Answering questions with line and bar graphs

Liang Dong, Robert St. Amant

Department of Computer Science, North Carolina State University

Abstract. This paper describes our work in progress toward a practical tool to improve the ability of data analysts to extract specific information from simple graphical presentations. Toward this goal, we have developed cognitive models that carry out the necessary perceptual and cognitive operations to answer specific questions about data presented either as a line graph or a bar graph, concerning point reading (quantity estimation or comparison to a known value), item comparison, and trend estimation. These models run in the SIMCog framework, which allows models to interact with a Web application running in a browser. The models are a reasonable match to human data gathered in a pilot study; generalization and model validation remain for future work.

Keywords: cognitive model, graphical presentation, eye tracking

1 Introduction

How well does a given graphical presentation of data convey specific information? Visual representations of data in some graphical form (we'll refer to these in general as visualizations) have become a pervasive part of interactive online systms. Line graphs and bar graphs, some of the simplest presentations available, show trends in the economy, election results, and comparable information. Although there are many popular libraries like d3.js and chart.js to generate graphs for presentation on the Web, these tools generally do not extend to assisting in the choice of a given presentation technique to convey specific information. The approach described in this paper is a first step toward producing recommendations to improve decisions about this choice.

It is well known that for specific tasks, some visualizations are superior to others; for example, bar graphs are superior to pie charts presenting the same information, in terms of efficiency and accuracy in general. What has been explored only to a lesser extent, however, is how the relative effectiveness of different visualizations depends on the data being displayed. Our long-term goal is to develop an automated system that can give the designer of a visualization a recommendation for the most effective type of visualization *for a given dataset.* We are working with intelligence analysts who look at a variety of visualizations, often in a sensemaking context.

Our approach involves developing ACT-R cognitive models of common tasks required to extract information from simple visualizations. Currently, at this stage in the project, our work is restricted to line graphs and bar graphs. From a practical perspective, we are also interested in making such results easily available to people who build visualizations. Our models run in SIMCog-JS [3] and interact with Web browser-based software, as shown in Figure 1.

Fig. 1. Model execution environment

In the remainder of this paper we will briefly describe related work, then the path we have followed to the present. First is a user study, with gaze tracking, in which users answered specific questions, looking at different types of graphs. Second is our modeling work to reproduce user performance. Our results are preliminary, with models having been fit to user performance, for a small set of users; they have not yet been validated in a larger experiment on more users and visualizations generated from different datasets.

2 Related Work

Visualization design, even for simple cases such as line graphs and bar graphs, has seen continuing interest among psychologists, computer scientists, statisticians, and data analysts. For example, Cleveland's work [1, 2] focuses on the graphical presentation of data for scientific or technical purposes, with an emphasis on accurately conveying large amounts of information so as as to make decoding easy and effective. Cleveland begins with basic principles of graph construction, looking at ways of making the data stand out.

Kosslyn [4], in contrast, targets the communicator-of-results rather than the pure statistician developing visualizations for expert colleagues or the artist producing visual effects that may create an emotional impact on the viewer. Tukey famously remarked that a good visualization "forces us to notice what we never expected to see;" Kosslyn's focus is on clear communication of what the analyst has already noticed.

Researchers have examined principles of good data presentation from a cognitive perspective. Influential early work is by Lohse [5], whose UCIE (Understanding Cognitive Information Engineering) system simulates graphical perception for simple visualizations. UCIE predicts response time to answer a question posed to a graphic display from assumptions about the sequence of eye fixations, short-term memory capacity and duration limits, and the degree of difficulty to acquire information at each glance. An empirical study compared actual performance to UCIE predictions over a range of display types and question types. The results yielded some support for the cognitive model. UCIE simulates how people answer certain questions posed to bar graphs, line graphs, and tables. It can process three types of queries: point reading, comparisons, and trends. Point-reading questions refer to a single datapoint . Comparison questions refer to a pair of adjacent data points. Trend questions refer to a range of successive data points.

Peebles and Cheng [6] describe an experiment and eye movement study, the results of which show that optimal scan paths assumed in the task analysis approximate the detailed sequences of saccades made by individuals. Their research demonstrates the computational processing non-equivalence of two informationally equivalent graphs and illustrates how the computational advantages of a representation can outweigh factors such as user unfamiliarity. Peebles and Cheng even describe ACT-R models of their tasks. The work described in this paper differs by considering a different type of graph; further, our modeling work is not as far advanced.

3 A Pilot User Study

We take line and bar graphs as representative of the type of familiar graphical information visualization relied on by analysts. Our review of the information visualization and graphical perception literatures did not lead to a set of tasks applicable to bar graphs, however. Using a generic set of primitive task operators identified by Lohse [5], and based on general guidelines in the visualization literature, we developed a plausible, small set of tasks that can be carried out with a line graph or bar graph. The tasks can best be understood as answers to questions. Consider the sample line graph shown (colored lines only) on the left of Figure 2:

- Point reading: Is the value of Product B at time 4pm greater than 3?
- Item comparison: At time 3pm, is the value of Product A less than Product C?

– Trend estimation: Is the trend of Product A from 5pm to 8pm up or down?

These tasks can be grouped or chained together to extract information relevant to an analysis from the display. More generally, point-reading questions refer to a single datapoint, requiring estimation and also potentially comparison with a fixed value, in a fixed location. Item comparison questions refer to pairs of data points, for comparison of their values. Trend questions refer to a range of successive data points.

We set up a pilot study with three participants. Each participant looked at a succession of questions of the type shown above, each question followed by a visualization. After the visualization was displayed, it was replaced by a set of choices for the answer. An Eye Tribe gaze tracking system was mounted below the monitor, to record data from which gaze fixations could be tracked. A sample set of fixations, with lines connecting them, is overlaid on the visualizations in Figure 2. In each round, participants answered three questions for each type of visualization, on different data; the rounds were repeated for a total of three rounds per participant.

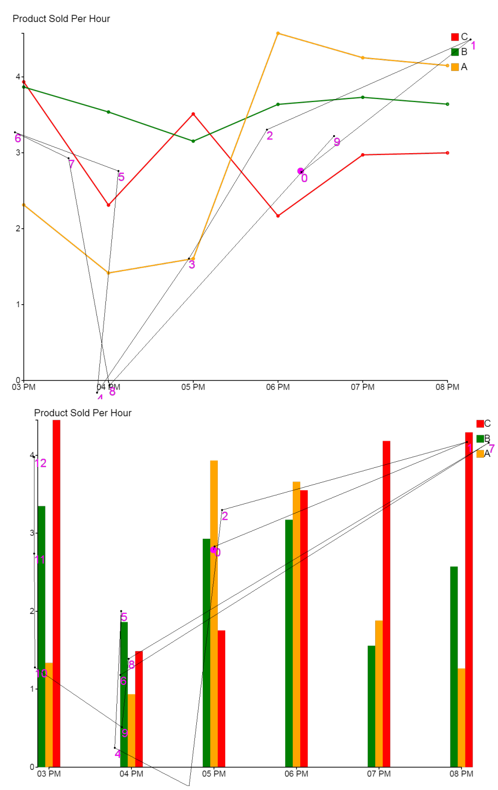

Fig. 2. Visualizations for point reading questions, with fixations overlaid

For our three pilot study participants, the bar graph resulted in a lower mean task duration for point reading and item comparison, while the line graph was superior for identifying trends. These issues are discussed in more detail below.

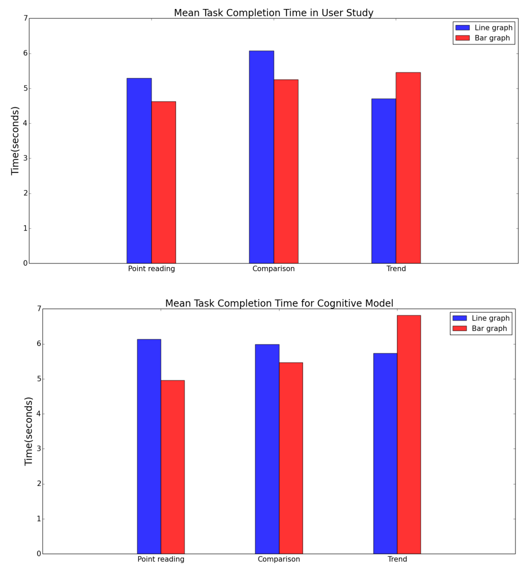

Fig. 3. Mean task duration

4 Cognitive modeling

Our models were built in the ACT-R cognitive architecture. ACT-R is a high-level computational emulation of human cognitive processing, generally including representations of memory, attention, visual and motor processing, problem solving, learning, and related phenomena—to a large extent, the phenomena that we are interested in with respect to analyst performance. A version of ACT-R runs in Java, on a server that can communicate with a client browser.

This arrangement is made possible by SIMCog-JS, a system due to Halverson [3]. SIMCog-JS (Simplified Interfacing for Modeling Cognition - JavaScript) allows models to interact with browser-based software, while requiring little modification to the task code. The modeler specifies how elements in the interface are translated into ACT-R chunks; the software allows keyboard and mouse interaction with JavaScript code, and it allows sending ACT-R commands from the external software. Our framework is based on SIMCog-JS, but we developed graphs based on d3.js and chart.js on the front end. The framework is shown in Figure 4.

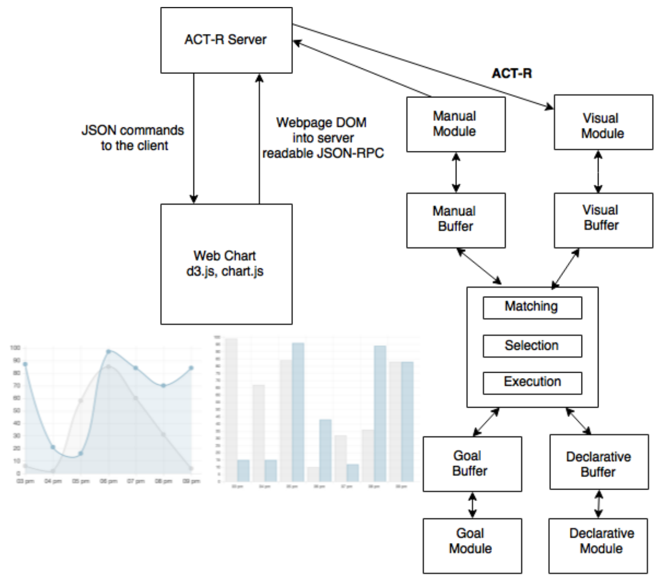

Fig. 4. Cognitive framework based on SIMCog-JS

The basic requirement for the cognitive models is to simulate patterns in how people read and understand line graphs and bar graphs, to derive answers to specific questions. Our pilot study shows preliminary evidence that people do follow general patterns. For types of graphs, fixations generally occur in the following sequence: Legend → X axis → Line or Bar → Y axis. Production rules in the model follow the same pattern, with state information recorded in the goal buffer.

The cognitive models we have developed are still relatively simple, but they roughly performance to answer questions about visualizations, as described in the previous section. The model is initialized with a set of chunks in declarative memory that represent state information (e.g., the first action that the model will take) and domain information (e.g., the mapping of characters such as 1, 2, 3... to an ordering of quantities). Given a task that concerns a specific item, the model begins by looking up and recording the color coding of the item in the legend to the visualization. Depending on the specific task, the model will find and attend to the locations of points in a line graph or values in a bar graph; it will find, attend, and read labels on the x-axis and y-axis as necessary for estimation and comparison.

One complication is that people's scans are not "ideal;" sometimes their fixations revisit the legend, as if they are probabilistically unable to retrieve an association between the symbol named in the question and the color of a line or bar in the graph. For example, on the right of Figure 2 the fixation sequence revisits the legend after moving to the correct bar. To reproduce this behavior, we define several productions that can fire (their conditions will be satisfied during the match) when the model finishes a specific step, such as reading the X axis. One production moves the gaze back to the legend, while another moves to the relevant line or bar. These two productions are randomly selected in the conflict resolution process; we use the utility parameter to tune the probabilities with which productions fire.

Means of task duration for the participants in the pilot study are shown on left of Figure 3 for the three types of questions. Means of model task durations, based on ten runs, are shown on the right. The results appear to be a reasonable match between the model and human performance. Several caveats apply, however: we have fitted these models to a small number of participants in the pilot study, who carried out a small number of repetitions for each type of question, to different datasets. The models have not yet been validated; a larger study with more participants and a greater number of datasets is needed.

We are currently refining the models to improve their match to observed performance—the general patterns, but "errors" as well. The overall goal, as described in the introduction, is to develop models that replicate human performance at a level detailed enough to produce recommendations for a given visualization, tied to the data being visualized.

5 Conclusion

In summary, we have developed cognitive models in ACT-R framework that can answer specific questions about a line graph or a bar graph. Our work can be divided into the following three parts.

For the user study, to better understand how people answer questions based on line graphs and bar graphs, we designed a small pilot study. An eye tracker was used to identify gaze fixations. This data provided a basis for modeling work; the experiment itself also provided a motivation for the integration of experiment and modeling software in a single environment (though this is incomplete.)

We developed ACT-R cognitive models to simulate the basic patterns of how people read line graphs and bar graphs to answer questions. The cognitive models can handle three types of questions: point reading, item comparisons, and trends. The models themselves are not novel; although they were developed within our lab, they replicate the structure of comparable models in the literature. We mainly used utility theory as represented in ACT-R to capture randomness in participant behavior, which we attribute to memory limitations.

As for the architecture, we employed the SIMCog framework (a solution allowing models to interact with web browser based software) to connect Web graphs and cognitive models. The server is built within the Java ACT-R task environment. We use d3.js library to generate graphs in the client; the server interprets JSON-RPC messages from the client about the current status and relays them to ACT-R model.

The paper is a first step in evaluating graph presentation, tied to specific datasets, in a real world Web application. The research is a work in progress, with several limitations. A formal experiment, with sufficient data for statistical analysis, remains to be done. A continuation of user evaluation will also be needed to support model validation. In the future, we will also make the system suitable for other graphs like pie graphs, other types of histograms, and so on.

6 Acknowledgments

Thanks to Tim Halverson for his contribution and assistance with respect to SIMCog. This research was support in part by the National Science Foundation (IIS-1451172) and by the Laboratory for Analytic Science.

References

1. W. S. Cleveland. *Visualizing Data* Hobart Press, 1993.
2. W. S. Cleveland. *The Elements of Graphing Data.* Hobart Press, 1994.
3. T. Halverson, B. Reynolds, and L. Blaha. SIMCog-JS: Simplified Interfacing for Modeling Cognition - JavaScript. *Proceedings of the International Conference on Cognitive Modeling, Groningen, The Netherlands*, April 9-11, 39-44.
4. S. M. Kosslyn. Graph Design for the Eye and Mind. *Oxford University Press*, 2006.
5. G. L. Lohse. A cognitive model for understanding graphical perception *Human-Computer Interaction* 8(4) (1993): 353-388.
6. D. Peebles and P. C.-H. Cheng. Modeling the effect of task and graphical representation on response latency in a graph reading task. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 45(1) (2003): 28-46.