Learning Users' Interests in a Market-Based Recommender System *

Yan Zheng Wei, Luc Moreau and Nicholas R. Jennings

Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton, UK.
{yzw01r,L.Moreau,nrj}@ecs.soton.ac.uk

Abstract. Recommender systems are widely used to cope with the problem of information overload and, consequently, many recommendation methods have been developed. However, no one technique is best for all users in all situations. To combat this, we have previously developed a market-based recommender system that allows multiple agents (each representing a different recommendation method or system) to compete with one another to present their best recommendations to the user. Our marketplace thus coordinates multiple recommender agents and ensures only the best recommendations are presented. To do this effectively, however, each agent needs to learn the users' interests and adapt its recommending behaviour accordingly. To this end, in this paper, we develop a reinforcement learning and Boltzmann exploration strategy that the recommender agents can use for these tasks. We then demonstrate that this strategy helps the agents to effectively obtain information about the users' interests which, in turn, speeds up the market convergence and enables the system to rapidly highlight the best recommendations.

1 Introduction

Recommender systems have been widely advocated to help make choices among recommendations from all kinds of sources [1]. Most of the existing recommender systems are primarily based on two main kinds of methods: content-based and collaborative. However, both kinds have their weaknesses: the former cannot easily recommend non-machine parsable items, whereas the later fail to accurately predict a user's preferences when there are an insufficient number of peers. Given this, it has been argued that there is no universally best method for all users in all situations [2].

In previous work, we have shown that an information marketplace can function effectively as an overarching coordinator for a multi-agent recommender system [3, 4]. In our system, the various recommendation methods, represented as agents, compete to advertise their recommendations to the user. Through this competition, only the best items are presented to the user. Essentially, our system uses a particular type of auction and a corresponding reward regime to incentivise the agents to bid in a manner that is maximally consistent with the user's preferences. Thus, good recommendations

* This research is funded by QinetiQ and the EPSRC Magnitude project (reference GR/N35816).

(as judged by the user) are encouraged by receiving rewards, whereas poor ones are deterred by paying to advertise but by receiving no rewards.

While our system works effectively most of the time, an open problem from the viewpoint of the individual recommenders remains: *given a set of recommendations with different rating levels, in what order should an agent advertise them so that it can learn the user's interests as quickly as possible, while still maximizing its revenue?* To combat this, we have developed a reinforcement learning strategy, that enables an agent to relate the user's feedback about recommendations to its internal belief about their qualities and then to put forward those that are maximally consistent with this.

Against this background, this paper advances the state of the art in the following ways. First, a novel reinforcement learning strategy is developed to enable the agents to effectively and quickly learn the user's interests while still making good recommendations. Second, from an individual agent's point of view, we show the strategy enables it to maximize its revenue. Third, we show that when all agents adopt this strategy, the market rapidly converges and makes good recommendations quickly and frequently.

2 A Market-Based Multi-Agent Recommender System

Different recommendation methods use different metrics and different algorithms to evaluate the items they may recommend. Thus, the internal rating of the quality of a recommendation can vary dramatically from one method to another. Here, we term this internal evaluation the method's *internal quality* (INQ). However, a high INQ recommendation from one method does not necessarily mean the recommendation is any more likely to better satisfy a user than a low INQ item suggested by another. Ultimately, whether a recommendation satisfies a user can only be decided by that user. Therefore, we term the user's evaluation of a recommendation the *user's perceived quality* (UPQ).

With these concepts in place, we now briefly outline our market-based recommender (as per Fig. 1). Each time when the marketplace calls for a number (S) of recommendations, each agent submits S recommendations and bids a price for each of them. Consequently, the marketplace ranks all items in decreasing order of price and displays the top S items to the user and, meanwhile, each corresponding agent pays for each displayed item an amount of credit equal to its bid for that item. The user then visits a number of the displayed items and gives each a rating (i.e. UPQ) based on how it satisfies him. Finally, the market rewards the agents with positive UPQ recommendations an amount of credit that is proportional to their UPQs. Thus, the system completes one round of operation and proceeds with another, following the same basic procedure.

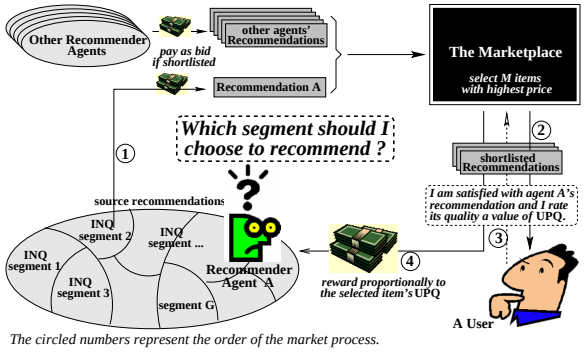

Fig. 1. An Agent's Learning Problem

We have shown that, to make effective recommendations, an agent needs to classify its recommendations into a number (G) of INQ levels (segments) and be able to correlate these segments to the UPQs [4]. Indeed, an agent that has sufficient experience of the user's feedback can learn the user's interests by correlating its recommendations (and their corresponding INQ segments) to the rewards (that reflect their UPQs). This enables a self-interested agent to consciously make recommendations from those INQ segments that correspond to high UPQs so that it can best satisfy the user and, thus, gain maximal revenue. To effectively compute the agents' revenue, we define an agent's *immediate reward* (made from a specific recommendation in one auction round) as the reward it received minus the price it has paid for the advertisement. With this, the agent needs to learn how much immediate rewards, on average, it can expect for items in each INQ segment. We term this average immediate reward for each INQ segment an agent's *expected revenue*. Thus, an agent can maximize its revenue by frequently bidding those recommendations from the segments with high expected revenue.

However, when an agent starts bidding, it has no information about the expected revenue for each segment. Therefore, the agent needs to interact in the marketplace by taking actions over its G segments to learn this information. In this context, the agent's learning behaviour is on a "trial-and-error" basis, in which good recommendations gain rewards whereas bad ones attract a loss. This kind of trial-and-error learning behaviour is exactly what happens in Reinforcement Learning [5]. Thus, to be more concrete, an agent needs an algorithm to learn the expected revenue over each segment as quickly as possible and still maximizing revenue.

3 The Learning Strategy

This section aims to address the problem of producing the expected revenue profile over an agent's G segments. In detail, an agent needs to execute a set of *actions* (bidding on its G segments), $(a_1, a_2, \cdots, a_G)$, to learn the expected revenue of each segment ($R(a_i)$, $i \in [1..G]$). Specifically, an action a_i that results in its recommendation being displayed to the user must pay some amount of credit. Then, it may or may not receive an amount of reward. We record the t^{th} immediate reward that a_i has received as $r_{i,t}$ ($t = 1, 2, \cdots$). From a statistical perspective, the expected revenue can be obtained from the mean value of the series of discrete immediate reward values, i.e. $r_{i,t}$. In this context, the Q-learning technique provides a well established way of estimating the optimality [6]. In particular, we use a standard Q-learning algorithm to estimate $R(a_i)$ by learning the mean value of the immediate rewards:

$$\hat{Q}_i := (1 - \frac{1}{t_0 + t}) \cdot \hat{Q}_i + \frac{1}{t_0 + t} \cdot r_{i,t} \ , \tag{1}$$

where $\hat{Q}_i$ is the current estimate of $R(a_i)$ (before we start learning, all $\hat{Q}_i$s are initialized with a positive value) and $\frac{1}{t_0+t}$ is the learning rate that controls how much weight is given to the immediate reward as opposed to the old estimate (t_0 is positive and finite). As t increases, $\hat{Q}_i$ builds up an average of all experiences, and converge to $R(a_i)$ [5].

To assist the learning algorithm, an exploration strategy is needed to decide which specific action to perform at each specific t. In fact, it is hard to find the absolutely

best strategy for most complex problems. In reinforcement learning practice, therefore, *specific* approaches tend to be developed for specific contexts. They solve the problems in question in a reasonable and computationally tractable manner, although they are often not the absolutely optimal choice [6]. In our context, knowing how much can be expected through each action, an agent can use a probabilistic approach to select actions based on the law of effect [7]: *choices that have led to good outcomes in the past are more likely to be repeated in the future*. To this end, a *Boltzmann exploration* strategy fits our context well; it ensures the agent exploits higher $\hat{Q}$ value actions with higher probability, whereas it explores lower $\hat{Q}$ value actions with lower probability [5]. The probability of taking action a_i is formally defined as:

$$P_{a_i} = \frac{\mathbf{e}^{\hat{Q}_i/T}}{\sum_{j=1}^{G} \mathbf{e}^{\hat{Q}_j/T}} \qquad (T > 0), \tag{2}$$

where T is a system variable that controls the priority of action selection. In practice, as the agent's experience increases and all $\hat{Q}_i$s tend to converge, the agent's knowledge approaches optimality. Thus, T can be decreased such that the agent chooses fewer actions with small $\hat{Q}_i$ values (meaning trying not to lose credits) and chooses more actions with large $\hat{Q}_i$ values (meaning trying to gain credits).

4 Evaluation

This section reports on the experiments to evaluate the learning strategy. We previously showed that our marketplace is capable of effectively incentivising good methods to relate their INQs to the UPQs and this capability is independent of the specific form of the correlation between the two qualities [4]. Here, we simply assume that there are four good recommendation methods in our system and they have a linear correlation between their INQs and the UPQs. To correlate these two qualities, all agents divide their INQ range into $G = 20$ equal segments. Q_{init} is set to 250, $T = 200$ and $t_0 = 1$ for all agents. The market each time calls for $S = 10$ recommendations. With these settings, we are going to evaluate the system according to the properties that we want the learning strategy to exhibit:

• **Q-Learning Convergence to Optimality:** $\hat{Q}$ values' convergences are important because, otherwise, an agent will have no incentive to bid. To evaluate this, we arranged 300 auctions. We find that an agent's $\hat{Q}$ values always converge (as per Fig. 2) such that high INQ segments' $\hat{Q}$s converge to high values and low INQ segments' $\hat{Q}$s converge to low values (because of the linear correlation between INQs and UPQs). This is because the recommendations from a segment corresponding to higher UPQs receive more immediate reward than those corresponding to lower UPQs.

• **Revenue Maximization:** All recommendation methods are self-interested agents that aim to maximise their revenue by advertising good recommendations and by receiving rewards. To demonstrate this property, we organized two set of experiments. One with four learning agents and the other with four non-learning agents (i.e. bidding randomly), with all other settings remaining the same. We find that the learning agents consciously raise good recommendations more frequently than non-learning ones. Thus, the former

can make, on average, significantly greater amounts (about 43%) of credit than the latter (see Fig. 3(a) and (b)).

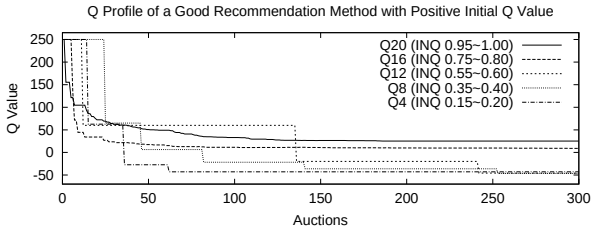

Fig. 2. Q-Learning Convergence

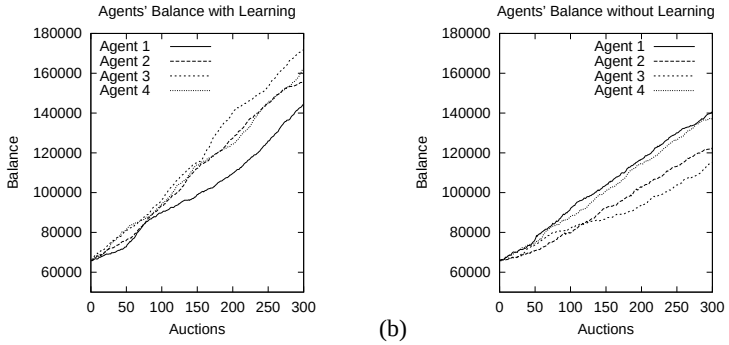

Fig. 3. Recommenders' Balance

• **Quick Market Convergence:** Market convergence (prices of recommendations of different UPQ levels converge to different price levels) enables the agents to know what prices to bid for recommendations with certain UPQs to gain maximal revenue [3, 4]. Thus, quick market convergence let agents reach this state quickly. To evaluate this, we contrast the learning market with the non-learning one using the same settings when assessing revenue maximization. We find that the former always converges quicker than the latter. Specifically, the former (Fig. 4(a)) converges after about 40 auctions, whereas the latter (Fig. 4(b)) does after about 120 auctions. Indeed, as the learning agents' $\hat{Q}$ profiles converge, more high quality recommendations are consistently suggested (since high $\hat{Q}$ values induce high probability to bid these items because of equation (2)) and this accelerates effective price iterations to chase the market convergence.

•**Best Recommendation's Identification:** To evaluate the learning strategy's ability to identify the best recommendation (with the top UPQ) quickly and bid it consistently, we use the same set of experiments that were used to assess the market convergence. We then trace the top UPQ item highlighted by a randomly selected learning agent and the corresponding one in the non-learning market in Fig. 4 (a) and (b) respectively (see the circle points). Fig. 4(a) shows that this item's bidding price keeps increasing till it

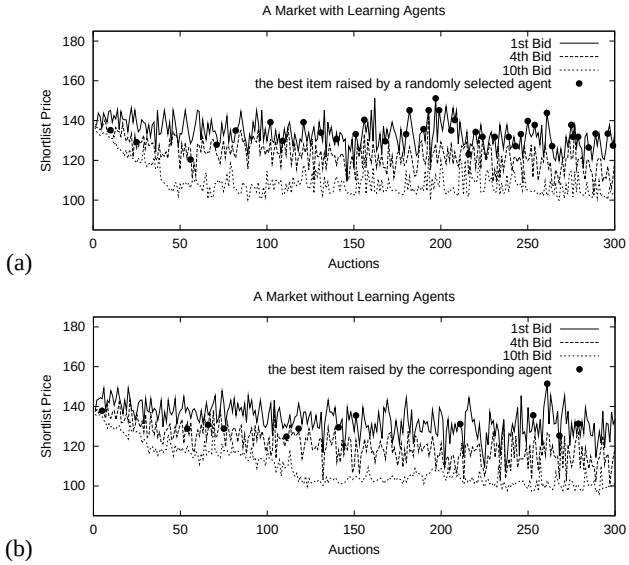

Fig. 4. Market Convergence

converges to the first bid price of the displayed items. This means that as long as the agent chooses this particular item to bid in an auction (after the market converges), it is always displayed in the top position. However, in contrast, this phenomenon proceeds slowly in a non-learning market (see Fig. 4(b)). Additionally, a learning agent raises the best recommendation more frequently (39 times, see Fig. 4(a)), about three times as much, compared to a non-learning one (13 times, see Fig. 4(b)).

5 Discussion and Future Work

The learning strategy presented in this paper significantly improves our previously reported market-based recommender system [3, 4] by speeding up the market's ability to make good recommendations. In terms of learning users' interests, most existing recommender systems use techniques that are based on two kinds of features of recommendations: objective features (such as textual content in content-based recommenders) and subjective features (such as user ratings in collaborative recommenders). However, many researchers have shown that learning techniques based on either objective or subjective features of recommendations cannot successfully make high quality recommendations to users in all situations [8, 9, 2]. The fundamental reason for this is that these existing learning algorithms are built *inside* the recommenders and, thus, the recommendation features that they employ to predict the user's preferences are fixed and cannot be changed. Therefore, if a learning algorithm is computing its recommendations based on the features that are relevant to a user's context, the recommender is able to successfully predict the user's preferences (e.g. a customer wants to buy a "blue" cup online and the recommendation method's learning algorithm is just measuring the

"colour" but not the "size" or the "price" of cups). Otherwise, if the user's context related features do not overlap any of those that the learning algorithm is computing on, the recommender will fail (e.g. the user considers "colour" and the learning algorithm measures "size"). To overcome this problem and successfully align the features that a learning technique measures with a user's context in all possible situations, we seek to integrate multiple recommendation methods (each with a different learning algorithm) into a single system and use an overarching marketplace to coordinate them. In so doing, our market-based system's learning technique encapsulates more learners and each learner computes its recommendations based on some specific features. Thus, our approach has a larger probability of relating its features to the user's context and so, correspondingly, has a larger opportunity to offer high quality recommendations.

To conclude, to be effective in a multi-agent recommender system (such as our market-based system), an individual agent needs to adapt its behaviour to reflect the user's interests. However, in general, the agent initially has no knowledge about these preferences and it needs to obtain such information. But, in so doing, it needs to ensure that it continues to maximize its revenue. To this end, we have developed a reinforcement learning strategy that achieves this balance. Essentially, our approach enables an agent to correlate its INQs to the UPQs and then direct the right INQ recommendations to the right users. Specifically, through empirical evaluation, we have shown that our strategy works effectively at this task. In particular, a good recommendation method equipped with our learning strategy is capable of rapidly producing a profile of the user's interests and maximizing its revenue. Moreover, a market in which all agents employ our learning strategy converges rapidly and identifies the best recommendations quickly and consistently. For the future, however, we need to carry out more extensive field trials with real users to determine whether the theoretical properties of the strategy do actually hold in practice.

References

1. Resnick, P., Varian, H.R.: Recommender Systems. Commun. of the ACM **40** (1997) 56–58
2. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Information Systems **22** (2004) 5–53
3. Wei, Y.Z., Moreau, L., Jennings, N.R.: Recommender systems: A market-based design. In: Proc. 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03), Melbourne (2003) 600–607
4. Wei, Y.Z., Moreau, L., Jennings, N.R.: Market-based recommendations: Design, simulation and evaluation. In: Proc. 5th International Workshop on Agent-Oriented Information Systems (AOIS-2003), Melbourne (2003) 22–29
5. Mitchell, T.: Machine Learning. McGraw Hill (1997)
6. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research **4** (1996) 237–285
7. Thorndike, E.L.: Animal intelligence: An experimental study of the associative processes in animals. Psychological Monographs **2** (1898)
8. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: Proc. Conf. on Human factors in computing systems. (1995) 210–217
9. Montaner, M., Lopez, B., Dela, J.L.: A taxonomy of recommender agents on the internet. Artificial Intelligence Review **19** (2003) 285–330