Using topic hierarchies with privileged information to improve context-aware recommender systems

Brazilian Conference on Intelligent Systems, 3th, 2014, São Carlos.
http://www.producao.usp.br/handle/BDPI/48628

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
Using Topic Hierarchies with Privileged Information to Improve Context-Aware Recommender Systems

Camila V. Sundermann*, Marcos A. Domingues*, Ricardo M. Marcacini† and Solange O. Rezende*

*Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, SP, Brazil
{camilavs, mad, solange}@icmc.usp.br

†Universidade Federal do Mato Grosso do Sul
Três Lagoas, MS, Brazil
ricardo.marcacini@ufms.br

Abstract—Recommender systems are designed to assist individuals to identify items of interest in a set of options. A context-aware recommender system makes recommendations by incorporating available contextual information into the recommendation process. One of the major challenges in context-aware recommender systems research is the lack of automatic methods to obtain contextual information for these systems. Considering this scenario, in this paper, we propose to use contextual information from topic hierarchies to improve the performance of context-aware recommender systems. Three different types of topic hierarchies are constructed by using the LUPI-based Incremental Hierarchical Clustering method: a topic hierarchy using only a traditional bag-of-words, a second topic hierarchy using a bag-of-words of named entities and a third topic hierarchy using both information. We evaluate the contextual information in four context-aware recommender systems. The empirical results demonstrate that by using topic hierarchies we can provide better recommendations.

Keywords—Recommender Systems; Text Mining; Topic Hierarchy; Named Entities; Context-Aware Recommender Systems

I. INTRODUCTION

With the wide variety of products and services available on the web, it is difficult for users to choose the product or service that most meets their needs. In order to reduce or even eliminate this difficulty, recommender systems have emerged. A recommender system is used in various fields to recommend items of interest to users. One of the main areas that currently use these systems is e-commerce that interacts directly with customers by suggesting products of interest with the aim of improving its sales. These systems have been used by e-commerce websites such as Amazon1 and Netflix2.

Most recommender approaches focus only on users and items to make the recommendations. However, in many applications, it is also important to incorporate contextual information into the recommendation process [1], [7]. Researchers began to realize that the quality of the recommendations increases when additional information such as time, place, and others are used in these systems. So, the integration of contextual information in recommender systems has become a topic of increasing importance in research [1], [10], [2], [8].

Context is a concept that can assume different definitions depending on the area in which it is inserted. In this paper, context is defined as any information that can be used to characterize the situation of an entity (e.g., a web page). A context-aware recommender system makes recommendations by incorporating available contextual information into the recommendation process. Although the use of contextual information for recommendation systems has received great focus in recent years [1], [10], [2], [8], [14], there is a lack of automatic methods to obtain such information for context-aware recommender systems. For this reason, the acquisition of contextual information is a research area that needs to be better explored.

In this paper, we combine two text mining techniques to capture the context of web pages and then we evaluate this context in web page context-aware recommender systems. The first technique consists of extracting named entities, i.e., entities that can be identified by a proper name like people, organizations, places, trends and products, besides temporal and numeric expressions [13]. The second one consists of using the contextual information extracted from topic hierarchies. We first extract named entities from the web pages and then construct three types of topic hierarchies: a first hierarchy of all textual content (i.e., bag-of-words), a second one only from named entities and a third one combining both information. We empirically evaluate the contextual information from these topic hierarchies, and the results demonstrate that better recommendations can be provided.

Comparing our method with the existing context acquisition methods, we consider the item’s context (web page’s context), while most methods consider the user’s context. Besides that, our method uses unsupervised techniques whereas some methods use supervised techniques, as Hariri et al. [8], that assume there are explicit labels representing context.

This paper is structured as follows: in Section II, we present the context-aware recommender systems used to evaluate the contextual information. In Section III, we present our proposal. We evaluate our proposal in Section IV. In Section V, we
II. CONTEXT-AWARE RECOMMENDER SYSTEMS

Context-aware recommender systems learn and predict the tastes and preferences of users by incorporating available contextual information in the recommendation process. According to Adomavicius and Tuzhilin [2], contextual information can be applied at various stages of the recommendation process. Following this criterion, these systems can be divided into three categories: contextual pre-filtering, contextual modeling and contextual post-filtering.

In this work, we evaluate the effects of using the contextual information, obtained from topic hierarchies, in four different context-aware recommender systems. In next sections, these systems, representing the categories mentioned before, are described in detail.

A. Contextual Pre-filtering Approach

In a pre-filtering approach, the contextual information is used as a label for filtering out those data that do not correspond to the specified contextual information. The remaining data that passed the filter (contextualized data) is used to generate the model.

In [1], the combined reduction approach (C. Reduction) uses the contextual information as label to segment the data. A segment is defined as a subset of the overall data selected according to the context or combination of its values.

Briefly, this approach consists of the following two phases. First, using the training data, a recommendation method is run for each contextual segment (e.g., accesses on Mondays would be a segment) to determine which segment outperforms the traditional recommendation model (using user and item from the whole dataset). Second, taking into account the context of the active session, we choose the best contextual model to make the recommendation. Here the best model is the one that has the highest F1 measure.

B. Contextual Modeling Approach

In contextual modeling approach, context is used in the recommendation model, i.e., the contextual information is part of the model together with user and item data.

Domingues et al. [7] proposed a contextual modeling approach, called DaVI-BEST, which considers contextual information as virtual items, using them along with the actual items in the recommendation model. Assuming that a website has multidimensional sections, each section \( s = (u, I, C) \) is a tuple defined by a user \( u \), a set of accessed items \( I \) and a set of contextual information \( C \). DaVI-BEST algorithm transforms each multidimensional section \( s \) into an extended two-dimensional section \( s' = (u, I \cup C) \), where the contextual information is used as virtual items with regular items.

Once we have a set of extended two dimensional sessions \( S' \), building/learning a contextual recommendation model consists of applying a traditional recommender algorithm on \( S' \). Note that regular items are used to build the model and make recommendations. On the other hand, virtual items are used in addition to build/improve the recommendation model but they can not be recommended.

C. Contextual Post-filtering Approach

In contextual post-filtering approach, the contextual information is used after the traditional recommendation model construction to filter or reorder the recommendations, i.e., the context is initially ignored. When the top-N recommendations are generated, this approach adjusts the list of obtained recommendations for each user using contextual information.

Panniello and Gorgoglione [14] proposed two contextual post-filtering approaches: Weight PoF and Filter PoF. These approaches first compute the probability of user’s access items under a given context. Then, the probability is used to reorder or filter out the recommendations, respectively. The probability that a active user \( u_a \) accesses an item \( i \) under the context \( c \), can be computed as:

\[
P_c(u_a, i) = \frac{\text{Num}_c(u_a, i)}{\text{Num}_c(u_a)}.
\]

where \( \text{Num}_c(u_a, i) \) is the number of users \( u \) that also accessed the item \( i \) in the context \( c \) and \( \text{Num}_c(u) \) is the total number of users that accessed any item in the context \( c \).

In Weight PoF approach, the score of the recommendations is multiplied by the probability \( P_c(u_a, i) \):

\[
\text{score}_c(u_a, O, i) = \text{score}(u_a, O, i) \times P_c(u_a, i).
\]

In Filter PoF approach, the recommendations are filtered based on a threshold value \( P^* \) of the probability \( P_c(u_a, i) \):

\[
\text{score}_c(u_a, O, i) = \begin{cases} 
\text{score}(u_a, O, i) & \text{if } P_c(u_a, i) \geq P^* \\
0 & \text{if } P_c(u_a, i) < P^*
\end{cases}
\]

III. OUR PROPOSAL

According to Adomavicius and Tuzhilin [2], the concept of context has been studied extensively in areas of computing and other disciplines. As already stated, context can be defined in many ways, depending on the field of application. The most widely accepted definition of context and that is used in this paper was proposed by Dey [6]: ”Context is any information that can be used to characterize the situation of an entity”. The entities are, in our work, web pages. The contextual information can be of different types, and each type can have a particular structure [2]. In [1] and [14], the authors consider context as a hierarchical structure that can be represented as trees. For example, in [14], the contextual attribute “period of the year” is represented as a hierarchical structure illustrated in Figure 1.

In this paper, we propose to use a hierarchical structure of the web page contents to capture the context of textual data from these pages. This structure, called topic hierarchy, organizes texts into topics and subtopics. To illustrate it, Figure 2 presents a dendrogram, which is a binary tree where each node represents a set of documents. In topic hierarchies, for each set of documents, descriptors or topics are extracted to indicate the content of these documents. Following, we discuss topic hierarchies in details.
Hierarchical text clustering is an unsupervised machine learning method that allows automatic knowledge extraction from large textual collections. It organizes the documents into topics and subtopics, where each topic is a group of documents related to each other. Most existing clustering methods assume that the textual collections are static, i.e., it is necessary to repeat the whole clustering process whenever new information is available. Besides, these methods usually represent the textual information by using only the terms of the documents, i.e., by using bag-of-words (technical information) [15]. However, there is potential additional information embedded in the texts that can be used to complement the traditional bag-of-words representation. This additional information is called “privileged information” [17].

In this work, we consider named entities extracted from the web pages as the privileged information for the clustering. The term “Named Entity (NE)”, widely used in Natural Language Processing applications, was born, according to Sekine [16], in the Message Understanding Conferences (MUC). Named entities are information units like names, including person, organization and location, and numeric expressions including time, date, money and percent expressions [16]. For instance, in the sentence, from [13], “Flavel Donne is an analyst with General Trends, which has been based in Little Spring since July 1998”, “Flavel Donne”, “General Trends”, “Little Spring” and “July 1998” are person, organization, location and time entities, respectively.

The main reasons for using named entities as privileged information are: (i) named entities are not explicitly available in the data, so their correct recognition requires some additional processing; (ii) named entities represent rich information about documents content; and (iii) named entities are available for only a fraction of the documents, what characterizes privileged information.

In order to exploit privileged information, we use an approach for topic hierarchy construction proposed by Marcacini and Rezende [11]. This approach, called LUPI-based Incremental Hierarchical Clustering (LIHC), extends the paradigm LUPI (Learning Using Privileged Information) [17]. The LUPI paradigm incorporates privileged information in the classification task, while the LIHC method incorporates privileged information in the task of text clustering. This type of information can also be treated as a second vision/description for the data, if we consider the traditional bag-of-words as the first sight.

For a better understanding of the LIHC method, let $D^t$ and $D^p$ two sets of features, where $D^t$ are technical information features and $D^p$ are privileged information features. Let $D_{pri} = \{d^p_1, \ldots, d^p_m\}$ and $D_{tec} = \{d^t_1, \ldots, d^t_m, d^t_{m+1}, \ldots, d^t_n\}$ the sets of documents with privileged information (totaling $m$ documents) and with technical information (totaling $n$ documents), respectively, where $d^p \in D^p$ and $d^t \in D^t$. The document subset with privileged information is represented by $Y = \{(d^t_1, d^p_1), \ldots, (d^t_n, d^p_m)\}$. Various clustering algorithms are run (or repeated runs of the same algorithm with different parameter values) to obtain several clusters from the subset $Y$. To aggregate the generated clusters, the LIHC approach obtains two co-association matrixes $M^t(i,j)$ and $M^p(i,j)$ which represent, respectively, the technical information (bag-of-words) clustering model and privileged information (named entities) clustering model. The combination of these two clustering models is performed by using a consensual co-association matrix:

$$M_c(i,j) = (1 - \alpha) M^t(i,j) + \alpha M^p(i,j).$$

for all items $i$ and $j$. In this case, the parameter $\alpha$ is a combination factor ($0 \leq \alpha \leq 1$) that indicates the importance of each feature space in the final co-association matrix. The initial model of the LIHC method is obtained by applying any hierarchical clustering algorithm from the matrix $M^t$. The remaining text documents, i.e., the documents without privileged information, are inserted incrementally into hierarchical clustering. For the construction of topic hierarchies, the topic extraction is based on selection of the most frequent terms of each cluster.

As already stated, topic hierarchies can be viewed as contextual information that characterize the items, and used to better characterize the user’s preferences with respect to the items. So, in this paper, we construct topic hierarchies from the web pages and then we use the topics and their granularities as contextual information for context-aware recommender systems.

IV. EMPIRICAL EVALUATION

The empirical evaluation consists of comparing C. Reduction, DaVI-BEST, Weight PoF and Filter PoF approaches against the uncontextual Item-Based Collaborative Filtering (IBCF) in order to demonstrate how much the results are influenced if we adopt hierarchical topics as contextual information.

A. Data Set

The data set used in the experiments is from a Portuguese website about agribusiness and consists of 4,659 users, 15,037 accesses and 1,543 web pages written in the Portuguese language. The textual content of the pages is used directly to obtain the set of entities and topics.
For the topic hierarchies, we consider the topics generated by the LIHC method as contextual information. For the topic hierarchy construction, two bag-of-words are created: a bag-of-words of the whole texts (traditional bag-of-words) and a bag-of-words with only named entities. Traditional text preprocessing tasks are executed, like stemming and stopwords removal. The bag-of-words are built using the term weighting measure TF-IDF (term frequency-inverse document frequency). The LIHC combination factor expresses the weight of privileged information in the consensus clustering solution, as described in Section III. We use three values of combination factor (CI): 0 (in this case, the weights of technical and privileged information are 100% and 0%, respectively, i.e., only the traditional bag-of-words is considered), 1 (the weight of technical information is 0% and the weight of the privileged information is 100%, i.e., only the bag-of-words of named entities is considered) and 0.5 (the weights of technical information and privileged information are both 50%).

We select subsets of topics to analyze the effect of the number of topics used as context in the recommendation task. The selected subsets are: {50,100}, {15,20}, {10,15}, {10,50}, {5,10}, {5,100} and {2,7}. In this configuration {x,y}, the parameter x identifies the minimum number of items allowed in the topic, while the parameter y identifies the maximum number of items per topic. For each combination factor, the subsets of topics generated by configurations above are different. Table I presents the number of topics for each generated subset.

<table>
<thead>
<tr>
<th>Granularities</th>
<th>Number of Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CI = 0</td>
</tr>
<tr>
<td>{50,100}</td>
<td>83</td>
</tr>
<tr>
<td>{15,20}</td>
<td>73</td>
</tr>
<tr>
<td>{10,15}</td>
<td>93</td>
</tr>
<tr>
<td>{10,50}</td>
<td>201</td>
</tr>
<tr>
<td>{5,10}</td>
<td>242</td>
</tr>
<tr>
<td>{5,100}</td>
<td>516</td>
</tr>
<tr>
<td>{2,7}</td>
<td>933</td>
</tr>
</tbody>
</table>

B. Supporting Tools

For the pre-processing and the hierarchical clustering of the items, we use the tools: JPretext\(^3\) and LIHC\(^4\). They are part of Torch [12], that is a set of tools developed to support text clustering and construction of topic hierarchies. JPretext transforms the collection of texts in a bag-of-words and LIHC tool implements the LUPI-based incremental hierarchical clustering method.

Named entity recognition is performed by using REMBRANDT\(^5\), a system for named entities recognition and for detection of relationship among entities. This system was designed to recognize classes of named entities, like things, location, organization, people and others, in texts written in Portuguese. REMBRANDT uses Wikipedia\(^3\) as knowledge base for the classification of entities and it has its own interface, the SASKIA, to interact with this base. The goal of this interface is to facilitate the navigation in the structure of categories, links and redirects of Wikipedia [4].

C. Experimental Setup and Evaluation Measures

The recommendation algorithms that are used in the experiments are based on the Item-Based Collaborative Filtering (IBCF) algorithm [5]. Let \( m \) be the number of users \( U = \{ u_1, u_2, ... , u_m \} \) and \( n \) the number of items that can be recommended \( I = \{ i_1, i_2, ... , i_n \} \). An item-based collaborative filtering model \( M \) is a matrix representing the similarities among all pairs of items, according to a similarity measure. In this paper, we use the cosine angle similarity measure, defined as:

\[
\text{sim}(i_1, i_2) = \cos(\vec{\mathbf{i}}_1, \vec{\mathbf{i}}_2) = \frac{\vec{\mathbf{i}}_1 \cdot \vec{\mathbf{i}}_2}{\| \mathbf{i}_1 \| \| \mathbf{i}_2 \|},
\]

where \( \mathbf{\vec{i}}_1 \) and \( \mathbf{\vec{i}}_2 \) are rating vectors and the operator “\( \cdot \)” denotes the dot-product of the two vectors. In our case, as we are dealing only with implicit feedback, the rating vectors are binary. The value 1 means that the user accessed the respective item, whereas the value 0 is the opposite.

Given an active user \( u_0 \) and his set of observable items \( O \subseteq I \), the \( N \) recommendations are generated as follows. First, we identify the set of candidate items for recommendation \( R \) by selecting from the model all items \( i \notin O \). Then, for each candidate item \( r \in R \), we calculate its recommendation score as:

\[
\text{score}(u_0, O, r) = \frac{\sum_{i \in K_r \cap O} \text{sim}(u_0, i)}{\sum_{i \in K_r} \text{sim}(i, r)},
\]

where \( K_r \) is the set of the \( k \) most similar items to the candidate item \( r \). The \( N \) candidate items with the highest values of score are recommended to the user \( u_0 \).

We use the 4 most similar items to make the recommendations and 0.1 as a threshold in Filter PoF to filter out the recommendations, since these values provided the best results for this experiment.

The protocol considered in this paper to measure the predictive ability of the recommender systems is the All But One protocol [3] with 10-fold cross validation, i.e., the set of documents is partitioned into 10 subsets. For each fold we use \( u_0 - 1 \) of these subsets for training and the rest for testing. The training set \( T_c \) is used to build the recommendation model. For each user in the test set \( T_e \), an item is hidden as a singleton set \( H \). The remaining items represent the set of observable items \( O \), that is used in the recommendation. Then, we compute Precision@\( N \), where \( N \) equals 1, 2, 3, 5 and 10 recommendations and Mean Average Precision (MAP@\( N \)), where \( N \) equals 5 and 10 recommendations. For each configuration and measure, the 10-fold values are summarized by using mean and standard deviation. We obtained a low standard deviation in our experiments and for reasons of space we will not discuss them. To compare the two recommendation algorithms, we applied the two-sided paired t-test with a 95% confidence level.

D. Results

Table II presents the results of our ranking evaluation by means of MAP@\( N \) for three context-aware recommendation algorithms (C. Reduction, Weight PoF and Filter PoF), and also for the Item-Based Collaborative Filtering approach (IBCF), which is used as baseline. The DaVI-BEST results are not considered because they are equivalent to the IBCF results.

---

\(^3\)http://sites.labic.icmc.usp.br/torch/mod2z01/jpretext
\(^4\)http://sites.labic.icmc.usp.br/torch/doceng2013
\(^5\)https://www.wikipedia.org

---

64
In Table II, its possible to note that, at most granularity levels, the context-aware techniques are able to obtain a statistically significant improvement over the baseline (IBCF). The Filter PoF algorithm has better results than the baseline in most of the experiments, but in a minority of cases this does not occur. The explanation for this fact is that the threshold used to filter out the recommendations is too high for this particular case, i.e., a huge amount of recommendations are filtered out, decreasing the accuracy of the recommender system. For most experiments, Weight PoF algorithm outperforms the other algorithms.

Regarding the granularity, there is not a standard behavior or a trend in the experiments. We can observe that for two values of the combination factor (0.5 and 1), the granularity [5,10] presents the best results. For the combination factor 0, the results of C. Reduction and Weight PoF algorithms are better at the granularity {15,20}, while the Filter PoF algorithm has better results for the granularity {2,7}. The baseline, in turn, has the same results regardless of the number of topics because it does not use contextual information. Examining the values of the combination factor, we can conclude that, in the most of the experiments, the factor equals 0 had the best results, i.e., the topics extracted from the hierarchical clustering built with the traditional bag-of-words provide results that outperform the results obtained by using the topics of the other two hierarchical clustering. However, the best value of MAP is obtained in the combination factor equal 0.5 by the Filter PoF algorithm.

In Figure 3, we compare the precision accuracy for a varying number of recommendations. There are three graphics in that figure, being a graphic for each combination factor that shows the best results for that factor. For the combination factor equals 0, the results are better at the granularity configuration {2,7} (Graphic I). The Graphics II and III present the results for the combination factor equals 1 and 0.5, both with granularity configuration {5,10}. Analyzing these graphics, we see that contextual-aware algorithms provide better results than the baseline.

V. RELATED WORK

According to Adomavicius and Tuzhilin [2], there are three different ways to acquire contextual information: explicitly, for example, a site may obtain contextual information prompting the user to fill out a form; implicitly, for example, time information such as day, time, and others, can be obtained from the web access timestamp; and inferred, when contextual information is obtained using data and text mining techniques. In this paper, we infer context from web pages using text mining techniques. Following, some related works are presented.

In [10], Li et al. proposed methods to extract contextual information from online reviews. They investigated available restaurant review data and four types of contextual information for a meal: the company (if the meal involved multiple people), occasion (for which occasions is the event), time (what time of the day) and location (in which city the event took place). They developed their algorithms by using existing natural language processing tools such as GATE tool6.

Hariri et al. [8] introduced a context-aware recommendation system that obtains contextual information by mining hotel reviews made by users, and combine them with user’s rating historic to calculate a utility function over a set of items. They used a hotel review data from “Trip Advisor website”7.

Ho et al. [9] proposed an approach to mine future spatio-temporal events from news articles, and thus provide information for location-aware recommendation systems. A future event consists of its geographic location, temporal pattern, sentiment variable, news title, key phrase, and news article URL.

VI. CONCLUSION

In this paper, we proposed to use contextual information from three different topic hierarchies, constructed by LIHC method, to improve the accuracy of context-aware recommender systems. One of the topic hierarchies was constructed with the traditional bag-of-words (combination factor equals 0). The second one was constructed with the bag-of-words formed only with named entities (combination factors equal 1). The third topic hierarchy was constructed combining both (combination factor equals 0.5). The empirical evaluation showed that by using topics from the topic hierarchies as contextual information, context-aware recommender systems can provide better recommendations. The contextual information obtained from the three topic hierarchies improved the recommendations in 3 out of 4 recommender systems evaluated in this paper: C. Reduction, Weight PoF and Filter PoF (in most of the experiments).

As future work, we will expand our findings by using other data sets as well as other context-aware recommender systems in order to evaluate the effects of using topic hierarchies as contextual information. We will also compare our proposal against other algorithms for generating contextual information.

ACKNOWLEDGMENT

The authors would like to thank FAPESP and CAPES/Brazil for financial support (grant #2010/20564-8, #2012/13830-9, and #2013/16039-3, São Paulo Research Foundation (FAPESP)).

REFERENCES


6http://gate.ac.uk
7http://www.tripadvisor.com
Comparing the context-aware recommendation algorithms against the IBCF algorithm. The values that are not statistically significant (p-value > 0.05) are in colored cells and the best results are in boldface.

<table>
<thead>
<tr>
<th>Granularity</th>
<th>MAP@5/10</th>
<th>MAP@5/10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HCF</td>
<td>cReduction</td>
</tr>
<tr>
<td>(50,100)</td>
<td>0.302924</td>
<td>0.4511504</td>
</tr>
<tr>
<td>(15,20)</td>
<td>0.302924</td>
<td>0.4892563</td>
</tr>
<tr>
<td>(15,15)</td>
<td>0.519021</td>
<td>0.5523409</td>
</tr>
<tr>
<td>(10,50)</td>
<td>0.476242</td>
<td>0.5222088</td>
</tr>
<tr>
<td>(5,10)</td>
<td>0.494566</td>
<td>0.5474966</td>
</tr>
<tr>
<td>(1,2,7)</td>
<td>0.4966097</td>
<td>0.5597789</td>
</tr>
</tbody>
</table>

Fig. 3. Comparison of considered recommendation algorithms at different number of recommendations. Graphic I: combination factor 0 and granularity [2,7]. Graphic II: combination factor 1 and granularity [5,10]. Graphic III: combination factor 0.5 and granularity [5,10].