Supporting Exploration and Serendipity in Information Retrieval

Nattiya Kanhabua

Department of Computer and Information Science
Norwegian University of Science and Technology

24 February 2012
Motivation

• Typical search engines
  – Lookup-based paradigm
  – Known-item search

Does this paradigm satisfy all types of information needs?
Beyond the lookup-based paradigm

Two tasks when searching for unknown:

1. Exploratory Search
   - Users perform *information seeking*
     - E.g., collection browsing or visualization
   - Human-computer interaction

2. Serendipitous IR
   - Systems *predict/suggest* interesting information
     - E.g., recommender systems
   - Asynchronous manner
The next generation of search


Trial lecture

Nattiya Kanhabua
PART I – EXPLORATORY SEARCH
Exploratory search

- Information-seeking task [Marchionini 2006, White 2006a]
  - Seek for *unknown*, or an open-end problem
  - Complex information needs
  - No knowledge about the contents
Exploratory search activities

Features of exploratory search

- Query (re)formulation in real-time
- Exploiting search context
- Facet-based and metadata result filtering
- Result visualization
- Learning and understanding support
Query (re)formulation

• Help users to *formulate information needs* in an early stage [Manning 2008]

• Query suggestion
  – Support by major search engines
  – Based on *query logs analysis*

• Query-by-example
  – Search using examples of documents
Leveraging search context

- Effective systems must adapt to contextual constraints [Ingwersen 2005]
  - Time, place, history of interaction, task in hand, etc.

- Types of context
  1. **Explicitly** provided feedbacks
     - E.g., select relevant documents
  2. **Implicitly** obtained user information
     - E.g., mine users’ interaction behaviors

![Figure 1. Implicit Query Interface](image)
Facet-based result filtering

- Facets are **properties** of a document [Tunkelang 2009]
  - Usually obtain from *metadata*

- Facet search provides an ability to:
  - Explore results via properties
  - Expand or refine the search
Facet-based result filtering

- Facets are **properties** of a document [Tunkelang 2009]
  - Usually obtain from *metadata*

- Facet search provides an ability to:
  - Explore results via properties
  - Expand or refine the search

- No metadata?
  - Categorization
  - Clustering
Result visualization

• **Provide overviews** of the collection and search results
  – To understand and support an analysis

• **Applications**
  – manyEyes [Viégas 2007]
  – Stuff I’ve seen [Dumais 2003]
  – TimeExplorer [Matthews 2010]
Result visualization

• Provide **overviews** of the collection and search results
  – To understand and support an analysis

• Applications
  – manyEyes [Viégas 2007]
  – Stuff I’ve seen [Dumais 2003]
  – TimeExplorer [Matthews 2010]
Result visualization

- **Provide overviews** of the collection and search results
  - To understand and support an analysis

- **Applications**
  - manyEyes [Viégas 2007]
  - Stuff I’ve seen [Dumais 2003]
  - TimeExplorer [Matthews 2010]
Support learning and understanding

• Provide facilities for **deriving meaning** from search results

Examples

– **Wikify!**: linking documents to encyclopedic knowledge [Mihalcea 2007]
– Learning to link with Wikipedia [Milne 2008]
– Generating links to background knowledge [He 2011]
Evaluation of exploratory search

- Evaluation metrics for exploratory search [White 2006b]
  1. Engagement and enjoyment
     - The degree to which users are engaged and are experiencing
  2. Information *novelty*
     - The amount of new information encountered
  3. Task success
  4. Task time
     - Time spent to reach a state of task completeness
  5. Learning and cognition
     - The amount of the topics covered, or and the number of insights users acquire
Future direction

• Collaborative and social search
  – Support of task division and knowledge *sharing*
  – Allow the team to move *rapidly* toward task
  – Provide *already encountered* information
PART II – SERENDIPITOUS IR
Serendipitous IR

- **Serendipity** [Andel 1994]
  - The act of encountering relevant information *unexpectedly*
- **Task:** Predict and suggest relevant information
  - E.g., recommender systems
Recommender systems

• Motivation [Adomavicius 2005, Jannach 2010]
  – Ease information overload
  – Business intelligence
    • Increase the *number* of products sold
    • Sale products from the *long tail*
    • Improve users’ *experience*

• Real-world applications
  – Book: Amazon.com
  – Movie: Netflix, IMDb
  – News: Yahoo, New York Times
  – Video & music: YouTube, Last.fm
Problem statements

• Given:
  – Set of *items* (e.g., products, movies, or news)
  – *User* information (e.g., rating or user preference)

• Goal:
  – Predict the relevance score of items
  – Recommend *k* items based on the scores

<table>
<thead>
<tr>
<th>Item</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0.8</td>
</tr>
<tr>
<td>I2</td>
<td>0.6</td>
</tr>
<tr>
<td>I3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Non-personalized recommendation
Problem statements

• Given:
  – Set of *items* (e.g., products, movies, or news)
  – *User* information (e.g., rating or user preference)

• Goal:
  – Predict the relevance score of items
  – Recommend *k* items based on the scores

<table>
<thead>
<tr>
<th>Item</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0.8</td>
</tr>
<tr>
<td>I2</td>
<td>0.6</td>
</tr>
<tr>
<td>I3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Recommender System

User information

Personalized recommendation
Personalized recommendation

- Two main approaches
  - Content-based
  - Collaborative filtering
Personalized recommendation

- Two main approaches
  - Content-based
  - Collaborative filtering

<table>
<thead>
<tr>
<th>Item</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0.8</td>
</tr>
<tr>
<td>I2</td>
<td>0.6</td>
</tr>
<tr>
<td>I3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Recommendation System

Collaborative filtering recommendation

User information

Community data
Content-based recommendation

• Basic idea
  – Give me “more like this”
  – Exploit item descriptions (contents) and user preferences
    • No rating data is needed
Content-based recommendation

• Basic idea
  – Give me “more like this”
  – Exploit item descriptions (contents) and user preferences
    • No rating data is needed

• Approach
  1. Represent information as bag-of-word
  2. Compute the similarity between the preferences and an unseen item, e.g., the Dice coefficient or the cosine similarity

\[
2 \times \frac{|\text{keywords}(b_i) \cap \text{keywords}(b_j)|}{|\text{keywords}(b_i)| + |\text{keywords}(b_j)|}
\]

User profiles

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Director</th>
<th>Writer</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Lord of the Rings: The Return of the King</td>
<td>Action, Adventure, Drama</td>
<td>Peter Jackson</td>
<td>J.R.R. Tolkien, Fran Walsh</td>
<td>Elijah Wood, Viggo Mortensen</td>
</tr>
<tr>
<td>The Twilight Saga: Eclipse</td>
<td>Adventure, Drama, Fantasy</td>
<td>David Slade</td>
<td>Melissa Rosenber, Stephenie Meyer</td>
<td>Kristen Stewart, Robert Pattinson</td>
</tr>
<tr>
<td>Harry Potter and the Deathly Hallows: Part 1</td>
<td>Adventure, Drama, Fantasy</td>
<td>David Yates</td>
<td>Steve Kloves, J.K. Rowling</td>
<td>Daniel Radcliffe, Emma Watson</td>
</tr>
</tbody>
</table>
Collaborative filtering (CF)

• Basic idea [Balabanovic 1997]
  – Give me “popular items among my friends”
  – Users with similar tastes tend to have also a similar taste

• Basic approach
  – Use a matrix of user-item ratings (explicit or implicit)
Collaborative filtering (CF)

• Basic idea [Balabanovic 1997]
  – Give me “**popular items** among my friends”
  – Users with similar tastes tend to have also a similar taste

• Basic approach
  – Use a matrix of user-item ratings (explicit or implicit)

  **Implicit rating**
  - Clicks
  - Page views
  - Time spent on a page
Collaborative filtering (CF)

• **Basic idea** [Balabanovic 1997]
  – Give me “**popular items** among my friends”
  – Users with similar tastes tend to have also a similar taste

• **Basic approach**
  – Use a matrix of user-item ratings (explicit or implicit)
  – Predict a rating for an **unseen** item
User-based nearest-neighbor CF

- Given the **active user** and a matrix of **user-item ratings**
- Goal: predict a rating for an **unseen** item by
  1. Find a set of users (**neighbors**) with similar ratings
  2. Estimate John’s rating of Item$_5$ from neighbors’ ratings
  3. Repeat for all unseen items and recommend top-N items

<table>
<thead>
<tr>
<th></th>
<th>Item$_1$</th>
<th>Item$_2$</th>
<th>Item$_3$</th>
<th>Item$_4$</th>
<th>Item$_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User$_1$</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User$_2$</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User$_3$</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Find neighbors

- Measure **user similarity**, e.g., Pearson correlation
  - $a, b$ : users
  - $r_{a,p}$ : rating of $a$ for item $p$, $\bar{r}_a, \bar{r}_b$ = users’ averaged ratings
  - $P$ : set of items, rated by both $a$ and $b$

<table>
<thead>
<tr>
<th></th>
<th>Item$_1$</th>
<th>Item$_2$</th>
<th>Item$_3$</th>
<th>Item$_4$</th>
<th>Item$_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User$_1$</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User$_2$</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User$_3$</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$\text{sim} = 0.85$
$\text{sim} = 0.70$
$\text{sim} = -0.79$
Estimate a rating

• Prediction function

\[ \text{pred}(a,p) = \bar{r}_a + \frac{\sum_{b \in N} \text{sim}(a,b) \times (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} \text{sim}(a,b)} \]

– Combine the rating differences
– Use the user similarity as a weight

<table>
<thead>
<tr>
<th></th>
<th>Item(_1)</th>
<th>Item(_2)</th>
<th>Item(_3)</th>
<th>Item(_4)</th>
<th>Item(_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4.87</td>
</tr>
<tr>
<td>User(_1)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User(_2)</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User(_3)</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

sim = 0.85
sim = 0.70
Item-based nearest-neighbor CF

- **Basic idea**
  - Use the similarity between items (instead of users)
  - Item-item similarity can be computed offline
- **Example**
  - Look for items that are similar to Item₅, *or neighbors*
  - Predict the rating of Item₅ using John's ratings of neighbors

<table>
<thead>
<tr>
<th></th>
<th>Item₁</th>
<th>Item₂</th>
<th>Item₃</th>
<th>Item₄</th>
<th>Item₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User₁</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User₂</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User₃</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Problems of CF

• Sparse data
  – Users do *not rate many* items

• Cold start
  – No rating for *new users or new items*

• Scaling problem
  – Millions of users and thousands of items
  – $m = \#users$ and $n = \#items$
  – User-based CF
    • Space complexity $O(m^2)$ when pre-computed
    • Time complexity for computing Pearson $O(m^2n)$
  – Item-based CF
    • Space complexity is reduced to $O(n^2)$
Possible solutions

• How to solve the sparse data problem?
  – *Ask users to rate* a set of items
  – Use other methods in the beginning
    • E.g., content-based, or non-personalized

• How to solve the scaling problem?
  – *Apply* *dimensionality reduction*
    • E.g. matrix factorization
Matrix factorization

• Basic idea [Koren 2008]
  – Determine *latent* factors from ratings
    • E.g., types of movies (drama or action)
  – Recommend items from the determined types

• Approach
  – Apply *dimensionality reduction*
    • E.g., Singular value decomposition (SVD) [Deerwester 1990]
Hybrid recommendation

• Basic idea
  – Different approaches have their shortcomings
  – Hybrid: combine different approaches

• Approach
  1. Pipelined hybridization
     • Use content-based to fill up entries, then use CF
Hybrid recommendation

• Basic idea
  – Different approaches have their shortcomings
  – Hybrid: combine different approaches

• Approach
  1. Pipelined hybridization
     • Use content-based to fill up entries, then use CF
  2. Parallel hybridization
     • Feature combination: ratings, user preferences and constraints
Future directions

- **Temporal dynamics** of recommender systems
  - Items have short lifetimes, i.e., dynamic set of items
  - User behaviors depend on moods or time periods
  - Attention to breaking news stories decay over time
  - Challenge: **how to capture/model temporal dynamics?**
    - TimeSVD++ [Koren 2009]
    - Tensor factorization [Xiong 2010]
    - Temporal diversity [Lathia 2010]
Future directions (cont’)

• Group recommendations [McCarthy 2006]
  – Recommendations for *a group of users or friends*
  – Challenge: **how to model group preference?**

• Context-aware recommendations [Adomavicius 2011]
  – Context, e.g., demographics, interests, time and place, moods, weather, so on
  – Challenge: **how to combine different context?**
Conclusions

1. Exploratory Search
   – Users perform information seeking
     • E.g., collection browsing or visualization
   – Human-computer interaction

2. Serendipitous IR
   – Systems predict/suggest interesting information
     • E.g., recommender systems
   – Asynchronous manner
References

References (con’t)