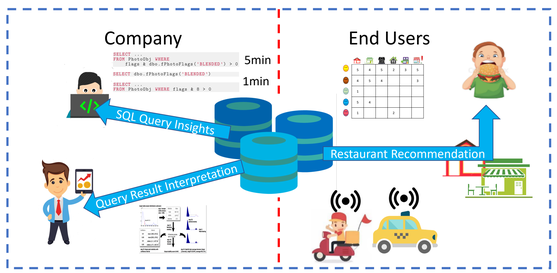
Facilitating Data Exploration, Query Composition and Analysis, and Query Answer Analysis

Zainab Zolaktaf
Email : zolaktaf@cs.ubc.ca
Department of Computer Science, University of British Columbia

Facilitating User Interaction with Data

Abstract

In many domains, users interact with data that is stored in large, and often structured, data sources. At a high level, this interaction involves three phases: 1. a **data exploration phase**, where the user explores the data to understand it, 2. a **query composition phase**, where the user writes and executes a query, 3. a **query answer analysis phase**, where the user analyses query answers produced by the system. My PhD research focuses on facilitating user interaction in these phases [1]. Specifically, for exploration, I study models that can find and suggest novel and relevant data items. For composing questions, I study models that can predict properties of a new query before it is submitted to the system. This includes insights such as the time it will take for the system to generate answers. To help them analyze the query results, we devise efficient mechanisms that estimate properties of the answers, such as the answer range. Overall, the solutions developed in my work aim to increase the efficiency and decision quality of users.



Research Questions

- Facilitating data exploration [3,4]
  - How do existing recommendation models perform with regard to exploration and how can they be modified to facilitate data exploration more rigorously?
- Facilitating query composition and analysis [5]
  - How can we help users better tune, optimize, and analyze SQL queries?
- Facilitating query answer analysis [2]
  - After a query has been submitted, how can we help the user understand and interpret the query answers?

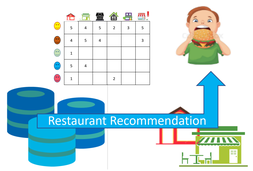
References

1. Zolaktaf, Zainab. "Facilitating User Interaction With Data." *PhD@ VLDB*. 2017.
2. Zolaktaf, Zainab, Jian Xu, and Rachel Pottinger. "Extracting aggregate answer statistics for integration." 18th International Conference on Extending Database Technology (EDBT), 2015.
3. Zolaktaf, Zainab, Reza Babanezhad, and Rachel Pottinger. "A Generic Top-N Recommendation Framework For Trading-off Accuracy, Novelty, and Coverage." *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 2018.
4. Zolaktaf, Zainab, Omar AlOmeir, and Rachel Pottinger. "Bridging the Gap Between User-centric and Offline Evaluation of Personalized Recommendation Systems." *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM, 2018.
5. Facilitating SQL query composition and analysis, submitted, 2019.

Facilitating Data Exploration

Problem Overview

- **Given a log of prior interaction data, e.g., user ratings on items, recommend a set of N unseen items to each user such that recommendations are accurate, novel, and Improve overall item-space coverage**



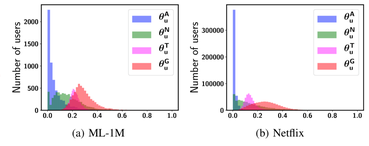
Solution Overview

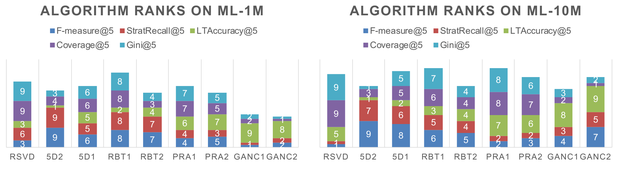
- **Automatically infer a long-tail novelty preference Θ_u for each user**
  - **Propose 4 different models, from simple heuristics like count of number of ratings (θ_u^A) to learning-based model (θ_u^G)**
- **Integrate Θ_u into a generic re-ranking framework to provide customized balance between coverage and accuracy**
- **Our framework, GANC, is generic**
  - **we can plug-in any base recommender, e.g., Most Popular (Pop)**

Selected Results

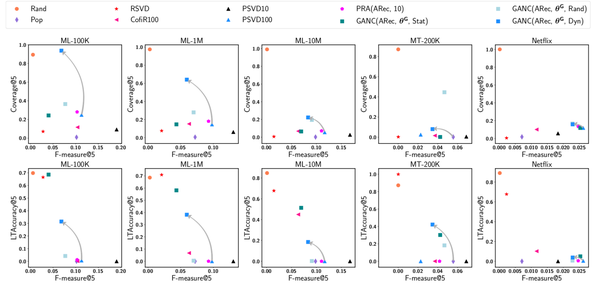
- 5 datasets
  - MovieLens (ML-100K, ML-1M, ML-10M), MovieTweetings (MT-200K), and Netfllix
  - ML-100k and ML-1M are dense datasets, while others are sparse

Histogram of 4 different long-tail novelty preference models.





Rating prediction in dense settings. RSVD is base recommender. Lower height is better and corresponds to better rank among metrics. GANC outperforms RSVD in 4 metrics, including accuracy.



Ranking prediction . Comparing Accuracy vs Coverage vs Novelty. The head of the arrow shows our algorithm, while the bottom shows the underlying accuracy recommender. Our algorithm increases coverage and novelty while maintaining reasonable levels of accuracy.

Facilitating Query Composition and Analysis

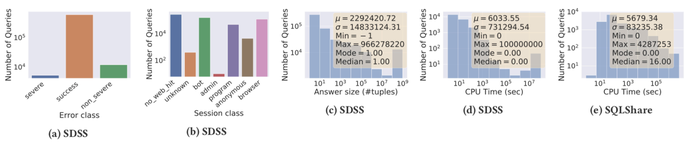
Problem Overview

- **Given a workload W consisting of prior SQL queries, provide insights about the performance of a SQL query Q_*, prior to submitting it to database.**
- **Do not rely on database statistics or query execution plans**
- **Problem settings**
  - **Homogeneous instance: Q_* and the queries in W are posed to the same database instance**
  - **Homogeneous schema: Q_* and the queries in W are posed to different database instances with the same schema in the same DBMS**
  - **Heterogeneous schema: : Q_* and the queries in W are posed to different databases with different schemas that run in the same DBMS**

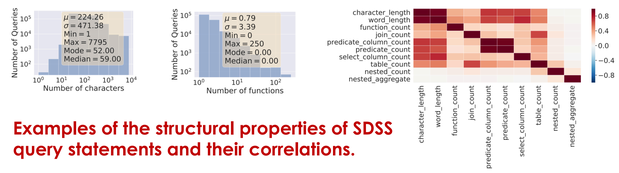
Insights:
- SQL error prediction
- CPU time prediction
- Answer size prediction
- Session Classification
  - Bot/Browser/Program/...

Workload Analysis

- 2 Query Workloads
  - SDSS (618,053l queries), used for homogeneous instance
  - SQLShare (26,728 queries), used for other problem settings



Label distributions for classification and regression problems.



Examples of the structural properties of SDSS query statements and their correlations.

Selected Results

- **Compared 7 baselines**
  - **Simple baselines**
    - **mfreq (predicts majority cla**
    - **median (predicts median)**
  - **ctfidf and wtfidf**
    - **Bag-of-ngrams and tfidf**
  - **ccnn and wcnn**
    - **Shallow convolutional neural network**
  - **clstm and wlstm**
    - **3-layer Long Short-term memory**
- **Objective function**
  - **Huber loss for regression**
  - **Cross-entropy for classification**

Model	v	p	Loss	F_{non_severe}	$F_{success}$	F_{severe}	Accuracy
mfreq	-	-	0.5951	0.0000	0.9863	0.0000	0.9730
ctfidf	500000	1500000	0.5860	0.7131	0.9888	0.0053	0.9778
ccnn	159	17403	0.1106	**0.7961**	**0.9897**	0.1669	**0.9797**
clstm	159	1944003	0.0830	0.6922	0.9893	**0.2206**	0.9786
wtfidf	500000	1500000	0.5836	0.6546	0.9885	0.0620	0.9773
wcnn	85942	8597953	0.1006	0.7441	0.9894	0.2006	0.9790
wlstm	85942	10522303	**0.0691**	0.6971	0.9887	0.0018	0.9776

Error classification on SDSS.

Model	v	p	Loss	p	Loss
			CPU Time		Answer Size
median	-	-	0.0675	-	1.6357
ctfidf	500000	500000	0.0668	500000	1.0400
ccnn	159	16801	0.0471	16801	**0.7517**
clstm	159	1943401	0.0452	529651	0.7678
wtfidf	500000	500000	0.0668	500000	1.0922
wcnn	85942	8595101	**0.0441**	8595101	0.8472
wlstm	85942	10521701	0.0443	9107951	0.8256

Query CPU time and Answer size prediction (right), in Homogeneous instance (SDSS).

MSE of answer size prediction by structural properties in Homogeneous instance (SDSS).

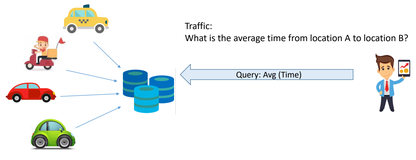
	Homogeneous Schema			Heterogeneous Schema	
Model	v	p	Loss	p	Loss
median	-	-	2.0049	-	2.1616
ctfidf	500000	500000	0.4742	500000	1.2360
ccnn	101	11001	**0.4625**	10901	**1.1547**
clstm	101	1937601	0.7935	1937501	1.6046
wtfidf	500000	500000	0.4898	500000	1.9702
wcnn	14843	1485201	0.5230	1395901	2.0416
wlstm	14843	3411801	0.8081	3322501	1.8546

Query CPU time prediction (SQLShare).

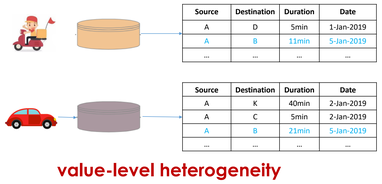
Facilitating Query Answer Analysis

Problem Overview

- **Answering aggregate queries in integration contexts**



- **Aggregate queries**
  - **group set of data values and calculate informative statistics Sum, Average, …**
  - **need to combine sets of data segmented across sources in integration contexts**



Source	Destination	Duration	Date
A	D	5min	1-Jan-2019
A	B	11min	5-Jan-2019
...	...	...	...

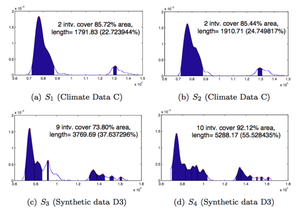
Source	Destination	Duration	Date
A	K	40min	2-Jan-2019
A	C	5min	2-Jan-2019
A	B	21min	5-Jan-2019
...	...	...	...

value-level heterogeneity

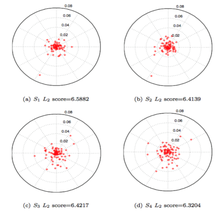
Solution Overview

- **Define aggregate answers as a distribution of viable answers**
- **Provide summary statistics for the viable answer distribution**
  - **Key point statistics**
  - **High coverage intervals**
  - **Stability**
- **Provide algorithms for the efficient extraction of above statistics**

Selected Results



By returning dense areas, the high coverage intervals cover a small percentage of the range of data.



Closer to the center, and the denser the points, the more stable the query result.