[
    Length[ComponentMeasurements[Erosion[finalCurve2, 1], "Area"]] == 1,
    finalCurve2 = Erosion[finalCurve2, 1];
    (*Print["erosion repeat"];*)
    ,
    (*Print["Breaking due to more than 1 component"];*)
    Break[];
   ]
  ];

  finalCurve3 =
   N /@ Function[{col}, Mean[First /@ Select[MapIndexed[{#2[[1]], #1} &, Reverse[col]], #[[2]] == 1 &]]] /@
     Transpose[ImageData[finalCurve2]];

  {ListLinePlot[finalCurve3, PlotRange → All], finalCurve2, finalCurve3}
]
```







{4., 4., 4., 4., 4., 4.5, 5., 5.5, 5.5, 5.5, 6., 6., 6., 6.5, 6.5, 6.5, 7., 7.5, 8., 8., 8., 7.5, 7., 6.5, 6.5, 6.5, 6., 6., 6., 5.5, 5.5, 5.5, 5., 4.5, 4., 4., 4.5, 6., 7.5, 11., 12.5, 14.5, 16.5, 18., 20., 21.5, 23.5, 25.5, 26.5, 30.5, 32., 33.5, 34., 34., 35., 36., 38., 39.5, 40.5, 41.5, 42.5, 43.5, 44.5, 45.5, 47., 49., 50., 51., 51., 51., 51.5, 52., 53., 53.5, 54., 54.5, 55., 55.5, 56., 56.5, 57., 58., 58.5, 59., 59., 59., 58., 57., 54., 53., 52., 51., 49.5, 48., 47., 46., 45., 42., 41., 40., 40., 40., 40., 40.5, 40.5, 41., 41., 41., 41., 41., 41., 41., 41., 41.5, 41.5, 42., 42., 42., 42., 42.5, 43., 44., 44., 44.5, 45., 45.5, 46., 46.5, 47., 47., 48., 48.5, 49., 49., 49., 48.5, 48., 47.5, 47.5, 47., 47., 46.5, 46., 46., 45.5, 45.5, 45., 44.5, 44., 45., 46.5, 48.5, 51., 57.,

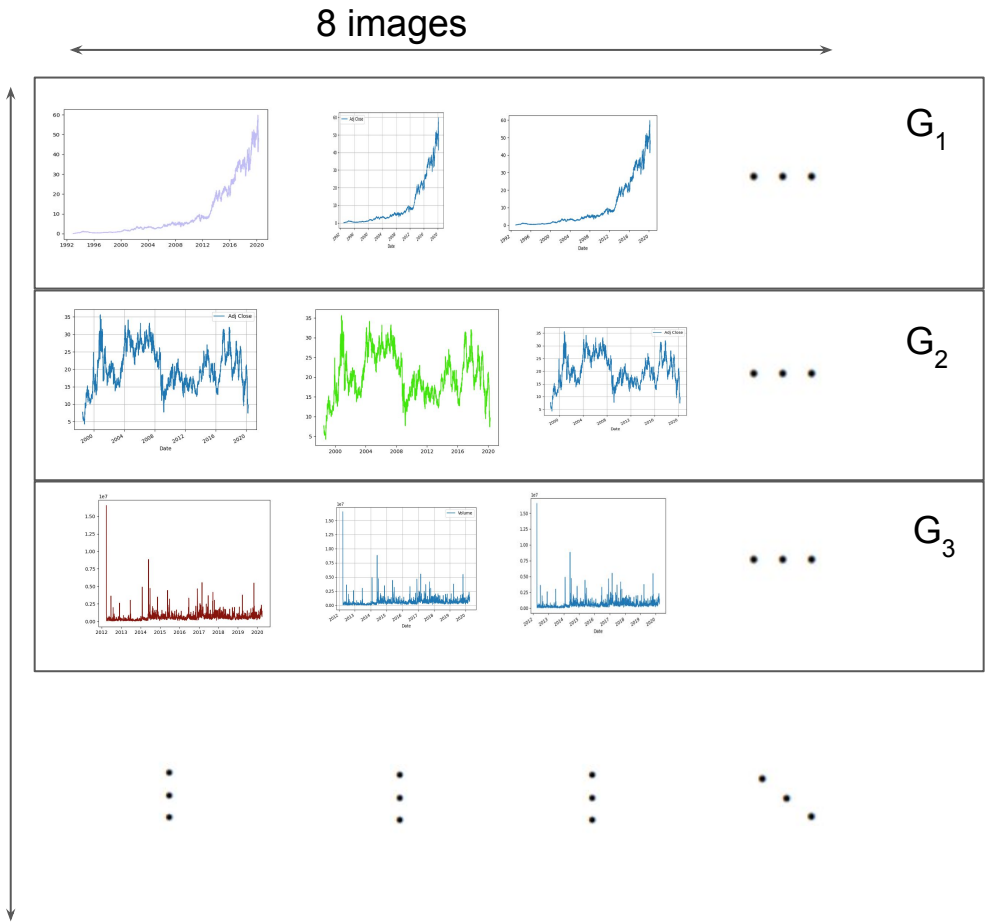
Dataset

kaggle → 50 unique data series

https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

G_i = Graphs from same underlying data

G = Union[{G_1, G_2, …}]



Metrics for Evaluations

(used on generated rankings)

$$Recall@k = \frac{\text{Number of relevant recommended items @k}}{\text{Total number of relevant items}}$$

k=10

matplotlib.png_directImage_EuclideanDistance.csv

Users > harshal > OneDrive - Harshal Gajjar > Semesters > sem11 > RE-ProfKong > Graphs > Results2_500 > AAON

```
1  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/matplotlib.png"
2  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/matplotlib_577_372.png"
3  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/sns.png"
4  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AWF.csv/matplotlib.png"
5  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/sns_353_248.png"
6  "/Users/harshal/Downloads/FinalDataset_sans_scatter/DDT.csv/matplotlib.png"
7  "/Users/harshal/Downloads/FinalDataset_sans_scatter/LHX.csv/matplotlib.png"
8  "/Users/harshal/Downloads/FinalDataset_sans_scatter/AWF.csv/matplotlib_579_327.png"
9  "/Users/harshal/Downloads/FinalDataset_sans_scatter/KNAB.csv/matplotlib.png"
10 "/Users/harshal/Downloads/FinalDataset_sans_scatter/LHX.csv/matplotlib_321_317.png"
11 "/Users/harshal/Downloads/FinalDataset_sans_scatter/SC.csv/matplotlib.png"
12 "/Users/harshal/Downloads/FinalDataset_sans_scatter/DDT.csv/matplotlib_566_312.png"
13 "/Users/harshal/Downloads/FinalDataset_sans_scatter/KNAB.csv/matplotlib_512_318.png"
14 "/Users/harshal/Downloads/FinalDataset_sans_scatter/SC.csv/matplotlib_515_457.png"
15 "/Users/harshal/Downloads/FinalDataset_sans_scatter/LOOP.csv/matplotlib_560_409.png"
16 "/Users/harshal/Downloads/FinalDataset_sans_scatter/AEE.csv/matplotlib.png"
17 "/Users/harshal/Downloads/FinalDataset_sans_scatter/KOF.csv/matplotlib.png"
18 "/Users/harshal/Downloads/FinalDataset_sans_scatter/EVSTC.csv/matplotlib.png"
19 "/Users/harshal/Downloads/FinalDataset_sans_scatter/KOF.csv/matplotlib_423_463.png"
20 "/Users/harshal/Downloads/FinalDataset_sans_scatter/BIF.csv/matplotlib_392_415.png"
21 "/Users/harshal/Downloads/FinalDataset_sans_scatter/LOOP.csv/matplotlib.png"
22 "/Users/harshal/Downloads/FinalDataset_sans_scatter/BIF.csv/matplotlib.png"
23 "/Users/harshal/Downloads/FinalDataset_sans_scatter/EVSTC.csv/matplotlib_381_252.png"
24 "/Users/harshal/Downloads/FinalDataset_sans_scatter/CPLP.csv/matplotlib.png"
25 "/Users/harshal/Downloads/FinalDataset_sans_scatter/AEE.csv/matplotlib_560_285.png"
```

Find all AAONs (its 8)

$$Precision@k = \frac{\text{Number of relevant recommended items @k}}{\text{k}}$$

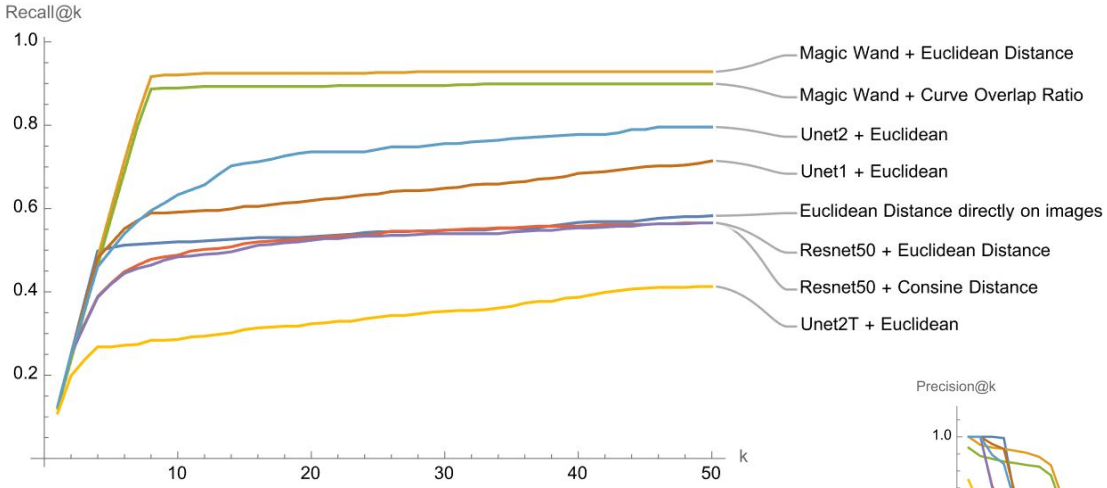
k=6

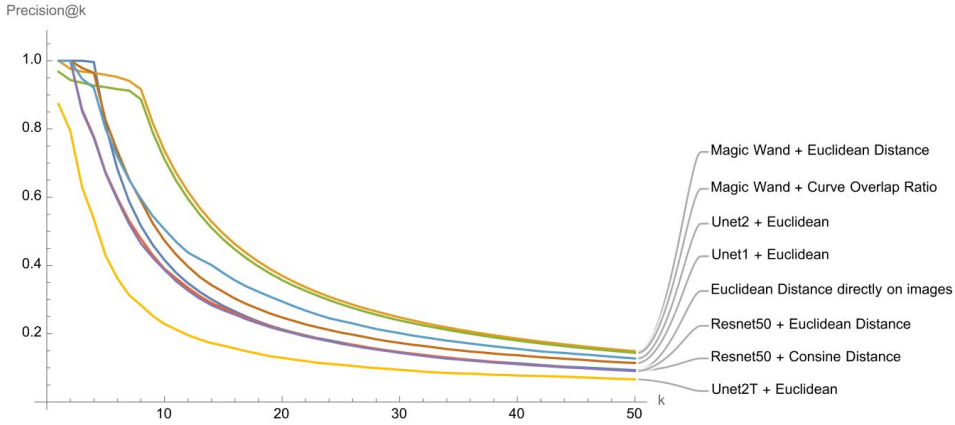
matplotlib.png_directImage_EuclideanDistance.csv

Users > harshal > OneDrive - Harshal Gajjar > Semesters > sem11 > RE-ProfKong > Graphs > Results2_500 > AAON

```
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/matplotlib_577_372.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/sns.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AWF.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AAON.csv/sns_353_248.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/DDT.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/LHX.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AWF.csv/matplotlib_579_327.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/KNAB.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/LHX.csv/matplotlib_321_317.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/SC.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/DDT.csv/matplotlib_566_312.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/KNAB.csv/matplotlib_512_318.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/SC.csv/matplotlib_515_457.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/LOOP.csv/matplotlib_560_409.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AEE.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/KOF.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/EVSTC.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/KOF.csv/matplotlib_423_463.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/BIF.csv/matplotlib_392_415.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/LOOP.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/BIF.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/EVSTC.csv/matplotlib_381_252.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/CPLP.csv/matplotlib.png"
"/Users/harshal/Downloads/FinalDataset_sans_scatter/AEE.csv/matplotlib_560_285.png"
```

Explain enough about your results for the listener to understand what you observed, at a high level.





Conclude with a restatement of your thesis.

Conclusion: 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, non uniform resizing of the image, plot background, etc.

Group 12

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

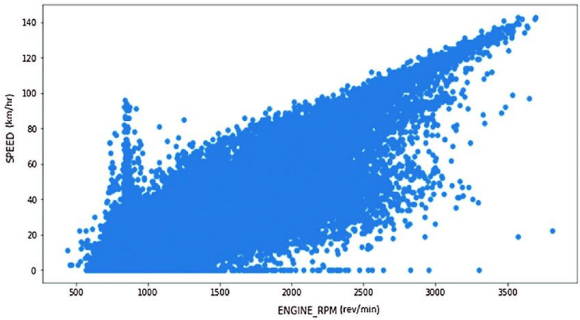
Johnny Nguyen and Andrew Zhao

Motivation

- **Discovering trends** is a major goal in exploratory data analysis (EDA)

- Datasets are exponentially growing in size and dimension

- Scatterplotting **large datasets** has the potential to be both
  - Hard to interpret (clutter, overlapping points, overwhelming data)
  - Computationally intense



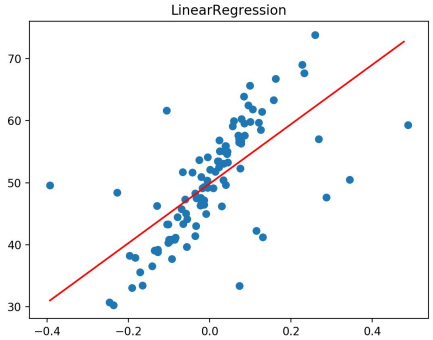
- Sampling can decrease both visual clutter and compute while preserving trends!

Questions

1. Do smaller **sampling rates** effectively preserves trends compare to larger sampling rates?

2. What **sampling methods** are better at trend preservation than others?

Background/Set up the Bit



- Numerical methods don't always correspond with human perception, which is the root of exploratory data analysis (Wang, 2019)
  - Many trend preservation analysis relies on statistical line of best fit, which can miss outliers or more complex relationships

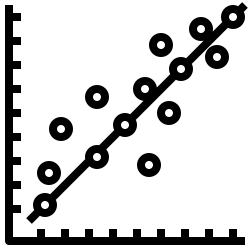
- There has been one known user study of effectiveness of sampling techniques on outlier identification, shape examination, and density detection, **but not trend analysis!** (Jun Yuan 2020)

- These studies chose to exclude trend analysis because the sampling methods didn't explicitly design for trend preservation

BitFlip

We propose a mixed methods approach to test the effectiveness of various sampling techniques for the previously unstudied task of exploratory scatterplot trend analysis

+



Quantitative analysis

High-Level Overview of Work

1. Sampling Methods and Datasets
2. Prestudy (How many points to sample per feature relationship?)
3. Hypotheses
    a. Corresponding Tasks
4. Preliminary Results

User Study Design - Sampling Methods

(1) Random Sampling

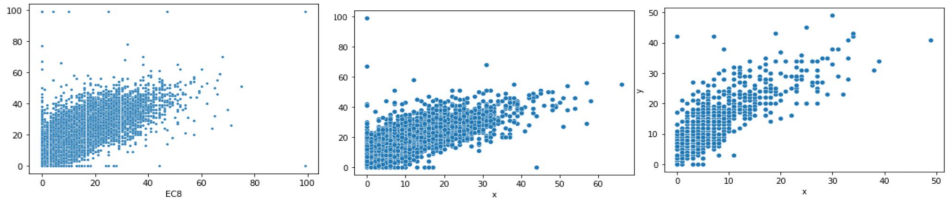
(2) Density-biased Sampling

(3) Blue Noise Sampling

(4) Outlier-biased Density-based Sampling

(5) Farthest Point Sampling

(6) Outlier-biased Blue Noise Sampling



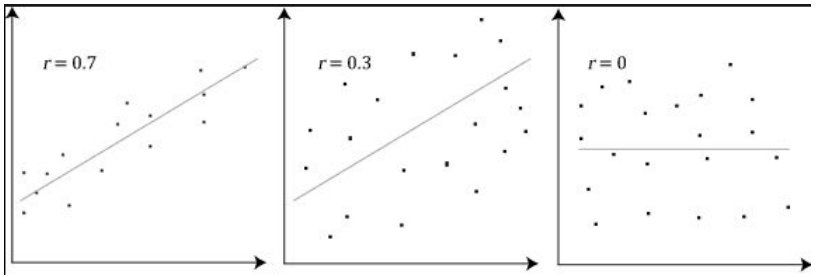
Original Blue Noise Random

Dataset Curation

Three Diverse, Large Datasets

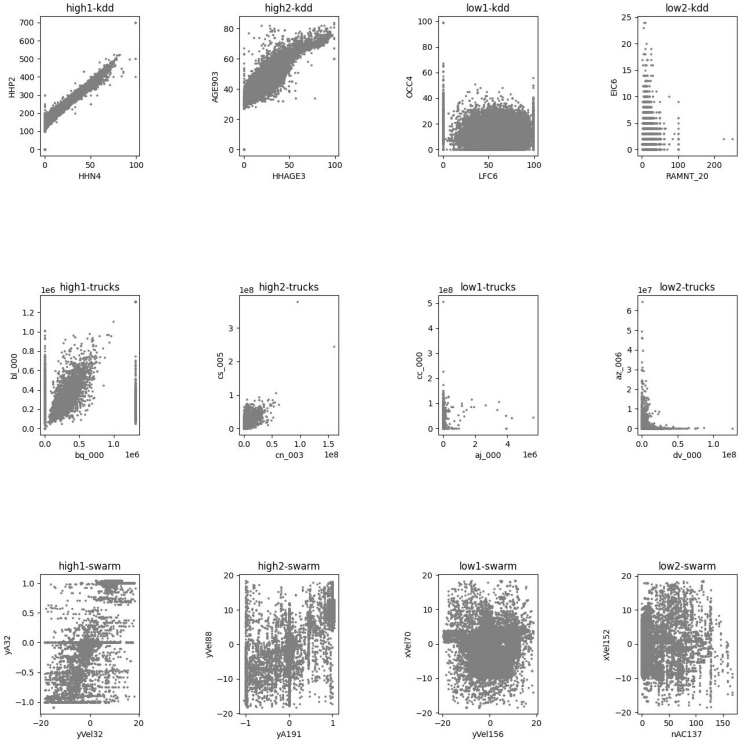
- (1) KDD Cup 1998 Data: 191779 data points, 481 features
- (2) APS Failure at Scania Trucks: 60000 data points, 171 features
- (3) Swarm Behaviour: 24017 data points, 2400 features

Randomly sampled pairs of features with **high** (>0.7) and **low** (<0.2) Pearson's correlation coefficient mapping to high and low linear correlation



Ultimately chose 2 scatterplots for each dataset with high linearity and 2 scatterplots each with low linearity (12 total feature pairs)

Dataset Curation



User Study Design - Pre-study

Motivation: Different feature pairs may have drastically different sampled % to preserve general visual similarity

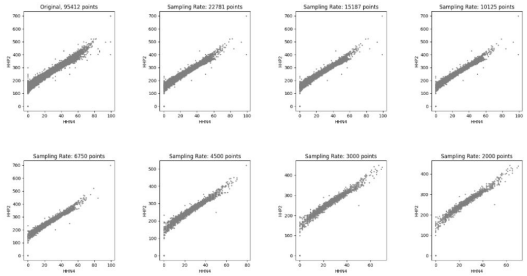
Goal: Find reasonable baseline sampled rate per feature pair

Used random sampling to get 2000, 2000*1.5, 2000*1.5^2… 2000*1.5^n points

Did not surpass <50% dataset size

Recruited 8 survey participants (Average age: ~22, Male%: 50%)

Please select the scatterplot that has the smallest number of points while being perceptually similar to the original scatterplot



Original

Option 2 (second highest points)

Option 3 (third highest points)

Option 4 (fourth highest points)

Option 5 (fifth highest points)

Option 6 (sixth highest points)

Option 7 (seventh highest points)

Option 8 (eighth highest points)

Pre-study Results

Feature Set	Minimum Preserved Point Count
HC-KDD-1	2000
HC-KDD-2	2000
HC-Swarm-1	4500
HC-Swarm-2	2000
HC-Trucks-1	3000
HC-Trucks-2	4500
LC-KDD-1	22781
LC-KDD-2	15187
LC-Swarm-1	3000
LC-Swarm-2	3000
LC-Trucks-1	15187
LC-Trucks-2	15187

Table 1: Feature sets and their resulting optimal sampled points using random sampling. We label based on high or low correlation (HC/LC) and dataset.

Hypotheses

H1. All sampling techniques will preserve general linear trends for sufficiently large datasets

H2. The smaller the sample, the less time it will take for users to determine a linear trend

H3. All other sampling techniques are better than random sampling at preserving linear trends

H4. Blue-noise sampling is better than other sampling techniques at preserving linear trends

Tasks

User Study Tasks

Task 1: Find minimum sampling rate to preserve perceptual linear correlation by random sampling 2000, 2000*1.5, 2000*1.5^2… 2000*1.5^n points, asking participants about the presence of linear correlation

Task 2: Detect visual presence of linear trends with different sampling methods using pre-study sample rates

Task 3: Determine visual preservation of line of best fit under different sampling methods by overlaying 6 different candidate lines to the scatterplot and asking participants the best fit. We use the users' chosen line of best fit for the unsampled pair as ground-truth

Qualtrics Survey Formal Study

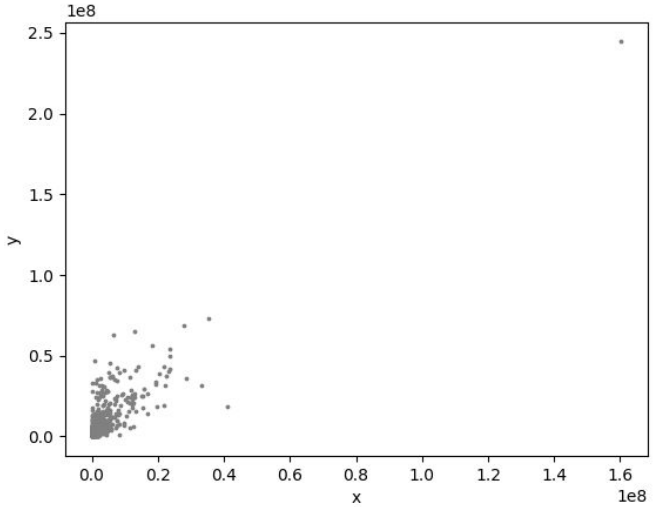
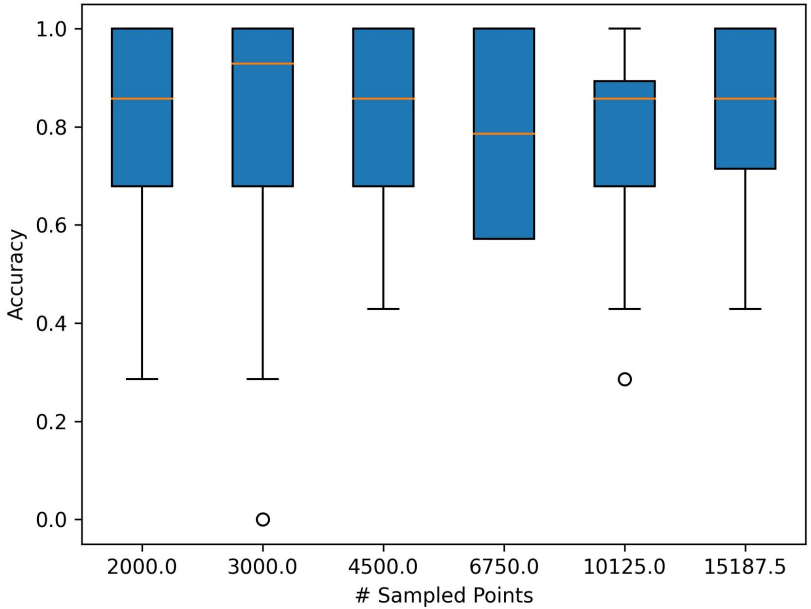


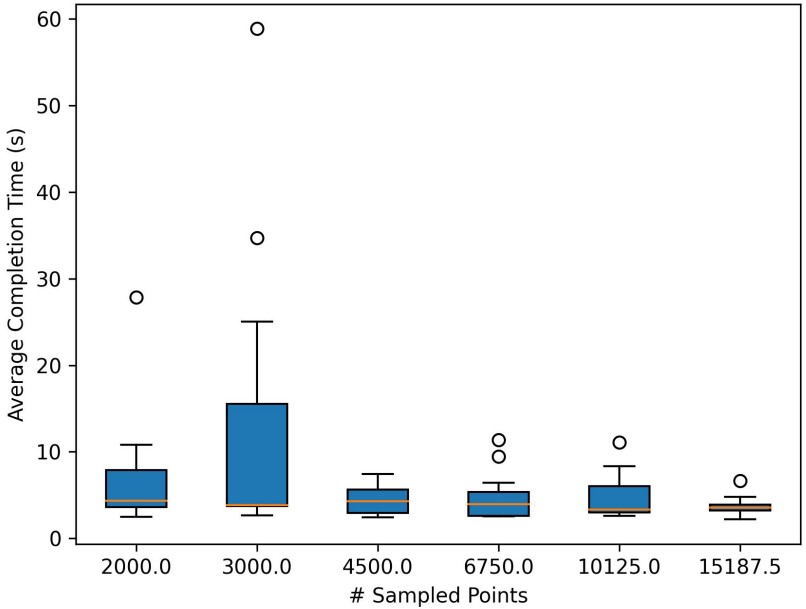
Image Source: 35.png

Question: Is the variable on the x-axis linearly correlated with the variable on the y-axis?

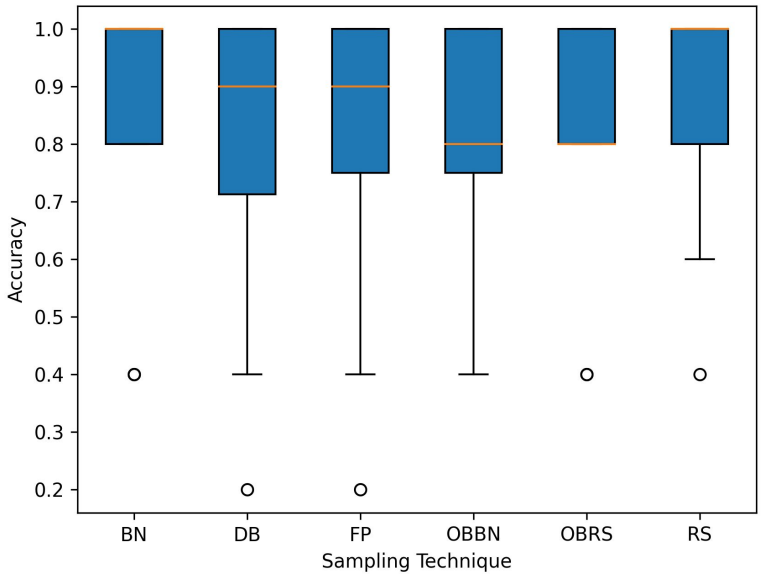
- ◯ Yes
- ◯ No

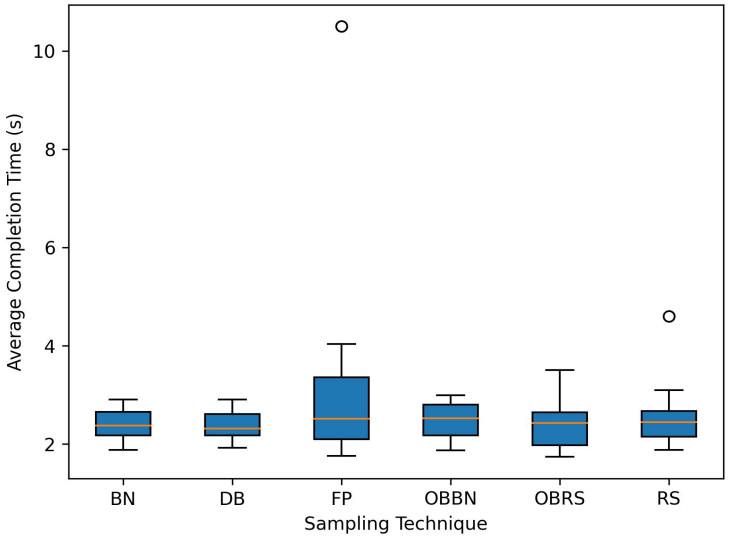
Results - Sampling Rates (Task 1)





Results - Sampling Rates (Task 2)





Results - Findings

No sampling rate is statistically better than another time-wise or accuracy-wise -> **H2 rejected**, **H1 partially accepted**

Blue Noise Sampling is better than Random Sampling accuracy-wise ($p = 0.04$) -> **H3 and H4 partially accepted**

Discussion and Conclusion

- **Sampling is a useful technique** for Scatterplot EDA that preserves trends up to 2000 points (2-8%) and reduces computation for interactive visualization

- We recommend Blue Noise sampling as a solid baseline, although further testing can be done with diverse datasets and novel sampling methods

- We will finish up work on Task 3, which will give insight on the degree of visual trend preservation of each sampling method

Bibliography

Wang Y, Wang Z, Liu T, Correll M, Cheng Z, Deussen O, Sedlmair M. Improving the Robustness of Scagnostics. IEEE Trans Vis Comput Graph. 2020 Jan;26(1):759-769. doi: 10.1109/TVCG.2019.2934796. Epub 2019 Aug 20. PMID: 31443018.

Jun Yuan, Shouxing Xiang, Jiazhi Xia, Lingyun Yu, and Shixia Liu. 2020. Evaluation of Sampling Methods for Scatterplots. https://doi.org/10.48550/ARXIV.2007.14666