Similarity Metrics for SQL Query Clustering

G. Kul, D. T. A. Luong, T. Xie, V. Chandola, O. Kennedy and S. Upadhyaya

IEEE Transactions On Knowledge And Data Engineering, December 2018

Submitted By:

Aiman Siddiqua – 2K18/MC/008
Apoorva – 2K18/MC/019

Analysis Of Paper

Database Management System (MC-302)

ABSTRACT

- Database access logs are the starting point for many forms of database administration, from database performance tuning, to security auditing, to benchmark design, etc.
- Clustering is the first step towards understanding the massive query logs, for an analyst to extract broad patterns.
- Most clustering methods use pairwise similarity, which can be difficult for SQL Queries, especially without underlying database schema or data.
- In the paper, query similarity is computed relying only on the query structure.
- Three query similarity heuristics were applied to query clustering.
- To improve the accuracy of the heuristic, a generic feature engineering method was applied using classical query rewrites to standardize query structure.
- The proposed strategy results in a significant improvement in the performance of all three similarity heuristics.

Summary of PocketData Dataset and Google+

	Pocket Dataset	Google+
All queries	45,090,798	2,340,625
SELECT queries	33,470,310	1,352,202
Distinct query strings	34,977	135

Summary of UB Exam Dataset

Year	2014	2015
Total number of queries	117	60
Number of syntactically correct queries	110	51
Number of distinct query strings	110	51
Number of queries with score > 50%	62	40

Summary of IIT Bombay Dataset

Question	Total number of queries	Number of parsable queries	Number of distinct query strings
1	55	54	4
2	57	57	10
3	71	71	66
4	78	78	51
5	72	72	67
6	61	61	11
7	77	66	61
8	79	73	64
9	80	77	70
10	74	74	52
11	69	69	31
12	70	60	22
13	72	70	68
14	67	52	52

INTRODUCTION

The three metrics were evaluated on the following datasets:

- A large set of student authored queries released by IIT Bombay.
- A smaller set of student queries gathered at the University at Buffalo, and released as part of this publication.
- SQL logs that capture all activities on 11 Android phones for a period of one month.

The paper accomplishes the following, A survey of existing SQL query similarity metrics, an evaluation of these metrics on multiple query logs, and applying query standardization.

LITERATURE REVIEW

Paper Title	Motivation	Features	Distance Function
Agrawal et al. (2006)	Q. Reply Importance	Schema, rules	Cosine Similarity
Giacometti et al. (2009)	Q. Recommendation	Difference pairs	Difference Query
Yang et al. (2009)	Q. Recommendation	Selection/Join, Projection	Jaccard coefficient on the graph edges
Chatzopoulou et al. (2011)	Q. Recommendation	Syntactic element frequency	Jaccard coefficient and cosine similarity
Aouiche et al. (2006)	View Selection	Selection/Join, Group-by	Hamming distance
Aligon et al. (2014)	Session Similarity	Selection/Join, Projection, Group-by	Jaccard Coefficient
Makiyama et al. (2016)	Workload Analysis	Term frequency of projection, selection/join, from, group-by, order-by	Cosine Similarity

REGULARIZATION RULES

Canonicalize Names and Aliases:
First regularization step attempts to create a canonical naming scheme for both attributes and tables.

Syntax Desugaring:
SQL's redundant syntactic sugar can be removed by basic pattern replacement. Such as x BETWEEN (a,b) can be replaced by a<x AND x<b.

EXISTS Standardization:
All SQL queries containing one of (EXISTS, IN, ANY and ALL) can be rewritten using only EXISTS.

DNF Normalization:
We normalize all boolean-valued expressions by converting them to disjunctive normal form(DNF).

Commutative Operator Ordering:
Standardizing the order of operands in an expression using commutative and associative operators by defining a canonical order of operands.

Nested Query De-correlation:
A common database optimization that converts some EXISTS predicates into joins for more efficient evaluation.

OR-UNION Transform:
We use a regularization transformation that exploits the relationship between OR and UNION.

Union Pull-Out:
Since the prior transformation may introduce UNION operator in nested subqueries, we push selection predicates down into the union as well.

EXPERIMENTAL METHODS

CLUSTERING VALIDATION MEASURES

They are used to validate the quality of a labeled dataset by estimating two quantities:
1. the degree of tightness of observations in the same label group and
2. the degree of separations between observations in different label groups.

SIMILARITY METRICS

We use the similarity metrics as defined in Aouiche et al. (2006), Aligon et al. (2014) and Makiyama et al. (2016).

SILHOUETTE COEFFICIENT

It is a measure of how similar a data point is to its own cluster in comparison to other clusters.

For a data point i it is measured as
$b(i) - a(i)/max(a(i), b(i))$

a(i) is the average distance from i to all other data points in the same cluster
b(i) is the average distance from i to all other data points in the closest neighboring cluster.

BETACV MEASURE

It is the ratio of the total mean of intra-cluster distance to the total mean of inter-cluster distance.

The smaller the value of BetaCV, the better the similarity metric characterizes the cluster partition of queries on average.

DUNN INDEX

It is defined as the ratio between minimum distance between query pairs from different clusters and the maximum distance between query pairs from the same cluster.

Higher values of Dunn Index indicate better worst-case performance of the clustering metric.

RESULTS

- The first experiment was to evaluate which similarity metric can best capture the task performed by each query.
- The black columns in Fig. 1 shows a comparison of three similarity metrics using each of the three quality measures.
- As can be seen in Fig. 1, Aligon seems to work the best for the workloads.
- Next experiment evaluated the effectiveness of regularization by applying it to each of the three metrics.
- Fig. 1 shows the values of three validation measures both with and without regularization.

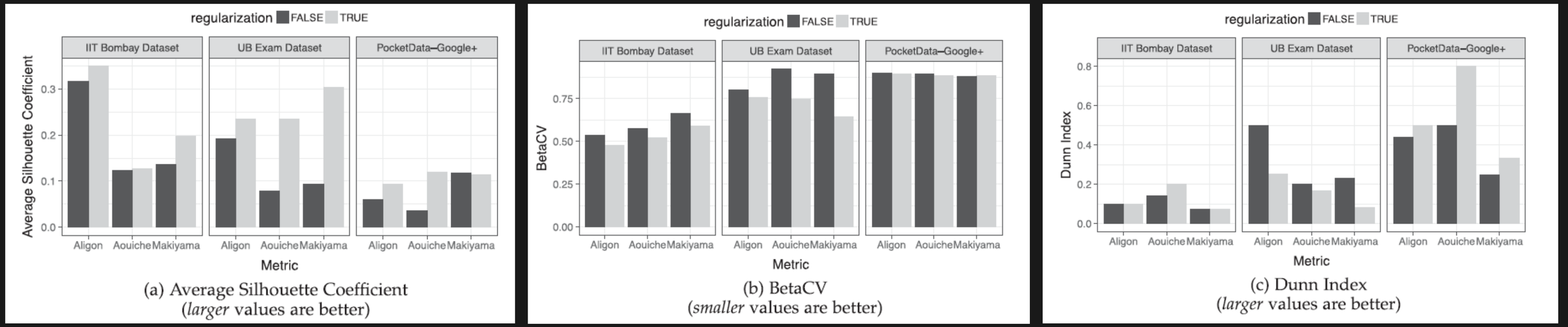
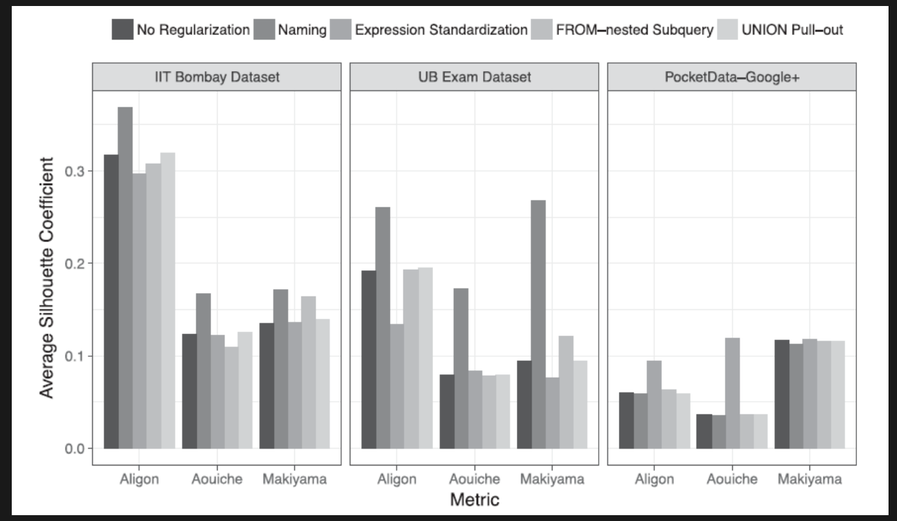
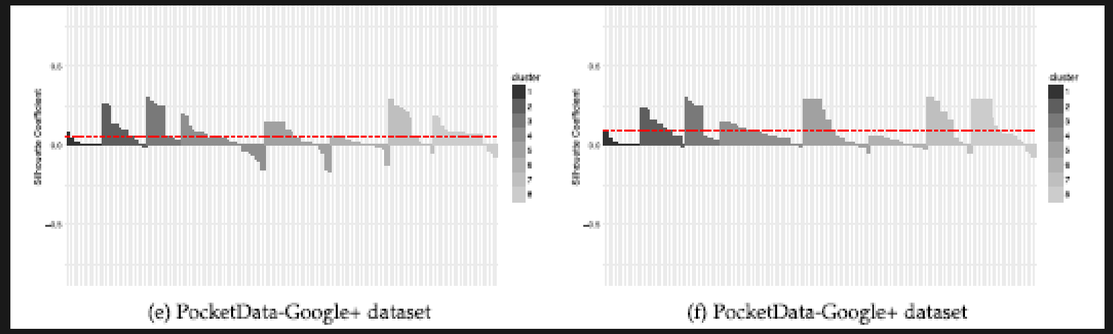


Fig. 1

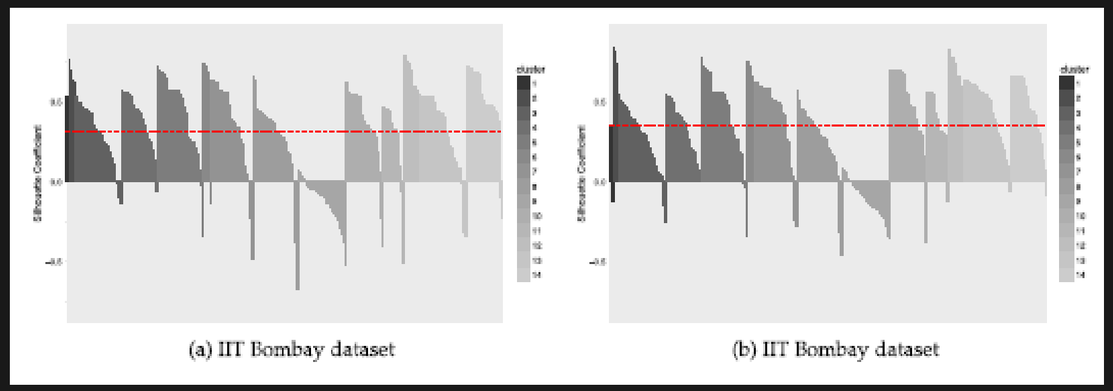
CONCLUSIONS

- Finally, the Silhouette Coefficient for Aligon Metric was evaluated.
- From the graphs, the overall effects of regularization could be seen.
- The feature engineering steps provided an improvement across the board because they addressed the error reasons that were identified.

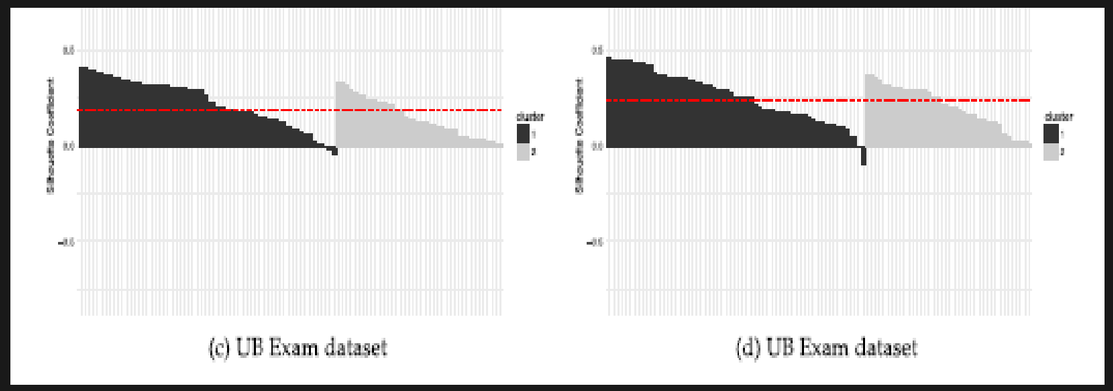




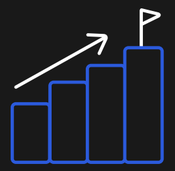
(e) PocketData-Google+ dataset (f) PocketData-Google+ dataset



(a) IIT Bombay dataset (b) IIT Bombay dataset



(c) UB Exam dataset (d) UB Exam dataset



APPLICATION SCENARIOS

1.

IMPROVE DATABASE PERFORMANCE

By creating views of the most frequent complex queries, the database performance can be improved substantially.

2.

DETECT OUTLIERS

By identifying the query clusters and partitioning the queries, all outliers can be detected.

3.

RESEARCH WORK

Using Query clusters and partitioning, the properties of the SQL query dataset and its schema can be reasearched.

THANK YOU !