Towards Visualization Recommendation Systems

Manasi Vartak[1] Silu Huang[2] Tarique Siddiqui[2] Samuel Madden[1] Aditya Parameswaran[2]

[1]MIT
{mvartak, madden}@csail.mit.edu

[2]University of Illinois (UIUC)
{shuang86, tsiddiq2, adityagp}@illinois.edu

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 significant 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 systems, namely visualization recommendation systems, that can automatically identify and interactively recommend visualizations relevant to an analytical task. We detail the key requirements and design considerations for a visualization recommendation system. We also identify a number of challenges in realizing this vision and describe some approaches to address them.

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 [30, 14]. 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* (VISREC) systems that automatically recommend 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 by their sheer number or unnecessary details. On similar lines, the goal of VISREC 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—in terms of number of records and number of attributes— has been rapidly increasing, the amount of human attention and time available to analyze datasets has stayed constant. With larger datasets, users must manually specify and examine a larger number of visualizations and must experiment with more attributes and subsets of data before arriving at visualizations showing patterns of interest.

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 and patterns, and surface relevant insights in the form of accessible and intuitive visualizations.

Limitations of Current Tools. Current visualization tools such as Excel and Tableau provide a powerful set of mechanisms to manually specify visualizations. However, as tools to perform sophisticated analyses 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 unfamiliar with attributes in the dataset.
- *Insufficient means to specify trends of interest.* Current tools lack the 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. Us-

ing 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 with the 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 the visualizations of a dataset.
- *Limited understanding of user preferences.* Apart from giving users the ability to re-create past visualizations, existing tools do not take past user behavior into account while 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.

Recent work by us and others has attempted to propose systems that address various aspects of visualization recommendations, e.g. [32, 40, 46, 37]. Commercial products are also beginning to incorporate elements of VISREC into their tools [1, 2]. However, all of these tools are far from being full-featured VISREC systems. This position paper aims to detail the key requirements and design considerations for building a full-feature VISREC 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 VISREC setting. Throughout this paper, we focus on the *systems-oriented* challenges of building a VISREC system. There are many challenging user interface and interaction problems that must be addressed to build an effective VISREC system; these are, however, outside the scope of this vision 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 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*.

I. Data Characteristics. In many ways, the goal of a visualization recommender system is to mine the data for interesting values, trends, and patterns to speed up data analysis. 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 VISREC system can consider while making recommendations, e.g.: *a*) *summaries*, e.g., histograms or summary statistics [43], providing an overview of the data distribution; *b*) *correlations*, e.g., Pearson correlation, Chi-squared test [43], providing an understanding of correlated attributes; *c*) *patterns and trends*, e.g., regression [43], association rules, or clustering, providing an understanding of what is "typical" in the dataset and enabling users to contextualize trends; *d*) *advanced statistics*, e.g., tests like ANOVA, Wilcox rank sum [43] aiding in deeper analysis.

II. Intended Task or Insight. Along with data, an important input to a VISREC system 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., chairs with high sales on October 15). While we may be able to obtain explicit task information from the user (e.g. via a drop-down menu or query language of sorts), we may also infer intent through user actions. Finally, if we have information about the user's assumptions or biases regarding the data or task, the VISREC system can also provide recommendations to counter these biases.

III. Semantics and Domain Knowledge. A large amount of semantic information is associated with any dataset—what data is stored, what information does each attribute provide, how are the attributes related, 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 possesses 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 that the OB-GYN hospital 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.

IV. 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 [27, 28] proposes techniques to choose visual encodings, while related work in information visualization includes a variety of techniques to visualize data with varying dimensionality and data types [26, 15, 23, 20].

V. User Preferences and Competencies. Multiple users analyzing the same dataset may have attributes of common interest, while the same user analyzing different datasets may prefer specific visualization types. Similarly, certain views of a particular dataset may be most intuitive or most relevant during a particular phase of analysis, leading most users to prefer these visualizations. A VISREC system also needs to account for the varying levels of visual literacy and statistical ability of the user. There is a large body of work on extracting user preferences (e.g., [16, 29]) as well as cognitive modeling (e.g., [13]), techniques from which can be adapted for VISREC. Furthermore, these techniques can be combined with assessments of visual and statistical literacy (e.g. [11, 9]) to tailor recommendations for each user.

Traditional recommendation systems focus mainly on User Preference and to some extent on Intended Task; however, the other axes enumerated above are tailored to VISREC systems.

3. RECOMMENDATION CRITERIA

The previous section discussed factors that contribute to the utility of visualizations. In this section, we discuss criteria 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 VISREC (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 relevance.
- *Surprise*: This metric measures the novelty or unexpected-ness of a recommendation. For product recommendations, this metric prefers items the user didn't explicitly ask for but are relevant. In VISREC, this corresponds to visualizations that show something *out of the ordinary*. 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, it becomes interesting.
- *Non-obviousness*: This metric is specific to VISREC. Non-obviousness measures whether the recommendation is expected given semantics and domain knowledge (as opposed to surprise which is defined with respect to data). For instance, the OB-GYN example discussed previously was surprising from a statistical point of view, but was, in fact, obvious to a user with minimal 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, along with measures related to coherence over time and visualization set quality.

- *Diversity.* This metric 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. A more subtle notion of diversity would capture the "informativeness" of a collection of visualizations relative to each other—the *conditional utility* of a visualization given others.
- *Coverage.* This metric measures how much of the space of potential visualizations and of the dataset 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 (and similarly whether the system examined 10% or 100% of the data) before recommending visualizations.

4. ADAPTING RECSYS TECHNIQUES

The task of building a VISREC 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 VISREC that can be solved using existing techniques, and those that require new techniques.

Existing methods for product recommendation broadly fall into three categories [7, 4]: (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 [24]). 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 [39], in contrast, does not depend on history and therefore, does not suffer from cold start problems.

VISREC differs from product recommendations in a few key areas that impact the techniques that can be used for recommendation. In VISREC, *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, VISREC 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 must be performed by knowledge-based filtering.

Applying knowledge-based techniques to VISREC 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.

VISREC systems also suffer from a problem not faced by product recommendations, namely one of *false discoveries*. When making automated visualization recommendations, a VISREC system evaluates many tens or hundreds of visualizations before making recommendations. Since a visualization can (roughly) be thought of as performing a hypothesis test, chances of finding spurious patterns increase with increasing number of visualizations. As a result, a VISREC system must account for potential false discoveries in recommendations using techniques such as Bonferroni [8] or FDR [6] correction.

Thus, while there is a rich body of work in recommender systems, the unique challenges of VISREC 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 VISREC requirements on system design and techniques that can be used to meet these requirements.

5. ARCHITECTURAL CONSIDERATIONS

Making visualization recommendations, particularly based on data, is computationally expensive. Therefore, we find that the most important consideration in making real-time recommendations is the *data processing engine*. While traditional disk-resident databases can accomodate large datasets, they cannot provide the interactive speeds necessary for visualization recommendation. As a result, a VISREC system must take advantage of main-memory using techniques such as operating on samples, pre-materializing views and using efficient indexes. We now elaborate on some of these strategies.

Pre-computation. Many real-world recommender systems perform complex and expensive computation (e.g. computations on the item-user matrix in collaborative filtering [24]) in an offline phase. The results of this computation are then used to make fast predictions during the online phase. Since VISREC systems must employ knowledge-based filtering and the set of potential visualization is not known upfront, 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* can be used to precompute and store aggregate views for visualization (e.g. Nanocubes [25]). 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. [22]) can be used to 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 happen online. To avoid latencies in the hours, online computation must perform aggressive optimizations while evaluating visualizations. Some of the techniques include: (i) *parallelism*: faced with a large space of potential visualizations that must be evaluated, we can evaluate visualizations in parallel to produce a large speedup; (ii) *multi-query optimization*: the computations used to produce candidate visualizations are often very similar; they perform similar operations on the same or closely related datasets. Consequently, multi-query optimizations techniques [34, 31] can be used to intelligently group queries and share computation; (iii) *pruning*: while the above techniques can increase the speed of execution, they do not reduce the search space of visualizations. Although hundreds of visualizations are possible for a given dataset, only a small fraction of the visualizations are actually useful. As a result, a significant fraction of computational resources are wasted on low-utility visualizations. Pruning techniques (e.g. confidence-interval pruning [43], bandit resource allocations [41]) can be used to discard low-utility views with minimal computation; (iv) *better algorithms*: finally, there are opportunities to develop better and faster algorithms to compute statistical properties.

Approximate Computation. Approximate query processing [3, 5] 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 [10], importance sampling [38]) can be used to further speed up computation, especially because users may be satisfied with imperfect results: both imperfect visualizations [22] and imperfect recommendations of visualizations. 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 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

We now briefly describe some of our efforts towards building VisRec systems and future work.

SeeDB. As a first attempt towards building a full-fledged VisRec system, we built SeeDB [40] (Figure 1). SeeDB is designed as a mixed-initiative [18] system that provides users the ability to both manually construct visualizations (component "B"), and receive recommendations (component "D"). In judging utility, SeeDB deems a visualization to be interesting if it displays a large deviation from a reference. For example, a visualization of sales of staplers over time may be interesting if a reference (e.g., sales of all products) is showing an opposite trend. Our user study comparing SeeDB with and without recommendations demonstrates that users are *3X more likely to find recommended visualizations useful as compared to manually generated visualizations.*

Figure 1: SeeDB Frontend: (A) query builder, (B): visualization builder, (C): visualization pane, (D) recommendations pane

zenvisage. Our new visualization recommendation tool is called Zenvisage—meaning to view (data) effortlessly [37, 36]. The goal of zenvisage (Figure 2) is to quickly identify interesting patterns or trends from large datasets via one of two mechanisms: a simple drag-and-drop based interactive interface with query sketching capabilities (e.g., find a visualization where there is a spike at a certain point simply by drawing the desired visualization on a canvas—Box 4 shows the result for the drawing in Box 3, while Box 2 shows other typical visualizations for context) and a visual data exploration language called ZQL for more complex requests (Box 5).

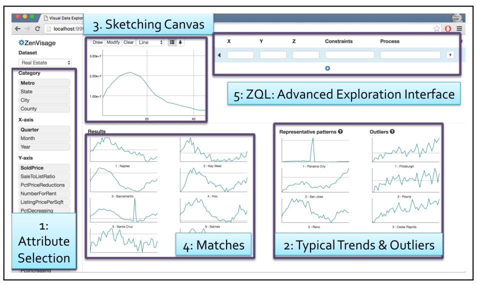

Figure 2: zenvisage Frontend

7. RELATED WORK

Partial Automation of Visualizations. Tools such as Spotfire and Tableau have recently started providing some features for automatically choosing visualizations for a data set [1, 2]; however these features are restricted to a set of aesthetic rules-of-thumb that guide visualization. Profiler [19] detects anomalies in data and provides some visualization recommendation functionality.

VizDeck [21] allows users to select from visualizations presented on a dashboard.

VISREC **Systems.** Work such as [28, 27] focuses on recommending visual encodings for a user-defined set of attributes, thus addressing the *visual ease of understanding* axis. Similar to SEEDB, [35, 45, 44] use different statistical properties of the data to recommend visualizations. [12] monitors user behavior to mine for intent and provides recommendations, while [42] uses task information and semantic web ontologies. Most recently, the Voyager system [46] has been proposed to provide visualization recommendations for exploratory search.

Finding patterns and trends. The data mining and machine learning community has developed a large swath of statistical analysis tools such as Knime, RapidMiner, SAS, and SPSS, and programming libraries [17, 33] for doing complex analytics tasks such as classification, clustering, and dimensionality reduction. While many of these tools and libraries can be employed in VISREC systems to mine for patterns in data, only expert users with detailed knowledge of algorithmic details and parameterizations can use these tools effectively.

8. CONCLUSION

With increasing interest in data science and large numbers of high-dimensional datasets, there is a need for easy-to-use, powerful visualization recommendation tools to support visual analysis. While we are in the early days of VISREC systems, we believe the directions outlined in this paper, as well as the analogies to and differences with traditional recommendation systems can lead to interesting, challenging, and impactful problems for the database research community.

9. REFERENCES

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

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

[3] S. Acharya et al. The aqua approximate query answering system. SIGMOD '99, pages 574–576, New York, NY, USA, 1999. ACM.

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

[5] S. Agarwal et al. Blinkdb: Queries with bounded errors and bounded response times on very large data. EuroSys '13, 2013.

[6] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

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

[8] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

[9] J. Boy, R. A. Rensink, E. Bertini, and J.-D. Fekete. A principled way of assessing visualization literacy. *IEEE transactions on visualization and computer graphics*, 20(12):1963–1972, 2014.

[10] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. SODA '13, pages 1434–1453, 2013.

[11] J. B. Garfield. Assessing statistical reasoning. *Statistics Education Research Journal*, 2(1):22–38, 2003.

[12] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. IUI '09, pages 315–324, New York, NY, USA, 2009. ACM.

[13] T. M. Green, W. Ribarsky, and B. Fisher. Building and applying a human cognition model for visual analytics. *Information Visualization*, 8(1):1–13, Jan. 2009.

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

[15] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *TVCG*, 6(1):24–43, 2000.

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

[17] G. Holmes, A. Donkin, and I. H. Witten. Weka: A machine learning workbench. In *Conf. on Intelligent Information Systems '94*, pages 357–361. IEEE, 1994.

[18] E. Horvitz. Principles of mixed-initiative user interfaces. CHI'99, pages 159–166. ACM, 1999.

[19] S. Kandel et al. Profiler: integrated statistical analysis and visualization for data quality assessment. In *AVI*, pages 547–554, 2012.

[20] D. Keim et al. Information visualization and visual data mining. *TVCG*, 8(1):1–8, 2002.

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

[22] 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.

[23] M. Kreuseler, N. Lopez, and H. Schumann. A scalable framework for information visualization. INFOVIS '00, pages 27–, Washington, DC, USA, 2000. IEEE Computer Society.

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

[25] L. Lins, J. T. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, 2013.

[26] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

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

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

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

[30] K. Morton et al. Support the data enthusiast: Challenges for next-generation data-analysis systems. *PVLDB*, 7(6):453–456, 2014.

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

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

[33] Pedregosa et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

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

[36] T. Siddiqui et al. Fast-forwarding to desired visualizations with zenvisage. In *CIDR*, 2017.

[37] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran. Effortless visual data exploration with zenvisage: An expressive and interactive visual analytics system. In *PVLDB*, 2016.

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

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

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

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

[42] M. Voigt et al. Context-aware recommendation of visualization components. In *eKNOW'12*, pages 101–109, 2012.

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

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

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

[46] K. Wongsuphasawat et al. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization & Comp. Graphics*, 2015.