SHIFTR: A Fast and Scalable System for Ad Hoc Sensemaking of Large Graphs

Duen Horng Chau, Aniket Kittur, Hanghang Tong,
Christos Faloutsos, and Jason I. Hong
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{dchau, nkittur, htong, christos, jasonh} @cs.cmu.edu

ABSTRACT

We present SHIFTR, a system that assists users in making sense of large scale graph data. Making sense of information represented as large graphs is a fundamental challenge in many data-intensive domains. We suggest the potential of strong synergies between the data mining, cognitive psychology, and HCI communities in matching powerful graph mining tools with insights into how people learn and interact with information, and here we present SHIFTR as one such application. SHIFTR adapts the Belief Propagation algorithm to target important sensemaking tasks such as flexibly reorganizing graph entities into multiple groups based on both positive and negative examples. SHIFTR scales linearly with the graph size through its fast algorithm, novel *mList* data structure, and externalization of graph meta data.

We demonstrate SHIFTR's usage and benefits through real-world sensemaking scenarios using the DBLP dataset that has almost 2 million author-publication relationships. A demo video of SHIFTR can be downloaded at `http://www.cs.cmu.edu/~dchau/shiftr/shiftr.mov`.

Categories and Subject Descriptors

H.2.8 [**Databased Applications**]: Data Mining

General Terms

Algorithms, Design

1. INTRODUCTION

Large graphs are ubiquitous. They are often used because of their rich expressiveness; they represent collections of entities and capture the relationships among them. However, by the same token, exploring an unfamiliar graph — or *making sense* of it — is hard. Larger graphs exacerbate the problem. In this work, we offer one solution to help people make sense of large graphs with millions of nodes and edges. We define the *problem of sensemaking on large graphs* as:

given
- a large graph with millions of nodes and edges, and some limited knowledge about only a few of the nodes,

how to
- find *more* relevant nodes?
- *group* them into *multiple* coherent categories?
- *iteratively* improve such categorizations with *positive* and *negative* examples?
- and importantly, do all these *quickly*?

We contribute with a system called SHIFTR that embodies an integrated approach to sensemaking, which is grounded in cognitive psychology theories and harnesses powerful graph mining algorithm to assist users in exploring and understanding large graphs. SHIFTR stands for ***S**upporting **H**eterogeneous **I**nformation **F**oraging for **T**ransient **R**eorganization*. Figure 1a gives an overview of its architecture. SHIFTR's contributions include: (1) *unifying* cognitive psychology theories on sensemaking with an adapted Belief Propagation algorithm to support the user in making sense of large graph data; (2) attaining *high speed* and *scalability* through employing the novel *mList* data structure, fast algorithm design, and externalization of graph meta data; (3) *fluidly mapping* sensemaking tasks to the internal operations of SHIFTR's algorithm, through an intuitive user interface.

2. A SENSEMAKING DEMONSTRATION

We will demonstrate SHIFTR's usage, user interaction, fast algorithm, and scalable system design through sensemaking scenarios on the DBLP dataset, which has over 1.9M author-paper relationships; from these, SHIFTR creates a bipartite graph of authors and papers.

Here, we illustrate one example scenario, where we make sense of the research history of Prof. Brad Myers, a prolific author in Human-Computer Interaction. This example demonstrates how SHIFTR can support the iterative generation and refinement of multiple ad-hoc conceptual representations of Myers' work. The goal of the scenario is to understand and represent Myers' different research areas, co-authors that he has worked with, and papers published in those areas. This task demonstrates strengths of the SHIFTR platform on a heterogeneous dataset involving many possible groupings, as Myers has worked in many areas over the years with overlapping authors who themselves have worked in multiple areas. We first use SHIFTR's search feature to bring up the item for Brad Myers (Figure 2a). Next, we create a group by dragging Myers' name to an empty

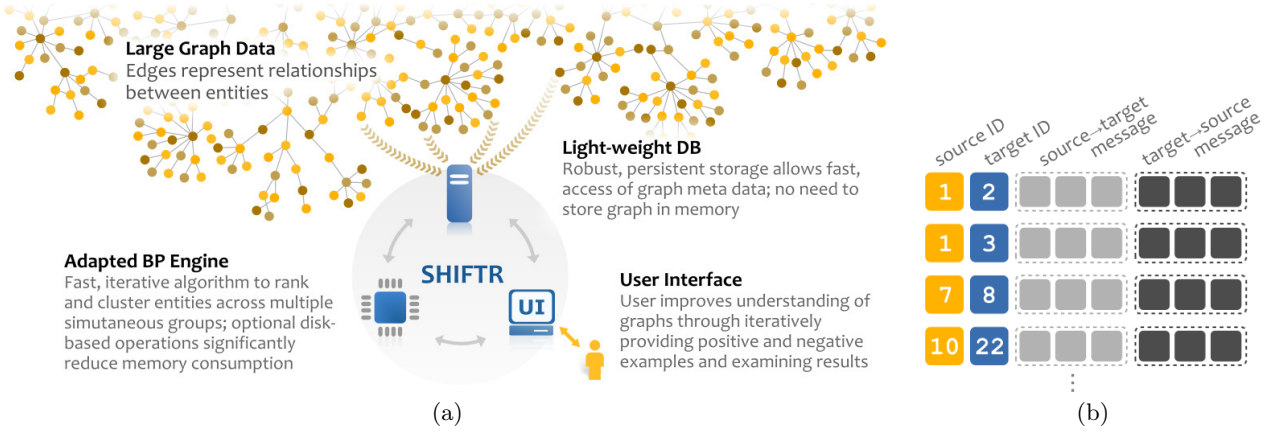

Figure 1: (a) Overview of the SHIFTR system. (b) The *mList* data structure used in SHIFTR, which significantly speeds up computations. It combines the adjacency list representation of a graph with messages associated with the edges. The message sent from node i to j always precedes that from j to i, which significantly reduces lookup time for messages in reverse directions. Source ID and target ID are `ints`; a message is an array of `floats`. This example assumes each node has three states.

group. In the new group Brad Myers is automatically selected as a seed example, as denoted by its bold appearance. Pressing the *Cluster Data* button generates a list of items that are most relevant to Myers, ranked by relevance from high to low. These items may be either authors or papers, and can be filtered by type using the drop-down box at the bottom left of each group's window. The number of recommendations generated can be adjusted using the slider at the top of the main window. Top relevant items for Myers include papers with topics such as *End User Programming*, *Text Entry*, and *Interface Generation*, and the names of related authors (see Figure 2b). For each of these topics we create a separate group using the *Add Group* button, and name each accordingly. We then drag Brad Myers into each group along with an example paper representing the topic, and recluster the data.

The above actions demonstrate SHIFTR's major features; the user can (1) create and refine groups; (2) use few examples to retrieve relevant items; (3) utilize heterogeneous data (authors or papers); and (4) use the same examples in multiple groups. Many of the newly generated items match their groups' topics. To refine the topics, we double-click on papers we identify as relevant, which selects them as positive, seed examples. Reclustering results in even better performance, with more relevant papers and authors in each group.

SHIFTR also support (soft) negative examples We can remove papers irrelevant to all the existing groups. This results in other similarly undesired articles being removed from the original source groups as they are more closely associated with the new negative example group. This strategy of supporting conceptual positive and negative examples greatly increases the power to reorganize and refine clusters to improve their quality. The same functionality can also make clusters more specific. For example, we can create a more specific *Debugging* group from papers in *End User Programming* (Figure 2c). Note that Andrew Ko now appears in both groups, reflecting that he authored papers in both areas. SHIFTR can also be run in partitioning mode where each item is assigned to the most likely cluster (i.e., maximum-a-posteriori assignments).

3. THE SHIFTR SYSTEM

This section describes how SHIFTR's algorithm and user interface work together to support sensemaking, and how its system design makes it fast and scalable.

The Algorithm. We view sensemaking as an inference problem, where we try to assess the marginal probabilities of each node being in each of the possible groups. To do this, we first apply the *pairwise Markov Random Field* (MRF) model over the graph, which enables us to interpret each node in the graph as a discrete random variable that takes on one of a set of specific states S. We map the notion of *state* to *group* in SHIFTR. That is, if the user has created two groups in SHIFTR, then all nodes have two states internally. From now on, we use *state* and *group* interchangeably. And we assume there are two groups in the following discussion. A node's *belief* is a vector of probabilities that describe how likely the node belongs to each of the possible states. For example, a node belief of $[0.2, 0.8]$ means it has 0.2 probability being in state 1 and 0.8 in state 2. Each node is also associated with a *prior belief* (same dimension as node belief), which represents what the user thinks the node's belief should be.

The Belief Propagation algorithm[1] (BP) [13] is a scalable algorithm that can solve this inference problem for sensemaking. BP functions via iterative message passing between nodes in the graph, where each message represents the source's opinion about the target's belief. At every iteration of the algorithm, each node updates its belief based on its prior belief and the messages from its neighbors. The node then turns its belief into a message that will "influence" its neighbors' beliefs. This process continues until the node beliefs converge (within some threshold), or after a maximum number of iterations has passed.

We used a *truncated* version of BP, which stops the algorithm after a certain number of iterations, to obtain a *gradient* of node beliefs that peaks at the seed nodes. Stopping the algorithm early corresponds to relaxing the error bound on the final node beliefs. And this belief gradient ranks all nodes based on their relevance relative to the seed

[1]See [13] for an excellent, detailed description.

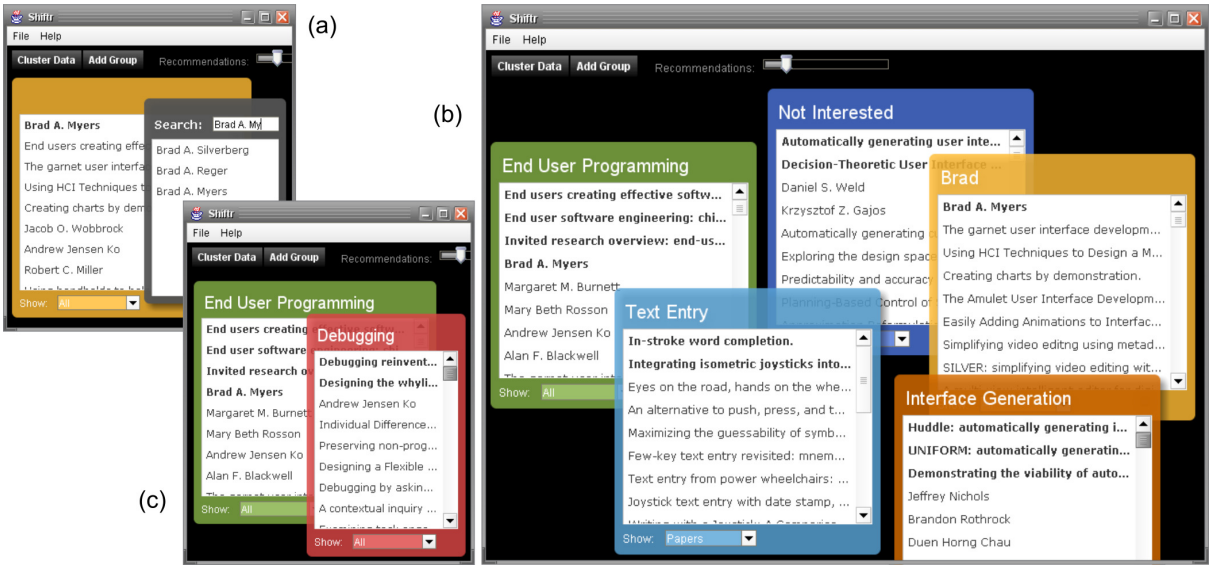

Figure 2: (a) The SHIFTR user interface showing a group populated with authors and papers relevant to Brad A. Myers, who has been located with prefix search. (b) The cluster results for three of Myers' research areas (*End User Programming*, *Text Entry*, and *Interface Generation*), the group *Not Interested* containing undesired negative examples from Text Entry, and the group *Brad* with items less relevant to the three research areas. The Text Entry group is set to show only papers, using the drop-down filter at the lower left. The group labels are provided by the user. (c) Andrew Jensen Ko shows up as a relevant author for both groups, reflecting the fact that he authored papers in both areas.

nodes. Intuitively, a node that is connected to a seed node with a number of short paths would be more related than one connected with fewer, but longer paths. BP is able to capture this "closeness" in its computation.

The User Interface. SHIFTR's user interface allows the user to perform sensemaking tasks by transparently modifying the internal states of the algorithm. Table 1 summarizes how the user's actions are mapped to the algorithm's internal states. When the user selects a node into a group, say group 1, as a seed, this biases the node's prior belief towards favoring group 1. Since BP computes *every* node's marginal group probabilities, we can pick the top k nodes most relevant to the seed nodes in a group by simply picking the ones that has the highest marginal probabilities for that group. The ranking of the nodes, based on marginal probabilities, helps the user evaluate how other nodes are relevant to the seeds, which in turn inform the user in picking more nodes that are relevant that will further improve the quality of the groupings. The MRF representation of a graph does not discriminate nodes of different types. Therefore, the user can freely select nodes of mixed types as examples.

Implementation. SHIFTR is written in Java, in over 5000 lines of code. SHIFTR is developed with Eclipse 3.4.2 on a Core 2 Duo E6750 desktop computer with 6GB of RAM that runs 64-bit Windows Vista. We have designed and implemented a non-trivial data structure for SHIFTR to enable both fast in-memory, and disk-based operations on large graph datasets, achieving high space and time efficiency, as we will explain in Section 5. Moreover, we use SQLite [7] to store our graph on the disk, because SQLite is compact (<500KB) and serverless. Our SQLite-based design allows for extensibility: analysts of datasets may optionally create more tables to store additional graph meta data.

4. RELATED WORK

Significant research in cognitive psychology has shown that people represent most concepts as *collections of exemplars* defining a prototype, rather than an all-or-none rule-based representation [5, 3]; people build up these concepts iteratively [6], which are often *flexible*, *ad-hoc*, and *theory-driven* rather than determined by static features of the data [3]. Drawing upon these research findings, SHIFTR, as a sensemaking system, should support *iterative*, *graded*, *exemplar-based categorization* of graph entities, and allow the users to *experiment* with *shifting*, *ad-hoc* representations of the data.

Several existing and well-known research summarizes document and terms (e.g., LSI[1]). Among them, the work most related to SHIFTR is constrained-based clustering (e.g.,[11]). However, such algorithms do not support iterative improvement of the clustering and some other functionalities in SHIFTR. The class of graph cuts algorithm [14] efficiently solves energy minimization problems in computer vision which often applies the MRF model on images. However, it does not rank nodes based on relevance within a cluster. Proximity-based algorithms, such as *Random Walk with Restart* (RWR), give graded node relevance [4, 9, 10], and some of them support user feedback [10], but all of them do not support clustering multiple groups. To summarize, none of the above research ideas supports all the functionalities of SHIFTR.

SHIFTR uses a *truncated* version of BP. Standard BP [13] has been successfully applied in many domains, such as computer vision [8] and error-correcting codes [2]. Gaussian BP is a variant where its underlying distributions are Gaussian [12]. Generalized BP [13] allows messages to be passed between subgraphs, which can improve accuracy in the computed beliefs and promote convergence.

Table 1: Sensemaking requirements for SHIFTR and how it supports them

	How SHIFTR's support through its...		
Sensemaking Req	**Algorithm**	**GUI**	**Sample Task**
Example-based	Setting nodes' prior belief	User picks nodes as examples	'Find items relevant to X'
Grouping relevant items	Threshold based on nodes' marginal group probabilities	Relevant nodes visually grouped and ranked	'Group relevant items together'
Multiple, Simultaneous Groups	#groups $\leftrightarrow$ #states of a node; can be arbitrarily many	User adds or removes groups in GUI	'Separate items relevant to X, Y, and Z'
Iterative Improvement Through Feedback	Modifying nodes' prior belief	User gives positive or negative relevance feedback with examples	'Find *more* items relevant to X and Y, but not Z'
Goal-Directed, Ad Hoc Sensemaking	Node relevance can propagate to neighboring nodes of different types	User can pick nodes of mixed-typed as examples	'Find items relevant to A, B, and C (each of a different type)'

5. SCALABILITY

SHIFTR's computation time scales *linearly* with the number of edges, $|E|$, in the graph, thanks to the underlying Belief Propagation algorithm, which has $O(|E|)$ time complexity. An iteration over the graph of DBLP, which has almost *2 million* edges, takes merely 2.7s. SHIFTR can keep the graph in memory if it fits, or leave it on the hard disk. Regardless of the storage methods, the graph is represented as an *edge file*, which *fully* describes the graph topology. We devised the data structure, called *mList*, for the edge file, to reduce its size to the minimum necessary for the algorithm to operate on. Figure 1b shows the *mList* data structure, which combines the adjacency list representation of a graph with the temporary messages that the algorithm passes along the edges. This special design of keeping a pair of directed edges pointing in opposite directions allows fast lookup of one edge given the other, which enables us to compute all new messages in each iteration with only *two sequential passes* through the edge file, eliminating the need for random access of edges.

6. CONCLUSIONS

We present SHIFTR, a fast and scalable system that assists users in exploring and understanding large graphs. We contribute SHIFTR as a whole; our contributions include:

- *unifying* cognitive psychology theories on sensemaking with an adapted Belief Propagation algorithm to support important sensemaking tasks on large graphs, such as flexibly reorganizing graph entities into multiple groups based on both positive and negative examples;
- delivering a *high speed* and *scalable* system through employing the novel *mList* data structure, fast algorithm design, and externalization of graph meta data;
- *fluidly mapping* sensemaking tasks to the algorithm's internal parameters and operations, through an intuitive user interface.

We will demonstrate SHIFTR's applicability and scalability on real-world sensemaking scenarios using the DBLP dataset that has almost 2M author-publication relationships.

7. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grants No. CNS-0721736 and under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Duen Horng Chau is supported by the Symantec Research Labs Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

8. REFERENCES

[1] P. W. Foltz and S. T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Comm. of ACM (CACM)*, 35(12):51–60, Dec. 1992.

[2] B. J. Frey and D. J. C. Mackay. A revolution: Belief propagation in graphs with cycles. In *In NIPS*, pages 479–485. MIT Press, 1998.

[3] R. L. Goldstone, M. Steyvers, and B. J. Rogosky. Conceptual interrelatedness and caricatures. *Memory & Cognition*, 31:169–180, 2003.

[4] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. *ACM SIGKDD*, Aug. 2004.

[5] E. Rosch and C. B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605, 1975.

[6] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The cost structure of sensemaking. In *CHI*, pages 269–276, New York, NY, USA, 1993. ACM.

[7] SQLite. http://www.sqlite.org/.

[8] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, pages 900–907, 2003.

[9] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622, 2006.

[10] H. Tong, H. Qu, and H. Jamjoom. Measuring proximity on graphs with side information. In *ICDM*, pages 598–607, 2008.

[11] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *SDM*, pages 1–12, 2008.

[12] Y. Weiss and W. T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[13] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. pages 239–269, 2003.

[14] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:437–444, 2004.