A Recommender for Active Preference Estimate

Andrea Malossini and Paolo Avesani

Fondazione Bruno Kessler

Abstract. Recommender systems usually assume that the feedback from users is independent from the way a suggestion is presented. Usually the learning effort to estimate the user model is aimed at recognizing a subset of items according to the user preferences. We argue that acceptance or rejection is not independent from the presentation of the recommended items. Shaping a recommended item means to decide what kind of presentation can be more effective in order to increase the acceptance rate of the item suggested. In this work we address the challenge of designing an active policy that recommends a presentation of an item, i.e. which features use to describe the item, that will produce a more informative feedback from the user. The main contribution of this paper is a first principled design of an algorithm to actively estimate the optimal presentation of an item. An experimental evaluation provides the empirical evidence of the potential benefits.

1 Introduction

A basic step in recommender systems is the estimate of user model. Learning the user preferences is the premise to deliver personalized suggestions and to design adaptive information filters. The challenge is how to support the user to make a choice when the space of items is so huge that an exhaustive search is not sustainable [1]. The usual strategy is to take advantage from the feedbacks of the users to learn their preferences while are browsing a sample of items [2].

The elicitation of preferences takes place as an iterative process where the users get explicit what kind of items like most or dislike. Conversational recommender systems [3,4] were introduced to manage this process as a dialogue to shorten the estimate of the user preferences and consequently the recommendation of a compliant item. Step by step the goal is to converge on a subsample of items that includes the user's choice.

Usually this kind of approaches share the simplifying assumption that the choice of the user is independent from the representation of the items. Let consider a scenario where the purpose is to choose a movie. The choice of the user may be affected whether the movies is described by features like title, years and genre, rather than title, actor and rating. What might happens is that the user would accept our recommendation in the first case while would reject the other even though the item, e.g. the movie, is the same [5].

The objective of this work is to address the problem of learning what features affect the user acceptance of a given item. The working hypothesis is that the

L. Telesca et al. (Eds.): DigiBiz 2009, LNICST 21, pp. 19–26, 2009.

process of decision making is biased by the information encoded in a subset of features. If such information is available the user may perform the assessment of an item and to decide whether s/he likes or dislikes. In the other case the user can't evaluate an item and the feedback may be perceived equivalent to a dislike option as above. We prefer to refer to this feedback as a third option called unknow. Therefore we may distinguish among three kinds of feedback: like, dislike and unknown.

It is worthwhile to remark that the two answers, dislike and unknown, are not equivalent. If we received an unknown feedback, it doesn't necessarily mean that the recommended item isn't suitable to the user. May be the same item, recommended with a different set of informative features, might be accepted at the next stage. If we received a dislike feedback the recommended item is equally rejected by the user but we may learn what kind of features affect the decision.

The ultimate goal is to design a recommender system that helps not only to select compliant items (items that users might like) but also to shape effective items (items that users might perceive as appropriate) . Our approach is oriented to active sampling. The idea is to recommend those features that will produce more informative feedbacks from the user [6]. This work differs from other approaches based on active policy [2]. Usually the task of estimate the user model aims to learn the threshold values that discriminate between acceptance and rejection. We focus the estimate task to learn only what are those features that are relevant to take a decision. It might be conceived as a shallow user model that requires less effort, e.g. short interactions.

The motivation of this work derives from a different perspective on e-commerce where the selection of an item is not driven only by the preferences of a single user but more often is matter of negotiation [7] between two parties: producer/consumer, buyer/seller, tenderor/tenderee. While the buyer is interested to find something s/he likes among huge range of items, the seller aims to recommend items that provide larger revenues. In this scenario, much more competitive rather than collaborative, the general habit is to reduce the amount of elicited information since it might represent a competitive advantage for the counter part. In the following we will refer to a framework of bilateral negotiation [8].

The main contribution of this paper is a first principled design of an algorithm to actively estimate user preferences. An experimental evaluation provides the empirical evidence of the potential benefits.

2 Formal Definition

In this Section, we formally define the behavioural model of users playing bilateral negotiations.

We model a user using a workflow which encodes the sequence of actions to be executed; in particular, we define three functions, the *pref* function, the *shape* function and the *like* function, which encode the user's preferences and decision making procedures. While *pref* and *like* functions are supposed to be stationary and defined at the beginning of the negotiation, *shape* is conceived

as an adaptive function whose definition evolves through the negotiation by a process of learning. The way to shape an offer change according to the feedback collected from the counter part. After each stage of interaction the hypothesis on issues relevance is revised with respect to the evidence collected so far by the rejected offers.

The *pref* function, given the offer number in the negotiation, returns an item from the ordered preference list of the user. The user is in charge of selecting an item for the next offer. We indeed suppose that each user has a fixed list of items ordered in accordance with his utility. Formally, let U be a set of users, T be the item set, and k be the maximum number of offers (for each user involved in such negotiation) before the negotiation expires, we define, for each user $u \in U$, the *pref* function as $\text{pref}_u : I_k \longrightarrow T$ where $I_k = \{1, 2, 3, \ldots, k\}$.

The *shape* function, given an item, returns an offer to send to the counterpart. Practically, it selects which features of an item to include in the offer. In the next Section we give a formal description of the item set and the offer set. Formally, let O be the offer set, for each user $u \in U$, we define the *shape* function as $\text{shape}_u : T \longrightarrow O$

The *like* function, given an offer, return a label (answer) which is related to the evaluatioLet $A = \{0, \perp, 1\}$ be the set of all possible answers to an offer, for each user $u \in U$, we define the *like* function as $\text{like}_u : O \longrightarrow A$ The like function is invoked every time we need to evaluate a new opponent's offer; we emphasize that both the values of the issues and the presence or absence of some issues play a role in the definition of this function.

After the definition of the *pref*, *shape*, and *like* functions, we can focus on the user's workflow. Firstly, A calls its *pref* function which returns the item to propose. Then, A calls its *shape* function which provides a subset of issues to include in the offer. Finally, A sends the offer to B. Once the offer is received, B calls its *like* function to decide if accept the offer. If B rejects it, B will call its *pref* and *shape* functions to generate a counter offer, that will then send to A. At this point A will evaluate it using its *like* function. The negotiation go on until one of the two users accepts an offer or the time expires.

3 Computational Model

In this Section we develop an active learning algorithm that will explore the space of agreements by selecting a set of issues that are "relevant" for the opponent or yield an high "benefit" when assessing the opponent's preferences. This search will be carried out inside the present negotiation and will react according to the counter-offers proposed by the opponent.

As a first step, let us define the item set T; we suppose that each item $t_i \in T$ can be described as a set of pairs of the type

$$t_i = \{(\text{attr}_1, a_{i1}), (\text{attr}_2, a_{i2}), \ldots (\text{attr}_n, a_{in})\},$$

where n, the number of features is fixed, and we denote with a_{ij} the value assumed by the j-th feature of the i-th item. The shape function defined in

Eq. 2 takes an item t_i and shapes the offer by selecting a subset of informative features. Formally this can be accomplished by setting some of the values a_{ij} to the special value `NA` (not available, i.e. missing value). Hence, we obtain the offer $o_i \in O$ as

$$o_i = \{(\text{issue}_1, v_{i1}), (\text{issue}_2, v_{i2}), \dots (\text{issue}_n, v_{in})\},$$

where

$$v_{ij} = \begin{cases} a_{ij} & \text{if the } j\text{-th feature is an issue,} \\ \texttt{NA} & \text{otherwise.} \end{cases}$$

Essentially, in order to define the shape function we need to attach a score to each item's feature. This score can be obtained by using a variant of the active learning algorithm for feature relevance [9]. In order to adapt such algorithm we need to introduce some definitions in the following.

As stated in the previous Section, we assume that the negotiation will end after a certain number of offer exchanging k. We can, then, define the *user's content matrix* at step m (i.e. after m offers exchanged) as

$$V_m = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{pmatrix}, \tag{1}$$

where $m \leq k$. Analogously, we can construct a similar matrix, the *opponent's content matrix* at step m, which contains the issues' values of the offers received by the opponent. We denote such matrix as W_m, where $m \leq k$. Note that these matrices contain also missing values. As long as we receive and make new offers, the matrices will increase row by row.

V_m will encode our strategy for making offers, whereas W_m will contain features' values which are of interest for the opponent.[1] During the negotiation process, the user is able to collect a lot of information related to the opponent's preferences analysing the content of the received offers. These data can be used to estimate probabilistic models by considering each issue as a discrete random variable with a multinomial distribution (for continuous issue we can think of transforming the original continuous distribution to a discrete distribution, essentially because we expect to have very few values for each issues, hence a continuous distribution will bring no information).

Let us define then the *issue matrix* F_m and the response vector A_m at step m as

[1] In fact, we argue that the opponent, when composing an offer, will use item of interest with potentially interesting features' values.

$$F_m = \begin{pmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{m1} & f_{m2} & \dots & f_{mn} \end{pmatrix} \tag{2}$$

$$A_m = (a_1, a_2, \dots, a_m)^T \tag{3}$$

where

$$f_{ij} = \begin{cases} \Pr(\text{issue}_j = v_{ij} | W_{i-1}) & \text{if } v_{ij} \neq \texttt{NA} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

represents the probability that the issue issue_j be equal to the correspondent value in the i-th offer we made given the content matrix W_{i-1} and a_i the i-th response to our i-th offer (which can assume values in $\{0, \perp\}$). Let g_j be a *relevance* function [2] of the issue issue_j computed using the matrix F_m, then, we can compute the *benefit* of issue issue_j in a new offer as[9]

$$\begin{aligned} &B(\text{issue}_j) = \\ &\sum_{a \in \{0,\perp\}} \Big[g_j(F_m, \text{issue}_j = v_{(m+1)j} | a_{m+1} = a) - \\ &\qquad g_j(F_m | a_{m+1} = a) \Big]^2 \cdot \Pr(a_{m+1} = a | F_m), \end{aligned} \tag{5}$$

where $v_{(m+1)j}$ is the j-th issue's value of the new offer we are willing to make. This strategy differs from a similar work [10] where the heuristic is to maximize the expected utility, while here the criteria is to maximize the average increment on expected utility.

The relevance function and the benefit function are used in the shape function to attach a score to each feature of an item. In particular, we could use this information to:

Exploration. Select the features which possess a high benefits B, which means that including those features in the offer would allow for a smart assessing of the opponent's preferences;

Exploitation. Select the features which possess a high relevance g, which means that using those features in the offer would allow the opponent to take a decision using his relevant features, hence minimizing the chance of the offer to be rejected.

In the following we restrict our attention to exploration strategy. We design a recommender engine that exploits the computation of B to support an *Active Offer Shaping* (AOS). AOS recommends features that are ranked at the top by B function.

[2] As a standard measure of relevance we use the mutual information, but other measures could be used, depending on the context.

4 Empirical Evaluation

As reference example to sketch the experimental framework we refer to a scenario of cooperative rental of a DVD movie. Two users engage an iterative process to select what kind of DVD movie to watch together.

To generate a data set suitable for the simulation of such a kind of bilateral negotiation as described in the previous sections, we start from the *Movielens*.

According to the simulation model adopted in Section 2, we need to model each user by a *pref* function, defined by the rating of *Movielens* users, and the *like* function, defined as the category of movies that discriminate between high and low ratings . These two functions completely describes the "virtual" users that will engage a bilateral negotiation to achieve an agreement on a movie that both of them like. The *shape* function is alternatively implemented as a random choice or as adaptive strategy according to the definition of Section 3.

We designed a simulation using a sample of 1500 user pairs to run negotiations. At each negotiation we have one user which always plays the random strategy, whereas the other one alternatively uses the random strategy or the AOS recommender.

From the simulations we obtained that the rate of success for random strategy is 0.81 (0.01), whereas for the AOS strategy is 0.93 (0.01). There is an increment of 12% (180 negotiations) in the number of successful negotiations when using the AOS recommender. It is interesting to analyze in detail the disaggregated results, as reported in Table 1.

The first row reports the mean number of counts (and the corresponding standard deviation) when adopting the AOS recommender the negotiation fails, whereas, using a random strategy the negotiation succeeds. The second row depicts exactly the opposite, when using the AOS recommender is beneficial and random strategy does not. The difference between these two counts is about 170 which explains the increase in the rate of success when using the AOS strategy. The fourth row depicts the situation where both the strategies produce a successful negotiation, but using AOS a faster agreement is reached. The fifth line is the opposite. Here we can notice that, typically, there are about 700 negotiations where the random strategy produces faster successful negotiations.

Notice that the remarkable performance of the random strategy is due to the fact that the problem, as is, is easy to solve; in fact, among the 18 features

Table 1. Mean number and standard deviation of counts for the different situation that may arise during the negotiations

AOS	RND	%	μ	σ
Unsuccessful	Successful	0.05	85.9	8.1
Successful	Unsuccessful	0.17	257.2	13.1
Unsuccessful	Unsuccessful	0.01	20.4	4.9
Faster	Slower	0.26	400.0	13.6
Slower	Faster	0.46	696.0	20.7
Same speed		0.02	41.3	6.2

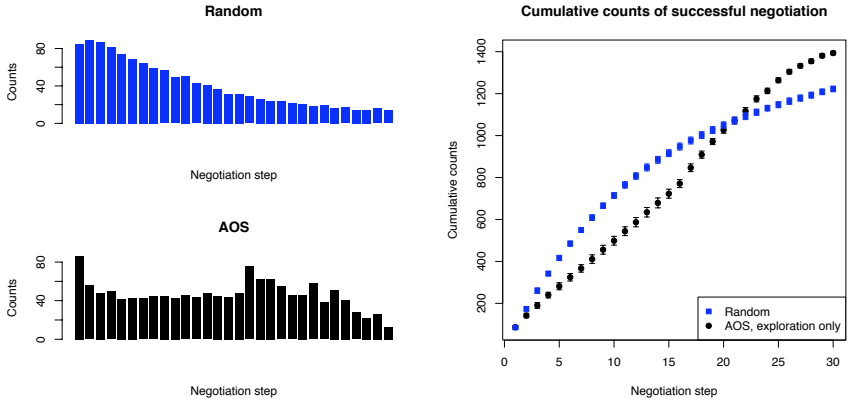

Fig. 1. On the left hand side the differences between negotiations with the AOS recommender and random strategy; on th eright hand side the distribution of the average negotiation lenghts with the AOS recommender and random strategy

available, you have only to find the right feature (probability 1/18) and its value, which may assume only two values, 0 or 1 (with probability which depends on the distribution of the genre). The problem can even be solved more efficiently by probing the same feature once one obtains as answer `DISLIKE`. However when the scenario becomes more complicated (for example when the user model implies a combination of right features/values and the cardinality of the features is higher than 2, or possibly continuous) we expect that the random strategy would perform very poorly, whereas, using the AOS recommender and its exploration/exploitation capabilities we expect to obtain far better results. The extremely simple scenario used, nevertheless, is useful for analyzing the behavior of the algorithm used for active learning. Another effect of this simple scenario is that the number of answers of type `DISLIKE` one obtains is very low (once discovered the right feature, being it binary, there is an high probability of ending the negotiation), and the computation of the relevance is deeply affected by it.

Another interesting results is the comparison of the speed in obtaining a successful negotiation.

In Figure 1 we show, in the left part, the histograms of the mean counts of successfull negotiations with the random strategy and with the AOS recommender. As you can see, the histogram of random strategy is a deacreasing function of the offer number, as expected, whereas, the AOS histogram presents a peak around the 20th step. The first peak in the AOS histogram is due to the fact that the first offer is shaped at random, not having any useful information to use in the active feature sampling procedure.

In the right part of Figure 1, we plot the cumulative counts of successful negotiations (and error bars associated), where we can clearly see that in the first part of the negotiation, the random strategy is better, but after the 20th offer the AOS recommender provides additional successes. This indicates that at

certain point in the negotiation, the "exploration power" of the random strategy stops to be fruitful, whereas the AOS recommender keeps being effective.

5 Conclusions

In this paper we proposed a computational model to support active learning of issues' relevance. The active sampling approach is particularly indicated because it allows to exploit useful information from ongoing negotiations and to intelligently explore the space of agreements. Furthermore, shortening the estimate of opponent's features relevance allows to speed up the achievement of an agreement in a bilateral negotiation.

Acknowledgments

This work was partially supported by the project 034744 EU-INFSO-IST ONE.

References

1. Burke, R.D., Hammond, K.J., Young, B.C.: The findme approach to assisted browsing. IEEE Expert 12(4), 32–40 (1997)
2. Viappiani, P., Pu, P., Faltings, B.: Conversational recommenders with adaptive suggestions. In: RecSys 2007: Proceedings of the 2007 ACM conference on Recommender systems, pp. 89–96. ACM, New York (2007)
3. Carenini, G., Smith, J., Poole, D.: Towards more conversational and collaborative recommender systems. In: Proc. of IUI 2003, pp. 12–18. ACM, New York (2003)
4. Cynthia, A., Thompson, M.H., Göker, P.L.: A personalized system for conversational recommender. JAIR 21, 393–428 (2004)
5. Winterboer, A., Moore, J.D.: Evaluating information presentation strategies for spoken recommendations. In: RecSys 2007: Proceedings of the 2007 ACM conference on Recommender systems, pp. 157–160. ACM, New York (2007)
6. MacKay, D.J.C.: Information-based objective functions for active data selection. Neural Computation 4(4), 590–604 (1992)
7. Kersten, G., Lo, G.: Aspire: an integrated negotiation support system and software agents for e-business negotiation. International Journal of Internet and Enterprise Management 1(3), 293–315 (2003)
8. Telesca, et al.: Open negotiation environment: An open source self-learning decentralised negotiation framework for digital ecosystems. In: Digital EcoSystems and Technologies Conference (DEST), pp. 186–191 (2007)
9. Olivetti, E., Veeramachaneni, S., Avesani, P.: 5. In: Computational Methods of Feature Selection. Chapman & Hall/CRC, Boca Raton (2008)
10. Chajewska, U., Koller, D., Parr, R.: Making rational decisions using adaptive utility elicitation. In: AAAI/IAAI, pp. 363–369 (2000)