Summarizing Provenance of Aggregate Query Results in Relational Databases

By: Panthi Patel, Adam Miller, Austin Kartheiser

Abstract/Summary

- Data provenance info is large
- Overwhelming to users
- Use summarization schemes to ease burden
- Experiments to show utility

What Is Provenance?

- Provenance is the process data went through
- “How did we get here?”
- Difficult to describe all data provenance
- Summaries would be useful
- This paper: summaries of aggregate provenance

Basic Summaries

- Set of k rules
- Summarize something “interesting”
- Rules have stars, only count real values
- Score is product of weight and coverage
- Only usable on single tables

Impact Summaries

- Similar to basic summaries
- Score based on coverage, weight, impact factor
- Impact factor can vary
  - Example in paper uses revenue
- Can involve multiple relations

ID	Gender	SumRev (M$)
a_1	Female	624.6
a_2	Male	3.374 B$

TABLE II: Results of Q_1

ID	Year	Genre	Rating	Gender	Country	Coverage	Impact
s_4	⋆	Action	8	Male	⋆	20.00	43.49
s_5	⋆	Comedy	7	Male	US	20.00	11.53
s_6	2015	⋆	⋆	Male	⋆	30.00	55.74

TABLE V: An impact summary

Comparative Summaries

- Summarize provenance tuples from two sets of tuples, unlike basic summaries that normally summarize a single set
- Cannot be reduced to basic summaries with union of two sets

Q_2 : SELECT Gender, Genre, COUNT(*) AS CountM
FROM MoviesDirectors GROUP BY Gender, Genre

ID	Gender	Genre	Count
a_3	Male	Drama	3
a_4	Male	Horror	2
a_5	Female	Comedy	2
a_6	Male	Comedy	2
a_7	Male	Action	3
a_8	Male	Comedy	2

TABLE III: Results of Q_2

ID	Rating	Director	Gender	Country	Cvg a_3	Cvg a_8
c_1	⋆	Steven Spielberg	Male	US	33.34	50.00
c_2	7	⋆	Male	US	66.67	50.00

TABLE VI: A comparative summary

Contributions

- Define provenance summaries, impact summaries, and comparative summaries for general form of aggregate SQL queries
- Formalize score function for these summaries and show summarization problems to find best provenance summaries
- Present algorithms that efficiently produce summaries with high score
- Show interaction methods and visualization techniques
- Conduct experiments and user surveys to show effectiveness of proposed solutions

Background : Data Summarization Rules

A summarization rule s over a relation R with schema R = {A1, ..., Am} is an n-ary tuple in which for every Ai∈R, s[Ai] ∈ Dom(Ai) ∪ {a wildcard value that matches every attribute value and allows the rule to summarize multiple tuples not in Dom(Ai)}.

Data Summarization Rules - Cover & Count

- [Cover & count] Given a relation R, a summarization rule s covers a tuple $r \in R$ denoted by $r \in s$ if for every Ai, r[Ai] = s[Ai] or s[Ai] = *.
  - Cover(s) is the set of tuples in R that are covered by s and Count(s) = |Cover(s)| is the number of tuples covered by s.
- [Marginal cover & marginal count] For a list of rules S = (s1, s2, ...) over relation R, MCover(si, S) is the marginal cover of si and is defined as the tuples that are covered by si and not any $sj \in S$ with $j < i$.
  - MCount(si,S) = |MCover(si, S)| is the marginal count of si.

Data Summarization Rules cont.

- [Score function and marginal score] The score of a list of rules S = (s1, s2, ...) is defined as follows: $\text{Score}(S) = \Sigma\ s_i \in S\ [\text{MCount}(s_i, S) \times \text{Weight}(s_i)]$.
  - Weight is a monotone function that returns a non-negative real number. For $\Sigma\ s_i \in S\ [\text{MCount}(s_i, S) \times \text{Weight}(s_i)]$ is its marginal score.
    - The weight function conveys how well a rule summarizes the non-wildcard values in a table.
- [Summary] A summary over a relation R is a set of summarization rules over R.
  - The score of a summary is the maximum score between all the possible lists containing the rules in the summary.

Data Summarization Rules - The Summarization Problem

Given a relation R and a fixed value k, the summarization problem is to find a summary S with |S|=k and maximum Score(S).

The summarization problem is NP-hard

Background: Best Rule Set

The **Best Rule Set** (BRS) is a greedy algorithm that finds a sub-optimal set of rules efficiently.

BRS has k steps starting with an empty rule set S. Each step it adds the best rule that maximizes the score function.

To find the rule s to add in each step, the algorithm computes the impact of every possible rule on the score function and prunes some of the rules.

Formalize provenance summarization rules : impact + impact Score

Impact is the relevance of a given rule to the output query.

Contingency set is the minimal set of tuples required to be removed from a query before it becomes sensitive to our tuple.

For each tuple, take the difference of the output and the output without that tuple's contingency set. Divide it by the size of the tuple's contingency set + 1.

Weight is the number of non-star attributes in the rule that are not in a tuple of the output query.

Impact Score is the Impact times the Weight.

Formalize provenance summarization rules : Comparative summarization Problem + CScore

Summarize the similarities between sets of provenance for two tuples in the output query.

Take the marginal number of tuple pairs per rule and multiply by Weight. Sum all these values to get CScore.

Summarize Algorithms: Impact Provenance algorithm

The algorithm follows the calculation from earlier. Returns the set of greedily selected rules.

Has subroutine called BestMarginalRule, which selects the next rule for inclusion in the set. It trims some suboptimal rules, then loops through the remainder and calculates their marginal impact score. The maximum is the one selected.

Summarize Algorithms: Comparative provence summarization

Returns a set of comparative provenance rules.

Has a subroutine called BestComparativeMRule, which performs the calculation described earlier. It computes the maximum marginal score and returns it.

Summarize Algorithms: IPS VS BRS VS CPS

IPS uses impact scores to prune faster than BRS. Use of impact scores also makes it more resilient to skewed data.

CPS differs from BRS by computing a score function for pairs of tuples.

Summarize Algorithms: Computing Impact and Contingency Set

Computing both sets is NP-hard for general queries.

Doing so for queries with a built-in aggregation, however, is not.

This allows for efficient computation of provenance on ASPJ queries.

Summarize Algorithms: Analysis of Summarization Problems

Running time of IPS is $O(k*n^2*m)$, where k is the number of rules, n is the size of the relation, and m is the size of the relational schema.

Running time of CPS is $O(k*n^3*m)$. However, this is when already given our two answer relations.

User Interface

- Provides users with facilities needed to explore provence summaries
- User writes a query and then can click on 1 or more tuple to see the provenance summary
- User gets impact summary if they click on a single tuple
- User get comparative summaries if they click on multiple tuples

User Interface - Impact Summaries

- Impact summaries are presented as rows in a table
- The quality measure for each rule is is one of the following
  - Score : contribution of each rule to summary's total score
  - Coverage: fraction of provence tuples that are covered by a rule
  - Impact : impact of a rule normalized by total possible impact
- If impact summary that involve multiple tables, the user can see separate summaries for each table

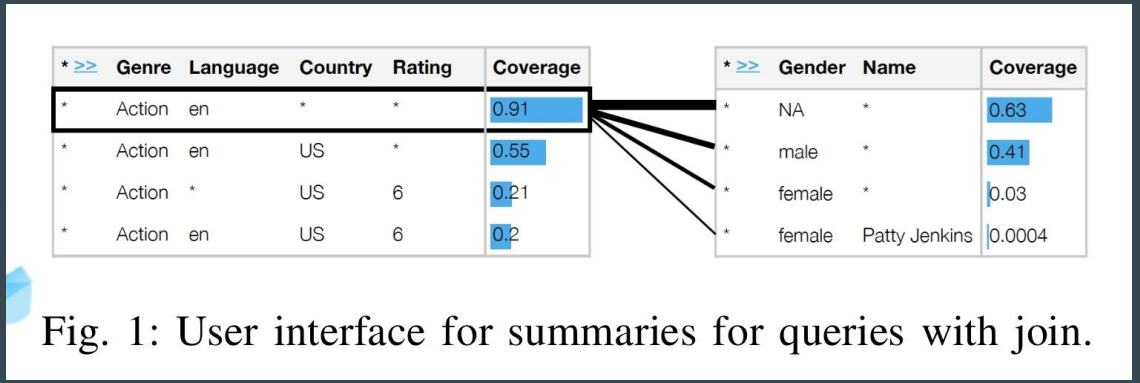


Fig. 1: User interface for summaries for queries with join.

User Interface - Comparative Summaries

- Similar to interface of impact summaries but user can also see how balanced a rule is in covering the provenance of two tuples
- Right click on any rule will expand it to see a set of super-rules contained within a rule

Experimental Evaluation

- produced summaries for a large number of randomly selected answers
- Ran different queries to analyze results of the summarization algorithm
- Algorithm was implemented in Python and ran on a machine with 3.3GHz and 16 GB RAM that used PostgresSQL 9.4
- Used three dataset : IMDB, Global Legal Entity Identifier, TPC-H
- To evaluate performance of IPS, compare its runtime with BRS

	GLEI	IMDB	TPC-H
$\|D\|$	250k	181k+323k	60k - 3m
N	1	2	8
m_D	12	15	61
q	5	6	6

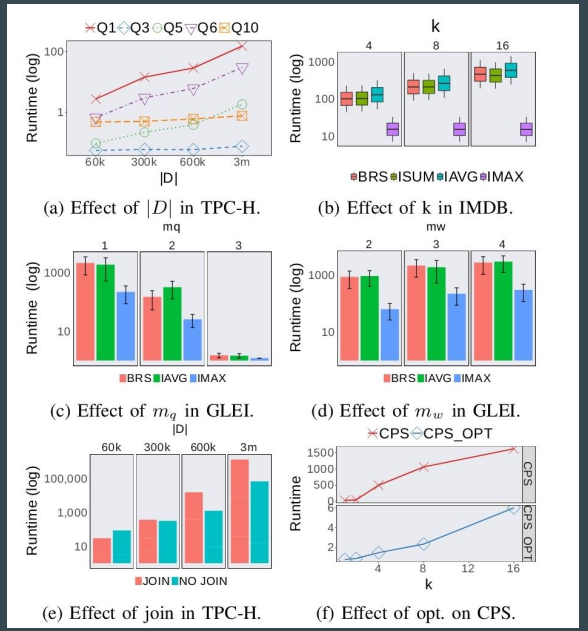
TABLE XI: Datasets characteristics.

Experimental Evaluation - Variables

- |D| : number of tuples
- N : number of tables
- Md: number of attributes
- Q : number of queries
- K : size of provence summaries (default = 8)
- Mw: max weight of rules (default = 4)
- Ma: number of group by attributes in queries (default = 1)

Experimental Evaluation - Findings

- Runtime of ISP increased with |D|
- But actual runtime is dependent of the query
- Runtime goes up if K is increased, increase is linear
- Runtime of ISP decreased id number of Mq (group by attributes) increased
- ISP performed similarly for all queries if data size was small
- Implementation of optimized CPS that prunes rules with low coverage runs 250% faster than unoptimized CPS



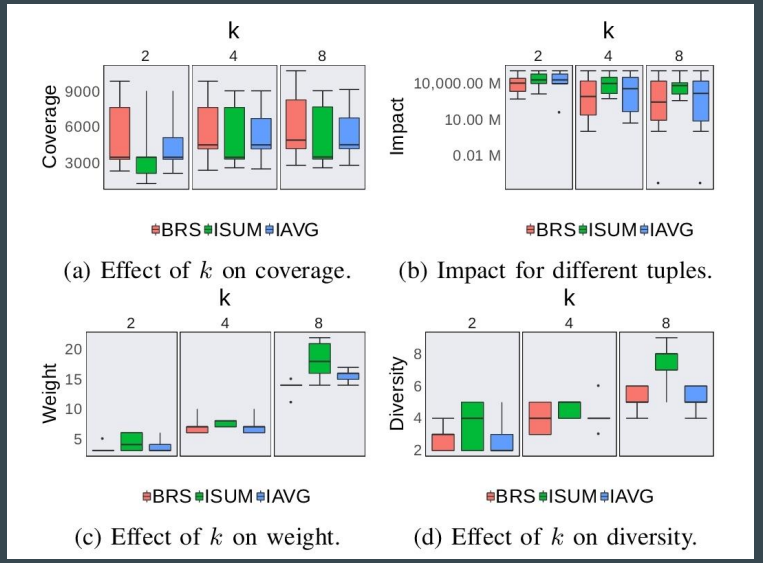
(a) Effect of $|D|$ in TPC-H. (b) Effect of k in IMDB.

(c) Effect of m_q in GLEI. (d) Effect of m_w in GLEI.

(e) Effect of join in TPC-H. (f) Effect of opt. on CPS.

Experimental Evaluation - Quality of provence Summaries

- Metrics used:
  - Impact : distribution of impact values for rules in summary
  - Coverage : total number of provenance tuples that are covered by rules in summary
  - Weight : total weight of rules in summary
  - Diversity : number of unique values for attributes present in summary
- BRS generates rules with high impact and a few high converge rules
- IPS constantly picks higher weight rules than BRS
- IPS outperforms BRS in diversity



(a) Effect of k on coverage. (b) Impact for different tuples.

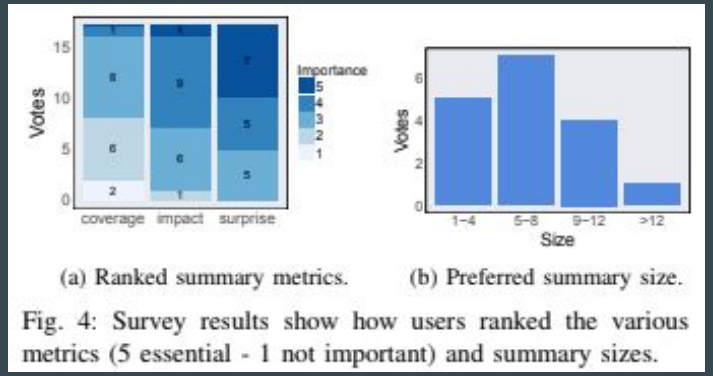
(c) Effect of k on weight. (d) Effect of k on diversity.

User Survey

The authors surveyed computer science students on their expectations of a data summarization system, designing their questions to be generic by presenting a use-case scenario that general users would be familiar with. They minimised bias by giving a generic example of the IMDB dataset.

17 participants completed the survey and results show participants favor high impact rules over rules with high coverage and they favor surprising rules the most.

Notably, users commented that high impact rules were most interesting and that they preferred shorter summaries.



(a) Ranked summary metrics. (b) Preferred summary size.

Fig. 4: Survey results show how users ranked the various metrics (5 essential - 1 not important) and summary sizes.

Critique

1. The author's conducted a user survey whose results should be taken with a grain of salt as the sample size was a miniscule 17 people.
2. Experiments were only run on three datasets, would like to see a larger sample of datasets

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.