Effects of relevant contextual features in the performance of a restaurant recommender system

Blanca Vargas-Govea, Gabriel González-Serna, Rafael Ponce-Medellín
Centro Nacional de Investigación y Desarrollo Tecnológico
Interior Internado Palmira S/N, Col. Palmira, Cuernavaca Mor., México
{blanca.vargas,gabriel,rafaponce}@cenidet.edu.mx

ABSTRACT

Contextual information in recommender systems aims to improve user satisfaction. Usually, it is assumed that the complete set of contextual features is significant. However, identifying relevant context variables is important as increasing their number may lead the system to dimensionality problems. In this paper, relevant contextual attributes are identified by using a simple feature selection approach. Once the features has been identified, it is shown their impact in different performance aspects of the system. This approach was applied to a semantic based restaurant recommender system. Results show that feature selection techniques can be applied successfully to identify relevant contextual data. These results are important to model contextual user profiles with meaningful information, to reduce dimensionality, and to analyze user's decision criteria.

Keywords

contextual information, feature selection

1. INTRODUCTION

When we want to know new places to eat, it is common to ask friends for suggestions. Touristic and gastronomic guides are other alternatives to find good restaurants. However, friends are likely to know our taste, current location and favorite environment. Consequently, their suggestions would be more precise than those provided by the guides. Actually, recommender systems are common online services that help users to cope with information overload by retrieving useful items according to their preferences. Collaborative Filtering (CF) is a successful technique, which automatizes the social recommender scheme. CF predicts user's preferences considering opinions of users with similar interests, whereas content-based recommendation systems build a model only from the user's favorite items. Common recommendation approaches only take into account user-item-rating data, ignoring contextual information. The drawback of this scheme is the lack of personalization; thus, a tourist visiting Africa could receive a recommendation of a restaurant located in Brazil. Information such as time, location, or weather can generate tailored recommendations according to the current user's situation.

CARS-2011, October 23, 2011, Chicago, Illinois, USA.

Contextual information has become a key factor to improve user satisfaction. However, not the totality of the contextual information given to the system is relevant. Moreover, if the system requires explicit information, asking for a huge amount of data can be intrusive. On the contrary, a lack of information can lead the system to generate poor recommendations. A careful selection of relevant information could improve efficiency and predictive accuracy of recommendation algorithms. To deal with this problem, feature selection techniques have been used in different domains but have not been widely exploited in contextual recommender systems. Usually, contextual information provided to the system is chosen by experience and it is assumed to be important.

This paper presents an analysis about the effects of contextual attributes in the predictive ability of a recommender system. The study focuses on *Surfeous*, a contextual recommender system prototype based on CF and semantic models. Relevant variables were identified by applying a simple feature selection approach. Once the meaningful variables have been identified, the effects of each relevant contextual variable in the predictive performance of the system was analyzed. Results of this research have impact on three aspects: i) identification of relevant contextual attributes that users take into account when selecting a restaurant, ii) reduction of dimensionality, and iii) providing new insights about the effects of contextual variables in the predictive performance of recommender systems.

This paper is structured as follows. Section 2 presents an overview of relevant work about the effects of context and feature selection techniques applied to recommender systems. Section 3 describes *Surfeous* and its contextual features. Section 4 presents general concepts about feature selection techniques and describes the approach applied to our analysis. The experiments and results are presented in Section 5. Finally, conclusions and future research directions are given in Section 6.

2. RELATED WORK

Our literature review focuses on two topics: the importance of context, and the use of feature selection techniques in recommender systems. As an emerging and under-explored research area, context-aware recommender systems are receiving an increasing attention [10]. Although the effect of context has not been widely studied, related work reveals that contextual information is important [1]. For example, in [8] the effect of context variables in a content-based system was analyzed. Several clusters were built according to the statistic dependence between pairs of context variables. Next, the system was evaluated over each cluster to observe changes in recommendations. Results showed that context variables improve predictive accuracy. In [13] it is shown the benefits of contextual information, both in precision and using an ontology to exploit semantic concepts. Another interesting strategy consists on splitting the user profile into several sub-profiles [2]. Each sub-profile represents the user in a time period of the day. Using sub-profiles, the system could recommend music precisely as it took into account time. It was showed that accuracy could increase by making recommendations using sub-profiles instead of a single profile.

As a consequence of the integration of contextual information, adaptability and dimensionality reduction requirements on recommender systems have increased. To deal with these problems, machine learning and data mining techniques have proved to be effective. However, feature selection and machine learning techniques have mainly been applied to content-based systems. For example, in [9] it is shown an interactive recommender system based on reinforcement learning. When the system returns a huge amount of results, a feature selection algorithm is applied to reduce the results list. Another approach to feature selection is presented in [3], where similarity between users is calculated taking into account the subset of common items that best describes the user preferences. The results show that predictive performance can be improved by a careful selection of item ratings. The recommender system is based on CF and does not include contextual information. In [4] the authors tested their methodology with a recommender system based on Semantic Web technologies and collaborative filtering. Their work focuses on the assessment of relevant model features using decision trees and feature selection techniques. However, the system was not evaluated with the new learned models.

In contrast to the reviewed works, in this paper, our primary goal is to identify relevant attributes and then evaluate their effects in the system's predictive performance. A semantic recommender system that fuses social and contextual aspects was used as test bed.

3. SURFEOUS: WHERE TO EAT?

Surfeous is a recommender system prototype that uses social annotations (e.g., tags) and contextual models to find restaurants that best suit the user preferences. The recommendations are shown as an ordered list (top-n).

In regard to the social aspect, *Surfeous* uses an item-based collaborative filter approach. Its prediction process is based on the Tso-Sutter [12] extended technique that includes tags. In contrast to the common two-dimensional relationship item-attribute, tags are represented as a three-dimensional relation user-item-tag. These three dimensions are arranged as a three two-dimensional problem: user-tag, item-tag and user-item by augmenting the standard user-item matrix horizontally and vertically with user and item tags correspondingly. Thus, user tags are considered as items, and item tags are viewed as users in the user-item matrix. After the extension, user and item based CF have to be recomputed with the new matrix.

In this paper, semantic Web technologies are exploited to manage the contextual information by using ontologies to model the user and restaurant profiles [11]. *Surfeous* specifies three context models, each one with the set of attributes shown in Table 1. The models are described as follows:

Table 1: Context attributes

Service model (23 attributes)
latitude,longitude,address,city,state,country,fax,ZIP, alcohol,smoking,dress,accessibility,price,franchise, ambiance,space,services,parking,cuisine,phone,accepts, days,hours
User model (21 attributes)
latitude,longitude,smoking,alcohol,dress,ambiance,age, transportation,marital-status,children,interests, personality,religion,occupation,favorite-color,weight, height,budget,accepts,accessibility,cuisine
Environment model (2 attributes)
time,weather

1. *Service model.* It describes the restaurant characteristics. The model has 23 attributes; 6 of them: cuisine, alcohol, smoking, dress, accepts (type of payment) and parking were defined according to `http://chefmoz.org`, an online dining guide. Values are selected by the user from several possible options showed by a GUI when he/she rates a new restaurant.
2. *User model.* It describes the user profile. The model has 21 attributes; 19 of them are provided by the user when he/she signs into the system the first time or modify his/her personal information.
3. *Environment model.* It refers to the time and weather of the user's location; their values are acquired from Web services. This information restricts the search to available restaurants that have appropriate installations. In this paper, *Surfeous* considers a 3 km ratio from the user's location to select the restaurants.

To generate the recommendations, *Surfeous* gets the user location and searches for the closer restaurants from a spatial database. With this information, an ontology is created at execution time. The closer restaurants become the instances that populate the ontology. Then, to match the context models, the Semantic Web Rule Language (SWRL) is applied to a set of semantic rules. This set of rules was defined based on a market study of consumer behavior.

From the attributes of the restaurant profile (i.e., service model) a relation is created to determine if its value matches the corresponding value in the user profile. Based on a space-temporal attribute, an antecedent and a consequent are created to describe the situation. For example, if the user does not smoke, the recommended restaurants need a no-smoking area. The rule is as follows:

$$\text{smokingArea}(R, no) \ \wedge \ \text{restaurant}(R) \ \rightarrow \ \text{noSmoking}(R, true)$$

Some examples of the rules and relations are shown in Table 2. Semantic Web rules use the ontology and infer the places that fulfill the premises. Results are ranked based on the number of context rules that hold for each user query: for each different restaurant a score is computed by counting and normalizing the rules that hold. The social results are added to this score considering weights between 0.1 and 0.9 with intervals of 0.1, where 0.0 stands for context-free (i.e., only tags) and 1.0 is 100% context (i.e., only rules). In this paper, fusion is the average of the intervals between 0.1 and 0.9. Our analysis is focused on the service model to explore what do the users are looking for to select a restaurant.

Table 2: Some rules and relations

user - service profile
person(X) $\wedge$ hasOccupation($X, student$) $\wedge$ restaurant(R) $\wedge$ hasCost(R, low) $\rightarrow$ select(X, R)
user - environment profile
person(X) $\wedge$ isJapanese($X, true$) $\wedge$ queryPlace(X, USA) $\wedge$ restaurant(R) $\wedge$ isVeryClose($R, true$) $\rightarrow$ select(X, R)
environment - service profile
currentWeather($today, rainy$) $\wedge$ restaurant(R) $\wedge$ space($R, closed$) $\rightarrow$ select(R)
Relations
likesFood(X, Y) X: person, Y: cuisine-type currentWeather(X, Y) X: query, Y: weather space(X, Y) X: restaurant, Y: $\{closed, open\}$

4. FEATURE SELECTION

Feature selection techniques have proven their usefulness in machine learning to improve predictive performance, to relief storage requirements, to provide a better model understanding, and to ease data visualization.

Attributes are relevant with regard to a class if their values can separate one class from the others. For example, if a tourist has to chose a restaurant to have breakfast, attributes such as the fax number are likely to be irrelevant. Conversely, attributes such as cuisine and location are frequently a decision criterion. When the attributes can be derived from other attributes, they are redundant and can be removed. For instance, the restaurant's address can be calculated from the latitude and longitude values.

Feature selection finds the minimum subset of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using the whole attribute set. There are two main methods [5] to feature selection, one is the filter method, which makes an independent subset evaluation considering general characteristics of the data such as distance, information gain, dependency and consistency. The second one is the wrapper method, which evaluates the attribute subset using the learning algorithm; its computational cost is high. In this paper, the filter method was chosen because of its independence of the algorithm and its low computational cost in contrast to the wrapper method.

To select the relevant context features Las Vegas Filter (LVF) algorithm [7] was chosen. LVF algorithm generates a random subset S of N attributes. If the number of attributes C is less than the best (C_{best}) then the algorithm computes their evaluation measure based on an inconsistency criterion; if the inconsistency criterion is satisfied, C_{best} and S_{best} are replaced in the *Solutions* list. Otherwise, if the number of attributes C is equal than the best (C_{best}) and the inconsistency criterion is satisfied, S is added to the *Solutions*. Max was defined from experimentation ($77 \times N^5$) (see Algorithm LVF).

Inconsistency criterion. The algorithm considers that two instances are inconsistent if their attributes have the same values except for their class labels. For the matching instances, regardless of the class labels, the inconsistency count IC (Eq.1) of an instance $A \in S$ is the number of instances in S equal to A minus the number of instances of the most frequent class (k) with the same attributes of S.

Algorithm LVF (Las Vegas Filter)

Input: maximum number of iterations (Max), dataset (D), number of attributes (N), allowable inconsistency rate (γ)
Output: sets of M features satisfying the inconsistency criterion ($Solutions$)
$Solutions = \emptyset$
$C_{best} = N$
for $i = 1$ to Max **do**
  $S = randomSet(seed)$; $C = numOfFeatures(S)$
  if $C < C_{best}$ **then**
    if InconCheck(S,D) $< \gamma$ **then**
      $S_{best} = S$; $C_{best} = C$
      $Solutions = S$
    end if
  else if $C = C_{best}$ and InconCheck(S,D) $< \gamma$ **then**
    append($Solutions, S$)
  end if
end for

$$IC_S(A) = S(A) - \max_k S_k(A) \tag{1}$$

The inconsistency rate IR of S (Eq.2) is the sum of the inconsistency counts divided by the total number of instances.

$$IR(S) = \frac{\sum_{A \in S} IC_S(A)}{|S|} \tag{2}$$

To characterize our feature selection problem, *Surfeous* can be seen as a classifier that predicts if a restaurant will be high rated by the user. Contextual attributes are represented as a vector, where the class of the training instances is labeled with the rating values (i.e., 0,1,2). The goal is to find the minimum set of contextual attributes that obtain at least the same predictive performance as with the whole attribute set. Once the minimum attribute subset has been found, the next step is to analyze the effects of different contextual attributes.

5. EXPERIMENTS

The experiments have three purposes: i) to identify relevant contextual attributes, ii) to show that with the minimum attribute subset, the predictive performance is as least the same as with the whole attribute set and, iii) to analyze the effects of relevant contextual attributes.

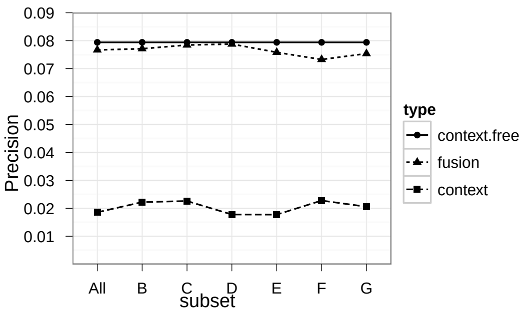
Data description. The experiments have been conducted using the data collected during a seven months period (i.e., from July, 2010 to February, 2011). Test users added and rated new and existing restaurants; they filled the attribute values described in Section 3. Data comprises 111 users that contributed with information about 237 restaurants and accumulated 1,251 ratings. Possible rating values are 0, 1, and 2, where 0 indicates that the user does not like the restaurant, and 2 denotes a high preference. Rating average is about 11.2 ratings per user; half of the ratings concentrates on the 38 best rated restaurants. There are numerous restaurants with few ratings (i.e., 65 restaurants have 1 rating), whereas less than 5 restaurants gathered more than 15 ratings. Although our sample is not in the range of thousands of instances, it presents a power law distribution usually found on recommender systems: a small number of items dominates the ratings whereas many items obtains only a few.

Attribute selection. As input to the feature selection algorithm we built a set of 5,802 instances. For each rated item, several instances were created by replacing their attribute values with different possible values. Each instance was a vector consisting of the 23 restaurant attributes described in Table 1, and rating values were given as nominal class labels. The consistency selector algorithm [7] was taken from WEKA [6], it involved a best-first search with a forward approach and 3-fold cross-validation. The remaining parameters were set to their default values. The output was a subset consisting of the following 5 attributes: cuisine, hours, days, accepts and address; 18 features were removed from the original set (i.e., 78.26% from the whole set).

Tests with Surfeous. Experimental setup is based on a leave-one-out scheme: an instance of each user was randomly taken to build the test subset (111 instances) while the remaining instances became the training subset (1,140 instances). Seven different datasets were defined: the subset All consists of the original 23 attributes. B is the minimum attribute subset (5) calculated with the feature selection algorithm (i.e., accepts, cuisine, hours, days, address). The reminder subsets (C-G) were built by removing one different attribute from B. For each set, 10 executions with normalized data were performed.

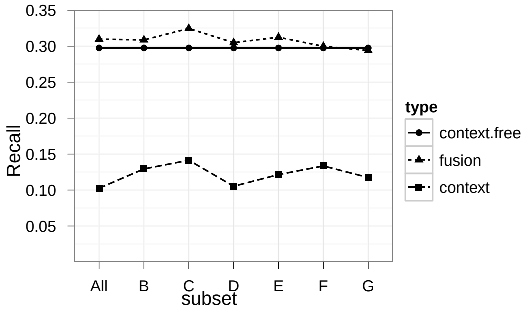
A test consists in executing *Surfeous* with each attribute set and measuring its performance. Evaluation was performed over three types of recommendations: those generated by the system without contextual features (i.e., context-free), those generated by the fusion of social and contextual aspects, and those produced only by the set of semantic rules. Furthermore, two facets of the system's performance were evaluated: its capacity to retrieve relevant items and its effectiveness to show the expected items in the first positions of the recommendation lists.

Figure 1 shows the results for precision. For fusion, the highest value was obtained with the subset D (0.0788). Although context-free precision was not outperformed, in comparison with the best result there is only a difference of 0.756%. It is a trade-off between feature reduction and performance. For semantic rules, the subset F (i.e., address, cuisine, hours and accepts) got the best value (0.0228). Although semantic rules do not show a good performance, they contribute with personalized features to the social approach with similar precision results. The relevant features of the best subsets are: hours, days, accepts and cuisine.



subset	C-free	Fusion	Rules
All	0.0794	0.0767	0.0186
B	0.0794	0.0771	0.0222
C	0.0794	**0.0785**	0.0226
D	0.0794	**0.0788**	0.0178
E	0.0794	0.0758	0.0177
F	0.0794	0.0733	**0.0228**
G	0.0794	0.0754	0.0206

Figure 1: Precision. Using the subset D (hours, days, accepts, address), fusion performed similar to the context-free model.



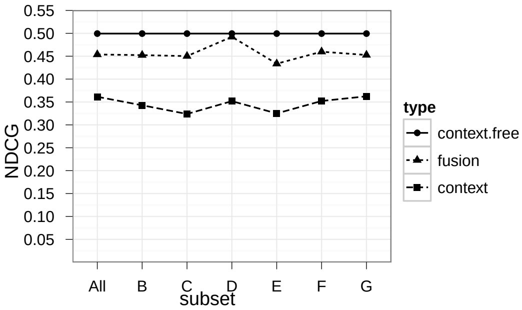
subset	C-free	Fusion	Rules
All	0.2975	0.3097	0.1025
B	0.2975	0.3086	0.1293
C	0.2975	**0.3246**	**0.1414**
D	0.2975	0.3048	0.1053
E	0.2975	0.3124	0.1215
F	0.2975	0.2997	0.1335
G	0.2975	0.2939	0.1174

Figure 2: Recall. Fusion outperformed the context-free model with most of the subsets.

Figure 2 shows the results for recall. For fusion, the majority of the subsets outperformed the context-free performance (0.2975). Subset C generated the best recall value both for fusion (0.3246) and for semantic rules (0.1414). As with precision, the most important attributes are: cuisine, hours and days. However, recall does not take into account the item's position in the recommendations list. Consequently, a system that shows the useful items in lower positions could obtain the same recall value than other system, which presents the expected items in higher positions. Since recall is unable to measure this aspect, *Surfeous* was evaluated with NDCG (*Normalized Discounted Cumulative Gain*). To compute the value, the top-k list is represented as a binary vector of 10 positions. A value of 1 is assigned to the position where the expected item appears; otherwise its value is 0. When the expected restaurant appears in the first position, it achieves the optimal score of 1 (i.e., $\frac{1}{log_2\ 2}$). Results were averaged over 23,294 queries.

Figure 3 shows that for all the attribute sets, *Surfeous* presented the expected items in the top-5 list. For fusion, with the subset D it is obtained a very similar value (0.4923) to the context-free performance (0.4994). For semantic rules, the subset G (i.e.,cuisine, hours, days, accepts) got the best score. Even though the attribute address appears as important in the subsets, *Surfeous* selects the recommended restaurants based on the user's location. For this reason, address is an implicit feature that can be discarded.

To sum up, the best subset for precision and NDCG is D (i.e., hours, days, accepts), whereas C (i.e., cuisine, hours, days) is the best only for recall. Results suggest that the restaurant opening times and its type of payment are likely



subset	**C-free**	**Fusion**	**Rules**
All	0.4994	0.4537	0.3614
B	0.4994	0.4523	0.3428
C	0.4994	0.4502	0.3238
D	0.4994	**0.4923**	0.3518
E	0.4994	0.4336	0.3249
F	0.4994	0.4598	0.3522
G	0.4994	0.4526	**0.3624**

Figure 3: NDCG. For fusion, the best NDCG score was achieved when using the subset D (hours, days, accepts, address).

to be the most important factors to make a choice. The majority of the test users are students with irregular meal hours, thus, they prefer a restaurant with flexible open hours and several types of payment even if it does not offer their favorite food.

Results for recall show that a context-free approach can be improved with the use of context features. Although the performance achieved by the semantic rules is low, they provide the social approach with features that enriches the decision process. A deep analysis of the set of rules is needed to determine the reason of their weak performance.

Identification of relevant contextual features facilitates a better understanding of the decision criteria of users. This knowledge is potentially useful to model user/item profiles with meaningful information, to design efficient user interfaces, and to improve services based on people preferences.

6. CONCLUSIONS AND FUTURE WORK

In this work a feature selection approach was applied to a recommender system that fuses social annotations and contextual models to recommend restaurants. The feature selection problem was characterized as a classification task. A subset evaluator was applied to the complete feature set, then, the effects of the relevant features in the system's performance were evaluated. It was shown that by using the reduced subset of attributes, the system's performance was not degraded. Feature selection techniques can contribute to improve the efficiency of a contextual recommender system.

As future research direction we want to extend the approach to another application domain towards deepening the understanding of contextual information in recommender systems. Turning the feature selection approach into a context-oriented technique is an interesting open issue that we would like to follow. Also, taking into account context variables in an evaluation methodology is part of our future work.

The authors thank the reviewers for their useful comments. This research was sponsored by CONACYT under grant 290586.

7. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23:103–145, January 2005.

[2] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. In *RecSys'09: Workshop on context-aware systems (CARS-2009)*, 2009.

[3] L. Baltrunas and F. Ricci. Locally adaptive neighborhood selection for collaborative filtering recommendations. In *Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 22–31, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] A. Bellogín, I. Cantador, P. Castells, and A. Ortigosa. Discovering relevant preferences in a personalised recommender system using machine learning techniques. In *Proceedings of the 8th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Preference Learning Workshop, pages 82–96, 2008.

[5] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11:10–18, November 2009.

[7] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *13th International Conference on Machine Learning*, pages 319–327, 1996.

[8] S. Lombardi, S. Anand, and M. Gorgoglione. Context and customer behavior in recommendation. In *RecSys'09: Workshop on context-aware systems (CARS-2009)*, 2009.

[9] T. Mahmood and F. Ricci. Learning and adaptivity in interactive recommender systems. In *Proceedings of the ninth international conference on Electronic commerce*, pages 75–84, New York, USA, 2007. ACM.

[10] B. Mobasher. Contextual user modeling for recommendation. In *RecSys'10: Workshop on Context Aware Recommender Systems (CARS-2010)*, Barcelona, Spain, 2010.

[11] R. Ponce-Medellín, J. G. González-Serna, R. Vargas, and L. Ruiz. Technology integration around the geographic information: A state of the art. *International Journal of Computer Science Issues*, 5:17–26, 2009.

[12] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1995–1999, New York, USA, 2008.

[13] D. Vallet, M. Fernández, P. Castells, P. Mylonas, and Y. Avrithis. A contextual personalization approach based on ontological knowledge. 17th European Conference on Artificial Intelligence - Contexts and Ontologies: Theory, Practice and Applications Workshop, Riva del Garda, Italy, 28 August, 2006.