Automatically Building Research Reading Lists

Michael D. Ekstrand¹  Praveen Kanaan¹  James A. Stemper²
John T. Butler²  Joseph A. Konstan¹  John T. Riedl¹

ekstrand@cs.umn.edu

¹GroupLens Research
Department of Computer Science and Engineering
University of Minnesota

²University of Minnesota Libraries

RecSys 2010
Recommending Research Papers

Way too many papers to read — need to find the right ones so we can get on to research!

Many papers, many reviewers — who is most qualified to judge the work?
Recommending Research Papers

Way too many papers to read — need to find the right ones so we can get on to research!

Many papers, many reviewers — who is most qualified to judge the work?

Working on a bibliography or reference list — who should have cited you?
Alice is a new grad student who took a class on adaptive web technologies. The professor assigned some readings, and she wants to learn more.

Building a Reading List

Alice is a new grad student who took a class on adaptive web technologies. The professor assigned some readings, and she wants to learn more.


How can we find important ones to read?
The **Reading List** Task

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th><strong>Output</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>List of papers on topic of interest</td>
<td>List of important papers to read</td>
</tr>
</tbody>
</table>

Magic happens here

---

Ekstrand et al. (GroupLens/UMN)
Domain Characteristics

Traditional CF methods are blind to item content or relationships.

The citation web has a defined structure.

Can we harness this to improve recommendation?
Questions

- Can we harness domain structure to improve recommendation of research papers?
- What algorithms perform well at building reading lists?
- How can we measure this performance?
Approach

1. Analyze Task and Domain
2. Design Candidate Algorithms
3. Prune Algorithm Pool (offline)
4. Evaluate Performance (user study)
Approach

Analyze Task and Domain

Design Candidate Algorithms

Prune Algorithm Pool (offline)

Evaluate Performance (user study)
Designing Candidate Algorithms

1. Analyze Task and Domain
2. Design Candidate Algorithms
3. Prune Algorithm Pool (offline)
4. Evaluate Performance (user study)
We adapt and combine off-the-shelf algorithms.

**Collaborative filtering (CF)**
- Item-item CF over sets of items [Karypis, 2001]

**Content-based filtering (CBF)**
- Lemur toolkit [Ogilvie and Callan, 2002] in BM25 mode with recommended baseline parameters

**Graph ranking**
- PageRank [Page et al., 1999]
- HITS authority scores [Kleinberg, 1999]
- SALSA [Lempel and Moran, 2000]
- Relative algorithms [White and Smyth, 2003]
  - Biased HITS
  - $k$-step Markov importance
Applying Item-Item CF

No users - papers are both users and items [McNee et al., 2002].

\[
\begin{array}{ccc}
 & A & B & C \\
A & ✓ & ✓ & ✓ \\
B & ✓ & ✓ & \\
C & ✓ & ✓ & ✓ \\
\end{array}
\]

Becomes a form of co-citation analysis.
Inspired by [Karypis, 2001]: user influence in item similarity does not need to be the same!

Karypis’s approach: normalize user purchase vectors to unit vectors so users with few purchases influence item similarity more.

Our adaptation: weight citation vectors by graph rank. Highly-ranked papers have more influence.

\[ \hat{a} = r(a)a \]

Use weighted citation vectors to compute cited paper similarity.
Using Weights with CBF

Two approaches

Subgraph ranking

Build subgraph from CBF results and rank its nodes.
Using Weights with CBF

Two approaches

Linear blending

Blend CBF scores with node weights in the global graph.

Learned coefficients with multivariate logistic regression (more on this later).
Several ways of blending algorithms

- Use CF as input to CBF
- Use CBF as input to CF
- Blend output from CBF and CF to produce result

Basic algorithms + all hybrids = 177 algorithms.
Hybridizing Recommenders [Burke, 2002]

Several ways of blending algorithms

- **Use CF as input to CBF**
- Use CBF as input to CF
- Blend output from CBF and CF to produce result

Basic algorithms + all hybrids = **177 algorithms**.
Hybridizing Recommenders [Burke, 2002]

Several ways of blending algorithms
- Use CF as input to CBF
- **Use CBF as input to CF**
- Blend output from CBF and CF to produce result

Basic algorithms + all hybrids = 177 algorithms.
Hybridizing Recommenders [Burke, 2002]

Several ways of blending algorithms

- Use CF as input to CBF
- Use CBF as input to CF
- Blend output from CBF and CF to produce result

Basic algorithms + all hybrids = 177 algorithms.
Pruning the Algorithm Pool

Analyze Task and Domain

Design Candidate Algorithms

Prune Algorithm Pool (offline)

Evaluate Performance (user study)
Designing an Evaluation Strategy

Goal: Measure algorithm performance at supporting a specific task.

Accuracy isn’t enough [McNee et al., 2006].
Offline Evaluation

Use metadata from ACM Digital Library.

Simulate introductory reading list with hold-out test on articles in *ACM Computing Surveys*.

- Hold out 5 items from each citation list
- Attempt to recommend back the 5 items
- Skip surveys with less than 15 resolved citations

Result: 220 survey articles for training and testing.
Use half the articles to train the CBF blend.

Learned multivariate logistic regression with response of 1 for articles in the holdout set, 0 for other articles.

\[ s(a) = \alpha L(a) + \beta r(a) \]

Since we’re ranking, intercept and log don’t matter.
Measure using half-life utility metric [Breese et al., 1998]:

\[ R_a = \sum_i \frac{u_{a,i}}{2^{(i-1)/(\alpha-1)}} \]

- \( \alpha = 5 \), the length of our reading lists
- \( u_{a,i} = 1 \) if article \( i \) is cited by survey \( a \), 0 otherwise

Aggregate to compute fraction of potential utility achieved (in range \([0, 1]\)).

\[ R = \frac{1}{nR_{\text{max}}} \sum_a R_a \]
Offline Results — CF

Half-life utility

<table>
<thead>
<tr>
<th>Method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf-hits</td>
<td>0.20</td>
</tr>
<tr>
<td>cf-plain</td>
<td>0.25</td>
</tr>
<tr>
<td>cf-full-unit</td>
<td>0.30</td>
</tr>
<tr>
<td>cf-unit</td>
<td>0.30</td>
</tr>
<tr>
<td>cf-pagerank</td>
<td>0.30</td>
</tr>
<tr>
<td>cf-salsa</td>
<td>0.30</td>
</tr>
</tbody>
</table>
Offline Results

Half-life utility

- CF
- CBF
- CBF–CF
- CF–CBF
- Fusion

Ekstrand et al. (GroupLens/UMN)  Automatically Building Reading Lists  #recsys2010
Salsa performs like PageRank
Chose 3 algorithms that performed well but have differing structures.

- CBF with Biased HITS in subgraph ranking configuration
- CBF fed into PageRank-weighted CF
- PageRank-weighted CF
Evaluating Performance

1. Analyze Task and Domain
2. Design Candidate Algorithms
3. Prune Algorithm Pool (offline)
4. Evaluate Performance (user study)
User-Based Evaluation

Asked graduate students to provide query sets of 5-10 papers and evaluate 3 5-item reading lists.

- Relevance of individual papers
- Importance of individual papers
- Quality of reading list as a whole
- Relative ranking of reading lists

All questions were set in the context of introducing a new researcher to the topic.
Use this page to provide us with 5-10 papers on a topic in computer science. We will use these papers as the starting point for building your reading lists. You can search for papers using the search box. Most papers in the ACM Digital Library, as well as some computer science papers from IEEE, Springer, and other publishers are available.

You may use multiple searches to build your list of papers. Once you have 5-10 selected papers, click "Continue" to proceed with the survey.


In this portion of the survey, we will present you with up to 5-item reading lists. Evaluate each list as a whole with the questions provided at the end of the list.


Nie, L., Davison, B., Qi, X.  *Topical link analysis for web search*.  In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (Aug. 2006)*. SIGIR ’06


Feedback

If you were preparing a 5-item reading list to enable another researcher to understand the essentials of your topic, how many of the items above would you include?

[ ] 0  [ ] 1  [ ] 2  [ ] 3  [ ] 4  [ ] 5

Overall, how good is this set of papers for introducing a new researcher to your topic?

[ ] Very Good  [ ] Good  [ ] Fair  [ ] Poor  [ ] Very Poor
You will find below a selection of papers that our algorithms have identified as candidates for a reading list. Beside each paper, we are asking you to rate it on the following three criteria:

**How relevant is the paper to your topic of interest?**
- Exactly captures the topic
- Mostly relevant
- Somewhat relevant
- Mostly irrelevant
- Entirely irrelevant

**How important do you think the paper is to your topic?** Measure this in terms of how long a reading list needs to be before you would include this paper. We will only ask you to rank the importance if the paper is at worst "Mostly irrelevant".
- Would include on a list of 5 papers
- Would include on a list of 10 papers
- Would include on a list of 25 papers
- Would include on a list of 50 papers
- Would omit or only include on very long lists

**How familiar are you with this paper?**
- I (co-)wrote it
- I have cited it
- I have read it
- I have heard of it (or am familiar with the authors)
- I have never heard of it

**Evaluate Individual Articles:**

<table>
<thead>
<tr>
<th>Paper</th>
<th>Relevance</th>
<th>Importance</th>
<th>Familiarity</th>
</tr>
</thead>
</table>
Below are summaries of the reading lists you have evaluated. Use the arrow buttons to put them in order from best to worst.

<table>
<thead>
<tr>
<th>Reading list A</th>
<th>Reading list B</th>
<th>Reading list C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web (May. 2006). WWW '06.</td>
<td>CSCW '94.</td>
<td>Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. GroupLens:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>an open architecture for collaborative filtering of netnews. In Proceedings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSCW '94.</td>
</tr>
<tr>
<td>Conference on Web Intelligence (Sep. 2005). WI '05.</td>
<td></td>
<td>annual international ACM SIGIR conference on Research and development in</td>
</tr>
<tr>
<td>Nio, L., Davison, B. D., Qi, X., Topical link</td>
<td></td>
<td>Shardanand, U., Maes, P., Social information filtering: algorithms for</td>
</tr>
<tr>
<td>analysis for web search. In Proceedings of the 29th</td>
<td></td>
<td>automating “word of mouth”. In</td>
</tr>
<tr>
<td>annual international ACM SIGIR conference on</td>
<td></td>
<td>Proceedings of the 22nd annual international ACM SIGIR conference on</td>
</tr>
<tr>
<td>(Aug. 2006). SIGIR '06.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
User-Based Results

Ekstrand et al. (GroupLens/UMN)  Automatically Building Reading Lists  #recsys2010
CF performed the best (surprisingly)

How do we do?
CF’s Recommender Systems Recommendations

- Shardanand and Maes. Social information filtering: algorithms for automating “word of mouth”. In *Proc. CHI 1995*.
Task-Driven Design and Evaluation

Design recommenders to support human information needs, not just improve prediction error [McNee et al., 2006].

Task and context inform design of both recommenders and evaluation strategies.

Opportunity to harness unique characteristics (or “personalities”) of specific algorithms.
Conclusion

Contributions

- Design and evaluation of recommenders for reading lists
- Method for biasing CF with authority metrics
- CF works well for reading lists (surprising)
Conclusion

Contributions

- Design and evaluation of recommenders for reading lists
- Method for biasing CF with authority metrics
- CF works well for reading lists (surprising)

Open Questions

- How to tell user that they have an important paper?
- How to more accurately operate within topic scope?
- Is SALSA misused?
Conclusion

Contributions

- Design and evaluation of recommenders for reading lists
- Method for biasing CF with authority metrics
- CF works well for reading lists (surprising)

Open Questions

- How to tell user that they have an important paper?
- How to more accurately operate within topic scope?
- Is SALSA misused?

Questions?

ekstrand@cs.umn.edu

Funded by NSF grants IIS 05-34939 and IIS 08-08692