Computational Models of Reflection

Joshua B. Gross
Naval Postgraduate School
Monterey, CA 93940
gross.joshua.b@gmail.com

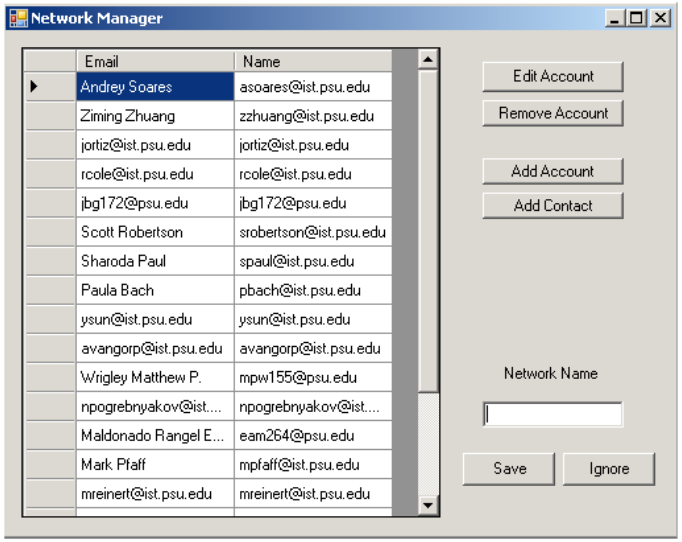

Figure 1. Example Social Cluster

CHI 2010, April 10 – 15, 2010, Atlanta, USA
ACM 1-xxxxxxxxxxxxxxxxx.

Abstract
In this paper, we describe ongoing work on deriving computational models of digital data that can be used to derive information for personal reflection. This paper focuses on the information and its use, rather than the mechanisms used to derive it. The experiments described are based upon two concepts important to reflection: novelty and reinforcement. The first experiment involved extracting social clusters from email, and the second involved extracting time signatures for computer usage.

Keywords
Email, communication, datetime, user studies

Introduction
An individual's computer obviously contains much information about the user, including readily identifiable data such as name, address, and credit card numbers. However, at a deeper level, a user's hard drive contains more information about that user than perhaps any other set of documents or records (Garfinkel & Shelat, 2003). From a psychological perspective, the hard drive is an imperfect mirror that reflects many (although not all) aspects of a person's life. In order to best use this information, the data must be evaluated and analyzed at as sophisticated a level as possible to provide the best possible reproduction of that imperfect reflection.

The focus of this work is to represent two crucial aspects of reflection: novelty and reinforcement. These aspects are not represented by different data, or even data types, but rather sit in juxtaposition to support exploration. Exploration must combine both the familiar, to encourage comfort and trust, and the new, to provide opportunity for learning and growth.

Computational Modeling for User Interaction

The goal of computational modeling (both generally and for user interaction) is explicitly *not* to derive information from a dataset using purely algorithmic approaches. The algorithmic approach assumes neither knowledge about the target information nor how that information will be used.

Instead, computational modeling is designed to structure information into a form useful for the end user. We cannot do this purely algorithmically; we need to understand the data and the user in order to develop computational models.

However, computational models are essential because of information overload. Presenting users with large amounts of information is unlikely to stimulate reflection. It is always possible to present only part of the information, but computational models allow *effective* information compression. Some of the common models of information compression are visualization, sampling, and clustering.

Visualization is attractive because it presents the information in a novel format, typically with one or more scales to allow the user to contextualize the information (Tufte, 1983). Unfortunately, visualization is only a partial solution, as it can present derived data, but the use of graphs and charts challenges the user to comprehend the visualization itself (Green & Petre, 1996). Also, if the visualization does not allow the user to focus on elements and explore, then reflection is unlikely to happen (Zhang, 2009).

Sampling is attractive, because it allows the software to present only *some* of the information, which is much more likely to be comprehensible to the user. However, sampling is a complex process. A quick analysis of email, for example, can show the most common addresses a user sends or receives some, but to what purpose? If attempting to stimulate reflection via sampled data, it may be critical to develop a model of the information that provides a statistical distribution, and then effectively uses that distribution. For example, in a Zipf (also called log-linear) distribution (Zipf, 1935/1965), such as used in information retrieval (Salton, 1987), one would look to the center of the distribution, whereas with a normal distribution, outliers become more important. Everything rests on one essential question: how is the data modeled, and how does that model lead to comprehension.

Clustering is perhaps the most powerful model, because it allows for both visualization and sampling. In a typical clustering process, some set of related data is spread into n-dimensional space, and an algorithm finds data elements that occupy nearby space. The outcome is a set of groups (clusters) of data that are related by the dimensions specified in the space creation, which is really a process of specifying interesting and/or useful information.

Because clustering finds common data, it is able to sample the data based not purely upon frequency, but

on the importance emphasized in the space creation. Once the clusters are created, they can be visually represented, navigated, and applied for different purposes.

SCuF: Social Cluster Filtering

The first set of experiments related to the use of clustering in social media came from an exploratory study in the use of email. Several high-throughput email users (Gross, 2007) were interviewed about their habits. Perhaps unsurprisingly, the issue of groups of communicators was omnipresent, as was the desire to more effectively search large repositories of email.

The computational tool developed to support this work is called Social Clustering Filters (SCuF) (Gross, 2008). SCuF was developed as an Outlook 2003 plugin that uses a custom clustering tool (also developed for this project) called Keystone.

SCuF operates by first producing an index of email messages and the senders and recipients for these messages. The user can select particular subsets of their email to be indexed, but the default is to index all. This index is used to generate clusters; the user specifies the number of clusters and maximum size of each cluster.

In the experimental validation of SCuF, several participants had their email downloaded into Outlook 2003. The software then indexed the participant's email, and allowed the user to generate clusters. Each cluster was presented as a list of email addresses and associated names. The user could then choose to edit the cluster (add or remove people), and then save the cluster, giving it a relevant name. The user could also choose to ignore the cluster. An example cluster can be seen in Figure 1.

The saved clusters could then be used to search for messages. Each cluster became a button (on a toolbar) with the user-specified name; pressing the button would show a new view of email messages to or from members of the cluster. The user could choose to include all or some subset of folders, allowing an orthogonal view of messages.

The perceived value of this tool was to create these groups quickly, and on the fly (clustering took only a few seconds, even with 10,000-100,000 messages indexed). However, the presentation of the clusters clearly showed users groups that existed in their data, and some of these groups surprised the users. The surprise groups show one aspect of reflection (novelty), while the unsurprising groups show another aspect (reinforcement).

Ultimately, this juxtaposition of reinforcement and novelty shows the importance of designing for reflection. By showing new information alongside expected information, SCuF afforded a new perceptual mechanism to comprehend their email corpus, along with an appropriate tool to make exploration possible.

Drivetime: Time Signatures in Media

Hard drives contain massive amounts of datetime information. A typical installation of a modern operating system (MacOS, Windows, or Linux) leaves hundreds of thousands of files on the hard drive, each with three separate datetime stamps. These data represent the *context* of the files. In addition, many files contain datetime stamps in the *content*. Of course, many files

are added by the user, incidentally by installing software, or intentionally by saving files; both of these processes create more datetime data.

Of course, the immediate question for this problem is; of what use are these data? Why is time an interesting dimension? Again, we must consider the two crucial aspects of reflection: novelty and reinforcement.

The power of time twofold: first, it is ubiquitous, and second, it can reflect both novelty and reinforcement. A user may well know that she checks email at 7:30 AM each weekday morning, but she may not realize that she tends spend more time reading personal email in the afternoon than in the morning.

The goal of this research, which is in early stages, is to identify ways in which time reflects user behavior. The value of this is not simply to track the user (although see below), but to provide the user with an understanding of their habits. One can, given a point in time, show the types of activities in which the user was engaging, at that point and nearby it.

One possible outcome for this is a "search" tool, using perhaps multiscale scrolling, allowing a user to see all of the activity on their computer. Densely populated regions of this timeline could then afford zooming; showing websites, email messages, files, other associated activities.

Again, the goal is to support novelty and reinforcement. Regular activities (checking cnn.com), shown alongside viewing a friend's photos, may allow for recollections that fell below some threshold without the appropriate reinforcement.

Future Directions

The power of gathering and analyzing data must always be tempered with the purpose of making something useful. Reflection and the principle of juxtaposing the novel with the familiar is a driving commitment in this work, designed to maintain the connection between the computational and the real.

Citations

[1] Garfinkel, S. L., & Shelat, A. (2003). Remembrance of data passed: A study of disk sanitization practices. *IEEE Security & Privacy, 1,* 17-27.

[2] Green, T. R. G., & Petre, M. (1996). Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages and Computing, 7*(2), 131-174.

[3] Gross, J. B. (2007). *Defining high-throughput email users.* Paper presented at the ACM Conference on Human Factors in Computing Systems (CHI) - Extended Abstracts (Student Research Competition Semi-Finalist), San Jose, CA.

[4] Gross, J. B. (2008). Intelligent interactions with email using social networks and ai, *ACM Conference on Human Factors in Computing Systems (CHI) Extended Abstracts - Doctoral Consortium*. Florence, Italy.

[5] Salton, G. (1987). Expert systems and information retrieval. *SIGIR Forum, 21*, 3-4.

[6] Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

[7] Zhang, X. L. (2009). Multiscale traveling: Crossing the boundary between space and scale. *Virtual Reality, 13*(2), 101-115.

[8] Zipf, G. K. (1935/1965). *The psycho-biology of language: An introduction to dynamic philology*. Cambridge, MA USA: MIT Press.