Exploiting User Feedback to Improve Quality of Search Results Clustering

Inbeom Hwang, Minsuk Kahng and Sang-goo Lee
Department of Computer Science and Engineering
Seoul National University
Seoul 151-742, Republic of Korea
{inbeom,minsuk,sglee}@europa.snu.ac.kr

ABSTRACT
Search result clustering provides an intuitive overview toward information contained in the search result. The goal of this research is to implement a clustering engine to provide search result clustering for various search tasks retrieving items, or objects whose contents do not contain descriptive text. Content-based similarity measures used for traditional clustering engines are not suitable for general measure, because of its domain-specific nature and lack of descriptive- ness. To remedy the problems, we exploit user feedback information to measure similarity between items. As the first approach to use user feedback information to measure similarity between general items to cluster them, we explore similarity models and algorithms suitable for clustering.

To realize usefulness of the presented clustering method, performance of the clustering is evaluated using some real-world data sets. The presented method produces more accurate clusters than clustering methods based on traditional content-based measures do. After optimizing the method, a web-based application based on the clustering method is implemented. The flexibility of the implemented system and the application enables search results clustering to be applied to search results containing various type of objects.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous

General Terms
Design, Human Factors

Keywords
Search results clustering, user feedback, interactive search

1. INTRODUCTION
As product search in e-commerce environments, search engines in practice face broader range of objects to handle. Most of the objects, such as multimedia objects including music and movies, could not be formalized and viewed as documents which traditional search engines have been retrieving.

Besides types of objects, amount of data is also increasing exponentially. As a result, number of search results had been increased significantly. It is quite common to see hundreds, even thousands of objects are returned by a single query. Search engines should present numerous results in a manner which is recognizable and informative enough. In most cases, too small faction of them can be fit in one screen, and users are given false implication about the whole set of returned objects if proper ranking method is not available. Traditional search engines took some alternative approaches to remedy this problem. Search results clustering, which clusters returned documents, is the most preferred alternative approach for re-organizing search results. By finding natural groupings of the documents in the result, document clustering methods discovers hidden relationships between documents. Clusters of documents consist a hierarchy which conceptualizes relationships between subtopics. Users can focus on a general topic by an abstract query and take more knowledge about the topic of his/her interest by drilling down to a specific cluster. Many commercial search engines and research projects succeeded to compose a better presentation method for search results by clustering them. Search results clustering is known to provide useful information to users without much domain knowledge.

The goal of this work is to implement a search results clustering engine for general objects. It includes proposing a flexible method for search results clustering and to implement a clustering engine for general objects. The proposed method should be able to be applied consistently to various types of them. To realize the flexibility, we exploit users’ interaction with objects. As did some information retrieval systems and recommender systems, we exploit users’ interaction with objects to find their relationships. Exploiting user feedback, these systems were not only able to find latent relationships between objects, but also able to compose a domain-free method. Clusters taken as result from our method are visualized by Si-Fi (http://vega.snu.ac.kr/sifi/) [8]. The web-based application is revised to visualize clusters given from the proposed method.

In this work, we realize effectiveness of our clustering method by assessing its performance. We show that clustering objects using a similarity measure based on user feedback information provides similar or well-formed clusters, compared to clustering method using content-based similar-
2. RELATED WORK

Document clustering has been studied extensively to cluster web documents. Both in research and commercial search engines, document clustering is the most preferred method among several post-retrieval result processing techniques. Although the topics regarded to this problem started getting attention in early days with [3], practical clustering engines gained popularity in relatively recent days. Commercial clustering engines including Vivisimo (www.vivisimo.com), WebClust (www.webclust.com), Fluster (www.funnelback.com) provides more friendly user experience by clustering the search results. Many researchers have conducted studies on results clustering, including [7, 13, 12, 9, 14, 6, 5]. By applying document clustering methods to web snippets returned by search engines, clustering engines gave a better topic understanding about the given query and made systematic exploration possible [2]. Previous work on this field is studied extensively in [2]. Document clustering methods measure similarity between results using document models, and it is not applicable if the target domain does not contain items represented by descriptive text. We try to expand the application range of search results clustering by using similarity measures computed with user feedback information.

User feedback has been widely used to complement documents’ or objects’ contents. By exploiting user feedback, many information retrieval systems were able to find hidden meanings and relationships between documents. Recommender systems based on collaborative filtering algorithms are one of many examples which take user feedback information largely into account. Using explicit or implicit feedback, they try to find similarly preferred items or users with similar tastes. Collaborative filtering can be applied to general items and users interact with them flexibly, regardless of items’ types. Through ten or more years of practice, these methods have shown its validity and usefulness. We measure similarities between general objects by using similar method with item-based collaborative filtering algorithm proposed in [10].

Some previous work tried to exploit user feedback to provide better-formed clusters. An early work [1] used an agglomerative clustering method to find related search queries and result pages. Though the method is too domain-specific and too simplified form, it has shown that user feedback can provide viable information for clustering. Some tried adding several constraints to clustering by taking users’ feedbacks, making a semi-supervised classification method. However, none of previous work has exploited user feedback to find relationships between objects. Clustering method we propose in this work is the first clustering method which uses user feedback information to measure similarities between items. By showing its effectiveness, we expect to show some possibilities of performance gain on traditional document clustering methods.

3. CLUSTERING WITH A SIMILARITY MEASURE BASED ON USER FEEDBACK

3.1 Advantages of User Feedback-based Similarity Measures

Similarity measure affects the quality of clustering significantly. It should be able to show highly diverse relationships between various objects. Moreover, to achieve the goal of this work, similarity measure for our method should be flexible.

Traditional clustering engines mostly used content-based measures. It is effective and efficient method for document clustering. However, for some objects such as multimedia objects (i.e. movies and music), actual contents of the objects are hard to obtain. Even if so, it is hard to justify that their content reveals the all significant features of them. One can point out that extracting or formalizing content of the objects is too domain-specific. Extraction of contents and comparison of them would not be reliable enough to be used as basis of the clustering algorithm.

Metadata is often used as an alternative, when contents of the objects are not available or not reliable. One of popular examples of metadata would be product catalogs. Product search engines rely on product catalogs to discover properties of the products and meaningful information. However, we have observed that product catalogs or other metadata are often insufficient to be used as features of the items. The similarity between them does not reveal the similarity between actual items in many cases. Take an example of a comparison of product catalogs between Amazon Kindle and Samsung Galaxy Tab¹, presented in Table 1. Product catalogs of these products do not have the same values on any properties, and the items belong to different product categories. However, their target market shares the area of tablet PC, and potential buyers of these products would consider them as similar items.

Moreover, it is possible in many cases other than the presented example, items in comparison would belong to different categories with different feature sets. It is hard to deliver a consistent similarity measure that can be applicable to various cases of comparison.

User feedback can be an alternative or complementary information for content-based information. As it is shown in the experiments, we could observe some performance gain by using user feedback information to compute the similarity

³Catalogs of two products are collected from Amazon.com product specification.
between items in our concern. Exploiting user feedback to calculate item similarity is a common approach taken by most of the collaborative filtering recommender systems. To the best of our knowledge, however, the effectiveness of using them as method for similarity measure of clustering has not been well studied yet.

3.2 Item Similarity Model

From an $n \times m$ user-item matrix $R$, the similarity between two items can be defined as cosine similarity between two $n$-dimensional vectors $R_i$ corresponding to item $i$ and $j$.

\[
similarity(i, j) = \cos(R_i, R_j) = \frac{R_i \cdot R_j}{||R_i|| ||R_j||}
\]

Cosine similarity is widely used by recommender systems, and has proven its validity by many years of practice. Although similarity measures could affect the accuracy and performance of clustering methods, cosine similarity is known to reveal the similarities between items as well as other similarity measures do, and the difference of performance was not significant than difference produced by implementation details such as formalization of preference scores or sampling of users.

By applying similarity measure to all significant pair of items, we can get an item-to-item similarity matrix. Entries having values under certain threshold can be filtered out for better performance.

3.3 Clustering Algorithm

Since the clustering is performed online, computational efficiency of the clustering algorithm is one of the major issues for search results clustering methods. It should be able to construct item clusters and corresponding labels in sub-second response time to interactively react to users’ input. Partitional clustering algorithms have lower time complexity compared to hierarchical clustering algorithms. It is reasonable to use partitional clustering methods instead of hierarchical clustering algorithms which suffer severe performance degradation with large number of items.

Labeling the clusters is another important issue to consider. Traditional document clustering engines rely on words, $n$-grams or phrases shared by documents inside clusters to label them. However, item sets in our concern cannot always share properties even if they are related to other items because of their heterogeneity or their unreliable content. Instead of making a summary or a prototype that reveals common properties of a cluster, we label clusters with a representative item contained in the cluster. A representative item of a cluster exemplify other items in the cluster, which somehow related to it.

We have chosen the k-medoids algorithm to perform clustering of the items. As a kind of partitional clustering algorithm, k-medoids algorithm provides reasonable performance and clustering accuracy. k-Medoids is preferable because it is not needed to compute additional similarity score other than values in the previously materialized item-to-item similarity matrix. Online computation of item similarity should be prohibited, since computing the similarity could be a large overhead if the density of the item-to-item matrix is high. The algorithm tries to find $k$ medoids, or representative items which maximize the sum of similarity between center of clusters and items in them. Starting from initial set of representative items, it finds local optimal solution by changing the representative items iteratively. Medoids found by the algorithm is shown to be the best set of items that represents the whole data set [11].

4. EXPERIMENTS

In this section, we evaluate the performance gain obtained from using the user feedback-based similarity measure. Using two real-world datasets, we compare the accuracy of user feedback-based similarity measure with that of a content-based similarity measure.

4.1 Datasets

We use two datasets consisting of multimedia objects. Domain of multimedia objects is good for assessing performance of user feedback-based similarity measure, because their contents are not able to obtained and compared with each other easily and reliably.

The first dataset we used has been obtained from BugsMusic\(^3\), one of the biggest music streaming service providers in Korea. It contains 1.5 billion listening logs over 17 million tracks collected from 17 million users’ interaction behavior. We sampled 1% of users after filtering out users who listened to tracks by less than 5 different artists. We aggregated track listening logs according to track’s artist and counted number of artist listening logs of each user, producing a user-artist matrix of listening counts. Users’ preference toward artists are given by implicit feedback, which users prefer music of artists they have listened more times. We have normalized value of each entry in the matrix, to prevent skew of similarity affected by small fraction of users who are overly active than others.

\[
preference(a, i) = \frac{\text{count}(a, i)}{\text{arg max}_{j \leq m} \text{count}(a, j)}
\]

We use public dataset of MovieLens\(^2\) as our second dataset. The dataset contains 10 millions of ratings that 7,000 users have done over 10,000 movies.

4.2 Measuring Quality of Clusters

We measure the effectiveness of our clustering method by re-classifying a set of pre-classified items, as it is done in

\[\text{http://bugsmusic.co.kr}\]

\[\text{http://www.movielens.org/}\]

### Table 1: A comparison between product catalogs of Amazon Kindle 3 and Samsung Galaxy Tab

<table>
<thead>
<tr>
<th>Name</th>
<th>Kindle 3</th>
<th>Galaxy Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Amazon</td>
<td>Samsung</td>
</tr>
<tr>
<td>Type</td>
<td>e-Book Reader</td>
<td>Tablet media</td>
</tr>
<tr>
<td>Operating</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Freescale 532MHz, ARM-11</td>
<td>IGHz Application Processor</td>
</tr>
<tr>
<td>Storage</td>
<td>4GB</td>
<td>16 or 32GB</td>
</tr>
<tr>
<td>Display</td>
<td>6 in</td>
<td>7 in</td>
</tr>
<tr>
<td>Input</td>
<td>USB 2.0</td>
<td>Multi-touch touch screen</td>
</tr>
<tr>
<td>Dimensions</td>
<td>7.5 * 4.8 * 0.34 in</td>
<td>7.5 * 4.7 * 0.47 in</td>
</tr>
</tbody>
</table>
The evaluation of the similarity measure was done on a number of test sets, which were categorized by genres, subjects, and/or other characteristics. We composed several test result sets for each dataset. Each test set consists of movies and musical artists classified according to their characteristics, including genre of their music, their nationality, their preferred instruments and so on. Movies directory also hierarchically classifies movies and related pages by genres, subjects and/or other characteristics. We composed several test result sets for each dataset. Each test set consists of movies and musical artists belonging to 4 to 5 different classes. Entries included in the test sets appear in one or more directories in hierarchy and have enough number of popularity to compute the similarity with others without a sparsity problem. More popular entries are selected if there are too many number of entries satisfying the selection criteria. Classes in directory hierarchy overlap with each other. We excluded entries appearing in multiple classes in each test set to build more objective ground truth. Brief explanations about each test set are presented in Table 2.

### 4.3 Effectiveness of Similarity Measure based on User Feedback

We compared performance of user feedback-based similarity measure with content-based similarity measure to prove effectiveness of our approach. For content-based similarity measure, we have used a well-known Bayesian Belief Network model to calculate similarities between metadata of objects. Previous work used Bayesian Belief Network model to process product catalog datasets for auto-classification, searching and duplication detection of catalogs. We used duplication detection method to measure similarities between metadata of objects.

F-measure of clusters obtained from both similarity measures are depicted in Figure 2 and Table 3. In all test sets, clustering algorithm with user feedback-based similarity measure produced as accurate as or more accurate set of clusters than clustering algorithm used content-based similarity measure.

### 4.4 Discussions

Similarity based on user feedback provided clusters with higher F-measure scores from six out of eight test sets. Two test sets which content-based similarity measure was more effective were more general classification of objects than classes contained in other test sets. It is related to the quality of metadata of the objects. Features needed for coarser separation of objects tends to appear on metadata, and content-based measure works better in these situations. However, content-based measures did not show their effectiveness on test sets which more conceptual classification is needed. We have observed that similarities computed by content-based similarity measure have rather discrete than continuous value, and it does not perform well when finer classification is needed. Assuming that keeping metadata suitable for most practical situations is not easy task where large number of objects exist, clustering engines would be better to use user feedback for their similarity measure, be-

### Table 2: Test sets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Test set</th>
<th>Subclasses</th>
<th>Objects</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BugsMusic</td>
<td>B1</td>
<td>5</td>
<td>105</td>
<td>Musicians by genres</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>4</td>
<td>85</td>
<td>Rock bands by subgenres</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>4</td>
<td>34</td>
<td>Instrumental artists by instruments</td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>4</td>
<td>64</td>
<td>Jazz musicians by subgenres</td>
</tr>
<tr>
<td>MovieLens</td>
<td>M1</td>
<td>5</td>
<td>52</td>
<td>Comedy movies by subgenre</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>5</td>
<td>42</td>
<td>SF movies by subjects</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>4</td>
<td>28</td>
<td>Action movies by subgenres</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>4</td>
<td>32</td>
<td>Crime movies by subjects</td>
</tr>
</tbody>
</table>

Figure 2: Quality of clusters
Table 3: Quality of clusters

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Test set</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Content-based</td>
</tr>
<tr>
<td>BugsMusic</td>
<td>B1</td>
<td>3.79</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>2.17</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>1.89</td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>1.66</td>
</tr>
<tr>
<td>MovieLens</td>
<td>M1</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>1.76</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>1.59</td>
</tr>
</tbody>
</table>

cause it reduces effort of keeping records and is able to realize various relationships between objects.

From the experimental results, we can see effectiveness of user feedback information for clustering method. Although the generality of the presented clustering method is not well shown in this work, the experiments show a possibility of user feedback information used for clustering task. Testing the performance of the presented method on more real world datasets will help to assess the generality.

5. CONCLUSION

We proposed a clustering method which uses users’ feedback information to compute similarity between objects. As the first work tried to use users’ feedback for clustering, we have shown that the proposed clustering method provides clusters fit in common sense. Effectiveness of the proposed method is realized by the experiments which compare the performance of user feedback-based similarity measure to content-based similarity measure. Compared to content-based similarity measure using metadata related with objects, clustering with similarity measure based on users’ feedback delivered better-formed clusters when fine-grained differentiations are needed.

As an ongoing work, this work showed a possibility of user feedback information used for clustering by delivering results as accurate as content-based clustering method. The flexibility of the presented similarity measure will be the basis for generalized clustering engine, which is the goal of our research. The next step of our research would be applying the presented clustering method to more domains where various types of objects are included.

5.1 Future Work

We expect user feedback information to be useful to traditional document clustering engines. Document clustering engines will gain more accuracy by taking user feedback information into account, along with document features. Moreover, user feedback information can be used to measure similarity between web documents that have dynamically changing contents.

Search results clustering should provide meaningful labels which users can understand, to make them anticipate contents of documents inside clusters. Document clustering engines have been summarizing documents in a cluster or extracting phrases shared by them to label the clusters. Our approach was to pick representative items to label clusters. It could be improved by adding contents shared by items contained in the clusters.

6. REFERENCES