Quarry: Picking From Examples to Explore Big Data

Rhema Linder
Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110 USA
rlinder@adobe.com

Eunyee Koh
Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110 USA
eunyee@adobe.com

CHI'15 Extended Abstracts, Apr 18–23, 2015, Seoul, Republic of Korea.
ACM 978-1-4503-3146-3/15/04.
http://dx.doi.org/10.1145/2702613.2732933

Abstract

Analysts use scripts, visualization tools, and spreadsheets as they process and understand data. We focus on two phases of analysts' work: discovery, where the field definitions are understood, and profiling, where assumptions are tested by searching, observing, and running counts on data. Lack of data exploration and understanding can lead to faulty assumptions and misinterpretation. In practice, analysts use SQL queries and scripts to subset big data, reducing it for visualization or spreadsheet pivots. However, due to large-size and high-dimensional data, it is challenging to determine precise subsets of interest without thorough data exploration and discovery.

We reduce the cost of previewing subsets by combining search with an information rich visualization of high-dimensional data. To enable discovery and profiling, Quarry supports (1) rapid query generation and visualized search; and (2) defining and previewing subsets of data for potential export for further processing. This work presents the design of Quarry and results from a formative study involving 11 analysts/data scientists and a dataset with 80 columns and 15 million rows.

Author Keywords

visualization; query generation; big data; search

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction

Enterprise analysts and data scientists spend time observing and manipulating data to understand it [6]. Kendel et al. interviewed enterprise analysts about how they worked with big data. They describe archetypal analyst types: hacker, advanced, scripter, and application user, as well as five phases in a non-linear workflow. *Hackers* and *scripters* perform this task with a combination of SQL queries, writing small scripts, visualizing statistics using tools such as R [3], and pivoting tables via Excel. Once data size scales to many gigabytes or terabytes, they become impractical for workstation tools to load, process, and visualize data. Before using many workstation tools (e.g. Excel, R), analyst may need to export a subset of data.

The present research focuses on hacker and scripter analysts during the *discovery* phase, where the field definitions are understood, and *profiling*, where analysts test assumptions by 'playing' with data. We designed Quarry to support analysts by enabling (1) *rapid query generation and search for exploring data and testing assumptions*; and (2) *defining and previewing subsets of data for potential export into the program of choice.* We visualize high-dimensional data to afford discovery, which helps analysts understand fields. We combine visualization and search, which enables seeing examples of the data before preview and export during profiling.

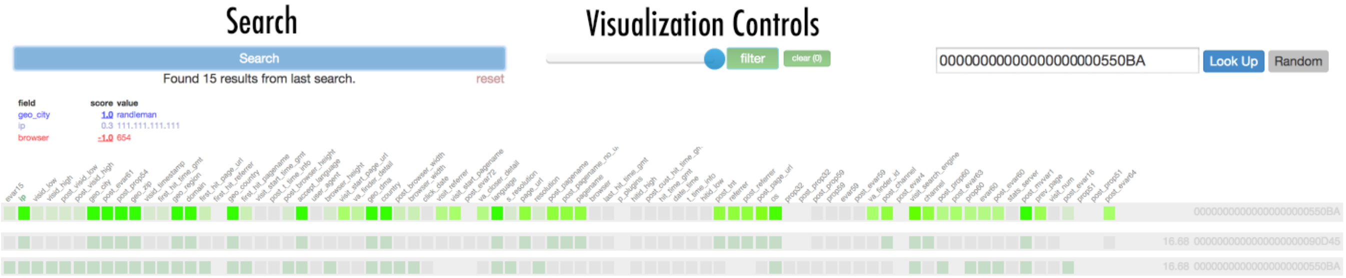
We introduce *picking*, a novel query generation technique which builds queries using fine-grained selection of example cells. We represent data as interactive cells in rows and fields (columns). Cells reveal their value when brushed, showing details on demand. Picking generates a query for matching a cell's field and value. When repeated on multiple cells, queries are integrated to form a search that can add, remove, and rank results. This provides similar expressiveness to very basic SQL and HIVE [1], but can be generated more quickly with the mouse.

This work makes the following contributions: *(1) A novel technique for generating queries quickly, using values from example data. (2) A visualization system for high-dimensional data to support profiling and discovery tasks.* In our formative study, we collected feedback from 11 data scientists that used Quarry to search through clickstream data with 15 million rows and 80 fields. We discuss their feedback on how the Quarry might support them in working with big data.

Related Work

Inasmuch as search results are visualized, Shneiderman's classic mantra, *overview first, zoom and filter, then details-on-demand* [8] applies. Visual attention and cognition is limited. Systems are beginning to push interactions beyond the "query response" paradigm [11] to better support exploration. We introduce a method for building queries from fine-grained example selection.

We see the goals of discovery and profiling phases [6] of analysts' to be similar to searchers'. Searchers may lack domain expertise to know what to look for and express queries effectively [11]. Hearst [5] argues that visualizing search results can help motivated people understand nominal data. Wilson et al. surveyed search interfaces and visualization techniques [12]. Typically, they include a set of documents arranged by a a handful of facets, such as year and citations for scholarly articles.



Found 15 results from last search.

field	score	value
geo_city	1.0	randleman
ip	0.3	111.111.111.111
browser	-1.0	654

00000000000000000000550BA

	00000000000000000000550BA
16.68	000000000000000000090D45
16.68	00000000000000000000550BA

Figure 1: The Quarry visualization system. The top right shows the look up section, which set the start row. The top left shows the current combined query. Right now, geo_city is set to be most important, followed by an IP address query. The browser is not included, thus, results show results only when fields geo_city is "randleman" and browser is not equal to 654. The 15 results are further ordered by ip address. In a browser, there are many results that can be shown by scrolling down. See Figures 2, 3, 4, and 5

.

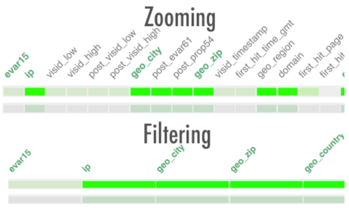


Figure 2: Zooming (affected by the slider) and filtering (activated by click field names, the "Filter").

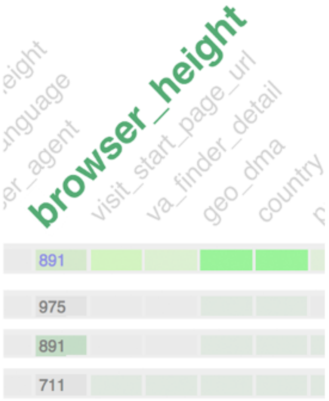


Figure 3: Users can brush over columns to see details-on-demand for each cell.

Visualized search systems typically summarize document based results. Our approach makes highly dimensional data fields and details visible. Search by example systems can take a document or multimedia as queries [12]. Quarry users can select cells to directly build and combine search terms, a more fine-grained approach in both query generation and visualization.

Quarry might be compared to other more complex systems such as Splunk [9] or Tableau [4]. At this stage of research, we focuses on the profiling and discovery stages of analytics. Future work could better integrate Quarry as an internal tool or export interoperable formats. Splunk is a data ingestion system used for developing reports based on machine data. It is useful for monitoring system load, or the number of events of a particular kind. Splunk supports scripts and hand crafted queries for search, rather than generate queries through example cells.

Quarry

Quarry has three main components: a search service, bulk importer, and an interactive visualization (see Figure 1).

We created an importer for converting data from CSV into Elasticsearch [2]. For each unique ID, we create a multi-field 'document' with multiple fields. This structure enables searching by ranking and filtering individual fields. While this structure support any CSV data, we use clickstream data with over 15 million records and 80 fields. Because of grouping, this creates 3.5 million documents.

Quarry's visualization is built with jQuery and the D3 framework. Generally, Quarry supports three actions: look up, visualizing high-dimensional data, and search. *Look up* enables users to gather a sample by ID or load in a row of data randomly. This provides a starting point for investigation. To get started, users can type or paste in an ID then click 'Look Up' or use 'Random' to load an example row of data. We call this the *start row*. *Visualizing high-dimensional data* enables transforming

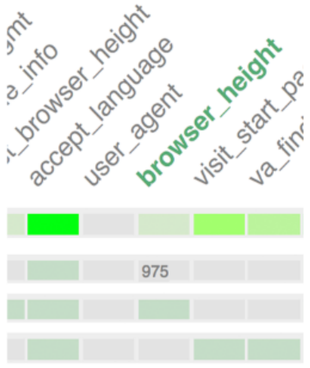


Figure 4: Users can brush over individual cells.

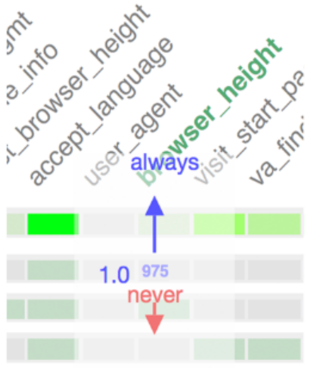


Figure 5: Seeing the value 975 on the browser field, the user clicks, then drags up to add $browser = 975$ as a term to the query.

views to zoom, filter (Figure 2), or see details-on-demand (Figures 3, 4). *Search* enables generating queries by clicking, dragging, and releasing one or more cells.

The visualization system represents all pieces of data as rectangular cells. Rows are grouped by ID. Each row contains a interactive cell for each of its fields. We map the color of each cell relative to the start row. Cells are green when their value matches the start row, grey when they do not match, and missing when the value is NA. The start row works as a heatmap. Its rectangles are bright green when the value matches the fields beneath it. When a rectangle is brushed with its cursor, it shows its value. Users can quickly brush through to investigate values, potentially using them in new queries.

The search system uses Elasticsearch [2]. Elasticsearch is an open source search platform developed by Apache. We take advantage of its multi-query feature, which affords filtering (boolean), scoring (ranking), and boosting among fields. Boosting changes the rank of results when multiple fields are searched simultaneously. In ranked search, documents are ranked by an overall score. Boosting changes how much each field affects a document's overall score. For example, setting a boost of 2.0 on title, and 1.0 on body introduces a bias toward matches in the title. By given users direct control of boosting, Quarry enables adjusting the bias of rank to reflect user interest across all fields.

Quarry's *picking* technique enables fine-grained construction of queries sourced with example data. *Picking* refers to the actions of miners in search of valuable resources that pick to sample and choose where to dig. To pick, a user brushes over a cell (Figure 4), clicks on a value, then moves the mouse up or down (Figure 5), and finally releases the mouse. This adds that value as a search query on that column, to the current query (see under "Search" button of Figure 1). For example, brushing over browser_height, clicking on "975", dragging up, then releasing adds the query (see Figure 5). Vertical movement controls the magnitude of ranking. Inspired by the in-context-slider [10], picking uses a single click to select and weight a term. This can be important when search occurs among multiple fields.

Users can integrate queries by picking multiple cells. After clicking on a cell, enough mouse movement up or down activates an ALWAYS or NEVER operation respectively. An ALWAYS (boolean AND) operation filters out documents without that field and value. A NEVER (boolean NOT) operation filters out documents that have that field and value. Moving from the middle to high will produce scores between 0 and 1, which affect the strength of the score for that term. While using ALWAYS and NEVER operations affords creating subsets, ranking helps order results. Once a user clicks Search, the top ranked results from the current query are visualized. Thus, it uses examples to drive new queries.

For example, the query in Figure 1 uses an ALWAYS on geo_city=randleman, a NEVER on browser=564, and a score of $.3$ on ip=111.111.111.111, resulting in 15 results. While these results are ranked by ip=111.111.111.111, "documents" that have ip≠111.111.111.111 are still included. Note: IP addresses are obscured in our dataset. 111.111.111.111 is used for illustration.

Formative Study

We gathered feedback from data scientists at a major tech company. In this, we sought to answer: *How did people use picking? Did the system perform in an understandable way?*

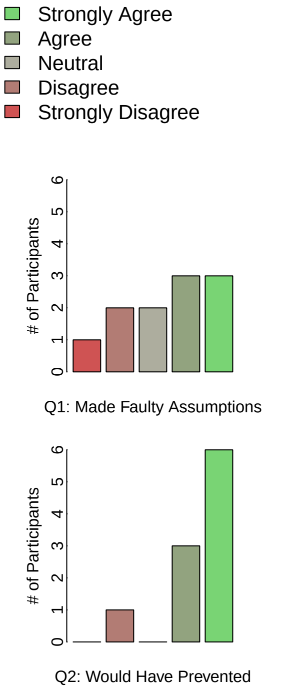


Figure 6: Results from our survey questions. Q1 and Q2 show participants benefit from more testing and exploration. Q1: *In working, I have made assumptions about the data or schema that I later found were wrong*. Q2: *Exploring the data more could have prevented me from making these wrong assumptions*.

9 data scientists and 2 PhD students (majoring data science related field) participated in a 30 minute user study. Participants had a median of 4.5 years of big data experience. After giving a brief tutorial, we asked participants to explore the system and data freely. Once they were comfortable, we asked them to perform search tasks. We ended sessions with a short discussion and survey about their experience. As they used Quarry, we observed their actions, made notes, and recorded audio as they narrated their actions.

Our participants performed multiple searches through picking. They looked at the highlighted fields and brushed through the columns and cells of data. The dense, yet explorable visual presentation of data surprised people.

> U6: This is a good way to visualize a large number of columns and labels across the top.
>
> U3: I like that it gives you a summary. Looking at just this record, you can see how different or similar [values are].. . . Your eyes are much more used to these colors. They can make sense of these colors much more quickly than they can make sense of these numbers.

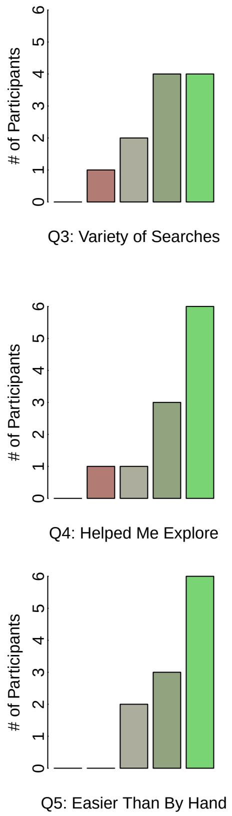
All participants used picking to answer and test assumptions and get a feel for the data. Once participants were comfortable picking, we gave them search tasks. Starting from a prepared row of clickstream data, we asked to to find: (1) the number of visitors that shared the same IP address, and (2) the number of people from the same city who used a different browser. The tasks were designed to ensure participants could use Quarry and to provoke discussion. As the participants performed searches through picking, they related the searching to SQL statements or HIVE queries. Most thought Quarry supported them in exploring a variety of searches, that would be slower when typed by hand. At the same time, while potentially faster for many queries, Quarry is not as flexible as hand crafted code.

> U2: Everything [Quarry does], I could write code and get anything that way. But this type of thing helps you ask these questions really quickly. So, you can get a feel for what's in your data for these types of queries.
>
> U8: So you issue SQL command here, you get some results. . . . Then, we click on the different field . . . and we hover here, we get even more values . . . We are expanding on a SQL statement by [picking].

Despite this advanced knowledge for working with data, a majority of our participants recalled making faulty assumptions. The survey (Figure 6) shows our participants could have avoided faulty assumptions by better exploring the data. To become familiar with data, our participants reported ordinarily using Hive and SQL queries. Our participants saw Quarry as most helpful for quickly issuing a variety of SQL like queries and Quarry made it easier to search and explore data (Figure 7).

Conclusion

Quarry's visualization system is designed to help users explore high-dimensional data before writing scripts or exporting. In current workflows, analysts develop SQL or Hive queries and scripts to subset data, before loading it in their preferred analysis tool [6]. Instead, our results suggest fine-grained search is a helpful alternative for better understanding data before export. Our approach



Q3: Variety of Searches

Q4: Helped Me Explore

Q5: Easier Than By Hand

Figure 7: Results from our survey questions. Q3, Q4, and Q5 show participants felt Quarry supported them in exploring quickly. Q3: *I was able to perform a variety of searches.* Q4: *Searching by example (picking) helped me explore the data set.* Q5: *Searching by example (picking) would be easier than typing search queries by hand.*

takes advantage of visualization and search techniques, making them more fine-grained.

Our participants were able to get "U2: a feel for the data". Thus, Quarry enables query generation from natural mappings. The participants searched clickstream data with over 15 million rows and 80 columns. In a study, participants reported Quarry helped them explore this data quickly by enabling a variety of different queries.

Most of our study participants said they made assumptions about the data or schema that they later discovered to be wrong, and more data exploration could have prevented from making faulty assumptions. As analysts need to manage bigger and higher dimensional datasets nowadays, we believe better big data exploration tools are critical for accurate data analysis and interpretation.

Our study shows encouraging potential for Quarry to support big data exploration for analysts, specially during the "discovery" phase, where the field definitions are understood, and "profiling", where analysts test assumptions by "playing around" with the data. This work can be extended for other datasets and contexts.

Future work could also add more functionality, such as supporting aggregate statistics of numeric data, structured data, and filtering ranges. Alternatively, a Quarry like view and functionality could be added to existing tools. This would enhance existing palettes [7] used for exploratory search.

References

[1] Apache hive. https://hive.apache.org/.

[2] Elasticsearch. http://www.elasticsearch.org/.

[3] The r project for statistical computing. http://www.r-project.org/.

[4] Tableau software. http://www.tableausoftware.com/.

[5] Hearst, M. *Search user interfaces.* Cambridge University Press, 2009.

[6] Kandel, S., Paepcke, A., Hellerstein, J. M., and Heer, J. Enterprise data analysis and visualization: An interview study. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (2012), 2917–2926.

[7] schraefel m.c., Wilson, M., Russell, A., and Smith, D. A. mspace: Improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM 49*, 4 (Apr. 2006), 47–49.

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

[9] Stearley, J., Corwell, S., and Lord, K. Bridging the gaps: joining information sources with splunk. In *Proceedings of the 2010 workshop on Managing systems via log analysis and machine learning techniques*, USENIX Association (2010), 8–8.

[10] Webb, A., and Kerne, A. The in-context slider: a fluid interface component for visualization and adjustment of values while authoring. In *Proc working conference on Advanced visual interfaces*, ACM (2008), 91–99.

[11] White, R. W., and Roth, R. A. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services 1*, 1 (2009), 1–98.

[12] Wilson, M. L., Kules, B., Shneiderman, B., et al. From keyword search to exploration: Designing future search interfaces for the web. *Foundations and Trends in Web Science 2*, 1 (2010), 1–97.