Graphs / Networks
Interactive applications

Duen Horng (Polo) Chau
Georgia Tech

Partly based on materials by Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Le Song
Building an interactive application

Will show you an example application (Apolo) that uses a "diffusion-based" algorithm to perform recommendation on a large graph

- **Personalized PageRank**
  (= Random Walk with Restart)

- Belief Propagation
  (powerful inference algorithm, for fraud detection, image segmentation, error-correcting codes, etc.)

- “Spreading activation” or “degree of interest” in Human-Computer Interaction (HCI)

- Guilt-by-association techniques
Building an interactive application

Why diffusion-based algorithms are widely used?

• **Intuitive to interpret**
  uses “network effect”, homophily, etc.

• **Easy to implement**
  Math is relatively simple

• **Fast**
  run time linear to #edges, or better

• **Probabilistic** meaning
Human-In-The-Loop Graph Mining

Apolo: Machine Learning + Visualization

CHI 2011

Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning
Finding More Relevant Nodes

Citation network

HCI Paper

Data Mining Paper
Finding More Relevant Nodes

HCI Paper

Data Mining Paper

Citation network
Finding More Relevant Nodes

Apolo uses guilt-by-association (Belief Propagation, similar to personalized PageRank)

Citation network
**Demo: Mapping the Sensemaking Literature**

**Nodes:** 80k papers from Google Scholar (node size: #citation)

**Edges:** 150k citations

The cost structure of sensemaking

*PDF 1993*


245 citations 8 versions
The cost structure of sensemaking


245 citations 8 versions
Key Ideas (Recap)

Specify **exemplars**

Find **other** relevant nodes (BP)
Apolo’s Contributions

1. Human + Machine
   It was like having a partnership with the machine.

2. Personalized Landscape
Apolo 2009
Apolo 2010
Apolo 2011

22,000 lines of code. Java 1.6. Swing. Uses SQLite3 to store graph on disk

The cost structure of sensemaking
245 citations 8 versions
User Study

Used citation network

**Task**: Find related papers for 2 sections in a survey paper on *user interface*

- Model-based generation of UI
- Rapid prototyping tools

Past, Present and Future of User Interface Software Tools

Brad Myers, Scott E. Hudson, and Randy Pausch

Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
Between subjects design
Participants: grad student or research staff
“Model-based”

“Prototyping”

10 papers for each section
Apolo

Google Scholar

“Model-based”

“Prototyping”

10 papers for each section

Expert judges rated papers

\[
\begin{align*}
1 + 0 &= 1 \\
1 + 1 &= 2 \\
0 + 0 &= 0 \\
\vdots
\end{align*}
\]
Higher is better.

Apolo wins.

Statistically significant, by two-tailed t test, \( p < 0.05 \)

* Judges’ Scores

<table>
<thead>
<tr>
<th></th>
<th>Apolo</th>
<th>Scholar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-based</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>*Prototyping</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>*Average</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

Higher is better. Apolo wins.
Apolo: Recap

A mixed-initiative approach for exploring and creating personalized landscape for large network data

Apolo = ML + Visualization + Interaction
Feldspar

Finding Information by Association.

CHI 2008

Polo Chau, Brad Myers, Andrew Faulring

YouTube: http://www.youtube.com/watch?v=Q0TIV8F_o_E&feature=youtu.be&list=ULQ0TIV8F_o_E

Feldspar

A system that helps people find things on their computers when typical search or browsing tools don’t work
Feldspar

A system that helps people find things on their computers when typical search or browsing tools don’t work

An example scenario...
“Find the webpage mentioned in the email from the person I met at an event“
“Find the **webpage** mentioned in the **email** from the **person** I met at an **event**“

If I can’t remember the specifics, such as any text in the webpage, email, etc.

→ **Can’t search**
“Find the **webpage** mentioned in the **email** from the **person** I met at an **event**“

If I can’t remember the specifics, such as any text in the webpage, email, etc.

→ **Can’t search**

If I haven’t bookmarked the webpage

→ **Can’t browse**
“Find the **webpage** mentioned in the **email** from the **person** I met at an **event**“
“Find the webpage mentioned in the email from the person I met at an event“

But I can describe the webpage with a chain of associations.
“Find the **webpage** mentioned in the **email** from the **person** I met at an **event**“

But I can describe the **webpage** with a chain of associations.

**webpage** – **email** – **person** – **event**
“Find the **webpage** mentioned in the **email** from the **person** I met at an **event**“

But I can describe the **webpage** with a chain of **associations**.

```
webpage – email – person – event
```

The psychology literature has shown that people often remember things exactly like this.
Natural question:

Can I find things by associations?
Natural question: Can I find things by associations?

Can I find the webpage by specifying its associated information (email, person, and event)?
Natural question:

Can I find things by associations?

Can I find the *webpage* by specifying its associated information (*email*, *person*, and *event*)?

We created Feldspar, which supports this associative retrieval of information.
Feldspar stands for....

http://youtu.be/Q0TIV8F_o_E
Feldspar stands for....

Finding Elements by Leveraging Diverse Sources of Pertinent Associative Recollection

http://youtu.be/Q0TIV8F_o_E
Implementation: Overview

Create a graph database to store the associations among items on the computer

Develop an algorithm that processes the query and returns results
Creating an Association Database (a graph)

Install Google Desktop and let it index all the items on the computer
Creating an Association Database (a graph)

Install **Google Desktop** and let it index all the items on the computer.

**Focus on 7 types**

- **Dates**
- **Events**
- **People**
- **Emails**
- **Files**
- **Folders**
- **Webpages**

**filetype:calendar**
**filetype:email**
**filetype:doc, etc.**
**filetype:web**
Creating an Association Database (a graph)

Identify **associations** and build our database, which is a **directed graph**

Focus on 7 types

Install **Google Desktop** and let it index all the items on the computer

**Google Desktop Database**

**filetype:calendar**
**filetype:email**
**filetype:doc, etc.**
**filetype:web**
Creating an Association Database (a graph)

Identify **associations** and build our database, which is a **directed graph**.

Focus on 7 types

Install **Google Desktop** and let it index all the items on the computer.

**Association Graph** (small subgraph shown)

- **from**
- **attach**
- **in email**
- **in folder**

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.

**Identify** **associations** and build our database, which is a **directed graph**.

**Focus on 7 types**

**Install Google Desktop** and let it index all the items on the computer.
Creating an Association Database (a graph)

Identify **associations** and build our database, which is a **directed graph**

Focus on 7 types

Install **Google Desktop** and let it index all the items on the computer

**Association Graph** (small subgraph shown)

- **from**
- **in email**
- **attach**
- **in folder**
- **attend**

Google Desktop Database

- **Dates**
- **Events**
- **People**
- **Emails**
- **Folders**
- **Files**
- **Webpages**

**filetype:calendar**
**filetype:email**
**filetype:doc, etc.**
**filetype:web**
Practitioners’ guide to building (interactive) applications

Think about scalability early

- e.g., picking a scalable algorithm early on

When building interactive applications, use iterative design approach (as in Apolo)

- Why? It’s hard to get it right the first time
- Create prototype, evaluate, modify prototype, evaluate, ...
- Quick evaluation helps you identify important fixes early (can save you a lot of time)
Practitioners’ guide to building (interactive) applications

How to do iterative design?

What kinds of prototypes?
  • Paper prototype, lo-fi prototype, high-fi prototype

What kinds of evaluation? Important to involve REAL users as early as possible
  • Recruit your friends to try your tools
  • Lab study (controlled, as in Apolo)
  • Longitudinal study (usage over months)
  • Deploy it and see the world’s reaction!

• To learn more:
  • CS 6750 Human-Computer Interaction
  • CS 6455 User Interface Design and Evaluation
If you want to know more about people…

http://amzn.com/0321767535
GLO-STIX

Graph-Level Operations for Specifying Techniques and

Chad Stolper, Minsuk Kahng, Zhiyuan “Jerry” Lin, Florian Foerster, Aakash Goel, John Stasko, Polo Chau
Force-directed layout commonly used, but often does not lead to deep insights.
The visualization (vis) community has created many helpful graph visualization techniques …

**Semantic Substrate**  
[B Shneiderman, A. Aris, TVCG’06]

**Pivot Graph**  
[M. Wattenberg, CHI’06]
But these tools are often not immediately available for use in high-level tools like:
We can re-create them using low-level libraries, but that takes (much) time and effort.
Our Goal:
We provide

... and you get
Our Goal:
We provide **Graph-Level Operations (GLO)**

... and you get **Visualization Techniques**
- Substrate on X
- Substrate on Y
- Show Links as Curved
- Aggregate
- (Size Nodes by Count)
- Show X Axis
- Show Y Axis
Identifying GLOs
Identifying GLOs

1. Align Nodes
2. Evenly Distribute Nodes
3. Evenly Distribute Nodes by Attribute
4. Substrate Nodes by Attribute
5. Evenly Distribute Nodes within Substrates
6. Position Nodes Relatively
7. Evenly Distribute Nodes Radially by Attribute
8. Evenly Distribute Nodes Radially
9. Position Nodes Radially by Attribute
10. Substrate Nodes Radially by Attribute
11. Evenly Distribute Nodes Along Plot Radius
12. Evenly Distribute Nodes Along Plot Radius
13. Position Nodes Along Plot Radius by Attribute
14. Substrate Nodes Along Plot Radius
15. Position Nodes Along Plot Radius by Constant
16. Apply an Algorithm to the Nodes
17. Size Nodes by a Constant
18. Size Nodes Relatively by a Continuous Attribute
19. Display All Links
20. Display Selected Links
21. Hide Links
22. Display Links as Straight
23. Display Links as Curved
24. Display Links as Circles
25. Clone Active Generation
26. Select Generation k
27. Set Source Generation k
28. Set Target Generation k
29. Remove Generation k
30. Aggregate by Attribute
31. Aggregate by Attribute and Attribute
32. Deaggregate Generation k
33. Show Axis
34. Hide Axis

34 Operations
5 categories (using card-sorting)
GLO-STIX Benefits

- For analysts:
  - Summon state-of-the-art visualization tools on demand (no need to switch tools)

- For engineers:
  - Implement and share GLO as “middle-level need to “reinvent the wheels”

- For researchers:
  - Discover new visualization techniques through combining GLOs
GLO-STIX Summary & Next Steps

- Published as a InfoVis’14 paper (top vis conference)
- Forms foundation of PhD thesis of Chad Stolper (co-advised by John Stasko, Polo Chau)
- Next steps:
  - Apply operations on subgraph
  - Open-source

Chad Stolper, Minsuk Kahng, Zhiyuan “Jerry” Lin, Florian Foerster, Aakash Goel, John Stasko, Polo Chau