ISSN 1870-4069

Contextual Variables Relationships and their Effect on Recommender Systems

Aram González, Jorge A. Ramírez

Tecnológico de Monterrey, Campus Estado de México, Mexico

{a00965707@itesm.mx, juresti}@itesm.mx

Abstract. The internet in our daily lives has made a great impact in the availability of information. However, the quality and relevance of the data may vary depending on what each person is looking for. Since every person has different interests and priorities, data can mean nothing to someone while being a top priority for someone else. Information retrieval and recommender systems were created to alleviate this phenomenon. Traditional recommenders do this by considering the users' ratings and preferences over a set of items. Following the success of collaborative filtering and other methods, and as part of their natural evolution, recommender systems began to include more information into their learning process. Data about the users, items and their surroundings are common trends to improve the recommendation process. The inclusion of new data sources lead to what today is known as context-aware recommender systems. This type of systems include data regarding the situations in which the system, user and item are involved when dealing with the recommendation process in previous and current interactions. In this paper, we present a model for contextual data capable of finding and modeling the relationships between the context variables, the users' preferences and the items' characteristics. The system generates recommendations with specific data in the queries requested by the users. Our model considers a general situation of the user's contextualized preferences. After we generate the initial model, we use collaborative filtering techniques to deal with specific characteristics presented by each item to generate our recommendation.

Keywords: context-aware recommender systems, Bayesian model, user model.

1 Introduction

A recommender system's objective is to give the best available suggestions to users over a set of items. This is often achieved with methods such as collaborative filtering algorithms or regression models.

Thus, the main goal is simplified and becomes the prediction of the item's score. However, the metrics for evaluating such algorithms have been previously discussed and found that they are not entirely effective and meaningful [6].

As we know, data from different sources has been used to achieve this goal. However, most of the acquired information has been used as a filtering condition. Even with the current models generated by the mentioned data, they tend to only describe the user's data and do not explicitly consider the relation between the data.

The evolution of context-aware recommender systems is a good example of the inclusion of surrounding data with the hope that it will improve the system's recommendations. From the start, and because of collaborative filtering's popularity, the contextual data is commonly used as part of the filtering process.

The data inside recommender systems is traditionally represented as matrices. This allows researchers to treat them as input data for collaborative filtering algorithms. The problem with this approach is that it tries to look for similarities between users or items in order to predict a score, while it ignores or, in the best case scenario, implicitly finds the relations between the factors used to compare the similarities between the users or items.

When using this approach, few things can be done with that relation's knowledge. Using a different approach may prove useful. By having the concrete model, it can be adjusted and refined so that it becomes a better representation of the user.

The information from the users about their ratings over a set of items can be seen as an abstract representation of them, giving us simple user models to work with. However, by excluding contextual information, we lose the interpretive power that is available to us to determine the user's goals and objectives. This will also prevent us from giving the best choice in the best possible way [5].

The relevance of a proper user model is based on the fact that users decisions are not only affected by the presence of distinct factors, but also by the interaction between them. That is, for example, the weather context variable may greatly affect the user's mood when rain is present. But in a sunny day, the weather context variable may not have the same impact on the mood of the user. Both cases have an impact on the user's final decision, but the same context variable has a different weight depending on the current condition.

There are comparative studies between representative members of context-aware recommender systems' approaches. Paniello *et al.* [11] conclude that contextual modeling approaches are some sort of best bet when deciding for an approach, even if they do not exceed the rest in every aspect considered in their evaluation. We decided to follow the contextual modeling approach to take advantage of an existing explicit model for our users.

Being part of the decision process, the relations between the context variables seem to be a relevant part to be considered. This requires a structure that is capable to represent the relations.

For this purpose, we decided to use Bayesian networks as the base for our user model. Bayesian networks are a graphical representation of conditional probabilities tables. They can be used to establish a causal relation between its components.

2 Previous Work

Collaborative filtering and techniques that combine it with other approaches are the most common approach to traditional recommender systems. Collaborative filtering recommender systems generate suggestions based on the similarity existing between the users in the system. It comes from the human behavior of rating items based on the opinions of other people [13].

Collaborative filtering methods are designed to produce recommendations personalized for users based on patterns of ratings without external information for items or users. So, by design, the basic data for collaborative filtering methods mostly depends on the ratings.

Adomavicius and Tuzhilin [1] established that collaborative filtering recommender systems try to predict the utility of a specific item based on the scores given by other users. This a broad definition of this type of approach. This definition means that every system that uses data from distinct users to help determine a specific user's rating for an item can be associated with collaborative filtering.

This is a common approach given the lack of data (ratings), given by a single user over a extended period of time. For our approach, we use the data from different users to establish our initial model that is shared by all the users. This model changes over time by both, the implicit and explicit feedback from users. The implicit part requires users natural interaction with the system, this means it needs them to use and rate the items presented. This allows the system to recalculate both the network's structure and its probabilities over time.

The explicit part comes from the user directly modifying the set of rules presented to them. This can modify the way in which each query is procesed. While this doesn't directly modify the calculated probabilities, it changes the form in which they are used.

Up until this point we could consider our approach to be partially collaborative filtering. However, the algorithms commonly presented as collaborative filtering focus their research on solely on improving their accuracy on the predicted ratings.

This has proven to be a common research area that allows recommender systems to be very accurate on their predictions. In fact, it's still a relevant research area that is presented in the works of Liu *et al.* [8], Nilashi *et al.* [9], and Ramezani *et al.* [12], just to mention some recent works.

While we can produce recommendations and inference over the possible ratings for current items. Its main focus is on the relations that exist between the data it considers, it being a combination of user, item, and contextual characteristics. And the second main focus is on the personalization of the experience by giving two ways to modify the current process.

This gives our approach the chance to present useful information both to the traditional recommender system's user, but it also allows to generate a data analysis on the background information that it uses in order to produce its results.

The explicit relation between context variables has also been a subject of study. A recent case is the one proposed by Baltrunas *et al.* [3]. They also came to the conclusion that context variables have a different relevance. While it still lacked the focus on the relations between context variables, it ends up being another step into the course of action that we follow.

Ono *et al.* [10], proposed a Bayesian network approach as the main user model for a context-aware recommender system. In their paper, the model included data from the user, the context and the item. They managed to obtain a big number of attributes for each system's element. While the overall process is similar to our approach, the construction of the Bayesian networks is different. Ono *et al.* focus their efforts in dealing with a lot of attributes. In contrast, our main objective is to adjust the Bayesian network's structure and conditional probabilities so that both match the current user and context. Also our approach adds a feedback process to let each user further the personalization of his model.

3 Data

The lack of context representation in recommmender systems may lead to a loss in predictive power. Adomavicius and Tuzhilin [2], expose this point. By acknowledging this, we decided to work with a context-aware recommender system as our base.

While looking for a dataset, we were not able to find one with enough number of context variables and enough records to support the presented approach. The LDOS-CoMoDa dataset [7], satisfies most of the contextual requirements, but it lacks the amount of records required to show the actual differences between the created user models.

For this reason, we decided to develop our own dataset. We are currently working a vacation dataset. At the time this paper was written, we have 240 surveys collected. The answers are related to a group of people with similar characteristics as our movies dataset.

This dataset is focused in vacation recommendation. We asked about the place to which every user last traveled, the dates, companions, experiences, activities, lodging's characteristics as well as a rating for both the place and the experience. On this new dataset, we tried to focus more on the information that involves the entire trip and that can affect the traveler's experience.

The dataset is aimed at describing in the most detailed way possible the circumstances in which a person decides to take a vacation. So we included traditional contextual variables used on other works like season of the year, weather and time spent. But we also included data about the conditions during the trip, characteristics of the visited places and we also included the user's opinion about the lodging choices and what the elements people use to choose and rate them.

4 Bayesian Network as Model for Contextual Data

A Bayesian network allows us to visually represent the conditional probabilities of our system. A Bayesian network, even when data is missing, can learn the causal relationships between the domain's problems and characteristics. It also helps to predict future events by inferring how likely it is for them to appear [4].

We decided to use Hill Climbing and the Bayesian Information Criterion to obtain our conditional probabilities with the structure of our network. We decided to stick to the Bayesian Information Criterion because, after testing methods like the Akaike Information Criterion and Gaussian log-likelihood, we found no evidence that they outperformed the original score. In fact, Akaike results were really close to Bayesian information Criterion.

By analyzing the data, we found the same tendency in the expected error. This is, that it tends to diminish as we get more data. However, we also found that a correlation exist between the number of variables and the rate in which the expected error diminishes. Still, by using our approach, we are able to generate recommendations and generate some information about the decision process, albeit it might not be as accurate as when more data is available.

The rate in which the expected loss decreases is shown in Table 1 and in Figure 1. After some tests with 10 fold cross validation, the system showed a great amount of instability when tested with less than 150 answers.

Table 1. Expected loss function from our vacation dataset.

Data quantity	Expected loss
50	1104.629
75	∞
100	118.9803
125	1807.76
150	81.91782
175	64.09887
200	64.67456
225	73.79942
240	82.79571

The source of this behaviour can be explained as:

1. Amount of variables - The number of variables used included in this dataset is around 90. This includes the item's features as well as context variables. This makes it harder for the Bayesian network to accurately represent and predict the behaviour of the user.
2. Data outliers - While collecting our data, we found that some of the points are spread as shown in Figure 2. We have some data outliers that complicate the learning process of our model. Since it seems to be less identifiable patterns, the amount of data required for stabilization increases.

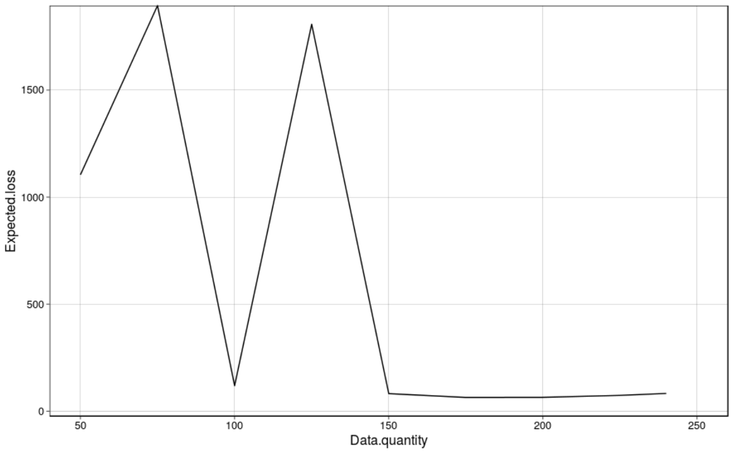

Fig. 1. Expected loss function from our vacation dataset.

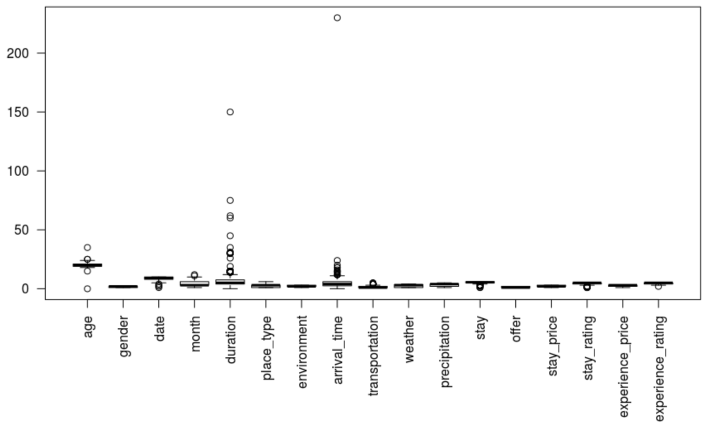

Fig. 2. Vacation dataset current status.

As an initial result, we obtain the Bayesian network's structure and the conditional probabilites behind it. Table 2 shows an example of the conditional probabilities of the types of places that users go on vacation.

We can begin to observe some of the effects of the variables by looking at the conditional probabilities tables generated with our Bayesian network. For example, Table 3 shows how the quality and rating of the place in which a person sleeps has a direct impact on the score the user gives to the whole trip. While some of this relations may seem obvious, it is relevant to ensure that our approach can determine them and that it can also find some new relationships that we might not really account for.

Table 2. Probabilies of the type of places that users will visit on vacation.

Place type	Probability
City	0.3064667
Mountain	0.1419333
Beach	0.1760667
Town	0.1562667
Remote place	0.08686667
Historical site	0.03406667

Table 3. Probability of experience scores given a move score of 4 out of 5 or more given the stay rating.

Stay rating	Probability
Between 0 and 1	0.155595
Between 1 and 2	0.306841
Between 2 and 3	0.4776664
Between 3 and 4	0.7134526
Between 4 and 5	0.9094099

4.1 Getting the First Step of the Recommendation

The first phase prediction of this particular recommender system is based on the inference process of the Bayesian networks. The inference is done by calculating the posterior probabilities from our conditional probabilities. The posterior probabilities can be calculated by applying the Bayesian rule (and the chain rule).

So we feed our particular queries with the information that is currently available. By using this type of inference, we can create complex queries for the system. The queries may include any of the variables we used for building our network and it will return us the probability of that being true. For example, if we wish to know if a user would like to have a swimming pool in his hotel in a winter vacation, we generate a query like this:

$$p(vacation\_rating >= 4 | season == summer | hotel\_pool == True).$$

An advantage of this approach is that we can get information about any of the variables presented in our network. So if we like to know which season is preferred by our users on winter, we can make a query to ask that. The queries can be as general or specific as we require them to be.

As a result we get a probability that indicates us how probable it is that our current user will enjoy a trip with those characteristics. While this approach is particularly suitable to ask for specific situations, it requires a query for each posibility when searching for each possible recommended item.

This gives our system a flexibility that is not present in other approaches. We can do queries with as little as no contextual data, and also do queries with up to every variable specified so that it can find the best match for our user.

This approach can also lead to obtain a better analysis of the users behaviours and the effect that contextual variables have in them. From a commercial view, this can also help a company to obtain a better understanding of their current user base so that they can offer services specially focused for them.

As flexible as this approach is, it is clear that this method does not recommend items in particular. The output are the categories that the user may enjoy. To deal with this issue, we use a different algorithm as an external filter.

5 Getting a Specific Item Recommendation

We follow two paths to get a specific item after the first phase of our recommendation process. We look for the item's characteristics and apply one of the following approaches:

– Rules - This are rules defined by the users that help the system personalize their results. Let's say our system knows that our user likes to spent winter vacations on the beach. This would lead our first phase to suggest such environment. However, to decide between Miami and Acapulco, we need more information. If the user sets that he likes to travel to a foreign country during vacation, then we have a new filter that will help us decide between our possible items. Of course this has the limitation that for being useful, it needs to have direct feedback from our user. This is were our second approach comes into play.
– Collaborative filtering - Collaborative filtering are a group of algorithms that can predict the score a user would give to a certain item based on the scores that similar users gave to similar or the same item. By looking at this, it alleviates our previous problem. We already have the score information for some items given by some users. We now need to apply the collaborative filtering to the set of items that meet the characteristics defined by our first phase. This helps us to reduce the search space and get a personalized result.

An advantage of dividing our approach in two phases is that we can work on both of them to either improve the predictions accuracy, or to get more information about the elements involved in the recommendation process.

In general, our model can be seen as shown in Figure 3. We first aquire our user's preferences and the context in which he is placed. Then a Bayesian network is used to determine the best characteristics of the items that can be recommended. Last, we use a filtering algorithm to determine the best match on the items that meet the same characteristics defined by our Bayesian network.

6 Conclusions

We presented a method based on Bayesian networks to represent and handle the different variables that can interact and affect the user's decision when looking for

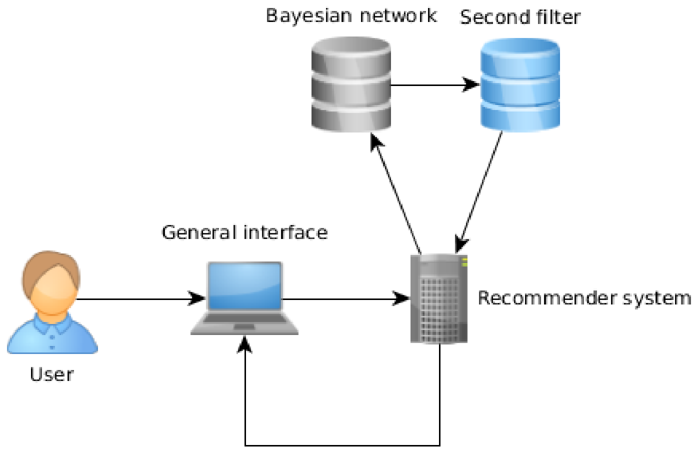

Fig. 3. General structure of our model.

a recommendation. We finish the recommendation process by using traditional methos such as collaborative filtering.

While the proposed approach can be used with datasets without contextual data, it will lose most of its benefits compared to other approaches.

The predictions about the user's preferences given a context are done directly over the Bayesian network. The query delivers the best options while still considering the selected values for the request. We can include as many constraints as we want or as data we have available at the moment.

Another advantage of our approach is that we can evaluate queries that do not necessarily involve the users' scores. We can learn more from their interactions, so that we can infer and even predict what they will do or buy giving the current context (or an hyphothetical one).

Finally, another advantage of having a visual method such as a Bayesian network as the base of the model is that it can be presented to the final user to let him do some adjustments if he does not feel satisfied by the current state of his model.

We also acknowledge that recommender systems are only helpers in the user's decision processs. Some users may want to have a more proactive role in the process, such as Van Setten *et al.* [14] expose, some users like to think and decide for themselves. For this reason we intended to give the user more control with feedback process. This way, the user is certain that his own beliefs are really considered and that the system covers his needs.

The work presented in this paper can be further improved and explored. We consider this as just a first step into better understanding the relationships between the available data inside a recommender system. Acknowledging this, we would like to present the following points as possible steps into further developing this approach:

– Complete the vacation dataset - While we have some data collected from our suveys. Our dataset can be improved by getting more users and by including follow-up questions in order to test the evolution of our model.
– Adding more than one domain - We focused on a single domain for this paper. This is a common approach on recommender system's research since some domains might not share the same preferences. Finding the preferences and variables that can be shared might prove useful to build the models used for the predictions.
– To further test our personalization approach, a better tracking of specific users along an expanded time window would allow us to improve the system's models and feedback measures. This could also lead to conclude good parameters that indicate the frecuency in which the network should be relearned. This could also mean that the base case could be altered depending on the system's evolution.

The future work is not necessarily limited to the subjects we mentioned. We consider them as the natural progression from our work and some viable points to improve our current results.

The main concern in this type of approaches is the impending need for contextual datasets that can capture the detailed information of the interactions between users and items. Since we are representing the interaction, we can also evaluate the satisfaction that it meant to the user, not limiting our approach only to the items in the dataset.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17(6), 734–749 (2005)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender systems handbook, pp. 217–253. Springer (2011)
3. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. Personal Ubiquitous Comput. 16(5), 507–526 (Jun 2012), http://dx.doi.org/10.1007/s00779-011-0417-x
4. Ben-Gal, I.: Bayesian networks. Encyclopedia of statistics in quality and reliability (2007)
5. Fischer, G.: User modeling in human–computer interaction. User modeling and user-adapted interaction 11(1-2), 65–86 (2001)
6. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
7. Košir, A., Odic, A., Kunaver, M., Tkalcic, M., Tasic, J.F.: Database for contextual personalization. Elektrotehniški vestnik 78(5), 270–274 (2011)

8. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems 56, 156–166 (2014)
9. Nilashi, M., bin Ibrahim, O., Ithnin, N.: Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system. Knowledge-Based Systems 60, 82–101 (2014)
10. Ono, C., Kurokawa, M., Motomura, Y., Asoh, H.: A context-aware movie preference model using a bayesian network for recommendation and promotion. In: User Modeling 2007, pp. 247–257. Springer (2007)
11. Panniello, U., Tuzhilin, A., Gorgoglione, M.: Comparing context-aware recommender systems in terms of accuracy and diversity: which contextual modeling, pre-filtering and post-filtering methods perform the best (2012)
12. Ramezani, M., Moradi, P., Akhlaghian, F.: A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains. Physica A: Statistical Mechanics and its Applications 408, 72–84 (2014)
13. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The adaptive web, pp. 291–324. Springer (2007)
14. Van Setten, M., Pokraev, S., Koolwaaij, J.: Context-aware recommendations in the mobile tourist application compass. In: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. pp. 235–244. Springer (2004)