Context and Customer Behavior in Recommendation

S. Lombardi
Università di S. Marino – DET, Str. Bandirola 46, Rep.of San Marino
sabrina.lombardi@unirsm.sm

S. S. Anand
University of Warwick - Dpt.Computer Science, Coventry CV4 7AL, UK
s.s.anand@warwick.ac.uk

M. Gorgoglione
Politecnico di Bari - DIMEG
V.le Japigia 182 Bari 70126, Italy
m.gorgoglione@poliba.it

ABSTRACT
The last few years have seen an increased interest in incorporating context within recommender systems. However, little empirical evidence has emerged to support the premise that context can actually improve recommendation accuracy. Indeed little agreement exists as to what represents the context of a user or indeed how such context should be used within a recommendation strategy. In this paper we study the effect of incorporating contextual variables, both observable and derived from past user behavior, on the accuracy of a content based recommender system. The system was evaluated using data from an Italian online retailer. Results suggest a significant improvement in performance when using contextual variables.

Categories and Subject Descriptors
H.2 [**DB Management**]: H.2.8 DB Application–*Data Mining*

General Terms
Algorithms, Performance, Modeling

Keywords
Recommender System, Context

1. INTRODUCTION AND PRIOR WORK
Personalized offers for individual customers are crucial for many kinds of businesses because they enable companies to service the unique needs and preferences of individual customers, help them build customer loyalty and finally increase their competitiveness in the marketplace. A recommender system (RS) is a personalization system that help users to find items of interest based on some information about their historical preferences.

Various classifications of RS have been proposed [2,3]. Most of these systems focus on recommending items to users based on customer's historical preferences (either explicitly stated or implicitly through purchases) and other available data on users (such as demographics) and items (such as content descriptions). These recommendation strategies are usually based on collaborative filtering [14], content-based filtering [13] or a combination of these two methods [6].

Collaborative Filtering recommends products based on the similarity of the preferences of a group of customers known as neighbors. The assumption is that users that have had common interests in the past, defined by feedback on items consumed, will have similar tastes in the future. Content based filtering systems have their roots in information retrieval [5] and information filtering [8] research. The approach is based around the analysis of items previously rated by a user and generating a profile for a user based on content item descriptors. It is usually computed by extracting a set of features from items rated previously by the user and is used to determine the appropriateness of the item for recommendation to the user. It can be varied as a classification problem. There has already been a work that treated recommending as a classification task [7].

RSs traditionally operate on a user-item matrix. As the user enters new ratings or makes new purchases, their user profile is updated by simply adding the new information to the current rating vector for the user. This additive approach to modeling the user simply ignores the notion of "situated actions" [15], that is, the fact that users interact with systems within a particular "context" and ratings for items within one context may be completely different from the rating for the item within another context. It is therefore not surprising that stories of inappropriate recommendations abound, such as the male customer buying a pregnancy book from Amazon.com as a present, persistently receiving recommendations on pregnancy related topics [10]. In fact, several studies have maintained that a change in the context makes the behavior of a customer change [9]. Nowadays, the variability of customer decision-making process decreases the capacity to predict their behavior. Furthermore, it is difficult to identify a contextual variable that affects the purchasing behavior of all customers in the same way. This is more difficult on-line.

Recent experimental research on customer modeling has proved that including contextual information in a customer's behavior model can increase predictive accuracy [12]. By considering contextual information, the customer transactions pertaining to a particular context become more homogenous, making it easier to predict customer behavior more accurately in similar contexts. Adomavicious et al. [1] showed that including the context in which transactions occur in a multidimensional model improves the ability of recommending items to users. As opposed to the use of explicit contextual variables, Anand and Mobasher [4] suggested the use of implicit contextual cues to retrieve relevant preference information from a user's long term profile and use it in conjunction with the information stored in the short term profile, generated within the current interaction, for generating recommendations.

One of the main criticisms of contextual approaches to recommendation is the reduction in the amount of data available for learning each of the contextual model. There is hence a belief that the advantage gained in incorporating context within the user model will be offset by the increased sparsity of the contextual data. This is further exacerbated by the multidimensional representations of context. In this paper, we study the effect of incorporating contextual variables, both observable and derived from past user behavior, on the performance of content based

RSs. In particular, we evaluate which contextual variable affects more the performance. Given a set of users, each with a set of transactions, first we consider the whole database and then we cluster the users based on the interactions between their contextual variables in order to understand whether the performance improves. We then build models aimed at selecting a subset of items relevant to a customer that presents himself at the e-tailors store displaying a certain context to his visit. The models are built for each context identified and for each cluster of users. In addition to the context, these models incorporate item features. We experiment with two different approaches to describing items based on structured and unstructured data.

2. PROBLEM FORMULATION

We are given a set of *n* users, *U*. The users transact with an e-tailer, purchasing items, *I*. Each item i_j may have a structured description based on a set of *s* attributes $\{ia_1,ia_2,...,ia_s\}$, that describe its characteristics such as Super_Category, Price, Brand, etc or an unstructured textual description which can be transformed into a standard vector space model developed for information retrieval. Let *T* represent the set of transactions by the users where each transaction is said to have been conducted within a context $k_r \in K$, the set of contexts. The context itself is defined by a set of contextual attributes $\{ka_1,ka_2,\ldots ka_k\}$.

Given a user, u_a, that presents himself at the e-tailer's store, we would like to be able to accurately predict the subset of *I* that would be candidates for purchase by the user. To this end, data pertaining to the set of transactions, T_a, conducted by the user, u_a in the past, is used to build a user preference model that is then used to recommend items, not currently consumed by u_a. A typical example of such a user preference model is that generated by a RS in the form of a rating function, r_a: $I \rightarrow [1,R]$, specific to a user (or set of users). Alternatively, content based models may learn a function, f_a: $I \rightarrow [0,1]$, where each item is mapped onto a likelihood of purchase by the customer (or group of customers). We refer to this model as the *uncontextual model*. In addition to the users transactions, the preference model may incorporate item attributes (descriptions) as is the case in content-based filters.

In the presence of context, we suggest that there exists a separate preference model for the user within each context. The transactions of the user are further partitioned into sets of contextual transactions, T_{kr}, based on each context, k_r, within which the user has transacted with the e-tailer. This results in a reduction in the amount of data available for learning each of the contextual preference models which has a derogatory effect on the accuracy of the function learned. To address this problem we opted to build aggregate user profiles instead of individual user profiles. There has been work on learning aggregate profiles for recommenders [11], that builds profiles that can apply across groups of users who may have similar interests. In this paper, we build a classifier not on individual purchase data but on purchase data from subsets of users. We achieve this by clustering users by their transactional behavior. The transactional behavior of a user is represented by a binary vector of size k(k-1)/2 where k represents the number of contextual variables in our model. Each element of the vector contains a 1 or a 0 for each pair of contextual variables depending on whether the pair of variables are independent of or dependent on each other. These binary vectors are clustered to produce a set of clusters of users that are affected by the same contextual variables in similar ways. For each cluster, we now have a set of "contextual states" and for each of them, we build a model that predicts the likelihood of a product being purchased by customers within that context. These models are built by using only the products purchased by customers in a particular context and cluster. Each group preference model is defined as a function, f_{ck} : $I \rightarrow [0,1]$ that map each item in I to the probability of it being purchased by a customer within the cluster and within the given contextual state c_k. For each cluster, the contextual model that gives the best performance is selected. Alternatively, if the performance of the un-contextual model is significantly better than that of the contextual model we can assume that for that cluster no contextual state is relevant.

3. EXPERIMENTAL SETUP

The experiments were conducted using a dataset from an Italian e-commerce portal which specializes in selling electronic products. The data consists of 108,238 transactions made between 2001 to 2007 and carried out by 32,429 customers. The attributes included in the dataset are: customer's ID, age, date of registration, transaction date, product purchased, product brand, product price, product category, purchased quantity, form of payment. The item table is composed of 14,955 items and the attributes included are: ID, Super_Category, Category, Big and Small Description of the item, Big and Small Photo, Price and Brand. Furthermore, each item has an associated small and big textual description where its characteristics such as color, size, shape, etc. are described. To compare the contextual and un-contextual approaches, we conducted experiments across the following experimental settings: 1) Context Model, 2) Baseline Model, 3) Clusters, 4) Item profiles and 5) Classifiers and Evaluation Metrics.

1. *Context Model.* We defined five contextual variables:

- *Time of day.* Time of the day a customer makes a transaction.
- *Time of year*. Time of the year a customer makes a transaction.
- *Weekday*. Day of the week a customer makes a transaction.
- *Super_Category*. Super Category of a product purchased.
- *Price_Sensitivity*. Analysis of the transactional data suggested that there was a strong negative correlation between "time since the first purchase of a product" and the user price for the product. This isn't that surprising as an "early adopter" or "technology enthusiast" would want to buy a new product without much consideration to price whereas a "laggard" would tend to be more price sensitive. Clearly the price sensitivity of a user may be dependent on other factors. This suggested a new contextual variable, P_{Sij}, that could be derived from the transactional data, as defined below:

$$PS_{ij} = \frac{\mathrm{PD}_{ij} \text{ - FPDate}_{\mathrm{j}}}{\mathrm{Lifetime}_{\mathrm{j}}}$$

where, $\mathrm{Lifetime}_j$ is the time elapsed between the first and last purchase of a item i_j, PD_{ij} is the time of the purchase of i_j in transaction t_i and FPDate_j is the date of the first purchase of i_j. If the expected value of PS_{ij} for customer u_i is close to one, the customers can be classified as price sensitive because he/she usually makes a purchase at the end of product life cycle when the price is lower. If it is close to zero the customers can be classified as not price sensitive. The same customer may show a different behavior in different times of day, periods of the year and days of the week or depending on Super_Category of product purchased and sensitivity to price. For instance, if the contextual variable is *Time of day*, the customer can purchase more products in evening because he/she has more time to decide and less during the rest of the day.

2. *Baseline Model.* As our experiments are aimed at showing the value of contextual recommendation, we used two baseline un-contextual and contextual models:

- A model built for the entire user base (Whole_DB), representing the non-personalized, un-contextual predictive model. We also built a set of baseline *contextual* models that were built on the entire user base but incorporated the contextual information, i.e. we built a separate model for each of the contextual states as defined above but assuming all users belong to one cluster. In this case, we recommend the same items for all customers and therefore, we do not personalize the offerings.
- A model for each cluster, personalized to a group of customers, but ignoring all contextual information. We also built a set of contextual models, for each cluster as described below.

3. *Clusters*. A simple model of context would assume that all contextual attributes are independent of each other. However, interactions between the contextual variables may exist and not accounting for these can have a derogatory effect on the predictive accuracy of the contextual model. On the other end of the spectrum, we can assume that no independence relationship exists between the contextual attributes. However, this would lead to the generation of $\Pi_k|ca_k|$ partitions of the user base, leading to sparsity in the training data set. In this study we took the middle ground of allowing for pairs of contextual variables to interact. Each user in the customer base was represented by a 10 dimensional vector, each dimension corresponding to a pair of contextual attributes and assigned one of two values {IND, DEP} depending on whether they were independent or dependent of each other. The Chi-Square test at 95% confidence was used to determine whether a given pair of contextual attributes was independent of not. These customer vectors were then clustered. Of the clusters learned, 10 clusters were deemed to have a sufficient number of transactions for further analysis (Table 1).

Table 1. User base, clusters, contextual attribute dependency

	Dependencies	Size	
		Users	Transactions
Whole_DB	None	32,429	108,238
Cluster_1	None	29,627	87,859
Cluster_2	{Time of Day, Time of Year}	1,001	7,399
Cluster_3	{Time of Day, Weekday}	413	2,665
Cluster_4	{Time of Year, Weekday}	361	2,549
Cluster_5	{Time of Year, Super_Category}	279	1,942
Cluster_6	{Time of Year, Weekday}, {Time of Day, Weekday}, {Time of Day, Time of Year}	244	1,936
Cluster_7	{Price Sensitivity, Super_Category}	98	1,110
Cluster_8	{Time of Day, Super_Category}	194	1,157
Cluster_9	{Super_Category, Weekday}	132	862
Cluster_10	{Time of year, Price Sensitivity}	80	759

4. *Item profiles*. The next task was to build a classifier for the whole customer base and for each cluster to predict whether an item would be purchased by a user within the whole database and the cluster or not. We considered two different approaches to describe items based on structured and unstructured data. In the first case, the independent attributes that described an item were Super_Category, Category, Price, Big and Small Description, Big and Small Photo and Brand. In the second case, unstructured text descriptions for the items were used as the independent variables. These texts were converted into a bag of words representation, with stop word removal, and the using the top 250 words selected based on document frequency. Hence each item was represented as a vector of 250 dimensions with values indicating the occurrence count of the corresponding word within the item description. These are used as independent variables for a classifier learned to predict the likelihood of purchase of the item by the user (or group of users).

5. *Classifiers and Evaluation Metrics*. The Weka classifier [16] that was used for building predictive models is Cost Sensitive that reweights training instances according to the total misclassification cost assigned to each class. The base classifiers that we used were J48 and JRIP for the models that used the item attributes as independent attributes, and we used Naïve Bayes Multinomial for the models that used the unstructured item description as the independent attribute, as the Naïve Bayes is known to perform well in text classification tasks. The predictive power of the models was evaluated via two performance measures: percentage of correctly classified instances (number of correctly classified instances among all cases, CCI), and True Positive rate (number of the items purchased that are predicted correctly, TP_rate). Ten fold cross validation was used to estimate the performance of the models.

4. RESULTS

In this section we present the results of the effect of context on RSs across all the experimental conditions. The aim is to analyze experimentally under which conditions a contextual RS outperforms the un-contextual. In particular, the results show that the contextual RS almost always performs better than the traditional RS for both approaches used. Table 2 (a and b) reports the contextual model that gives the best performance in terms of both CCI and TP_rate for J48 and JRIP in the case the item models include the attributes and for Bayes Multinomial in the case the unstructured text descriptions for the items were used as independent variables. In particular, for both approaches used the contextual models are built by using only the products purchased by customers in a particular context and cluster. Therefore, for each cluster we compared the performance of the five contextual models with each other and selected the best.

Table 2. Contextual attributes with the best performance

(a)

	Context that gives the best CCI		Context that gives the best TP_rate	
	J48	JRIP	J48	JRIP
Whole_DB	Super_Category	Super_Category	Super_Category	Super_Category
Cluster_1	Super_Category	Super_Category	Super_Category	Super_Category
Cluster_2	Super_Category	Price_Sensitivity	Super_Category	Super_Category
Cluster_3	Super_Category	Super_Category	Super_Category	Super_Category
Cluster_4	Super_Category	Super_Category	Super_Category	Super_Category
Cluster_5	Time of Year/Super_Category	Time of Year/Super_Category	Time of Year/Super_Category	Time of Day
Cluster_6	Super_Category	Time of Year/Weekday	Super_Category	Super_Category
Cluster_7	Price_Sensitivity/Super_Category	Weekday	Price_Sensitivity/Super_Category	Price_Sensitivity/Super_Category
Cluster_8	Super_Category/Time_of_day	Price_Sensitivity	Super_Category/Time_of_day	Time of Year
Cluster_9	Super_Category/Weekday	Time of Day	Super_Category/Weekday	Price_Sensitivity
Cluster_10	Super_Category	Super_Category	Super_Category	Weekday

(b)

	Context that gives the best CCI	Context that gives the best TP_rate
	Naïve Bayes Multinomial	Naïve Bayes Multinomial
Whole_DB	Super_Category	Super_Category
Cluster_1	Super_Category	Super_Category
Cluster_2	Super_Category	Super_Category
Cluster_3	Super_Category	Super_Category
Cluster_4	Super_Category	Super_Category
Cluster_5	Time of Year/Super_Category	Time of Year/Super_Category
Cluster_6	Super_Category	Super_Category
Cluster_7	Price_Sensitivity/Super_Category	Price_Sensitivity/Super_Category
Cluster_8	Super_Category/Time_of_day	Super_Category/Time_of_day
Cluster_9	Super_Category/Weekday	Super_Category/Weekday
Cluster_10	Super_Category	Super_Category

The contextual model that provides the best performance for the whole database and almost any clusters is either Super_Category or the one that is composed of dependence of Super_Category with other contextual attributes, when we use J48 and Naïve Bayes Multinomial. The result depends on the cluster and the evaluation metric when we use the JRIP algorithm.

Table 3. CCI and TP_rate for J48 and JRIP algorithms

		Uncontextual		Context	
		J48	JRIP	J48	JRIP
W_DB	CCI	75.503±0.246	69.636±0.339	96.667±0.052	91.861±0.120
	TP_rate	0.856±0.014	0.888±0.014	0.953±0.015	0.903±0.015
Cl_1	CCI	76.610±0.287	68.566±0.345	96.674±0.052	88.516±0.163
	TP_rate	0.840±0.013	0.885±0.014	0.952±0.015	0.915±0.015
Cl_2	CCI	81.043±0.246	72.036±0.323	96.392±0.056	84.513±0.210
	TP_rate	0.788±0.013	0.798±0.013	0.878±0.014	0.852±0.014
Cl_3	CCI	80.174±0.188	83.32±0.255	96.297±0.057	85.335±0.201
	TP_rate	0.682±0.011	0.758±0.012	0.792±0.013	0.848±0.013
Cl_4	CCI	86.540±0.187	79.805±0.258	96.763±0.050	88.646±0.161
	TP_rate	0.677±0.011	0.745±0.012	0.832±0.013	0.823±0.013
Cl_5	CCI	86.406±0.188	80.434±0.252	97.172±0.044	84.375±0.211
	TP_rate	0.657±0.010	0.779±0.012	0.635±0.010	0.753±0.012
Cl_6	CCI	87.115±0.180	81.043±0.246	96.676±0.052	82.778±0.228
	TP_rate	0.650±0.010	0.770±0.012	0.769±0.012	0.816±0.013
Cl_7	CCI	90.505±0.138	83.738±0.218	96.933±0.048	84.773±0.207
	TP_rate	0.619±0.010	0.756±0.012	0.725±0.012	0.804±0.013
Cl_8	CCI	89.214±0.154	78.964±0.266	97.216±0.043	83.578±0.220
	TP_rate	0.580±0.009	0.763±0.012	0.627±0.010	0.751±0.012
Cl_9	CCI	90.879±0.133	84.427±0.211	97.162±0.044	84.400±0.211
	TP_rate	0.582±0.009	0.745±0.012	0.592±0.009	0.757±0.012
Cl_10	CCI	92.036±0.177	85.945±0.194	97.127±0.045	90.930±0.132
	TP_rate	0.565±0.009	0.746±0.012	0.645±0.010	0.743±0.012

Table 4. CCI and TP_rate for Naïve Bayes Multinomial

		Uncontextual	Context
		Naïve Bayes Multinomial	Naïve Bayes Multinomial
W_DB	CCI	77.987±0.275	89.955±0.149
	TP_rate	0.511±0.008	0.916±0.015
Cl_1	CCI	78.656±0.269	89.684±0.148
	TP_rate	0.516±0.008	0.916±0.015
Cl_2	CCI	85.965±0.193	90.210±0.142
	TP_rate	0.618±0.176	0.896±0.014
Cl_3	CCI	87.469±0.176	91.849±0.120
	TP_rate	0.638±0.010	0.874±0.014
Cl_4	CCI	87.362±0.177	93.842±0.093
	TP_rate	0.652±0.011	0.919±0.015
Cl_5	CCI	86.747±0.184	92.638±0.109
	TP_rate	0.623±0.010	0.822±0.013
Cl_6	CCI	86.954±0.182	93.155±0.102
	TP_rate	0.666±0.011	0.881±0.014
Cl_7	CCI	88.405±0.164	93.546±0.097
	TP_rate	0.728±0.012	0.885±0.014
Cl_8	CCI	88.151±0.167	92.402±0.113
	TP_rate	0.652±0.010	0.904±0.014
Cl_9	CCI	88.439±0.164	91.901±0.119
	TP_rate	0.677±0.011	0.858±0.014
Cl_10	CCI	88.773±0.160	92.617±0.110
	TP_rate	0.695±0.011	0.923±0.015

Table 3 reports the CCI and the TP_rate for Whole_DB and all Clusters considered for un-contextual model and the contextual model that provides the best performances. Almost all contextual models perform better than the un-contextual in terms of both CCI and TP_rate. Only the contextual models built for Cluster 5 perform worse using either algorithm than the un-contextual model in terms of true positive rate. The values of both performance measures are higher when we use J48 algorithm. It is worth noting that the performance obtained for the whole customer base are either similar or higher than the ones obtained for clusters. In particular, the accuracy of the Whole_DB obtained by using J48 algorithm is either similar or lower than the one of Clusters while the true positive rate is always higher than the one of Clusters. Both the accuracy and true positive rate of the Whole_DB obtained by using JRIP algorithm are almost always higher than the ones of Clusters.

Table 4 reports the CCI and the TP_rate for Whole_DB and all Clusters considered for un-contextual model and the contextual model that provides the best performances. This is the case where the unstructured text descriptions for the items were used as the independent variables. Table 4 shows that there is at least a contextual model that leads a better performance. Furthermore, we can observe that the accuracy of Whole_DB is always lower than the one of Clusters while the true positive rate is almost always higher than the one of Clusters.

Comparing the results reported in Tables 3 and 4, it is worth noting that when we use item models that have the unstructured item description as the independent variables, the context affects more on the true positive rate than on the percentage of correctly classified instances. The contrary occurs when we use item models that include the attributes. Furthermore, in both tables, CCI values are almost equal for all user categories, but TP_rates are rather different. This may depend on the fact that the dependent variable is unbalanced. In fact, the accuracy may be biased to favor the majority class (product is not purchased) and therefore, it could assume values too high.

Table 5. Items purchased in each Super_Category

	Accessory_telephony	Audio-Home Hi-Fi	Car Audio	Photography	Hardware	Telephony	Video	Videogames
Whole_DB	463	401	195	484	1434	870	254	136
Cluster_1	460	386	189	474	1377	864	252	128
Cluster_2	257	110	66	190	421	589	83	37
Cluster_3	161	62	39	110	210	391	45	13
Cluster_4	143	61	27	98	143	383	36	10
Cluster_6	143	50	26	105	135	319	48	0
Cluster_10	53	18	0	36	55	173	0	0
Total items	670	1,061	502	1,181	9,052	1,119	920	450

Table 6. Items purchased within each contextual model for different Super Categories and Clusters

(a)

	Accessory_telephony	Audio-Home Hi-Fi	Car Audio	Photography	Hardware	Telephony	Video
Autumn	43	22	0	32	54	173	17
Spring	52	12	0	39	39	132	13
Summer	39	18	0	36	36	138	0
Winter	41	20	19	27	60	98	0
Total items	670	1,061	502	1,181	9,052	1,119	920

(b)

	Accessory_telephony	Photography	Hardware	Telephony
Not_Price_Sensitive	60	0	0	93
Price_Sensitive	49	31	47	188
Total items	670	1,181	9,052	1,119

(c)

	Accessory_telephony	Photography	Hardware	Telephony
Afternoon	57	33	55	125
Evening	36	28	29	92
Morning	30	38	41	157
Total items	670	1,181	9,052	1,119

(d)

	Accessory_telephony	Photography	Hardware	Telephony
Weekday	49	24	49	207
Weekend	43	26	42	55
Total items	670	1,181	9,052	1,119

Most results show that the contextual attribute that provides the best performance is Super_Category for both approaches used. Tables 5 and 6 report the total number of items belonging to each Super_Category and the number of items purchased within each Super_Category for the whole customer base and for each cluster. Since there is a dependence between Super_Category and other contextual attributes for four clusters (see Table 2) Table 6 reports the number of items purchased in the following contextual models: (a) Super_Category/Time of Year for Cluster_5, (b) Super_Category/Price_Sensitivity for Cluster_7, (c) Super_Category / Time of Day for Cluster_8, (d) Super_Category / Weekday for Cluster_9. The goal is to explain the importance to use the clusters rather than the whole customer base. In fact, the tables show that the number of items purchased decreases moving from Whole_DB to each Cluster. Although the performances obtained for the Whole_DB are almost always better than the ones obtained for all Clusters, it is attractive to use clusters because it is possible to personalize the offers and decrease the number of items to recommend. In fact, the user is more interested in examining a small set of recommended items rather than a long list of candidates.

5. CONCLUSION

This research aims at studying the effect of incorporating context with the process of predicting a set of items relevant to the user by using two different approaches.

The main conclusions of our study can be summarized as follows. Firstly, the contextual model almost always performs better than un-contextual in terms of both accuracy and true positive rate when we consider item models that include attributes. In particular, the effect of context is stronger on the percentage of correctly classified instances. Secondly, there is at least a contextual model that leads a better performance when the unstructured text descriptions for the items were used as the independent variables. The effect of context is stronger on the true positive rate. Thirdly, the performances of Whole_DB is almost always better than the ones of all Cluster for both approaches used. However, the total number of items that can be recommended is too high. While a threshold on likelihood may be used to reduce the number of recommendations, the fact remains that the recommendations are not personalized. For this reason, it is attractive to consider the clusters in order to personalize offerings with a set of items relevant to a lower number of users.

The results cannot be generalized to every dataset and to all industry sectors, but this experimental study may represent a meaningful initial step showing that the performance of RS improves in terms of both predictive accuracy and the true positive rate by exploiting contextual information. Further research is required in order to better understand the use of context in RS. In particular, it will be necessary to improve our method because the un-contextual model is affected by the contextual information when we build the clusters in a way. In addition, we plan to do the same experiments by using a collaborative filtering approach and to build an ensemble model in order to improve the performance of the contextual models. In fact, it was noticed that an ensemble of individual predictors performs better than a single predictor on the average. Our goal is to combine the results of each contextual model that will have a different weight depending on the results obtained in our previous experiments, to make better recommendations.

6. REFERENCES

[1] Adomavicius, G., Sankaranarayanan, R., Sen S., and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. T. on Inf. Sys. 23, 103-145.

[2] Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE TKDE 17, 6, 734-749.

[3] Anand, S.S., and Mobasher, B. 2005. Intelligent techniques in web personalization. In Mobasher, B. and Anand, S. S. (eds.) Intelligent Techniques for Web Personalization, Lecture Notes in Computer Science 3169, Springer-Verlag, Berlin Heidelberg, 1-37.

[4] Anand, S.S. and Mobasher B. 2007. Contextual recommendation. In B. Berendt, A. Hotho, D. Mladenic, and G. Semeraro (eds.) From Web to Social Web: Discovering and Deploying User and Content Profiles, LNAI 4737, Springer-Verlag, Berlin Heidelberg, 142–160.

[5] Baeza-Yates, R., and Ribeiro-Neto, B. 1999. Modern Information Retrieval. Addison-Wesley, Wokingham, UK.

[6] Balabanovic, M., and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. CACM 40, 3, 66-72.

[7] Basu C., Hirsh H. and W. Cohen. 1998. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proc. 15th of AAAI-98 (Madison, WI, USA, July 26-30), 714-720.

[8] Belkin, N., and Croft, B. 1992. Information filtering and information retrieval: two sides of the same coin? CACM 35, 12, 29-37.

[9] Bettman, J.R., Luce, M.F., and Payne, J.W. 1998. Constructive consumer choice processes. J. Consumer Research 25, 3, 187-217.

[10] Chi, E.H. 2004. Transient user profiles. In Proceedings of the Workshop on User Profiling, 521–523.

[11] Mobasher B., Dai, H., Luo, T., Nakagawa, M. 2002. Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. Data Mining and Knowledge Discovery 6, 1, 61-82.

[12] Palmisano C., Tuzhilin, A., Gorgoglione, M. 2008. Using Context to Improve Predictive Modeling of Customers in Personalization Applications. IEEE TKDE 20, 11, 1535-1549.

[13] Pazzani M.J, Muramatsu, J., Billsus, D.1996. Syskill & Webert: Identifying interesting web sites. In Proc. of AAAI-96 (Portland, OR, USA, August 4-8), 54-61.

[14] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proc. of CSCW 94 (Chapel Hill, NC, USA, October 22-26) , 175-186.

[15] Suchman, L.A. 1987. Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge University Press, New York, NY, USA.

[16] Witten, I.H., and Frank, E. 2005. Data mining. Practical machine learning tools and techniques. Elsevier, San Francisco, CA, USA.