VizDeck: Self-Organizing Dashboards for Visual Analytics

Alicia Key
University of Washington
akey7@uw.edu

Bill Howe
University of Washington
billhowe@cs.washington.edu

Daniel Perry
University of Washington
dbperry@uw.edu

Cecilia Aragon
University of Washington
aragon@uw.edu

ABSTRACT

We present VizDeck, a web-based tool for exploratory visual analytics of unorganized relational data. Motivated by collaborations with domain scientists who search for complex patterns in hundreds of data sources simultaneously, VizDeck automatically recommends appropriate visualizations based on the statistical properties of the data and adopts a card game metaphor to help organize the recommended visualizations into interactive visual dashboard applications in seconds with zero programming. The demonstration allows users to derive, share, and permanently store their own dashboard from hundreds of real science datasets using a production system deployed at the University of Washington.

1 Introduction

Science has been transformed by automated high-throughput data acquisition technology [2, 8, 21]. DNA sequencers, high-resolution simulations of the Earth, satellite imagery, terabytes of telescope imagery [19], and a national network of oceanographic sensors [6, 11] have transformed research from data poor (data *acquisition* was the bottleneck) to data rich (data *analysis* is the bottleneck).

As datasets grow in size and complexity, human attention becomes the limited resource. Figure 1 illustrates the problem: Data sizes and computing resource are both growing exponentially, but human cognitive capacity remains essentially flat[1]. This effect is measurable: In a recent survey, our collaborators report that the ratio of time they spend "manipulating data" as opposed to "doing science" is approaching 9 to 1 [10]. As a result, researchers increasingly rely on visual techniques for exploratory analysis to quickly identify patterns and generate hypotheses. We originally suggested sophisticated visual analytics tools (*e.g.*, Tableau s [22]), but found that scientists were not able to self-train quickly and reverted back to simpler tools such as Excel.

[1]This illustration derived from a slide by Cecilia Aragon at the University of Washington

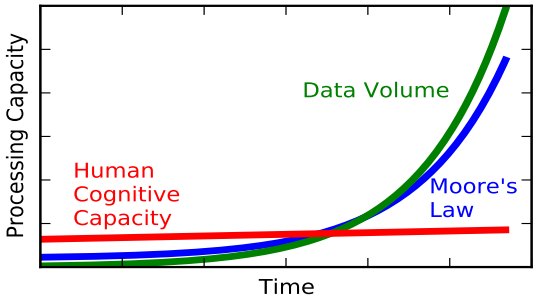

Figure 1: Data sizes and computing resource are both growing exponentially, but human cognitive capacity remains essentially flat.

We explore a different approach. Informed by the statistical properties of the underlying dataset, VizDeck generates a "hand" of ranked visualizations (and various control widgets), and the user plays these "cards" into a dashboard template, where they are automatically synchronized into a coherent web application that can be saved and shared with other users. By manipulating the hand dealt — playing one's "good" cards and discarding unwanted cards — the system learns statistically which visualizations are appropriate for a given dataset, improving the quality of the hand dealt for future users.

Dashboards can be saved and shared among collaborators simply by exchanging URLs. Server-side, a saved dashboard is represented as a *replay log* of the actions a user took to contruct it. Besides affording a useful undo/redo feature, this technique allows a collaborator to review the steps taken by the original author to create the dashboard, which we anticipate will improve cross-training and communication between users.

VizDeck is designed to address critical open problems in the intersection of visualization and data management. Objective assessment of visualization quality is considered one of the grand challenges in the visualization community [12], and is a prerequisite for automatic visualization of data. Seminal approaches to this problem involved heuristics based on human perception [14]. We allow developers to encode such heuristics when they exist to control the recommendations, but augment them by training a model that learns the relationship between statistical features of the data and visualization properties. Our system is the first to make such an association and show that it can be used to improve

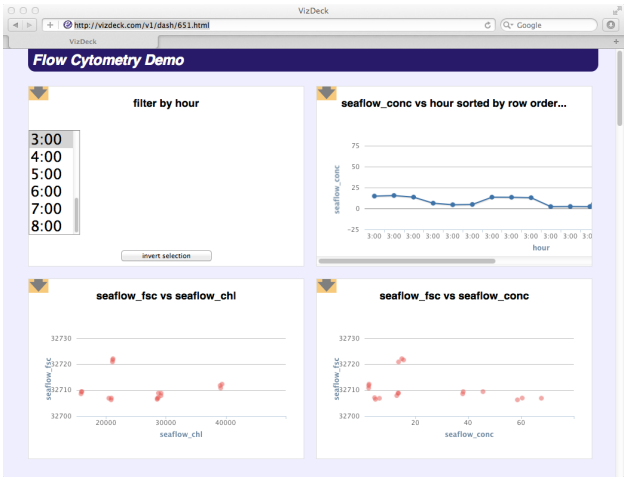

Figure 2: Users can create interactive dashboards in seconds with zero programming by allowing the system to automatically analyze data and recommend a set of appropriate visualizations. We show that with VizDeck, users can complete visualization tasks faster and with higher accuracy compared to existing visual analytics tools.

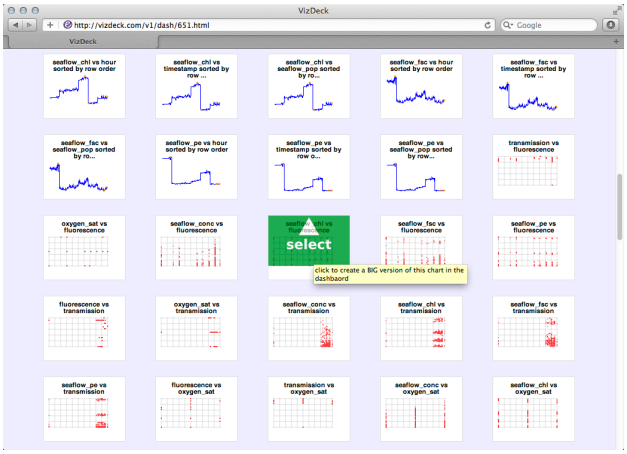

Figure 3: When users hover the top of a thumbnail a green arrow appears. They can click the green arrow to send the thumbnail to the dashboard as a full-sized version.

quality. Moreover, VizDeck offers a partial solution to the "schema chaos" problem we encounter frequently working with scientists [9], where a database may consist of hundreds or thousands of unorganized tables with inconsistent types, unknown relationships, and unreliable quality. By automatically recommending visualizations and streamlining the creation of mashups, we have found that users are able to perform exploratory analysis tasks significantly faster and with higher accuracy than using existing tools [17].

VizDeck is deployed in production and uses 100% real data from researchers in astronomy, oceanography, biology, and other fields. The system is currently deployed to interoperate with SQLShare, a production database-as-a-service application with hundreds of active users in the science community that de-emphasizes pre-defined schemas and focuses on ad hoc integration, query, sharing, and visualization [10]. Figure 2 shows a screenshot of a dashboard created with VizDeck used to analyze real environmental flow cytometry data. A flow cytometer measures the optical scatter and absorption patterns of particles in a narrow capillary to identify microganisms. Invented for medical diagnostics, the technology has been adapted for environmental observation and deployed on commercial vessels to characterize microorganism populations in the open ocean. The goal is to correlate the concentration of various microorganisms with physical observations such as temperature and salinity. The challenge is that very little is known about how these quantities relate, necessitating iterative, trial-and-error visual analytics — the researchers do not always know what they are looking for. With VizDeck, researchers can browse a variety of relationships simultaneously with zero overhead, then create interactive dashboards illustrating interesting relationships to share with their colleagues. A recent user study has shown that this approach outperforms other visual analytics tools that require more up-front training, in particular Tableau [22, 20] and IBM's ManyEyes [15].

2 What Will the User See and Do?

Figures 1-4 show the steps to create a VizDeck dashboard. Visitors will be guided through these steps as part of the demonstration. All public datasets in the SQLShare database are available for analysis with VizDeck. Visitors will also be free to explore these datasets and create and share their own dashboards.

Here are the steps to create the dashboard of Figure 2:

1. *Step 1:* Users will select one or more datasets from a list of real public datasets in the SQLShare system. In the running example, we select the *flow cytometry* dataset.
2. *Step 2:* Users are presented a ranked list of thumbnail-sized vizlets (Figure 3). The rank is determined by training a model to predict a score based on statistical features of the dataset plus validity heuristics provided by the vizlet developer. This process is described in Section 3. Users can *promote* thumbnails to the dashboard for closer inspection, or *discard* them to send them to the bottom of the list (not shown).
3. *Step 3:* VizDeck supports keyword search to help users browse the recommended vizlets. On the server, each protovizlet contains an array of keywords as a property. The keywords are the names of the attributes, the type of the visualization (bar, line, etc), and any other keywords the vizlet author deems appropriate. Multiple keywords delinited by spaces are treated with OR semantics. An autocompletion menu suggests keywords as the user types.
4. *Step 4:* As a dashboard is created, users can interact with it to explore the data (Figure 2). All the vizlets (both in the dashboard and in the deck) respond in real-time to user input on the dashboard.
5. *Step 7:* At any time, the user can share their dashboard by copying the URL found in the upper-right corner of the dashboard and sending it to others. When another user accesses the URL, they also get access to the creator's replay log, enabling them them to not only see the final dashboard, but also to replay or undo

the steps used to create it.

Dashboards created by visitors are saved in the system permanently and will be accessible over the web during and after the conference — visitors will be able to browse a list of other dashboards created by other users during the conference. To encourage users to create compelling dashboards, visitors can rate other users' dashboards; this list will be visible during and after the conference.

3 Dashboard Model

Our approach to (semi-)automatic visualization does not involve synthesizing one "perfect" visualization; rather, VizDeck facilitates creating interactive ensembles of visualizations by selecting from a list of candidates. We believe a composition of of interactive visualizations of 1-2 dimensions is more communicable than a single complex visualization superimposing multiple dimensions. To this end, VizDeck streamlines the trial and error process of creating these ensembles by enabling users to create dashboards and test them quickly. In particular, the user can browse a ranked list of visualizations and see what they are adding to the dashboard before they add them. This "knowledge in the world" [16] delivers improved speed and accuracy over more complex tools in preliminary user studies.

Vizlet Model A Vizlet is a 4-tuple (f_t, f_i, p, s), where f_t and f_i are *render functions* $Dataset \rightarrow Visualization$ producing thumbnails and interactive versions of the vizlet, respectively; p is a *predicate* $Dataset \rightarrow Bool$ indicating compatibility with a given dataset; and s is a *scoring function* $Dataset \rightarrow [0, 1]$ that is one of several signals used for ranking. This model is very general. In particular, both conventional visualizations and UI controls can be considered vizlets: both display data, and both optionally accept user input. This approach allows us to rank and recommend UI widgets alongside visualizations to facilitate the construction of interactive dashboards.

Event processing As you select thumbnails and work with full-size visualizations, the thumbnails collectively respond to the input. Specifically, highlighting points in one chart highlights the corresponding points in other charts, and establishing a data filter in one chart filters the display in other charts. Multiple filtering widgets are interpreted as a conjunctive expression, while multiple selections in a single widget are interpreted as a disjunction. Each vizlet type must respond to these filter and highlight requests, and each vizlet has methods to do this; however, the exact semantics of these events differ depending on the visualization, and vizlet designers are free to define the behavior of their vizlets in response to these events as they see fit.

Vizlet Types The vizlet types supported in the current prototype are scatter plots, histograms, bar charts, pie charts, timeseries plots, line plots, maps, and two kinds of UI widgets: drop down boxes and multi-select boxes. VizDeck is designed for extensibility to allow third-party development of new vizlet types (see Section 4). We envision a community resource where visualization designers can register their code with the VizDeck server, express the compatibility rules for their visualization through the functions p and s, and VizDeck can begin to automatically generate and recommend the visualization going forward.

Ranking Visualizations To make recommendations, we train a model of visualization quality that relates statistical features of the dataset to particular vizlets. We use log data collected by the system as ground truth. Each time a user promotes a vizlet, we record a vote for the vizlet — it was considered important enough to inspect more closely. We then attempt to predict this score from a set of features extracted from the data. Specifically, for each pair of columns (x, y), we extract the following for both x and y.

- Distinct Values: The number of distinct values. (Intuition: Visualizations such as bar charts assume a small number of distinct values.)
- Entropy: The number of distinct unique values in a column divided by the cardinality. (Intuition: Visualizations such as line charts assume that each x-value is unique.)
- Coefficient of Variation: The variance divided by the mean. (Intuition: Visualizations that are too "clumped" have a low coefficient of variation.)
- Kurtosis: The fourth moment about the mean. (Intuition: Kurtosis is an indicator of how outlier-prone a distribution is; visualizations that fit the axes to the data are sensitive to outliers.)
- Periodicity: The variance of the gap length between successive values. (Intuition: Line charts assume approximately regular sampling of the data; otherwise, the line between two successive points is misleading.)

4 System Implementation

The VizDeck system is comprised of a javascript client and a database-backed server written in Ruby. The server is responsible for retrieving data, analyzing it for compatibility with vizlets, ranking the results, and transmitting the ranked vizlets to the client for display. The client renders the vizlets in the browser and manages interaction. The server passes information to the client via compact representations of vizlets called *protovizlets*. A protovizlet is a set of (key,value) pairs that provide enough metadata to describe the visualization or UI widget so it can be rendered on the client side. Vizlet designers write both the javascript and Ruby components, and are therefore free to specify the content of the protovizlet.

Search VizDeck allows keyword search over the recommended vizlets to facilitate browsing. Each vizlet is indexed on its title, the attribute names involved, and a vizlet-specific description provided by the designer. The prmary use we have seen of the search box is to quickly find vizlets involving specific attribute names. To this end, the search box autocompletes the names of attributes. We chose this method rather than an explicit list of attributes to save real estate and capitalize on user familiarity with keyword interfaces. The search box suggests another application for VizDeck that hope to explore: a global search engine for visualizations — including those that have not yet been created, but could be upon request.

Replay Log and Saved Dashboards The client logs all user actions: promotions, discards, filter events, and highlight events. These entries are bundled and periodically sent to the server. These observations are stored on the server and are used for two purposes. First, we mine the UI actions to see what users do with various types of vizlets to inform later recommendations. Second, these UI actionas serve as a replay log to recreate a previous dashboard saved at a particular URL.

When the user retrieves a URL for a saved dashboard, the server retrieves the data and creates protovizlets as usual. It also sends back the replay log for the original dashboard that is being retrieved. The client "replays" this log using the UI actions that originally created the dashboard. The result is the state the user left the dashboard in before. A user can create a private copy of a dashboard at any time to prevent conflicts with other users.

5 Initial Results

An initial user study with 32 participants compared VizDeck with IBM ManyEyes, Google Fusion Tables, and Tableau. The test data were properties of the chemical elements that participants could explore with the various tools. Participants were asked to create a dashboard based on the data and to answer questions regarding basic trends of properties of chemical elements.

Users were able to complete tasks in less time and with higher accuracy with VizDeck than with either ManyEyes or Tableau [17]. We were competitive with Google Fusion Tables, but found that in many cases users were relying on the spreadsheet-like raw table interface in Fusion Tables rather than the visualization features. With larger-scale datasets, this technique becomes less feasible.

6 Related Work

The VizDeck system builds on seminal work on automatic visualization of relational data using heuristics related to visual perception and presentation conventions [18, 13]. More recent work on intelligent user interfaces attempts to infer the user's task from behavior and use the information to recommend visuaizations [7]. Dörk et al. derive coordinated visualizations from web-based data sources [3]. Mashup models have been studied in the database community [1, 4, 5], but do not consider visualization ensembles and assume a pre-existing repository of mashup components.

7 Conclusions and Future Work

VizDeck aims to enable user to create context and task appropriate interactive visualization dashboards for their data. Our early use of this tool with data in domain science research has enabled our users to quickly produce numerous dashboards.

Future work includes identifying new applications, increasing use for the technology, refining our ranking function, adding new vizlet types, and connecting VizDeck to other data repositories such as Google Fusion Tables and Microsoft data market.

8 References

[1] S. Abiteboul, O. Greenshpan, T. Milo, and N. Polyzotis. Matchup: Autocompletion for mashups. In *ICDE*, pages 1479–1482, 2009.

[2] S. Baker, J. Berger, P. Brady, K. Borne, S. Glotzer, R. H. D. Johnson, A. Karr, D. Keyes, B. Pate, and H. Prosper. Data-enabled science in the mathematical and physical sciences. Technical report, National Science Foundation, March 2010.

[3] M. Dörk, S. Carpendale, C. Collins, and C. Williamson. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14:1205–1212, November 2008.

[4] H. Elmeleegy, A. Ivan, R. Akkiraju, and R. Goodwin. Mashup advisor: A recommendation tool for mashup development. In *ICWS '08: Proceedings of the 2008 IEEE International Conference on Web Services*, pages 337–344, Washington, DC, USA, 2008. IEEE Computer Society.

[5] R. J. Ennals and M. N. Garofalakis. Mashmaker: mashups for the masses. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 1116–1118, New York, NY, USA, 2007. ACM.

[6] B. Gegosian. Ocean observing initiative, 2005. `http://www.oceanleadership.org/ocean\_observing`.

[7] 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.

[8] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[9] B. Howe. Sqlshare: Database-as-a-service for long tail science. http://escience.washington.edu/sqlshare.

[10] B. Howe and G. Cole. SQL Is Dead; Long Live SQL: Lightweight Query Services for Ad Hoc Research Data. In *4th Microsoft eScience Workshop*, 2010.

[11] The integrated ocean observing system. `http://ioos.gov/`.

[12] C. Johnson. Top scientific visualization research problems. *IEEE Comput. Graph. Appl.*, 24(4):13–17, 2004.

[13] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5:110–141, 1986.

[14] J. Mackinlay. Automating the design of graphical presentations of relational information. In *Readings in information visualization: using vision to think*, pages 66–82. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[15] Many eyes. `http://services.alphaworks.ibm.com/manyeyes/home`.

[16] D. A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison Wesley, 1994.

[17] D. Perry, A. Key, B. Howe, and C. Aragon. Sparking visual discovery: Evaluating vizdeck as an interactive visualization system. In *CHI 2012 (submitted)*, 2012.

[18] S. F. Roth and J. Mattis. Data characterization for intelligent graphics presentation. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 193–200, New York, NY, USA, 1990. ACM Press.

[19] Sloan Digital Sky Survey. `http://cas.sdss.org`.

[20] C. Stolte and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8:52–65, 2002.

[21] A. Szalay and J. Gray. 2020 computing: Science in an exponential world. *Nature*, 440(7083):413, 2006.

[22] Tableau. `http://www.tableausoftware.com/`.