VisPilot: Navigating Through Data Subsets with Hierarchical Summary of Visualizations

Doris Jung-Lin Lee
University of Illinois
jlee782@illinois.edu

Himel Dev
University of Illinois
hdev3@illinois.edu

Huizi Hu
University of Illinois
huizihu2@illinois.edu

Hazem Elmeleegy
Google Inc.
elmeleegy@google.com

Aditya Parameswaran
University of Illinois
adityagp@illinois.edu

ABSTRACT

The abstract for your paper for the PVLDB Journal submission. The template and the example document are based on the ACM SIG Proceedings templates. This file is part of a package for preparing the submissions for review. These files are in the camera-ready format, but they do not contain the full copyright note. Note that after the notification of acceptance, there will be an updated style file for the camera-ready submission containing the copyright note.

PVLDB Reference Format:
Ben Trovato, G. K. M. Tobin, Lars Thørväld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, Julius P. Kumquat, and Ahmet Sacan. A Sample Proceedings of the VLDB Endowment Paper in LaTeX Format. *PVLDB*, 12(xxx): xxxx-yyyy, 2019.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

1. INTRODUCTION

Visual data exploration are accessible and intuitive means for helping analysts understand multidimensional datasets. Analysts often study data distributions at different levels of data granularity [1, 6, 2] to discover trends and patterns, generate or verify hypotheses, and understand complex causal relationships. While visualizations exploits analysts's perceptual capabilities for visual pattern recognition, the effectiveness of human perceptual reasoning breaks down as datasets increases in size and complexity.

Manual exploration of multidimensional dataset is a time-consuming and error-prone process. To explore different subsets of data in a multidimensional dataset, an analysts would need to manually perform *drill-downs* on the space of possible data subsets by adding one filter at a time, without knowing what data subset would lead to an insight. Even if visualizations for all possible data subsets has been constructed, there is no systematic way for analysts to make sense of or even navigate through this large space of possible visualizations. More alarmingly, without being able to contextualize the relationships between data subsets, analysts may mistaken the general cause of an observed deviation in trend for a specific cause—a fallacy coined as "drill-down fallacy" in our paper [3].

To this end, we present VisPilot, a visual analytic tool that addresses the challenges of manual drill-down exploration. VisPilot traverses the space of possible data subsets (hereafter known as the *lattice*) to *automatically* identify a *compact* network of *informative* and *interesting* visualizations that collectively convey the key insights in a dataset. Demo attendees will have the opportunity to interact with VisPilot to better understand and gain insights regarding of multiple provided datasets.

2. DEMONSTRATION SCENARIO

3. SYSTEM

3.1 Recommendation Objective

3.2 System Design

VisPilot is a web application built on top of a traditional in-memory database, PostgresSQL. Given a set of user selection via the frontend interface, the Interaction Manager relays the corresponding HTTP request to the Java backend. The backend runs within an embedded Jetty web-server and is responsible for managing the lattice (Lattice Manager) and querying data distributions for specified data subsets (Query Manager). The lattice manager consists of three modules: 1) construction of the lattice, 2) storage the lattice for subsequent reuse, and 3) traversal across the lattice in search of a maximal subgraph. Amongst the three modules, lattice construction is the most computationally intensive step, since the size of the lattice grows exponentially with the cardinality (across all attributes) in the dataset. VisPilot employs several optimization to ensure that the system generates a dashboard at interactive speeds.

1. Instead of sending large numbers of independent database queries for every single data subset, the Query Manager applies sharing-based optimizations from Vartak et al. [5], submitting only a single query to the RDBMS via the Database Communicator and combining multiple aggregates to obtain the aggregate values for the requested subset.

Proceedings of the VLDB Endowment, Vol. 12, No. xxx
ISSN 2150-8097.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

2. When generating the lattice combinatorially, we apply an early stopping criterion that limits the lattice to be no larger than 4 levels (default setting, further adjustable by users). This shell-fragment approach [4] is based on the observation that most analysts are only interested in data subsets with small numbers of filters, beyond that the filter becomes difficult to interpret.
3. To eliminate insignificant subsets with small population sizes, users can also select an *iceberg condition* [7] to adjust the extent of pruning on visualizations whose sizes fall below a certain percentage of the overall population size.

After the lattice is materialized, we cache the generated lattice both in-memory and in the database to facilitate further reuse. Upon a new dashboard request, VisPilot first performs a lookup in the Storage Module to see if a corresponding lattice has been generated for the selected set of {Dataset, x,y, aggregation function}. This ensures that dashboard refinement operations that requires low-latency, such as changing the number of requested dashboard (k) or requesting for a dashboard expansion, does not require lattice recomputation. The traversal module operates on top of the materialized lattice to search for the maximal weighted subgraph, via the fontier-greedy algorithm described in our paper [3]. Finally, the recommended subgraph is sent to the frontend in a JSON format. The Dashboard Renderer generates the recommended visualizations through Vega-Lite[1] and displays the visualizations in a connected graph layout through vis.js [2].

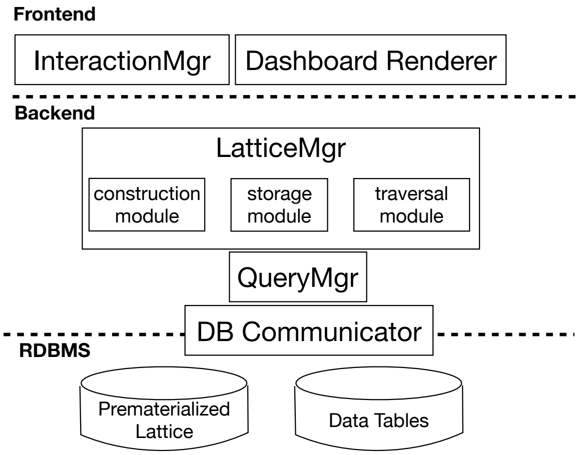

Figure 1: VisPilot System Architecture

4. CONCLUSION

Acknowledgments. We thank the anonymous reviewers for their valuable feedback. We acknowledge support from grants IIS-1513407, IIS-1633755, IIS-1652750, and IIS-1733878 awarded by the National Science Foundation, and funds from Microsoft, 3M, Adobe, Toyota Research Institute, Google, and the Siebel Energy Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies and organizations.

5. ADDITIONAL AUTHORS

6. REFERENCES

[1] A. Anand and J. Talbot. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626(c), 2015.

[2] J. Heer and B. Shneiderman. Interactive Dynamics for Visual Analysis. *Queue*, 10(2):30, 2012.

[3] D. J. Lee, H. Dev, H. Hu, H. Elmeleegy, and A. Parameswaran. Avoiding the drill-down fallacy with storyboard: Assisted and accelerated data exploration. *24th International Conference on Intelligent User Interfaces (IUI '19)*, 2019.

[4] X. Li, J. Han, and H. Gonzalez. High-dimensional OLAP: A minimal cubing approach. *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 528–539, 2004.

[5] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment*, 8(13):2182–2193, 2015.

[6] E. Wu and S. Madden. Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment*, 6(8):553–564, 2013.

[7] D. Xin, J. Han, X. Li, Z. Shao, and B. W. Wah. Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):111–126, 2007.

[1] `https://vega.github.io/vega-lite/`

[2] `http://visjs.org/`