Miller, Adam
A20418016

Patel, Panthi
A20417596

Kartheiser, Austin
A20436635

Summarizing Provenance of Aggregate Query Results in Relational Databases

Panthi Patel
Illinois Institute of Technology
ppatel122@hawk.iit.edu

Adam Miller
Illinois Institute of Technology
amiller13@hawk.iit.edu

Austin Kartheiser
Illinois Institute of Technology
akartheiser@hawk.iit.edu

ABSTRACT
In this paper we will summarize key points and major takeaways from the paper *Summarizing Provenance of Aggregate Query Results in Relational Databases*. We will also provide a critique of the paper.

KEYWORDS
Provenance, Query, Summary, Impact, Aggregate

1. Introduction
Provenance is the process by which data is gathered or modified. It is any information relating to the data's origin. In data systems, this focuses primarily on the sources that contributed to the results of a given query. However, this information can be large making its digestion difficult for curious users. The paper seeks to alleviate this burden, by presenting methods for efficiently constructing provenance information in a usable format.

Before delving into the meat of the research, it is important to lay out a foundation for the methods utilized. The first thing discussed are basic summaries, which are composed of a set of rules that describe the data provenance. These follow the same schema as the relation that was queried, with stars in certain columns that function as wild-card values. These wild cards match every tuple in the relation, while the other values match only with specific tuples. Moreover, rules each have a score, which is defined as the product of their weight and coverage. These values are simply calculated; weight is the number of non-star values in the rule, and coverage is the number of tuples that are included by the rule. Scores determine how relevant, or "interesting," a rule is to the overall summary.

The paper discusses two main summarization types: impact summaries and comparative summaries. The former is defined similarly to a basic summary as above, however, it also incorporates an "impact" factor, which will be discussed later. The latter considers the provenance of two or more queries, and in turn helps users determine similar tuples that contributed to the original query answers. This summarization rule, also, will be examined later on.

2. Contributions
There are four main contributions of this paper. The first is to define provenance summaries, comparative summaries, and impact summaries. Score functions are formalized for these summaries and the summarization problems are presented in order to find the best provenance summaries. The second is to demonstrate efficient algorithms to produce summaries with high scores. Third, a user interface is introduced. And lastly experiments are conducted along with a user survey to demonstrate the effectiveness of the proposed ideas and solutions.

3. Background
In this section we will establish some data summarization rules.

3.1 Cover and Count
Further concepts that are necessary for comprehension of this paper are cover and count. Cover was addressed earlier and is synonymous with coverage. It is the set of tuples that are summarized by a rule. Count is merely the size of the cover set.

3.2 Marginal Cover & Marginal Count
These parameters are similar to cover and count, however they are calculated for each rule compared to all that came before it. An example would be if the summary to this point included two tuples, A and B. A new rule that adds C to the cover would have that as its marginal cover. Additionally, the marginal count would be 1, since there is 1 tuple in the marginal cover of this rule.

3.3 Score Function & Marginal Score
The score function is the sum of all products of marginal counts and the weight of each rule.

Marginal score, much like the other marginal quantities, is how much a rule contributes to the total score of a summary.

3.4 A Summary

A summary over a relation R is a set of all rules with maximal score that explains the provenance of a relation.

3.5 Best Rule Set

Given this definition, finding a summary with a specific number of rules and maximal score is shown in the paper to be an NP-hard problem. Hence, approximations are necessary to accomplish a good summary in a reasonable amount of time. The Best Rule Set algorithm (BRS) uses impact scores to generate the best rule to include at the time. In order to keep runtime down, rules that obviously will not be included in the final set get pruned before consideration. It is important to note that the BRS algorithm does not produce optimal solutions, opting rather to generate them in an efficient timeframe.

4. Formalize Provenance Summarization Problem

4.1 Impact Score

Impact summaries summarize the provenance of a tuple in the form of an aggregate query. In the paper, the New Score Function, *IScore* (Impact Score), is defined for a list of summarization rules S. The impact score of S is the maximum impact score between every possible list that contains the rules in the set. The IScore function considers the impact in R^a of the provenance tuples covered by rules in S on a tuple a in the query result. The impact factor in the equation measures the impact of a rule using sensitivity analysis which is a technique that measures the sensitivity of a query with and without the tuples in the database.

4.2 Comparative Summarization Problem and CScore

Comparative summaries are used to best summarize similarities between the sets of provenance R^a and $R^{a'}$ for $a, a' \in Q(D)$. The *CScore* function measures how well rules $s_i \in S$ summarize R^a and $R^{a'}$. It is calculated using the marginal number of tuple-pairs from R^a and $R^{a'}$ that are covered by s_i and the number of non-wildcard values in s_i that do not appear in a and a'. It is used to give a summary that covers both R^a and $R^{a'}$ in a balanced way.

5. Summarization Algorithms

5.1 Impact Provenance Summarization Algorithm

The paper introduced an algorithm, impact provenance summarization algorithm (IPS), for finding impact summaries. It is an extension of the BRS algorithm that utilizes the new impact score function mentioned previously to summarize the provenance queries. It works by taking in a database, a relation, a query, an answer, and value k and it computes an impact summary with k rules. It should be noted that some rules are pruned if they cannot beat the rules from previous iterations. For each rule a marginal score is calculated and the best summary is returned.

5.2 Comparative Provenance Summarization Algorithm

Another algorithm was introduced to find comparative summaries, the comparative provenance summarization algorithm (CPS). Similar to IPS it extends the BRS algorithm. It takes in a database (D), a query (Q), a relation, a_1, $a_2 \in$ Q(D), and a value k and it returns a comparative summary with k rules. CPS calculates the provenance of a_1, a_2 and uses that to find the top k marginal rules.

5.3 Comparing IPS, BRS, and CPS

IPS varies from BRS in the sense that IPS generates provance and also takes into consideration the impact values while it is computing the marginal scores. While BRS also prunes rules, IPS utilizes impact values to prune faster in the case when the numerical attribute that is aggregated has skewed data. CPS is different from BRS in the way that it generates provence and computes the score function while taking into account pairs of tuples.

5.4 Computing Impact and Contingency Sets

Computing the impact of tuples is NP-hard and computing the impact of provenance tuples (t) on a query (Q) requires running Q^a (D\{t}) for every tuple.

This can be very costly, so to avoid it IPS computes the impact of provence tuples and stores them in *I*. In the proposed experiments, the impact of tuples can be computed without running the query. Furthermore, for Aggregate-Select-Project (ASP) queries with a MIN or MAX, to compute the impact, a contingency set must be found first. This can be NP-hard for general SQL queries but it can be efficiently calculated for queries of the general form for aggregate queries. Essentially, while the overall problem of computing impacts and contingency sets is intractable, this specific subset of the problem, namely ASP and ASPJ queries, is polynomially viable.

5.5 Analysis of Summarization Algorithms

The runtime cost of IPS is at most O(k x n^2 x m) where n is the number of rows in the relation and m is the size of the relational schema. The cost of CPS is O(k x n^3 x m). Both algorithms have an approximation ratio α = 1 -1/e. Simply put, the score of the result summary is greater than the optimal score multiplied by α.

6. User Interface

The interface provides users with the required facilities needed to explore provenance summaries. All the rules are presented in a way to help the user discover insights about the data. The first thing the user must do is input a query, being able to review the results upon completion. They can then click on one or more tuples to see a provenance summary. There are two different options for the interface depending on the type of query. The first one is for impact summaries when the user clicks on one tuple (impact summaries for the results of queries with joins are also included). The second option is for comparative summaries when the user clicks on multiple tuples.

As far as impact summaries are concerned, the rules are displayed as rows in a table that have a quality measure. The quality measures are score, coverage, and impact. Score reflects the amount of contribution of each rule to the total score of the summary**,** as detailed within the original paper. Coverage is the fraction of provenance tuples that are covered by a particular rule. Impact describes the impact of a rule normalized by the total possible impact. It should be noted that impact and coverage are not marginal and have no dependencies on other rules in the summary. When an impact query is involved with multiple tables, the user is shown separate summaries for each table. When a rule is clicked on, the user can view how it relates to the rules in another summary. The connection of the rule to others is demonstrated by a line connecting the two. The thicker the line the stronger the connection.

When comparative summaries are concerned the interface does not change much. With comparative summaries the user can see how balanced a rule is when covering the provenance of two tuples. For both impact and comparative summaries, the user can click on them to view a set of super rules that are contained within a rule. There is also an option that can allow attributes to be collapsed into one to reduce any clutter visually.

7. Experimental Evaluation

The goal of the experiments was to explore three items. First was to evaluate the performance of the algorithms to show how they generate summaries in real time. The second was to compare provenance summaries with basic summaries and demonstrate how provenance summaries have higher quality in certain metrics. The third was to show the relevance of metrics such as impact and diversity for provence summarization. Because performance of summarization algorithms vary depending on the selected tuples, for each experiment, summaries were produced for a large number of randomly selected answers. Furthermore, the results of the summarization algorithms are dependent on the query so the experiments were run for different queries. When queries with different aggregate functions were involved in the analysis the same query was used and the aggregate function was replaced. The algorithms were implemented in python and the experiments were run on a machine with 16GB RAM and 3.3GHz Intel CPU that used PostgreSQL 9.4. Perm was the provenance generation component.

Three data sets were utilized. The first dataset is Global Legal Entity Identifier (GLEIs) which is a real world financial dataset. It has 250,000 tuples, 1 table, 12 attributes, and 5 queries. The second dataset is IMDB which has 2 tables, *Movies* and *Directors*.

They have 181,000 and 323,000 tuples respectively, 15 attributes, and 6 queries. The third dataset is TPC-H which has 60,000 - 3,000,000 tuples in its 8 tables, a total of 61 attributes, and 6 queries. This dataset was utilized to show queries with multiple types of aggregation and joins are handled. For the experiments the default value of the size of the provenance summaries (*k*) is 8. The maximum weight (*mw*) of a rule is defaulted at 4 and the default group-by attributes (*mq*) in the queries is 1.

In order to evaluate the performance of IPS, its runtime was compared to BRS while the *k* and the number of tuples were changing in experiments 1 and 2. In experiment 3 *mq* and *mw* were changed. In experiment 4 the runtime of IPS for queries that have joins was investigated and experiment 5 did the same but for CPS. The results of experiment 1, performed with TPC-H, showed that runtime increased with more tuples but IPS scales differently for each query. Overall the actual runtime was dependent on the query itself. Experiment 2 analyzed the effects of *k* and found that runtime increased when *k* was increased. The increase in the runtime was linear, which was to be expected, given the runtime complexity of the algorithms. Experiment 3 found that the runtime of IPS decreased as mq increased, and increasing mw increased runtime. Experiment 4 looked into the effects of join queries, with TPC-H, and found that IPS performs consistently for all queries when the data size is small. But for larger datasets the join queries outperform those without. Finally experiment 5 explores CPS and optimization of it with the IMDB dataset. The optimized CPS prunes rules that have low coverage on provenance sets. It was discovered that the optimized CPS runs 250% faster than when unoptimized.

Some additional experiments were performed to measure the quality of a summary. There were four metrics used to measure the summaries produced by the IPS and BRS algorithms: impact, coverage, weight, and diversity. Impact is the distribution of the impact values for the rules. Coverage is the total number of tuples that are covered by a summary. Weight is the total weight of the rules in the summary. Diversity is the number of unique values for the attributes in the summary. Experiment 6 displayed the impact and coverage of the rules that were generated by different algorithms. It was found that BRS generates rules with high coverage and high impact. IPS finds rules with high impact that are also informative. Experiment 7 looked into diversity and weight. It was concluded that IPS picks rules with higher weights when compared to BRS. IPS with SUM generates rules with higher weights when compared to IPS with Average. Overall IPS is optimal in terms of diversity when compared to BRS. It can also be concluded that the higher weight rules of IPS are more informative.

8. User Survey

The authors conducted a user survey of 17 Computer Science students that concluded that users preferred high impact rules over rules with high coverage and surprising rules over high impact rules. The survey attempted to minimize bias through the use of an immensely generic example IMDB dataset. The users were asked to rank various summaries on a scale of 1 (unimportant) to 5 (essential) and to explain the reasoning behind their rankings. The only visible problem with the conducted survey was that it was only taken by 17 individuals. With such a limited dataset, there is plenty of room for uncertainty regarding the validity of its results.

9. Critique

While the authors do a good job of conveying their research without bias, there do arise some issues with the methods presented in the paper. The first comes from the user survey presented at the end of the paper. The authors only conducted their user survey with 17 participants, which is a miniscule amount of data. The information that they gathered was useful in illustrating a point, but it is prone to skew from such a small amount of information. Moreover, there is little information on the demographics of the students, other than the fact that they studied computer science. No mention was given to if these students knew each other, their experience with database information, or their relations to the authors. This comes together to make the user survey seem like a valid source of information free from bias, and it would have been better to conduct the analysis with a larger set of participants or possibly even omit this section entirely.

The second critique to be made about the methods presented in the paper is the small number of datasets used in the experimental analysis. The authors only ran their tests on three, which may not paint an entirely accurate picture of the true efficiency of the presented algorithms. Perhaps a greater variety of selections would have been useful, for one of the databases only has one relation, and another only has two. A wider spread of databases with more tables or fewer attributes might provide useful information. It might even be feasible to pose more queries instead. Whatever the case, the selection appeared sparse, and more overall information to gather would have been better.

10. Conclusion

Overall, the paper looked into provenance and summarization research and implemented new summarization techniques, impact summaries, and comparative summaries that are ideal for users with little knowledge of provenance semantics. They also validated their techniques with several experiments and a small user survey in order to show that they present summarized information that is relevant to users. The paper provides a useful insight and set of tools for developing comprehensible provenance information for aggregate queries. It seems likely that this paper will be used in the future as a foundation upon which to build systems that can convey detailed and digestible provenance metadata.