Towards Visualization Recommendation Systems

Citation: Vartak, Manasi, Silu Huang, Tarique Siddiqui, Samuel Madden and Aditya Parameswaran. "Towards Visualization Recommendation Systems." ACM SIGMOD Record, 45 (4), December 2016, 34-39.

As Published: 10.1145/3092931.3092937

Publisher: Association for Computing Machinery (ACM)

Persistent URL: https://hdl.handle.net/1721.1/121342

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Towards Visualization Recommendation Systems

Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, Aditya Parameswaran

Abstract— Data visualization is often used as the first step while performing a variety of analytical tasks. With the advent of large, high-dimensional datasets and strong interest in data science, there is a need for tools that can support rapid visual analysis. In this paper we describe our vision for a new class of visualization recommendation systems that can automatically identify and interactively recommend visualizations relevant to an analytical task.

1 Introduction

Data visualization is perhaps the most widely used tool in a data analyst's toolbox, but the state of the art in data visualization still involves manual generation of visualizations through tools like Excel or Tableau. With the rise of interest in data science and the need to derive value from data, analysts increasingly want to use visualization tools to explore data, spot anomalies and correlations, and identify patterns and trends [32, 18]. For these tasks, current tools require substantial manual effort and tedious trial-and-error. In this paper, we describe our vision for a new class of *visualization recommendation* (VizRec) systems that seek to automatically recommend and construct visualizations that highlight patterns or trends of interest, thus enabling fast visual analysis.

Why Now? Despite the widespread use of visualization tools, we believe that we are still in the early stages of data visualization. We draw an analogy to movie recommendations: current visualization tools are akin to a movie catalog; they allow users to select and view the details of any movie in the catalog, and do so repeatedly, until a desired movie is identified. No current tools provide functionality similar to a movie recommendation system which gives users the ability to intelligently traverse the space of movies and identify interesting movies without getting bogged down in unnecessary details. On similar lines, the goal of VizRec systems is to allow users to easily traverse the space of visualizations and focus only on the ones most relevant to the task. There are two reasons why such visual recommendation tools are more important now than ever before:

Size. While the size of datasets—both in terms of number of records and number of attributes— has been rapidly increasing, the amount of human attention and time available to analyze and visualize datasets has stayed constant. With larger datasets, users must manually specify a larger number of visualizations; experiment with more attributes, encodings and subsets of the data; and examine a larger number of visualizations before arriving at useful visualizations.

Varying Skill Levels. Users with varying levels of skill in statistical and programming techniques are now performing data analysis. As a result, there is a need for easy-to-use analysis tools for domain-experts who have limited data analysis expertise. Such tools can perform the heavy-lifting for analyzing correlations, patterns and trends, surfacing relevant insights in the form of accessible and intuitive visualizations.

Limitations of Current Tools. Current visualization tools such as Excel, Tableau, Spotfire provide a powerful set of tools to manually specify visualizations. However, as tools to perform sophisticated analysis of high-dimensional datasets, they lack several features:

- *Inadequate navigation to unexplored areas.* Due to the large number of attributes and values taken on by each attribute, exploring all parts of a dataset is challenging with current tools. Often some attributes of the dataset are never visualized, and visualizations on certain portions of the dataset are never generated. This focus on a tiny part of the data is especially problematic if the user is inexperienced or not aware of important attributes in the dataset.
- *Insufficient means to specify trends of interest.* Current tools lack means to specify *what* the analyst is looking for, e.g., perhaps they want to find all products that took a hit in February, or they want to find all attributes on which two products differ. Using current tools, the analyst must specify each candidate visualization individually and determine if it satisfies the desired criteria.
- *Poor comprehension of context or the "big picture".* Existing tools provide users no context for the visualization they are currently viewing. For instance, for a visualization showing a dip in sales for February, current tools cannot provide information about whether this dip is an anomaly or a similar trend is seen in other products as well. Similarly, the tool cannot indicate that another attribute (e.g. inclement weather) may be correlated to dip in sales. In current tools, users must generate related visualizations manually and check if correlations or explanations can be identified. There are also no means for users to get a high-level summary of typical trends in visualizations of the dataset.
- *Limited understanding of user preferences for certain types of visualizations, insights, or trends.* Apart from recording visualizations which the user has already generated and giving users the ability to re-create historical visualizations, existing tools do not take into past user behavior into account for identifying relevant visualizations. For instance, if the user typically views only a handful of attributes from a dataset, maybe it is worth recommending to this user other attributes that may be correlated or similar to these attributes or help explain interesting or anomalous aspects of these visualizations.

Recent work by us and others has attempted to propose systems that address various aspects of visualization recommendations, e.g. [34, 41, 48]. Commercial products are also beginning to incorporate elements of VizRec into their tools [2, 4]. However, all of these tools are far from being full-featured VizRec systems. This position paper aims to detail the key requirements and design considerations for building a full-feature VizRec system. While we are inspired by traditional product recommendation systems in developing the ideas in this paper, our primary focus will be on aspects that are unique to the visualization recommendation setting. We also describe our experiences in building precursors to these sophisticated VizRec systems. Throughout this paper, we focus on the *systems-oriented* challenges of building a VizRec system. There are many challenging user interface and interaction problems that must be addressed to build an effective VizRec system; these are, however, outside the scope of this paper. We begin by discussing *axes* or *dimensions* that are relevant in making a recommendation (Section 2), the criteria for assessing quality of recommendations (Section 3), and architectural considerations (Sections

- *Manasi and Sam are with Massachusetts Institute of Technology (MIT). E-mail: mvartak, madden@csail.mit.edu.*
- *Silu, Tarique, and Aditya are with the University of Illinois, Urbana-Champaign. E-mail: shuang86, tsiddiq2, adityagp@illinois.edu.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

4 and 5). We then describe our current work in this area (Section 6) and conclude with a brief discussion of related work (Section 7).

2 Recommendation Axes

Whether a visualization is *useful* for an analytical task depends on a host of factors. For instance, a visualization showing sales over time may be useful in a sales projection task, while a visualization showing toy breakdown by color may be useful in a product design task. Similarly, a visualization showing a dip in profit may be of interest for a salesperson, while a visualization explaining the upward trend in auto accidents would be of interest to auto-manufacturers. In this section, we outline five factors that we believe must be accounted for while making visualization recommendations: we call these *recommendation axes*. In designing VizRec systems, different systems may prioritize different axes differently, depending in the intended applications.

Data Characteristics. For the purpose of analysis, the most important, and arguably the richest axis, is data. In many ways, the goal of a visualization recommender system is to mine the data for interesting values, trends, and patterns to speed up the data analysis process. These patterns may be then presented to the user at different stages of analysis, e.g. when they first load the dataset, while performing some task, or viewing a particular visualization. There are a number of data characteristics that a VizRec system must consider, e.g.: *a*) *summaries*, e.g., histograms or summary statistics [45], providing an overview of the dataset and data distributions; *b*) *correlations*, e.g., Pearson correlation, Chi-squared test [45], providing an understanding of which attributes are correlated, and to what extent; *c*) *patterns and trends*, e.g., regression [45], association rules [8], or clustering [17], providing an understanding of what is "typical" in the dataset and enabling users to to contextualize trends; *d*) *advanced statistics*, e.g., tests like ANOVA, Wilcox rank sum [45] aiding in deeper analysis.

Intended Task or Insight. Another important input to a visualization recommender is the intent of the user performing analysis: This includes the following aspects: *a*) *style of analysis*: e.g. exploratory, comparative, predictive, or targeted. *b*) *subject of analysis*: subset of data and attributes of interest (e.g., adult males, sweater products, color); *c*) *goal of analysis*: e.g. explanations for a certain behavior (e.g., why is there a spike in february in sales), comparison between subsets of data (e.g., how are staplers doing compared to two years ago), finding unusual or outlier patterns (e.g., are there any toy colors doing "differently"), or finding specific patterns (e.g., are there any chairs that had high sales on October 15).

An important question is how to identify the user's intended task or insight. While we may be able to obtain explicit task information from the user (e.g. via a drop-down menu of tasks), we may also infer intent through the sequence of actions performed by the user.

Semantics and Domain Knowledge. A large amount of semantic information is associated with any dataset—what data is being stored, what information does each attribute provide, and how are they related to each other, how does this dataset relate to others etc. This semantic information determines, in part, whether a visualization is "interesting" or "unusual". For instance, if a user is analyzing a dip in profits, semantics would indicate that visualizations showing attributes such as sales, revenue, cost of goods sold, number of items sold would be useful. An even more significant factor—and much harder to capture—is domain knowledge. The user possess unique or domain-specific knowledge that guides the search for attributes, trends and patterns. For example, a recommendation system that only considers data and task may recommend a visualization showing patient intake by care unit because it finds that the OB-GYN unit has a disproportionately high percentage of female patients. A person with minimal domain knowledge would note that the trend shown in this visualization is obvious and therefore the visualization is unhelpful. Domain knowledge can include typical behavior of specific attributes or subsets of data (e.g., sales always goes up around christmas time, or electronics sales is always greater than stapler sales), or relationships between groups of attributes, (e.g., sales and profits are always proportional). It can also include external factors not in the dataset, e.g., an earthquake may have affected hard disk drive production.

Visual Ease of Understanding. A dimension that is completely unique to visualization recommendation is what we call *visual ease of understanding*. This dimension ensures that data has been displayed in the most intuitive way for easy understanding. Work such as [29, 30] proposes techniques to choose the best visual encodings for the particular data types. The rich area of information visualization includes a variety of techniques to visualize data with varying dimensionality, data types and interaction techniques [20, 27, 23]. We expect visualization recommendation systems to draw heavily from this literature.

User Preference. Multiple users analyzing the same dataset may have attributes of common interest, while the same user analyzing different datasets may prefer specific visualization types. For instance, an scientific user may always be interested in statistical significance of results while a novice user may be interested in qualitative results. Similarly, certain views of a particular dataset may be most intuitive and therefore preferred by most users. Another type of user preference is typical "patterns" of user behavior, e.g., the overview→zoom→detail-on-demand mantra [38]. Such preferences can provide important input in determining which visualizations would be most relevant for a particular user in a specific stage of analysis. There is a large body of work on extracting user preferences such as [21, 31], techniques from which can be adapted for visualization recommenders.

3 Recommendation Criteria

The previous section discussed factors that contribute to the utility of visualizations. In this section, we discuss criteria that can be used to measure quality of visualization recommendations. We find that some criteria are similar to traditional product recommendations (e.g. relevance) while others are unique to VizRec (e.g. non-obviousness) or are re-interpretations of existing criteria (e.g. surprise).

- *Relevance*: This metric measures whether the recommendation is useful to the user in performing the particular analytic task. As discussed in the previous section, many factors such as data, task, semantics etc., play a role in determining useful-ness of a visualization, and this is our primary metric for quality.
- *Surprise*: This metric measures the novelty or unexpected-ness of a recommendation. For product recommedantions, this metric prefers items the user didn't explicitly ask for, and therefore are novel. In VizRec, this corresponds to visualizations that show something *out of the ordinary* or unexpected. For example, a dip in sales of staplers may not be interesting by itself but when juxtaposed with the booming sales of other stationery items, becomes interesting due to the unexpected-ness of the result.
- *Non-obviousness*: This metric is specific to VizRec. Non-obviousness measures whether the recommendation is expected given semantics and domain knowledge. For instance, the OB-GYN example from the previous section was surprising from a statistical point of view, but was, in fact, obvious to a user with minimal domain knowledge. While surprise is defined with respect to data and history, non-obviousness is defined with respect to semantics and domain knowledge.

Since we expect the recommender system to recommend multiple visualizations, the quality of the *visualization set* is as important as the quality of individual visualizations. We note that the order of recommendations is also important in this regard and we expect order to be determined by relevance in addition to the metrics below that measure visualization set quality.

- *Diversity.* This metrics measures how different are the individual visualizations in the recommended collection. Diversity may be measured with respect to attributes, visualization types, different statistics, visual encodings, etc.
- *Coverage.* This metric measures how much of the space of potential visualizations is covered by recommendations. While users particularly value coverage during exploration, during

analysis, users seek to understand how *thorough* are the recommendations shown to them. For instance, the user would like to understand whether the system examined ten or ten thousand visualizations before recommending the top-5.

4 ADAPTING RECOMMENDER SYSTEM TECHNIQUES

The task of building a VIZREC system brings up a natural question: recommender systems is a rich area of research; how much of existing work can we reuse? Our goal in this section is to broadly identify problems in VIZREC that can be solved using existing techniques, and those that require new techniques.

Existing methods for product recommendation broadly fall into three categories [10, 6]: (i) *content-based filtering* that predicts user preferences based on item attributes; (ii) *collaborative-filtering* that uses historical ratings to determine user or item similarity; and (iii) *knowledge-based filtering* that uses explicit knowledge models to make recommendations. Collaborative filtering is probably the most popular technique currently used in recommender systems (e.g. at Amazon [28]). However, collaborative filtering (as well as content-based filtering) assumes that there is historical rating data available for a large number of items. As a result, it suffers from the traditional *cold start problems* when historical ratings are sparse. Knowledge-based filtering [40], in contrast, does not depend on history and therefore, does not suffer from cold start problems.

VIZREC differs from product recommendations in a few key areas that impact the techniques that can be used for recommendation. In VIZREC, *new* datasets are being analyzed by *new* users constantly. Furthermore, each new task on a dataset can produce an entirely new (and large) set of visualizations from which the system must recommend, i.e., not only is the universe of items large, it is generated on-the-fly. Consequently, VIZREC systems almost never have sufficient historical ratings to inform accurate collaborative or content-based filtering. Visualization recommenders must therefore rely on on-the-fly, knowledge-based filtering. This is not to say that techniques such as collaborative filtering cannot be used to transfer learning across datasets; it means that while such techniques can aid in recommendations, the heavy lifting is performed by knowledge-based filtering.

Applying knowledge-based techniques to VIZREC brings up several challenges that have not been addressed in the recommender systems literature: (i) Models must be developed for capturing the effect of each recommendation axis (Section 2) on visualization utility; (ii) Knowledge models must be such that they can perform online processing with interactive latencies. For example, along the data axis, several of the existing data mining techniques from Section 2 are optimized for offline processing. As a result, these techniques must be adapted to work in an online setting with small latencies; (iii) Efficient ranking techniques and ensemble methods must be developed for combining large number of models along individual axes, and multiple axes.

Thus, while there is a rich body of work in recommender systems, the unique challenges of VIZREC require the development of new, and in many cases, online and efficient recommendation techniques. In the next section, we discuss the implications of the unique VIZREC requirements on system design and techniques that can be used to meet these requirements.

5 ARCHITECTURAL CONSIDERATIONS

Making visualization recommendations based on the various dimensions in Section 2 can be computationally expensive. Recommendations based on data are particularly expensive since it involves mining the dataset for various trends and patterns, often exploring an exponential space of visualizations. Even for moderately sized datasets (1M rows, 50 attributes), a naive implementation of the underlying algorithm can result in latencies of over an hour [42]. As a result, a visualization recommender must be carefully architected to provide interactive latencies. We next discuss some strategies that can be used to make interactive latencies possible while making visualization recommendations.

Pre-computation. Many real-world recommender systems perform complex and expensive computation (e.g. computations on the item-user matrix in collaborative filtering [28]) in an offline phase. The results of this computation are then used to make fast predictions during the online phase. Since VizRec systems must employ knowledge-based filtering and the set of potential visualization is not known up-front, opportunities to perform complex computations offline may be limited. However, some types of pre-computation, drawn from the database systems literature, can be employed. For example, *data cubes* [19] can be used to precompute and store aggregate views of the data for subsequent use. A challenge with using data cubes is determining the dimensions on which to build a data cube and updating aggregates in response to tasks that query different parts of the data. Recent work such as [12] pushes forward the boundary on pre-computation by using models to predict accesses for views and pre-computing them. Along the lines of data cubes, a visualization recommender can also perform offline computation of various statistics and correlations that can inform subsequent explorations and construction of visualizations. Specialized indexes tailored to access patterns unique to visualization recommendations (e.g. [25]) can be used speed up further speed up online data access. Finally, traditional *caching* approaches that have been used with great success both on the client-side as well as the server-side can be used to further reduce recommendation latency.

Online Computation. As discussed previously, visual recommenders are in the unique position of having to produce the space of potential recommendations on-the-fly. As a result, a significant part of the computations must must happen online. To avoid latencies in the hours, online computation must perform aggressive optimizations while evaluating visualizations. Some of the techniques include: (i) *parallelism*: the first resort when faced with a large number of independent tasks is parallelism. Faced with a large space of potential visualizations that must be evaluated, we can evaluate visualizations in parallel produce a large speedup; (ii) *multi-query optimization*: the computations used to produce candidate visualizations for recommendation are often very similar — they perform similar operations on the same or closely related datasets; As a result, multi-query optimizations techniques [35, 33] can be used to intelligently group queries and share computations; (iii) *pruning*: while the two techniques above can increase the absolute speed of execution, they do not reduce the search space of visualizations. Often, although hundreds of visualizations are possible for a given dataset, only a small fraction of the visualizations meet the threshold for recommendation. As a result, a significant fraction of computational resources are wasted on low-utility views. Pruning techniques (e.g. confidence-interval pruning [45], bandit resource allocations [43]) can be used to discard low-utility views with minimal computation; (iv) *better algorithms*: finally, of course, there are opportunities to develop better and faster algorithms that can speed up the computation of a variety of statistical properties.

Approximate Computation. Approximate query processing [5, 7] has been shown to have tremendous promise in reducing query latencies on large datasets. Techniques based on different sampling strategies (e.g. stratified sampling, coresets [13], importance sampling [39]) can be used to further speed up computation. Sampling brings with it a few challenges. For a given computation, we must choose the right type of sample (based on size, technique etc). Additionally, for a given sampling strategy, we must provide users with measures of quality or confidence in the results (e.g. confidence intervals). These measures of quality are particularly important in data analysis since they inform users how much they can trust certain results. Finally, while sampling may be useful to compute many statistical properties, certain properties such as outliers cannot be answered correctly with a sample.

6 OUR PRIOR AND CURRENT WORK

Prior Work. As a first attempt towards building a full-fledged VIZREC system, we built SEEDB [42], a data-driven recommender system. SEEDB adopts a simple metric for judging utility of a visualization: a visualization is likely to be interesting if it displays a large deviation from a reference dataset (either generated from the entire dataset or by a user-defined comparison dataset). For example, in the case of stapler sales, a visualization of staplers over time would be judged as interesting if a reference dataset (e.g., sales over all products) showed the opposite trend, and would be judged uninteresting otherwise.

Deviation. To measure the deviation between two visualizations, SEEDB uses a *distance metric* that measures deviation between the two visualizations represented as probability distributions. This distance metric could be Jenson-Shanon distance, Earth Movers distance, or K-L divergence. Our user studies comparing SEEDB recommendations to ground truth show that users find visualizations with high deviation interesting.

Optimizations. SEEDB evaluates COUNT / SUM / AVERAGE aggregates over all measure-dimension pairs of attributes (or triples of attributes if three-dimensional visualizations are desired). Evaluating all such aggregates can be very expensive and can lead to latencies of over an hour for medium-sized datasets. As a result, SEEDB adopts the architectural aspects described earlier—including precomputation, parallel computation, grouped evaluation, as well as early pruning—to significantly reduce latency. We found that our suite of online optimizations led to a speedup of over 100X.

Design. SEEDB is designed as an end-to-end VIZREC tool that can run on top of any relational database. Figure 1 shows a screenshot of the SEEDB frontend. SEEDB adopts a *mixed-initiative design* that provides users the ability to both manually construct visualizations (component "B" in Figure 1) and also obtain automated visualization recommendations ("D").

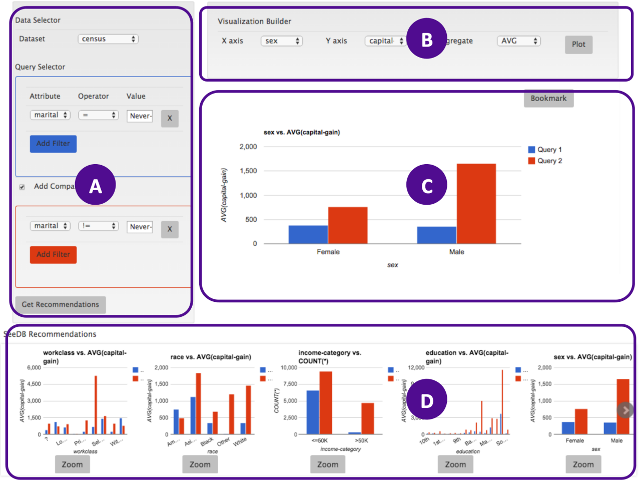

Fig. 1. SEEDB Frontend: (A) query builder, (B): visualization builder, (C): visualization pane, (D) recommendations pane

Results. With the aggressive use of optimizations, we were able to reduce SEEDB latencies from tens of minutes to a few seconds for a variety of real and synthetic datasets. Our user study comparing SEEDB with and without recommendations demonstrated that users are three times more likely to find recommended visualizations useful as compared to manually generated visualizations.

Current Work: Drawing on ideas presented in this paper, we are extending SEEDB in several ways:

- *Slices of data instead of attributes:* Currently, SEEDB focuses on recommending attributes of interest that help distinguish a given slice of data from the rest. One could instead envision recommending interesting slices of data instead of attributes. For instance, given that a user has an interest in Product = Staplers, can we show that Product = Pencils has a similar behavior on a given visualization?
- *Approximate Visualizations:* Given that users are often only interested in trends or comparisons as opposed to actual values in these visualizations, we can use this insight to recommend visualizations that are approximate, but depict correct trends and comparisons. We have started exploring this in recent work [26].
- *More expressive visualizations and queries:* Since SEEDB only supports 2-D visualizations, we are currently working on extending it to 3-D visualizations as well as visualizations on queries that span multiple tables. In addition, we are extending SEEDB to support more data types including time series data.
- *Incorporating other recommendation axes:* SEEDB is primarily a data-driven VIZREC tool. We are now adding features to SEEDB that can capture other dimensions of recommendations.

Our new tool, a generalization of SEEDB following these directions, is called ZENVISAGE (meaning to view (data) effortlessly).

7 RELATED WORK

Visualization Tools. The visualization community has introduced a number of visual analytics tools such as Spotfire and Tableau [1, 3]. While these tools have started providing some features for automatically choosing visualizations for a data set [2, 4], these features are restricted to a set of aesthetic rules-of-thumb that guide which visualization is most appropriate. Similar visual specification tools have been introduced by the database community, e.g., Fusion Tables [14]. These tools require the user to manually specify the visualizations to be created and require tedious iteration to find useful visualizations.

Partial Automation of Visualizations. Profiler detects anomalies in data [22] and provides some visualization recommendation functionality. VizDeck [24] depicts all possible 2-D visualizations of a dataset on a dashboard. Given that VizDeck generates all visualizations, it is meant for small datasets and does not focus on speeding up generation of these visualizations.

VIZREC Systems. Work such as [30, 29] focuses on recommending visual encodings for a user-defined set of attributes, thus addressing the *visual ease of understanding* axis. Similar to SEEDB, which is a data-driven recommendation system, [36, 47, 46] use different statistical properties of the data to recommend visualizations. [15] monitors user behavior to mine for intent and provides recommendations, while [44] uses task information and semantic web ontologies. Most recently, the Voyager system [48] has been proposed to provide visualization recommendations for exploratory search.

Data mining and Machine Learning. The data mining and machine learning communities have developed a large set of algorithms and techniques to identify trends and patterns in different types of data. These range from simple association rule and clustering algorithms [17] to sophisticated models for pattern recognition [9]. Many of these techniques can be used in VIZREC systems to mine for trends in data.

Recommender Systems. Techniques from recommender systems including collaborative filtering and content-based filtering [10, 6] can be used to aid visualization recommendations. As discussed previously, knowledge-based filtering techniques [40, 11] in particular are relevant for VIZREC systems. Finally, also relevant to VIZREC systems are techniques for evaluating quality of recommender systems as studied in [37, 16].

8 CONCLUSION

With the increasing interest in data science and the large number of high-dimensional datasets, there is a need for easy-to-use, powerful tools to support visual analysis. In this paper, we present our vision for a new class of visualization recommender systems that can help users rapidly identify interesting and useful visualizations of their data. We present five factors that contribute to the utility of a visualization and a set of criteria for measuring quality of recommendations. We also discuss the new set of challenges created by VIZREC systems and architectural designs to help meet these challenges. VIZREC systems are likely to become an important part of next-generation visualization software, and tools such as SEEDB and others are first steps in building such systems.

References

[1] Spotfire, http://spotfire.com. [Online; accessed 17-Aug-2015].

[2] Spotfire, http://www.tibco.com/company/news/releases/2015/tibco-announces-recommendations-for-spotfire-cloud. [Online; accessed 17-Aug-2015].

[3] Tableau public, www.tableaupublic.com. [Online; accessed 3-March-2014].

[4] Tableau showme. [Online; accessed 17-Aug-2015].

[5] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. The aqua approximate query answering system. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 574–576, New York, NY, USA, 1999. ACM.

[6] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[7] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: Queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys '13, pages 29–42, New York, NY, USA, 2013. ACM.

[8] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[11] R. Burke. Integrating knowledge-based and collaborative-filtering recommender systems. In *Proceedings of the Workshop on AI and Electronic Commerce*, pages 69–72, 1999.

[12] P. R. Doshi, E. A. Rundensteiner, and M. O. Ward. Prefetching for visual data exploration. In *DASFAA 2003*, pages 195–202. IEEE, 2003.

[13] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1434–1453. SIAM, 2013.

[14] H. Gonzalez et al. Google fusion tables: web-centered data management and collaboration. In *SIGMOD Conference*, pages 1061–1066, 2010.

[15] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 315–324, New York, NY, USA, 2009. ACM.

[16] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *The Journal of Machine Learning Research*, 10:2935–2962, 2009.

[17] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[18] P. Hanrahan. Analytic database technologies for a new kind of user: the data enthusiast. In *SIGMOD Conference*, pages 577–578, 2012.

[19] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD '96, pages 205–216, 1996.

[20] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[21] S. Holland, M. Ester, and W. Kießling. Preference mining: A novel approach on mining user preferences for personalized applications. In *Knowledge Discovery in Databases: PKDD 2003*, pages 204–216. Springer, 2003.

[22] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: integrated statistical analysis and visualization for data quality assessment. In *AVI*, pages 547–554, 2012.

[23] D. Keim et al. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.

[24] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: Self-organizing dashboards for visual analytics. SIGMOD '12, pages 681–684, 2012.

[25] A. Kim, E. Blais, A. Parameswaran, P. Indyk, S. Madden, and R. Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *Proc. VLDB Endow.*, 8(5):521–532, Jan. 2015.

[26] A. Kim, E. Blais, A. G. Parameswaran, P. Indyk, S. Madden, and R. Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *CoRR*, abs/1412.3040, 2014.

[27] M. Kreuseler, N. Lopez, and H. Schumann. A scalable framework for information visualization. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 27–, Washington, DC, USA, 2000. IEEE Computer Society.

[28] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[29] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, Apr. 1986.

[30] J. D. Mackinlay et al. Show me: Automatic presentation for visual analysis. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1137–1144, 2007.

[31] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, Aug. 2000.

[32] K. Morton, M. Balazinska, D. Grossman, and J. D. Mackinlay. Support the data enthusiast: Challenges for next-generation data-analysis systems. *PVLDB*, 7(6):453–456, 2014.

[33] C. H. Papadimitriou and M. Yannakakis. Multiobjective query optimization. In P. Buneman, editor, *PODS*. ACM, 2001.

[34] A. Parameswaran, N. Polyzotis, and H. Garcia-Molina. Seedb: Visualizing database queries efficiently. *PVLDB*, 7(4), 2013.

[35] T. K. Sellis. Multiple-query optimization. *ACM Trans. Database Syst.*, 13(1):23–52, Mar. 1988.

[36] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.

[37] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[38] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[39] S. T. Tokdar and R. E. Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

[40] S. Trewin. Knowledge-based recommender systems. *Encyclopedia of Library and Information Science: Volume 69-Supplement 32*, page 180, 2000.

[41] M. Vartak, S. Madden, A. G. Parameswaran, and N. Polyzotis. SEEDB: automatically generating query visualizations. *PVLDB*, 7(13):1581–1584, 2014.

[42] M. Vartak, S. Madden, A. G. Parameswaran, and N. Polyzotis. SeeDB: supporting visual analytics with data-driven recommendations. *VLDB 2015 (accepted)*, data-people.cs.illinois.edu/seedb-tr.pdf.

[43] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In *ECML*, pages 437–448, 2005.

[44] M. Voigt, S. Pietschmann, L. Grammel, and K. Meißner. Context-aware recommendation of visualization components. In *The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW)*, pages 101–109, 2012.

[45] L. Wasserman. *All of Statistics*. Springer, 2003.

[46] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *INFOVIS*, volume 5, page 21, 2005.

[47] G. Wills and L. Wilkinson. Autovis: automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[48] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization & Comp. Graphics*, 2015.