dpSmart: a Flexible Group based Recommendation Framework for Digital Repository Systems

Boyuan Guan, Liting Hu, Pinchao Liu, Hailu Xu, Zhaohui Fu, Qingyang Wang†

Florida International University
Louisiana State University†
OUTLINE

1. Introduction
2. Related Work
3. System Design
4. Evaluation
5. Conclusion & Questions?
INTRODUCTION
MAIN PROBLEMS

- **Limited Discoverability**
  - 12,376 (24.9%) out of the total 49,737
- **Lack of assistance to explore the system**
  - 700 clients out of 1,127 with a specific record when conducting search
- **Drop off visits**
  - 978,814 visits (58.8%) out of total 1,664,813 visits

General Problems for Digital Library Domain
What do the users **WANT** to know?
What do the users **NEED** to know?

Get to know the users & their region information

Awareness about the content

Awareness about the system condition (usages, statistic, user interactions)

Data Mining !!

Recommendation System
Intelligent Recommendation System (dpsmart)

**Challenges**

- Limited user information
- Multiple Recommenders available
- Side Effect (Noise)
- Impact on the system performance

**Building User vectors from web log data only**

**Allowing multiple recommenders work collaboratively**

**Grouping Users from stereotyping model**

**Optimizing the algorithm with multi-process programming**
THE TECHNICAL CONTRIBUTIONS

• Implementing a stereotype based recommendation system in real world system

• Avoiding the intensive labor works and automating the process from the log data extraction to model training.

• Avoiding the need of personal identifiable information and reducing the noise recommendation.

• Improving the system performing by using multi-process programming.
RELATED WORKS
• **200** research articles were published
• **62** methods proposed

INTRODUCTION | RELATED WORK | SYSTEM DESIGN | EVALUATION | CONCLUSION
ADDITIONAL METHODS

GLOBAL RELEVANCE

LOCATION BASED

QUERY/TERM SUGGESTION (QS / TS)
SYSTEM DESIGN
AUTOMATIC WEB SERVER LOG MINING MODULE

Web Server Log + Database Log + Metadata Engine

By linking the records to the metadata content will extend 20,000 dimensions

<table>
<thead>
<tr>
<th>Collections</th>
<th>Items</th>
<th>Search Terms</th>
<th>ID Involved?</th>
<th>Geo-Location</th>
<th>User Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>{number}</td>
<td>{number}</td>
<td>{number}</td>
<td>{True / False}</td>
<td>{Number}</td>
<td>{Nominal}</td>
</tr>
</tbody>
</table>
**SYSTEM IMPLEMENTATION – LOG DATA**

- **Web Server Log**
  - Pre-processing
    - Bots
    - Suffixes (.js, .css, etc)
    - Localhost
    - Internal IPs

- **Database Log**
  - Unique Users Matrix
    - Contain FI#
  - Cosine Similarity for CF
  - Location based Global Relevance
  - QS / TS

- **IP Decoding**
  - Dimension Expansion
    - Subject key words
    - Suffixes (.js, .css, etc)
    - Search Condition
    - # of visit (rating)
    - # of items
    - Location (Lat, Lng)

- **Metadata Engine**

---

**INTRODUCTION | RELATED WORK | SYSTEM DESIGN | EVALUATION | CONCLUSION**
User-group clustering module - Section 3.2

User Vector

Sub-space Clustering

Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4

Use Dominate Features

Internal
Academic
Library
Active
Passive
USE DOMINATE FEATURES TO MARK USER GROUPS

IP Address
- Metadata
  - Exist?
    - No: haven't review any record
    - Yes: Search Terms
      - Exist?
        - No: Library Users
          - On Starting Page
            - Recommendation: Popular items
            - On Item Default Page
              - Recommendation: Popular items
        - Yes: Active Users
          - Term/Query Suggestion
            - On Starting Page
              - Recommendation: Popular items
          - Passive Users
            - On Starting Page
              - Recommendation: Popular items

Search Terms
- Does identifier shows in search term
  - Yes: Internal Users
    - Hide recommendation on start page
      - Default
        - Recommendation: New Added
  - No: Academic Users
    - Default
      - Recommendation: CBF
SAMPLE USER RECOGNITION BY USING DOMINATE FEATURES

Basic User Vector: searches, item, idInvolved, numVisits, LocationIfMiami, numItems

- Cluster 0: 21238 (Active Web Users)
- Cluster 1: 2565 (Passive Web Users)
- Cluster 2: 97 (Digital Library Users)
- Cluster 3: 88 (Internal Users)
- Cluster 4: 667 (Academic Users)

More available here:
http://dpanther.fiu.edu/dpanther/dpsmart/usercluster/
# Recommendation Strategy

## Introduction

- **Related Work**
- **System Design**
- **Evaluation**
- **Conclusion**

<table>
<thead>
<tr>
<th></th>
<th>CBF</th>
<th>CF</th>
<th>Global Relevance</th>
<th>QS / TS</th>
<th>Location Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Users</td>
<td>🗒️</td>
<td>🗒️</td>
<td>👍</td>
<td>🍀</td>
<td>🍀</td>
</tr>
<tr>
<td>Academic Users</td>
<td>🍀</td>
<td>🐐</td>
<td></td>
<td>🗒️</td>
<td>🗒️</td>
</tr>
<tr>
<td>Digital Library Users</td>
<td>🐐</td>
<td>🐐</td>
<td></td>
<td>🐐</td>
<td>🐐</td>
</tr>
<tr>
<td>Active Users</td>
<td>🐐</td>
<td>🗒️</td>
<td></td>
<td>🐐</td>
<td>🐐</td>
</tr>
<tr>
<td>Passive Users</td>
<td>🐐</td>
<td>🗒️</td>
<td></td>
<td>🐐</td>
<td>🐐</td>
</tr>
</tbody>
</table>
Customized recommenders module – Section 3.4

Final Recommendations

- CBF
- CF
- Global Relevance
- QS / TS
- Location Based

Stereotyping
1. **Content based filter**: calculate the item-similarity among one specific user to find a threshold to provide CB recommendation

\[ SW(u, b) = \frac{\sum_{n=1}^{\infty} (A_n \cap B_n) \times W_n}{\sum_{n=1}^{\infty} (A_n \cup B_n) \times W_n} \] (2)

- Where SW is the weighted Jaccard Similarity
- \( a, b \) are any two items in dPanther system
- \( A, B \) are any pair metadata field from n selection metadata data fields between item \( a \) and \( b \)
- \( w \) is the weight assigned for a specific pair of one metadata field

2. **Collaborative filtering**: Calculate user similarity for current user to find if there is a similar user group. Calculation based on Cosine Similarity

\[ S_{(i,j)} = \frac{\sum_{u \in U} (R_{(u,i)} - \overline{R_u})(R_{(u,j)} - \overline{R_u})}{\sqrt{\sum_{u \in U} (R_{(u,i)} - \overline{R_u})^2} \sqrt{\sum_{u \in U} (R_{(u,j)} - \overline{R_u})^2}} \] (4)

- Where \( S \) is the cosine-based similarity
- \( U \) stands for the group of users who rate both item \( i \) and item \( j \)
- \( R_{(u,i)} \) or \( R_{(u,j)} \) means the score for such the item by the user
- \( \overline{R_u} \) is the average of the \( u \)-th user’s ratings

**Very Slow!!!**
3. Global Relevance (GR): Frequency of subject key works appearance in high occurred words plus the location information

\[ GR_u \propto N_i \quad \text{where} \quad L_i \in L_u \quad (6) \]

- Where GR is the Global Reference Score
- \( u \) is the current user
- \( i \) is the item in dPanther system
- \( N \) is the number of hits of item \( i \)
- \( L \) is the location information

4. Term Suggestion (TS) / Query Suggestion (QS): The similarity of the term to the metadata is generated by using the Jaccard index as suggestion by:

\[ S_{(a,b)} = \frac{|DS_a \cap DS_b|}{|DS_a \cup DS_b|} \quad (7) \]

- Where \( S \) is the Similarity Score between the search term and the item subject keywords
- \( a \) is the input search term
- \( b \) is the item in the system
- \( DS \) is the data set consist of the key works

Very Slow!!!
EVALUATION
• Whether the multiple-process programming algorithm improves the hosting Digital Repository Systems performance?

• When dpsmart integrated into the dPanther, what are the benefits of dpSmart regarding the page views, bounce rate & drop-off rate?

SYSTEM PERFORMANCE EVALUATION

Windows Machine
4 Cores of CPU
32 GB Memory

Randomly Selected 4,000 records to run the CBF with 1-process, 3-processes, 5-processes, and 7-processes respectively

(a) The Running Time Comparison.
(b) The CPU Usage.
(c) The Memory Usage.
SYSTEM PERFORMANCE EVALUATION

Windows Machine
4 Cores of CPU
32 GB Memory

Randomly Selected 4,000 records to run the CBF with 1-process, 3-processes, 5-processes, and 7-processes respectively

Fig. 4: The Average Usage of CPU, Memory and Virtual Memory by using 1-process, 3-processes, 5-processes, and 7-processes for the task of running 4000 records.
**SYSTEM IMPACT EVALUATION**

Experiment Time Line:

- **January, 2016**: First module, customized recommender, implement
- **2018**: Model rebuilding multiple times
- **January 2019**: Entire framework implement

---


(c) The Page View Stats from Jan. to Mar., 2019

Fig. 5: The Page View Stats for the year from 2015 to 2018, from January to March in year 2018, and from January to March in year 2019.
Experiment Time Line:

- **January, 2016**: First module, customized recommender, implement
- **2018**: Model rebuilding multiple times
- **January 2019**: Entire framework implement

Fig. 6: The Bounce & Drop-off Rate for the year from 2015 to 2018, from January to March in year 2018, and from January to March in year 2019.
Experiment Time Line: 

**January, 2016**  
First module, customized recommender, implement

**2018:** Model rebuilding multiple times

**January 2019:** Entire framework implement

---

Fig. 7: The system usability statistics from 2015 to 2018.

Fig. 8: The system usability statistics from Jan. to Mar., 2019.
1. Design **a flexible framework** that allow digital library can build their recommendation system purely from log data and metadata.

2. **Facilitate multiple popular recommenders** and implement it into a real-world Digital Repository System:dPanther (http://dpanther.fiu.edu).

3. Minimize the side effect (noisy recommendation) by **applying customizable group-based recommendation strategy**

4. The experimental evaluation shows that by applying the multi-process programming, the **model building time can be significantly reduced**.

5. The system usage statistics also indicate that during the evaluation time from January 2019 to February 2019, the Page Views have increased compared to 2018, **demonstrating the effectiveness of our proposed framework**.
QUESTIONS