UNIVERSITY OF SOUTHAMPTON

A Market-Based Approach to Recommender Systems

by

Yan Zheng WEI

A thesis submitted for the degree of

Doctor of Philosophy

in the

Intelligence Agents Multimedia Group

School of Electronics and Computer Science

March 2005

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

INTELLIGENCE AGENTS MULTIMEDIA GROUP
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Yan Zheng WEI

Recommender systems have been widely advocated as a way of coping with the problem of information overload for knowledge workers. Given this, multiple recommendation methods have been developed. However, it has been shown that no one technique is best for all users in all situations. Thus, we believe that effective recommender systems should incorporate a wide variety of such techniques and that some form of overarching framework should be put in place to coordinate the various recommendations so that only the best of them (from whatever source) are presented to the user. To this end, we show that a marketplace, in which the various recommendation methods compete to offer their recommendations to the user, can be used in this role. Specifically, our research is concerned with the principled design of such a marketplace (including the auction protocol, the reward mechanism and the bidding strategies of the individual recommender agents) and its evaluation in terms of how it can effectively coordinate multiple methods. In addition to the market mechanisms, a reinforcement learning strategy is developed to assist the individual recommender agents' bidding behaviour so as to learn the users' interests and still maximize their revenue. Finally, we evaluate our approach with a real market-based recommender system that is composed of a number of typical recommendation methods and that is evaluated with real users. The evaluation results show that our approach is indeed an effective means of coordinating multiple different recommendation methods in one single system and is an effective way of dealing with the problem of information overload.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend my heartfelt and sincere gratitude to my supervisors, Prof. Luc Moreau and Prof. Nick Jennings, for their invaluable advice, guidance, encouragement and patience through the completion of this project. I would also like to thank Prof. Wendy Hall and Prof. David De Roure for their inputs and suggestions on designing the user evaluation of our recommender system. Thanks should also be given to Dr. Steve Hitchcock for his help on the early trials of our recommender system and his valuable feedback on improving the system performance. Additionally, thanks should also be given to Miss Jingtao Yang for her great help on developing the recommender system's web services and many of the colleagues, such as Partha S. Dutta, Nishan C. Karunatillake, Nadim Haque and many other researchers, for their help on evaluating our system.

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

Chapter 1

Introduction

The World-Wide Web (the Web) [Berners-Lee et al., 1992] presents us with a vast array of information. Also, regardless of the metric used (i.e., growth in the number of networks, hosts, users, or traffic), the Internet is growing at least 10 percent per month and the content of the Web grows by an estimated 170,000 pages daily [Turban et al., 2000]p495. When taken together, these factors make it very hard to know what documents are out there and finding the right one is even more problematic. This phenomenon is known as *information overload*: here defined as the receipt of undesirable or non-relevant information that results in an economic loss for the recipient [Losee, 1989].

To address this problem a range of tools to assist with indexing, retrieving and searching techniques have been developed [Zamboni, 1998, Pinkerton, 2000, Howe and Dreilinger, 1997, Zuno, 1997]. However, the objective of efficiently and effectively delivering the right information, to the right people, at the right time, is still a fundamental research challenge. To this end, the research described in this thesis attempts to address the problem by designing, developing and evaluating an overarching system that incorporates and coordinates a number of different recommendation methods to suggest recommendations (section 1.1). In particular, a market-based approach is used to coordinate the behaviour of the constituent methods that make recommendations using a variety of techniques (section 1.2). By designing, simulating and implementing such a system, we have advanced the state of the art in a number of important ways outline in section 1.3. Finally, the whole document structure is listed in the end of the chapter (section 1.4).

1.1 Overcoming Information Overload

The most widely used tool to assist with information overload is that of a search engine. It provides a means of searching for existing information and it can exist in many different forms. For example, Lycos (`http://www.lycos.com`), one of the oldest search engines among those available today [Zamboni, 1998], extracts keywords, title, the words found in the first few lines of text, and the most frequently-occurring words in the rest of the document to provide users with relevant documents. WebCrawler [Pinkerton, 2000] was the first comprehensive full-text search engine for the Web. It performs index searching based on the document contents. But the number of processed documents is limited [Zamboni, 1998]. SavvySearch [Howe and Dreilinger, 1997] is designed to efficiently query other search engines by carefully selecting those engines most likely to return useful results and by responding to fluctuating load demands on the Web. SavvySearch learns to identify which search engines are most appropriate for particular queries, reasons about resource demands and represents an iterative parallel search strategy as a simple plan. Vrisko [Zuno, 1997] is a personal knowledge manager client, which provides facilities such as integrated searching over many search engines and relevance (pertinence to the matter at hand) ranking of results. Vrisko allows users to specify a context for the search. It also allows for user profiling. The profiles are updated according to user feedback and also automatically by search results. However, with all these searching tools, the user has to consider the large number of sources available, decide which one to access, and interact with each one individually. This is very tedious and inconvenient to the user [Levy et al., 1996]. Moreover, although these tools provide a means of searching for existing information, they lack a mechanism for informing the user about new information related to their interests. When cooped with the exploding volume of digital information, it is difficult for a user who is only equipped with a search capability to stay informed without sifting through huge amounts of incoming information.

To overcome this problem, it is possible to exploit information filtering techniques that can help in this context [Loeb and Terry, 1992]. Information filtering systems sort through large volumes of continuously arriving textual information and present to the user only those items that are likely to satisfy his[1] (or her) interest [Belkin and Croft, 1992]. Many such tools are now available over the Internet. For example, the Stanford Information Filtering Tool (SIFT) [Yan and Garcia-Molina, 1995] provides a service capable of directing information of users' interests to the subscribed users. It supports full-text filtering using well-known information retrieval models (such as boolean profiles, vector space and relevance feedback [Salton, 1989]). The SIFT filtering engine implements novel indexing techniques, capable of processing large volumes of information against a large number of profiles. Agentware i3 [Autonomy, 1997] applies pattern matching algorithms with contextual analysis in order to provide users with relevant documents based on information about the users' interests. SIGMA uses filtering agents to learn the user's interests when they query Usenet news articles (a high-volume and high-turnover discussion list service on the Internet) [Karakoulas and Ferguson, 1996]. However, in general, such tools still tend to have the weakness of either providing too much irrelevant information or missing relevant information [Shardanand and Maes, 1995, Resnick and Varian, 1997].

To overcome the above mentioned limitations of filtering, *recommender systems* have been advocated. A recommender system is one that assists and augments the natural social process of making choices among recommendations from all kinds of sources without sufficient personal experience of the alternatives [Resnick and Varian, 1997]. Thus, in this context, a recommendation is viewed as a reference to an item that will be directed to the user who is looking for information. A typical recommender system aggregates and directs recommendations to appropriate recipients. Given this view, it can be seen that a recommender system's main value lies in information aggregation and its ability to match the recommendations with people seeking information. It differs from conventional filtering systems in that recommendations are based upon subjective values assigned by people, namely the quality of items, rather than more objective properties (such as the text content of a document) of the items themselves [Resnick et al., 1994, Shardanand and Maes, 1995]. Compared to a system that only has searching or other simple information filtering functionalities, recommender systems require less experience on the part of the user and less effort to specify and restrain their interests when querying and operating the system [Resnick and Varian, 1997]. This is because recommender

[1] "His" represents "his or her" throughout the thesis. Similarly, "he" and "him" represent "he or she" and "him or her" respectively.

systems provide their users with recommendations that have been recognized as good (based on their previously expressed preferences or the preferences of other users with similar interests). Given these benefits, recommender systems have now been applied in many application domains, including music albums [Shardanand and Maes, 1995], video [Hill et al., 1995], Usenet news [Terveen et al., 1997, Resnick et al., 1994, Konstan et al., 1997], and Web navigation [Kahle and Gilliat, 1997, El-Beltagy et al., 2001].

Against this background, this research is concerned with the problem of information overload on the Web and how recommender systems can be used to help overcome this problem. In particular, it deals with the "where to go next" problem by presenting recommendations (represented as URLs) that are relevant to the users' current browsing context. This method is beneficial since users often ask questions such as "what else should I read?" and "where do other people go from here?". By convention, such recommendations are usually displayed in a separate window without interrupting a user's current navigation (Figure 1.1 is an example of the system that we have built for this task).

To date, two typical kinds of filtering approaches are used to produce recommendations: content-based and collaborative filtering (see section 2.1 for more details). The former makes recommendations by analyzing the similarity between the contents of the items that are ready to be recommended and those that have previously been marked as liked by the user. The latter makes recommendations by putting forward items that have been deemed appropriate by people who have similar interests to the user. Based on these two techniques, a large number of recommendation filtering methods have been developed (again see section 2.1 for more details). However, most conventional recommender systems share two major weaknesses:

- Each recommender system typically embeds some specific algorithm to compute correlations (the similarity of two relevant objects). However, there is no universally best way of doing this (and nor do we believe that there will ever be such a method). Rather, it is always the case that some methods are better in particular conditions and others are better in other conditions [Breese et al., 1998]. Given this, we believe the solution is to have a suite of recommendation methods available and to have the system automatically detect which one is the most appropriate

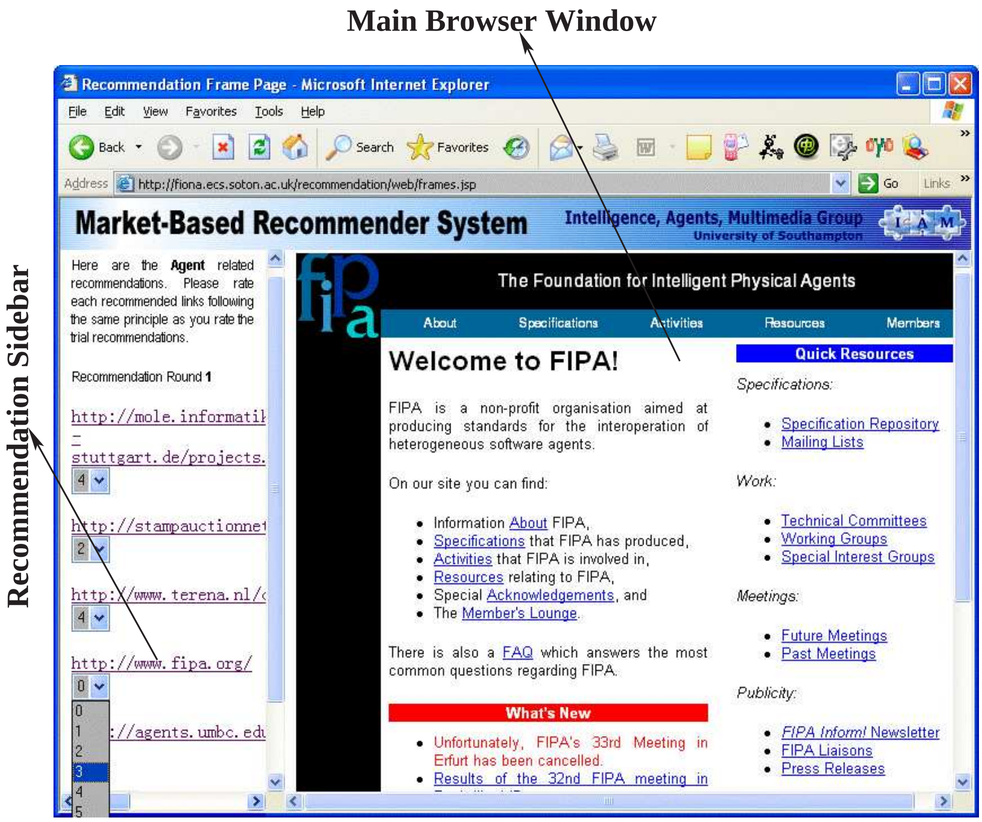


FIGURE 1.1: Browser with Recommendations

The main window displays the user's current context (the page being viewed). The side bar on the left is the output of the recommender system and displays a list of URLs in decreasing order of relevance to the user's current context.

in the prevailing context. However, such coordination is very difficult to attain, because the outputs from these diverse methods need to be compared.

- As ever more information is available on the Web, the pool from which recommendations can be made will continue to grow. However, users do not want correspondingly more recommendations to be presented (otherwise they will be overloaded). Thus, there is a need to be ever more selective and ensure that only the most appropriate recommendations are put forward.

Given these observations, with the growing number of recommendation methods and in face of evolving users' interests, we believe the best way forward in this area is to allow multiple recommendation methods to co-exist and to provide an overarching open system[2] that allows new methods to be added as and when they are developed and which coordinates their outputs such that only the best recommendations (from whatever source or method) are presented to the user [Wei et al., 2003b, To appearb]. Specifically, we believe a *market-based approach* is an efficient means of achieving such coordination because the problem of selecting appropriate recommendations to display in the sidebar space can be viewed as one of scarce resource allocation and markets are an efficient solution for this class of problems [Clearwater, 1996, Wellman and Wurman, 1998]. Moreover, the underlying economic theory provides an analytical framework for predicting aggregate behaviour and designing individual information providers [Mullen and Wellman, 1995]. Given this, this thesis is concerned with the systematic design and evaluation of such a market-based recommender system (see section 2.5 for the basics of market-based recommender systems). In particular, our system is capable of recommending documents relevant to the users' current browsing context as a way of dealing with the problem of information overload[3].

1.2 A Market-Based Approach to Recommendation

Moreau *et al.* [2002] showed that multiple recommendation methods can be incorporated into a single system, and that a marketplace is capable of coordinating multiple methods to direct the most valuable Web documents to the user's browser. In this system, each time a request for recommendations is made to the recommender system, a large number of recommendations may be put forward. However, from the point of view of the user, when seeking recommendations, it is ineffective to have too many items put forward. Moreover, the browser sidebar space is limited. Therefore, the question of how to accommodate so many recommendations into the limited sidebar space is the main concern.

[2] An open system is one in which the structure of the system itself is capable of dynamically changing [Jennings and Wooldridge, 1998]. The characteristics of such a system are that its components are not known in advance, can change over time, and may be highly heterogeneous (in that they are implemented by different people, at different times, using different software tools and techniques).

[3] In this work, we are not concerned with developing new recommendation methods. Our aim is to efficiently coordinate existing (and future) methods so that the overall system produces the best information to the user. We do not compare the relative performance of the methods. Rather our concern lies with the fact that different methods make recommendations simultaneously and we let the user decide which recommendations are good (irrespective of the specific methods they are provided by).

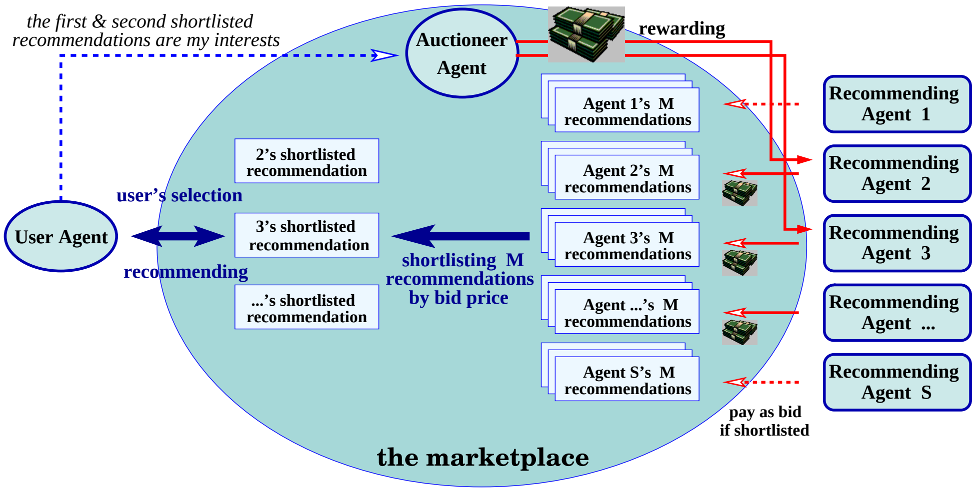


Figure 1.2: The Marketplace

Good recommendation providers are encouraged and bad recommendation providers will eventually go bankrupt.

At an abstract level, the problem of populating the limited space of the sidebar from the large number of potential recommendations can be viewed as an example of a scarce resource allocation problem. Given this observation, one of the best ways of allocating resources is to sell them using free market techniques and ideas [Clearwater, 1996]. More specifically, auctions are an excellent method of distributing resources to those who value them most highly [Klemperer, 1999]. Here, an auction can simply be viewed as a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants [McAfee and McMillan, 1987]. In a typical auction, there is an auctioneer, a seller and potential bidders. The auctioneer, acting on behalf of the seller, wants to sell the item and get the highest possible price, while the bidders, taking some bidding strategies to place bids, want to buy the item at the lowest possible price [Vickrey, 1961, Milgrom, 1989, Klemperer, 1999]. In this context, auctions are used to evaluate the recommendations, select the most valuable items and direct them to the browser sidebar (as illustrated in Figure 1.2).

However, there is not a universal auction design that is applicable to every context. Auctions vary from one another and these variations make the auctions more or less efficient in particular types of application. In our case, the marketplace operates according to the following metaphor. A user and the user agent acting on their behalf are selling sidebar space where information may be displayed. Information providers are keen to

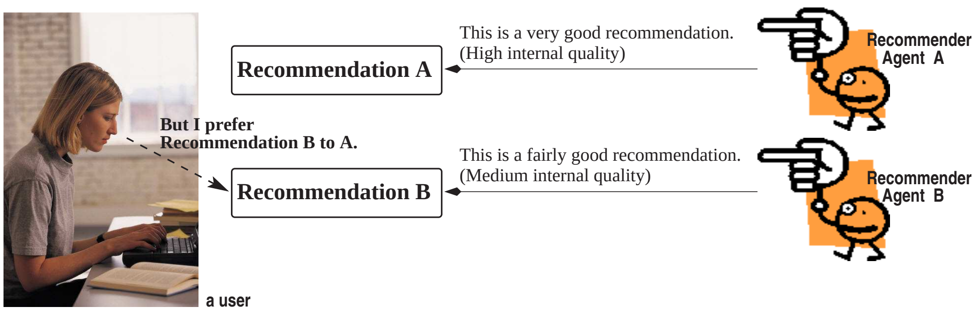


FIGURE 1.3: Different Valuations of Quality

get their recommendations advertised in the user's browser, and compete in the marketplace, ready to pay for such advertisements. Such information providers act as bidders. Each recommendation bears a bidding price acting as one bid. The recommender system acts as the auctioneer, ranking and selecting the most valuable items and recommending them to the user. The user will then choose some of them according to his interests as the next documents to be viewed. Those who provided such recommendations are the winners in this auction and will receive some reward in return (since such documents are deemed useful). Those documents not chosen by the user are deemed to have no relevance to the current document and will therefore receive no reward. Thus, over the longer-term, those agents that make good recommendations become richer and so are able to get their recommendations advertised more frequently than the methods whose recommendations are infrequently chosen by the user.

Given this background, however, none of the standard auction protocols can be used directly (for reasons outlined in more detail in section 2.4). Thus, this work had to develop a bespoke mechanism in order to realise this system (see Chapter 3 for more details).

Before delving into the details of the system design, it is important to clarify how the various measures of quality in the above discussion can be used to make the market function. In more detail, how good a recommendation is and whether a system works well are eventually decided by the user's opinion. To this end, all recommender systems share the same objective of improving recommendation quality. However, most of the existing systems lack a means of: *(i)* Specifically defining the quality of recommendations from the viewpoint of the user and various recommendation methods, since they may

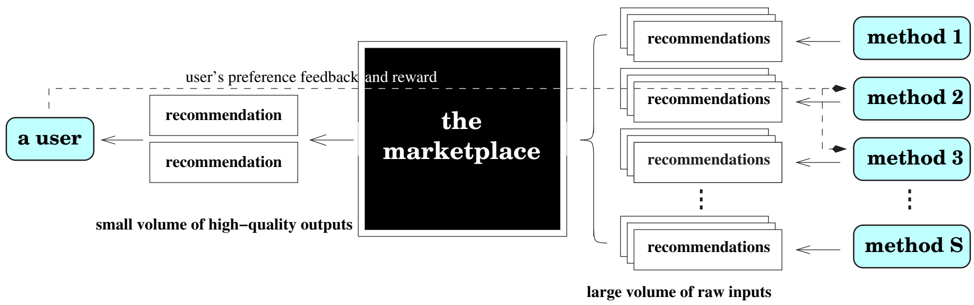


FIGURE 1.4: The Black Box Coordinating Various Methods

view the quality differently; *(ii)* Correlating these different qualities in a meaningful manner. In more detail, given a recommendation provided by a recommender agent with a specific recommendation method, a user's valuation of the recommendation may differ from that of the recommender agent. For example, in Figure 1.3, a recommender agent might highly rate a recommendation and therefore direct it to the user. However, the user may see this as a poor quality recommendation that is not very interesting. Given this situation, it is clear that the quality of a recommendation can be viewed from two aspects. From the viewpoint of the user, how well a recommendation satisfies the user is termed the *user perceived quality* (UPQ). From the viewpoint of a recommender agent with a specific recommendation method, the relevance score it computes for a particular recommendation is termed its *internal quality* (INQ). Moreover, the INQ value produced by different methods can vary significantly from one another. Therefore, without a systematic means of relating the UPQ to the recommendation methods' INQ, it is very difficult to provide high quality recommendations. In this research, the key challenge is to design a reward mechanism (that reflects the user's satisfaction of the recommendations) so that the marketplace can correlate the UPQ and the INQ.

In short, the key role of the marketplace developed in our system is to try and connect these two quality values by imposing a reward regime that incentivises the recommender agents to bid in a manner that establishes an appropriate correlation between these values and their bid price. In this way, the marketplace can be viewed as a black box with recommendations provided by different recommender agents as the input and only a few best recommendations passed through to the user as the output (see Figure 1.4).

In sum, the thesis designs and evaluates the market-based mechanism as a means of coordinating recommendations[4]. Through experiments and evaluations, we demonstrate that *the market-based approach to recommender systems is an effective means of coordinating multiple different recommendation methods in one single system and that it is an effective way of dealing with the problem of information overload by selecting only the best items from whatever methods to be displayed to users.* Specifically, it is not an information filtering technique, nor a hybrid recommender system, but a novel approach to making recommendations.

1.3 Research Contributions

The work described in this thesis advances the state of the art in the following ways:

(1) The market-based approach to recommender systems provides a novel method for effectively coordinating the behaviour of multiple recommendation methods with diverse measures of INQ. The market-based design exhibits the following properties:

- The analytical studies demonstrate that the bespoke marketplace we have designed is Pareto-optimal, maximizes social welfare and is individually rational.
- The empirical studies demonstrate that the market always converges, shortlists recommendations in decreasing order of UPQ, gives clear incentives and is fair to all constituent recommender agents, and is stable.
- The user studies demonstrate that the marketplace is an effective means of coordinating multiple different recommendation methods and the system is able to identify the best recommendations from the user's perspective and can suggest them frequently to users.

(2) We differentiate the quality of recommendations by two types of measurement: the UPQ and the INQ. No other recommender system uses UPQ or provides a mechanism for correlating INQ and UPQ.

[4]The credits paid by recommender agents for advertising their recommendations and the rewards awarded to agents to encourage them to put forward good suggestions are not a real currency. Thus, there is not a business model concerned with these credits and rewards, they are used only for the coordination of the recommender agents in our system.

(3) A novel reinforcement learning strategy is developed to assist recommender agents' bidding so that they can best satisfy the users while still maximizing their revenue.

These contributions have been published in the following papers:

- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Learning Users' Interests by Quality Classification in Market-Based Recommender Systems. IEEE Transactions on Knowledge and Data Engineering. To appear. This paper supports the above contribution point (3).
- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. A Market-Based Approach to Recommender Systems. ACM Transactions on Information Systems. To appear. This paper supports the above contribution points (1 & 2).
- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Learning users' interests in a market-based recommender system. In *Proc. of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04)*, pages 833–840, Exeter, UK, 2004a. Springer LNCS 3177. This paper supports the above contribution point (3).
- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Market-based recommender systems: Learning users' interests by quality classification. In *Proc. of the 6th International Workshop on Agent-Oriented Information Systems (AOIS-2004)*, pages 119–133, New York, US, 2004b. This paper supports the above contribution point (3).
- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Market-based recommendations: Design, simulation and evaluation. In *Proc. of the 5th International Workshop on Agent-Oriented Information Systems (AOIS-2003)*, pages 63–78, Melbourne, Australia, 2003a. Springer LNAI 3030. This paper supports the above contribution points (1 & 2).
- Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Recommender systems: A market-based design. In *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03)*, pages 600–607, Melbourne, Australia, 2003b. ACM Press. This paper supports the above contribution points (1 & 2).

- Luc Moreau, Norliza Zaini, Jing Zhou, Nicholas R. Jennings, Yan Zheng Wei, Wendy Hall, David De Roure, Ian Gilchrist, Mark O'Dell, Sigi Reich, Tobias Berka, and Claudia Di Napoli. A market-based recommender system. In *Proc. of the 4th International Workshop on Agent-Oriented Information Systems (AOIS-2002)*, pages 50–67, Bologna, July 2002. This paper supports the above contribution point (1).

1.4 Thesis Structure

The remainder of the document is structured in the following manner:

Chapter 2 presents a literature review on recommender systems, online auctions, and market-based recommender systems. This chapter provides a detailed discussion of these terms, and will discuss what others have done in these areas.

Chapter 3 presents the detailed design of the main components of our market-based recommender system. It deals with the evaluation techniques, auction protocol, reward mechanism, bidding strategy, and a discussion of the market equilibrium.

Chapter 4 establishes the experimental settings of the reward agent, the recommender agents and the user agent to simulate the marketplace empirically. This chapter then empirically evaluates the system design by simulations of the marketplace equipped with multiple recommendation methods.

Chapter 5 presents a recommender agent's problem of learning users' interests in terms of correlating the INQs to the UPQs of its recommendations in our marketplace, develops the strategy we build for the recommender agents, and evaluates the learning strategy.

Chapter 6 presents the user evaluation of our market-based approach to recommender systems. Specifically, this chapter sets out a number of evaluation metrics, presents a typical user trial of our system, configures the marketplace and three recommendation methods, and evaluates the system using a number of real users against the metrics we set up.

Chapter 7 draws some conclusions on the system design and presents the plan for future work.

Chapter 2

Literature Review

This chapter reviews the state of the art in conventional recommender systems (section 2.1) and agent-oriented approaches to recommender systems (section 2.2), outlines the fundamental principles and concepts of auction theory (section 2.4), outlines existing work on market-based recommender systems (section 2.5) and presents some basics of reinforcement learning (section 2.6).

2.1 Recommender Systems

To date, a large number of recommendation techniques have been developed. These are, however, based mainly on three kinds of filtering techniques: *content-based*, *collaborative* and *demographic* (although there is also some work on *hybrid* filtering techniques). Each of these categories will be discussed in turn in this section.

2.1.1 Content-Based Filtering

Conventional techniques to deal with information overload typically exploit content-based filtering techniques. Such filtering techniques recommend items for the user based on the descriptions of previously evaluated items. They have been widely used in making recommendations of information items. For example, Syskill recommends Web documents based on users' binary ratings ("hot" and "cold") of their interests [Pazzani et al., 1996] and Newsweeder helps users filter Usenet news articles by learning the user's profile based on his ratings [Lang, 1995]. Generally speaking, however, content-based filtering approaches have a number of weaknesses in recommending good items:

- A user's selection is often based on the subjective attributes (such as the quality, style or point-of-view of items) of the item [Goldberg et al., 1992], whereas content-based approaches are based on objective attributes (such as the text content of a document) about the items and do not take the user's perceived valuation of such subjective attributes into account [Montaner et al., 2003]. For example, these methods cannot distinguish between a well written and a badly written article if both happen to use the same terms.
- Either the items must be of some machine parsable form (e.g. text), or attributes must have been assigned to them by hand [Shardanand and Maes, 1995]. With current technology, media such as sound, photographs, art, video or physical items cannot be analyzed automatically for relevant attribute information. Moreover, in most cases, it is not practical or possible to assign these attributes by hand due to limitations of resources.
- The techniques do not have an inherent method for generating serendipitous finds [Shardanand and Maes, 1995]. They tend to recommend more of what a user has already seen. This is because content-based methods compare potential items with items that the user has already seen. However, the user's interests may beyond the scope of the previously seen items. Thus, such interesting items can hardly be recommended to the user.

2.1.2 Collaborative Filtering

A complementary technique that is also widely used is *collaborative filtering* [Goldberg et al., 1992] (or *social filtering* [Shardanand and Maes, 1995]). The basic idea of collaborative filtering is people recommending items to one another [Terveen et al., 1997]. Collaborative filtering essentially automates the process of "word-of-mouth" recommendations: items are recommended to a user based upon values assigned by other people with similar interests. The system determines which users have similar interests via standard formulas for computing statistical correlations (e.g., ratings on items of their interests) [Goldberg et al., 1992]. Collaborative filtering overcomes some of the limitations of content-based filtering in that the items being filtered need not be amenable to parsing by a computer. Moreover, recommendations are based on the users' perceived quality of items, rather than more objective properties of the items themselves [Shardanand and Maes, 1995].

In more detail, collaborative filtering techniques match people with similar interests and then recommend one person's highly evaluated items to the others [Goldberg et al., 1992, Resnick et al., 1994]. Thus, rather than computing the similarity of items (which relies on machine analysis of content [Herlocker et al., 2000]), collaborative filtering computes the similarity of user's interests. This means that subjective data about items can be incorporated into recommendations. This, in turn, facilitates serendipitous new finds because interesting items from other users can extend the current user's scope of interest beyond his already seen items. In addition, collaborative filtering techniques can be used to recommend both machine parsable items (such as textual articles [Terveen et al., 1997]) and non-machine parsable ones (such as audio and video files [Shardanand and Maes, 1995, Hill et al., 1995]). Indeed, perhaps the greatest strength of collaborative techniques is that they are completely independent of any machine-readable representations of the objects being recommended. Thus, they work well for complex objects such as music and movies where variations in taste are responsible for much of the variation in preference [Burke, 2002].

Collaborative filtering techniques have been widely used in many application domains (not only textual documents but also non machine-parsable media products). For example, GroupLens is a system for collaborative filtering of Usenet news to help people find articles they will like in the huge stream of available articles [Resnick et al., 1994, Konstan et al., 1997]. The system displays predicted scores (i.e. INQs) for the items suggested to its users and the users give their ratings (i.e. UPQs) to the articles after they read them. The system predicts items' scores based on the heuristic that people who agreed in the past will probably agree in the future with people who have similar interests. Based on similar techniques of predicting scores of items, MovieLens is a system for personalized recommendations for movies [Miller et al., 2003] and Ringo for music albums [Shardanand and Maes, 1995]. However, collaborative filtering recommender systems can do more than finding information items for people. They can even assist people to find other people (rather than documents) with similar interests. MEMOIR is such an example [DeRoure et al., 2001].

However, collaborative filtering approaches also have a number of shortcomings:

- Large numbers of people must participate so as to increase the likelihood that any one person will find other users with similar interests [Terveen and Hill, 2001] (the sparsity problem). The difficulty of achieving a critical mass of participants makes collaborative filtering experiments expensive.
- A user whose interests share little with others' will receive poor recommendations on a collaborative basis. An extreme case of this phenomenon happens when new users start off with nothing in their profiles of interests and must train a profile from scratch (the "cold start problem" [Resnick and Varian, 1997]). Even with a start profile, there is still a training period before the profile accurately reflects the user's preferences [Maltz and Ehrlich, 1995].
- These systems suffer from the "early-rater problem" [Montaner et al., 2003]: when a new item appears in the database, there is no way it can be recommended to a user until more information is obtained through another user either rating it or specifying which other items it is similar to.

2.1.3 Demographic Filtering

The demographic approach uses descriptions of people (such as occupation, age and gender) to learn the relationship between a single item and the type of people who like it [Krulwich, 1997]. For example, a mature, sophisticated woman is likely to prefer an expensive leather jacket, whereas a teenage schoolgirl may prefer a cheap denim one. To date, there is only one widely recognized system, LifeStyle, that uses such a technique [Krulwich, 1997].

However, this method has two principal shortcomings:

- It creates profiles by classifying users using stereotypical descriptions [Rich, 1979]. Thus, the same items are recommended to people with similar demographic profiles. However, in many cases, the stereotypes are too general to generate good quality recommendations [Montaner et al., 2003].

- If the user's interests shift over time, demographic filtering does not adapt their profile [Koychev, 2000]. For these reasons, demographic filtering is rarely used independently of the other filtering techniques.

2.1.4 Hybrid Approaches

As can be seen, both content-based and collaborative filtering have weaknesses. Moreover, these weaknesses tend to complement one another [Montaner et al., 2003]. Thus, hybrid filtering systems that integrate the two approaches have been advocated [Herlocker et al., 2000]. In a hybrid system, both objective and subjective properties of an item are taken into account in predicting its quality when making recommendations. For example, "collaboration via contents" filterbots integrate content-based filtering techniques to build virtual users in the GroupLens collaborative system when new recommendation items lack ratings [Sarwar et al., 1998], the Fab collaborative system maintains user profiles by using content-based analysis [Balabanovic and Shoham, 1997], Pazzani's system involves user collaborations to determine the ratings of predicted items and a content-based profile to compute similarity among users [Pazzani, 1999], Popescul's system uses secondary data (e.g. document contents) to predict users' preferences in collaborative recommendations when there is a lack of user ratings [Popescul et al., 2001], and Claypoole's system employs separate collaborative and content-based recommenders and uses an adaptive weighted average of the two in making its selections (as the number of users accessing an item increases, the weight of the collaborative component tends to increase [Claypool et al., 1999]). In addition, many business systems use a hybrid filtering approach to recommend their products and services, such as Amazon (`http://www.amazon.com`) and CDnow (`http://www.cdnow.com`).

While hybrid systems can sometimes overcome the shortcomings of pure content-based and pure collaborative systems, with respect to the objective and subjective properties of recommendations, they do so in a rigid and predetermined manner. Specifically, such systems try to use one of the recommendation properties (either objectiveness or subjectiveness) to complement the weaknesses of the other when the latter does not work effectively. However, there is no automated way of determining in what circumstances which kind of properties (objective, subjective or both) are relevant to a particular user in their current context.

2.1.5 Conclusion

Having an understanding of the basics of the different filtering techniques used in recommender systems, we can see that there is no universally best filtering technique for all kinds of users in all various situations. Therefore, a bespoke mechanism that incorporates multiple techniques and lets only the best items pass through is needed. The related work in terms of incorporating and coordinating multiple recommendation methods is discussed in section 2.5. However, before we do this, we discuss the agent-oriented approach to recommender systems.

2.2 Agent-Oriented Recommender Systems

As exemplified above, many recommendation methods have been developed and it is likely that many more will become available in the future. Moreover, we can see that many recommender systems incorporate various techniques to assist recommendations and interactions among various entities in a system. In this way, such systems will become very complex. For these reasons, from the point of view of software engineering, there is a need to maintain the system architecture so that it can work efficiently, when new components and more interactions amongst them become available [Jennings, 2001]. Additionally, many existing recommender systems are personalized to make the recommendations more effective to the users [Shardanand and Maes, 1995, Baclace, 1991, Sheth and Maes, 1993, Lieberman, 1995]. This is because, on the one hand, people's interests differ from one another, as well as their behaviour against the recommendations. On the other hand, the recommender can serve a user more effectively by learning his interests. To this end, agents are an effective conceptual model for various components of personalized environments [Maes, 1994].

To be more specific, here, an agent can be viewed as a computer system situated in some environment, that is capable of *autonomous actions* in that environment in order to meet its design objectives [Wooldridge and Jennings, 1995]. Agents have a variety of characteristics that make them an appropriate solution for complex systems in general: *(i)* They are capable of dealing with the dynamically changing characteristics of open systems. *(ii)* They represent a powerful tool for making complex, large or unpredictable systems modular [Jennings and Wooldridge, 1998]. In this way, the overall problem can be partitioned into a number of smaller and simpler components, which are easier to develop and maintain. *(iii)* They assist with the human computer interaction because they can be personalized to their owner. Agents can thus act as "expert assistants" with respect to some application, embodying knowledge about both the application and the users, and capable of assisting the users to achieve their goals [Jennings and Wooldridge, 1998].

In addition to these general reasons, we believe an agent-oriented approach is suitable in this particular context, because:

1. Users can naturally be represented in the system by user agents [Maes, 1994] that act autonomously on their behalf, finding information relevant to them, but also observing their activities, so that the system can tailor its answers to their needs.
2. Information sources can naturally be represented as active agents whose objective is to ensure their content is widely disseminated to appropriate users.
3. The open nature of a recommender system means the software components need to interact in flexible ways and that such interactions cannot be hand-crafted at design time.

These benefits have been recognized by a number of researchers and this has led to the development of a number of agent-based recommender systems. Relevant examples of such systems include:

- *Fab* combined content-based and collaborative filtering techniques to recommend items [Balabanovic and Shoham, 1997]. It is a hybrid system that seeks to incorporate the advantages of both methods. The *Fab* agents are used to search and index Web documents, and for profiling group interests. They enable the system to automatically identify emergent communities of interest in the user population, which enables enhanced group awareness and communications.

- *MEMOIR* uses trails, open hypermedia link services and a set of software agents to assist users in accessing and navigating vast amounts of information in Internet environments [DeRoure et al., 2001]. Additionally, it exploits trail data to support users in finding colleagues with similar interests, since users with similar interests follow similar paths and therefore they leave comparable trails. *MEMOIR* differs from previous systems in that it aims to find users rather than documents.

- *Linking in Context* provides a service that automatically recommends relevant Web document links to the user's current context, such that the user does not need to consult any external search facilities [El-Beltagy et al., 2001]. In more detail, the system coordinates different types of agents to extract document context, and generates and categorizes links into clusters according to different users' interests. When new types of users' interests become available, new link clusters are added into the system.

2.3 Improving the Quality of Recommendations

Although both agent and non-agent based recommender systems have a number of advantages in comparison to pure filtering systems, most of them still suffer from problems related to quality of recommendations. These include:

- The quality of the recommendations with respect to the user's preferences needs to be improved [Sarwar et al., 2000]. Users need recommendations they can trust to help them find what they like. If the recommendations provided by the recommender system are poor, then the system will not be used.

- It is difficult to match the various recommendation methods' INQ to the UPQ. This is because the INQ is based on a recommendation method's internal valuation of a recommendation, while the UPQ is based on the user's valuation of it. And, as discussed in section 1.2, these two valuations can be very different.

- There is not a universal criterion to evaluate which methods are better than the others. Therefore, there is not a universally best recommendation method that can always suggest the best recommendations in all circumstances. Moreover, it is difficult to compare and evaluate the different methods that are available [Breese et al., 1998]. This is due to the fact that different recommendation methods are based on different notions. For example, some are based on the correlation between the item contents (such as term frequency inverse document frequency [Salton and McGill, 1983] and weighting [El-Beltagy et al., 2001]), while others are based on the correlation between users' interests (such as votes [Goldberg et al., 1992] and trails [DeRoure et al., 2001]).

A number of recommenders have tried to address these research challenges. Specifically, Yu et al. [2001] introduced an information theoretic approach to measure the relevance of a user for predicting the preference for the given target concept. The learning approach [Sheth and Maes, 1993, Baclace, 1991, Billsus and Pazzani, 1998] to personalized information filtering using relevance feedback and genetic algorithms also helps in the context of improving the quality of recommendations. Finally, Sheth [Sheth and Maes, 1993] has demonstrated a system with a collection of information filtering interface agents which has learning capabilities of specializing users' interests and exploring new potential domains to predict the users' preferences so as to provide good recommendations. Now all these approaches make it possible to provide good recommendations from the viewpoint of one specific recommendation method itself. However, they are still weak in improving the quality from the user's point of view. Additionally, they still lack a regime that can seek the best recommendations across a number of different methods in a single system. Therefore, these approaches are still some distance away from meeting the objective of suggesting the best recommendations in all possible circumstances. In contrast, we believe that the best way of dealing with this problem is to allow multiple recommendation methods to co-exist and to provide an overarching system that coordinates their outputs such that only the best recommendations (from whatever source or method) are presented to the user. To make this a reality, however, a number of key challenges need to be overcome:

1. Given multiple recommendation methods, it is comparatively easy to provide the user with a multitude of recommendations. However, the challenge is to filter them to the user in a decreasing order of UPQ, given that there is no comparability among different methods' INQs. Moreover, another difficulty is that the notion of quality is not absolute, but rather it is user-defined and varies over time, typically according to the user's interests and activities.

2. Many methods for suggesting relevant information already exist and more will be developed in the future. Some may be specialized to specific media such as audio or video [Hill et al., 1995]; others may be particularly suited to process information in text documents [Maltz, 1993]; while yet others are more efficient in specific application domains [DeRoure et al., 2001]. From an architectural point of view, this means the challenge is to design a modular system that has the ability to accommodate multiple recommending methods, and to integrate them in a seamless and dynamic fashion as they appear on line.

As outlined in section 1.2, the means we have chosen to manage the multiple recommenders in a marketplace. To this end, section 2.4 explains the basic concepts of auctions in more detail.

2.4 Auction Theory

Auctions are an efficient and effective market mechanism. They are important in our research because they are an efficient means of allocating scarce resource. Moreover, the underlying economic theories are well-studied [Klemperer, 1999, Vickrey, 1961, Milgrom and Weber, 1982, McAfee and McMillan, 1987, Milgrom, 1989] and are suitable for modeling various market participants in experiments. In this context, we are not going to provide a detailed survey on auction theory (see [Klemperer, 1999] for such information). Instead, we report on the issues related to agent-mediated on-line auctions that are relevant to our context.

In designing an auction mechanism, two key issues are essential: the *protocol* and the agents' *strategy* [Jennings et al., 2001], each of which will be dealt with in turn.

Formally, a protocol is a set of norms that constrain the proposals that the auction participants are able to make [Jennings et al., 2001]. To date, thousands of different auction protocols have been devised [Wellman et al., 2001]. However, four standard auctions are widely used and analyzed: the ascending auction (also called the English auction), the descending auction (also called the Dutch auction), the first-price sealed-bid auction (sometimes called the Yankee auction), and the second-price sealed-bid auction (also called the Vickrey auction).

In the ascending auctions, the price is successively raised until only one bidder remains, and that bidder wins the object at the final price. This auction can be run by having the seller announce prices, by having the bidders call out prices themselves, or by having bids submitted electronically with the best current bid posted. The descending auction works in exactly the opposite way: the auctioneer starts at a very high price, and then lowers it continuously. The first bidder that calls out to accept the current price wins the good at that price.

In the first-price sealed-bid auction, each bidder independently submits a single bid, without seeing others' bids, and the good is sold to the bidder who makes the highest bid. The winner pays his bid. In the second-price sealed-bid auction, everything is the same as in the first-price sealed bid auction except that the good is sold to the bidder who makes the highest bid, but at a price of the second-highest bidder's bid.

In all of these cases, if there is more than one item to be sold, the items can all be sold at the same price (called price uniformity) or they may be sold at different prices (called price differentiation). In this work, we consider auctions with price differentiation since these allow a seller to obtain the maximum possible profit [Varian, 2003] p434-436 (refer to section 3.2 for more details).

Given a specific auction protocol, a bidder's strategy is the decision making model that they use in order to decide what actions to take [Jennings et al., 2001]. For example, a bidder's best strategy, irrespective of others', in an English auction is to bid actively until the price reaches the value of the good to him [Milgrom and Weber, 1982]. As another example, a bidder's best strategy in a second-price sealed-bid auction is to submit a bid equal to the value of the good to him [Milgrom and Weber, 1982]. Generally speaking, however, for most auction protocols the strategy is not so simple and significant work needs to be invested in order to design it.

To date, however, comparatively little work has been done on using auction-based techniques to implement recommender systems. The reason for this is that recommender systems have typically not incorporated multiple recommendation methods. Thus there is no need for a coordination mechanism. There are, however, two systems that have exploited this approach and these are discussed in section 2.5.

2.5 Market-Based Recommender Systems

Bohte *et al.* [2001] developed a system that used a competitive market-based allocation of consumer attention space from the user's perspective. In the marketplace, supplier agents compete with each other to attract the user's attention. The system uses an $(n + 1)$ auction, in which all winners pay the $(n + 1)^{th}$ price (i.e. there is no price differentiation) and n is the number of items for competition. This work investigates the user's behaviour in making choices when faced multiple items. Three user models are demonstrated: *1.* independent visits with several purchases, *2.* independent visits with one expected purchase, and *3.* search-till-found behaviour. According to [Bohte et al., 2001], model *1* shows very efficient allocation of recommendations, whereas in models *2* and *3* efficient allocation is difficult to obtain. This is because the recommendation provider's expected payoff is dependent on the allocation of the other recommendations. In this work, we believe the user's behaviour models when making decisions in the face of a list of recommendations are worth following (see section 4.1.3). However, this work has a number of limitations when it comes to being used in a recommender system context:

- This work is not concerned with the UPQ. Therefore, the recommenders cannot readily learn the user's actual interests.
- The auction protocol requires all recommender agents to pay identically for different selected items and comparatively little information is conveyed by the system (primarily a signal of selection). With only identical information of different selections, the system cannot differentiate between the various selected items and so

neither can the recommender agents. Thus, the recommender agents cannot learn to improve the quality of their recommendations according to a user's interest.

- This work is not targeted specifically at recommendations, but rather at the allocation of consumer's attention space. In other words, this work concerns the user's behaviour when facing a list of recommendations rather than producing high quality recommendations.

The second piece of work in this vein was our initial attempt at coordinating multiple recommendation methods in a single system as detailed in [Moreau et al., 2002]. In this work, an extensible modular recommender system suggesting relevant Web documents to the user's context is developed. The system is capable of integrating multiple recommendation methods. Moreover, a marketplace is developed in the system to coordinate various methods in making recommendations. The system also exhibited the efficient capability of incorporating multiple recommendation methods in a single system in a seamless manner. However, without modelling the user, the marketplace fails to meet the objective of making high quality recommendations with respect to the user's interests:

- Without investigating the user's valuation of the recommendations (the UPQ), the recommendations provided by various methods based on their individual valuation of the items (the INQ) are blindly made to the user (because the methods do not know whether the user likes the recommendations). Moreover, without defining the UPQ, the system fails to incentivise recommending agents about the user's interests. And the system has difficulty in knowing which recommendations are better given a user's context. Therefore, it is difficult to present the user with a list of recommendations in decreasing order of the user perceived quality.

- The marketplace is not economically efficient because of the way it limited the rewards. More specifically, there is only one recommendation rewarded in each auction. The rewarded item is always the first bid which is predicted to be the best. However, a real user may not select the first bid item. Moreover, he may select multiple recommendations in one auction in the real circumstances. Thus, the system should reward all user-selected items and more than one agent could learn the user's interests in each auction. Therefore, one reward regime limited the system learning capability of the user's interests. Moreover, the amount of each reward is based on the other agents' bids. In this way, the reward and the bidding price are based upon each other. Thus, the reward is in no way related to the user's valuation of a recommendation and therefore cannot incentivise the agents about the user's interests. In an efficient market, the reward and the price should indicate the value and cost of a good so as to incentivise the agents how the user values an item and how much to bid on it.

- The system cannot stop bad recommendations. This is because the reward regime is not based on the UPQ of the recommendations. Thus, bad recommendations can also bid to a suitable level to be shortlisted and rewarded continuously. However, we expect a good marketplace to have the capability to encourage good recommendations and discourage bad ones.

Against this background, none of the standard auction protocols are suitable to regulate our marketplace. This is because:

- Usually, there is only one bidder who wins in the standard auction protocols. However, there may be more than one bidder (the recommender agents) who wins in our auction.
- In the standard auction protocols, there is no notion of reward. However, in our auction, the system will reward the winners for making good recommendations.

Therefore, we need to develop a bespoke auction protocol, including a reward mechanism, to regulate the agents in our marketplace. With such an auction protocol, we also need to design a bespoke strategy for the agents. And these will be discussed in detail in Chapter 3.

2.6 Reinforcement Learning

In terms of suggesting good recommendations, an agent in our system needs to classify its recommendations into different INQ categories and correlate these categories to the user's preferences. This is because, by frequently suggesting items from those categories that best satisfy the users, a self-interested agent is able to maximize its revenue. To do this, an agent needs to make trials over these categories and see which make profits (by receiving rewards) and which lose credits (by paying for the bid, while not receiving rewards). Now this kind of "trial-and-error" learning behaviour is exactly what happens in *Reinforcement Learning* [Mitchell, 1997].

Generally speaking, in reinforcement learning, an agent is assumed to be situated in a multi-state environment where the agent's actions determine both its immediate reward and the next state of the environment. Moreover, this state transition affects the agent's future actions and, consequently, its future expected rewards. Thus, the expected future reward needs to be factored into the agent's current decision making. Therefore, in this kind of problem, an agent needs to learn which actions are desirable based on rewards that can be obtained arbitrarily far in the future. In this context, Markov Decision Processes (MDPs) are a typical model of this kind of reinforcement learning [Mitchell, 1997]. An MDP consists of a set of states (s), a set of actions (a), a state transition function ($\delta(s_t, a_t) = s_{t+1}$) and a reward function with respect to the transitions ($r = (s_t, a_t)$). The MDP model makes the complicated decision making processes intuitive and simplified by estimating the overall payoff (current reward plus the discounted delayed rewards in future) of all the possible state transitions (all combinations of the state-action pair (s_i, a_j)). After this estimation profile converges, an agent's optimal action selection strategy is choosing the action with the maximal overall payoff at any given state. However, in the context of our market-based recommender's learning, taking one action is independent of taking another (because future rewards are only based on future recommendations' UPQs and have no relationship to the current recommendation) (see details in section 5.2.1). In other words, the learning strategy we need does not have the concept of states and state transition. It considers the actions and their corresponding rewards only.

The most relevant work to our context in terms of reinforcement learning is the k-armed gambling problem [Berry and Fristedt, 1985]. In the k-armed gambling problem, an agent faces k gambling machines, each of which has a payoff probability of zero or one. The agent needs to learn the average payoffs that can be obtained from each gambling machine (or from each INQ segment in our case) as quickly as possible, while still maximizing its revenue. In this context, the solution to the k-armed gambling problem also suits our problem. Specifically, Berry and Fristedt developed a recursive algorithm to find the optimal strategy to gain the maximal payoffs in the case that the agent is permitted a fixed number of pulls [Berry and Fristedt, 1985]. However, in our context, the recommender agents do not have a limit on the number of interactions they can have with the marketplace. Thus, we aim to develop an unbiased optimal strategy by allowing the agents to gain sufficient experience. Gittins also tackles the k-armed gambling problem [Gittins, 1989]. His "allocation index" method indexes all the actions that an agent experienced with a combined value of the expected payoff of each action and the value of the information that can be obtained by choosing it. The agent then chooses the action with the largest index value and this is shown to guarantee the optimal balance between exploration and exploitation. However, this technique only applies if the expected future rewards are discounted, which is inappropriate in our context because future rewards are equally important as the current ones in our system (thus we do not discount rewards). Thrun also develops an exploration strategy for the k-armed gambling problem [Thrun, 1992]. Specifically, his strategy always chooses the action with the highest payoff. However, this strategy may produce a biased estimation from the true expectation, since the actions with negative signals received in the beginning may have insufficient experience to produce biased expectations (as we discuss in section 5.2.3). In contrast, we aim to build a strategy that can produce an unbiased expectation. Kaelbling's interval-based technique can be seen as an extension of Thrun's greedy strategy [Kaelbling, 1993]. Her approach computes the upper bound of the confidence interval on the success probabilities of all actions and chooses the one with the highest upper bound. However, this approach relies on an a priori analysis of the payoff distribution of each action. This approach also has the shortcoming that insufficient experience in the beginning of learning may produce biased confidence intervals and this, in turn, also induces a bias from optimum.

In sum, none of the above learning strategies is suitable for the recommender agents in our marketplace. Therefore, a specific strategy needs to be developed for our context, and this will be discussed in detail in Chapter 5.

2.7 Summary

We believe that with the growing number of recommendation methods and in the face of evolving users' interests, there is a need to develop a mechanism capable of accommodating multiple recommendation methods (both existing and forthcoming) and let them work together to provide high quality recommendations. In this context, agent-mediated online auctions appear to be a promising method for coordinating such system. Thus, the key challenge of this research is to develop the market mechanisms that have the capability to incentivise various recommending agents to bid according to the user's valuation of their recommendations such that only a small number of the most valuable items will be directed to the user. Additionally, a good learning strategy for recommenders to learn users' interests so that they can make good recommendations while still maximize their revenue is another research challenge.

Chapter 3

Auction Mechanism Design for the Recommender System

This chapter discusses the design of the auction for our recommender system. In this endeavor, there are two major issues that need to be addressed: the protocol and the strategies (as detailed in sections 3.2, 3.3 and 3.4).

This chapter contributes to the thesis with the auction mechanism being designed. The mechanism we developed for our system is theoretically proved to be able to give incentives of users' interests to recommender agents by rewarding the user-selected recommendations. A reward mechanism is also developed to incentivise recommender agents about the users' interests. The mechanism follows the principle of good recommendations making positive revenue by receiving rewards, whereas bad recommendations make a loss by paying to be shortlisted but not receiving any rewards. Additionally, the reward mechanism we developed is Pareto optimal and maximizes social welfare. With the auction protocol and reward mechanism in place, a set of rational bidding strategies are also developed for agents to make their recommendations. Finally, we theoretically proved that the marketplace is able to dynamically reach the equilibria. And at the equilibria, we show the recommender agents are able to reason about the relationship between the bidding prices and the rewards so as to maximize their revenue.

The remainder of this chapter discusses the auction protocol (section 3.2) and reward mechanism design (section 3.3), the agents' bidding strategies (section 3.4), as well as a theoretical analysis on how the marketplace works to relate the agents' bids to their corresponding rewards (section 3.5). First, however, some central evaluation criteria for mechanism design are presented.

3.1 Evaluation Metrics

Designing market mechanisms is an *engineering design task*, in which the rules should be developed in order to meet particular objectives, either for certain participants or for society as a whole [Roth, 2002]. In seeking to design the market mechanism for our recommender system, therefore, our first step is to identify the properties that we would like our auction to exhibit. This then gives us the requirements against which we can evaluate our design. In particular we would like to design a market that has the following standard properties [Sandholm, 1999, Varian, 2003, Dash et al., 2003]:

❶ *Pareto Efficiency:* A solution x is Pareto efficient if there is no other solution x' such that at least one agent is better off in x' than in x and no agent is worse off in x' than in x. Pareto efficiency provides us with a way of comparing alternative mechanisms and good mechanisms should be designed to maximize allocation efficiency [Roth, 2002, Sandholm, 1999, Varian, 2003]. This is important from the point of view of the individual agents because if a non-Pareto efficient mechanism is chosen then the design could be improved upon (for at least one agent) without making any of the other agents worse off.

❷ *Social Welfare Maximization:* In our context, social welfare is a numeric measure of the sum of all agents' utilities. In contrast to Pareto efficiency, social welfare provides a way to rank different social preferences over the various solutions and to indicate which is best for the group of agents as a whole [Kagel and Roth, 1995]. This is a supplement to the Pareto efficient criterion. From the viewpoint of individual agents, there may exist many Pareto efficient solutions to the given problem that cannot be distinguished between. In such cases, social welfare maximization provides a way of differentiating between them by determining which is the best from the social point of view [Sandholm, 1999, Varian, 2003].

❸ *Individual Rationality:* Participation in an auction is individually rational for an agent if its payoff in the auction is no less than what it would get by not participating. A mechanism is individually rational if participation is individually rational for all agents [Sandholm, 1999]. Individually rational protocols are essential in our context because all agents (representing the various recommendation methods) need a clear incentive to participate in the market so that the best possible recommendations can be picked by the market. Indeed, if the protocol is not individually rational for some agents, they would simply not participate in the auction and their recommendations would be lost.

❹ *Convergence:* If the prices for the goods being allocated converge after a number of rounds of auctions, the market is said to be convergent. This is important from the viewpoint of the bidding agents since it enables them to learn to bid rationally at a certain level for a given type of good (characterized by UPQ level in this case) in order to maximize their revenue [Roth, 2002]. Without convergence, an agent will never know how much to bid with respect to a given recommendation and therefore the marketplace behaviour will be chaotic.

❺ *Effective Shortlist in Decreasing Order of UPQ:* This is the common aim of all recommender systems [Herlocker et al., 2004]. The marketplace should be capable of shortlisting the recommendations in decreasing order of the UPQ after a number of auction iterations. This is important from the point of view of the users since they only want a small number of the best recommendations.

❻ *Clear Incentives:* A good mechanism design should give agents incentives to act in particular way, such that the system's global goal is attained despite the individual goal of the self-interested agents [Dash et al., 2003, Sandholm, 1999]. In our context, the protocol should be able to incentivise the recommender agents about the user's interests so that they can bid differently for different INQ levels. This is important because a recommender agent needs to relate its bids to the INQ of the recommendations through the feedback from the marketplace which reflects the user's preferences.

❼ *Fairness:* A good market mechanism should be fair to all participants [Roth, 2002, Dash et al., 2003]. In our context, a protocol is fair if it gives all constituent recommenders equal opportunity of suggesting their recommendations to users (irrespective of the agent or method that generates them). This is important because we want the system to shortlist the best recommendations in an unbiased manner meaning that the system does not degenerate to a single constituent recommender.

❽ *Stability:* A protocol is stable if it provides all agents with an incentive to behave in a particular way over time. The marketplace should be designed to be stable because if a self-interested agent is better off behaving in some other manner than desired, it will do so [Sandholm, 1999]. Thus, an unstable protocol allows agents to behave with intentions that make the system deviate from its best potential outcome [Roth, 2002]. Therefore, stability is important because without it the system behaviour is unpredictable.

With these metrics in place, sections 3.2, 3.3 and 3.4 detail the auction protocol we designed, the reward mechanism we established and the bidding strategies of the individual agents. Section 3.5 then analyzes how the market performs with such market mechanisms and the bidding strategies in place. Section 3.6 evaluates the auction mechanism against the first three metrics when analyzing the market equilibrium, whereas the evaluations against the last five metrics will be discussed in Chapter 4. We have organized the evaluation in such a way because the first three points are set from a pure economic point of view and, therefore, the marketplace can be evaluated against them at design time. The remaining five items relate to the quality of the system's output and can only be evaluated by experiments.

3.2 The Auction Protocol

This section defines the auction protocol for managing the multiple recommender agents (as per Figure 1.2 on page 7). To ensure recommendations are provided in a timely and computationally efficient manner, we choose a *generalized first-price sealed-bid auction* in which all agents whose recommendations are shortlisted pay an amount equal to their valuation of the advertisement (meaning we have price differentiation). We choose a sealed bid auction (in which agents will typically make a single bid) to minimize the time for running the auction and the amount of communication generated. We choose a first price auction with price differentiation because the relative ordering of the recommendations affects the likelihood of them being selected by the user. In particular, in the market, each information provider agent is keen to get their recommendations advertised to the user. Each agent has a valuation of the advertisement (which will be different between the different agents) and is willing to pay up to this amount to display its recommendations. When an agent gets its recommendations short-listed, and therefore advertised to the user's browser, it has consumed the advertisement service provided by the recommender system. In return, it needs to pay an amount of credit to the system.

In more detail, the market operates in the following manner. Each time when an auction is activated, the user requests recommendations of Web documents. In each such activation, the auctioneer agent calls for a number of bids, say M ($M > 0$) equal to the number of recommendations it is seeking. Then, each constituent recommender agent submits M bids to the auctioneer agent (each bid contains a recommendation and a price). After a fixed time, the auctioneer agent ranks all the bids it received by their bidding price, and directs the M bids with the highest prices to the user's browser (as recommendations in the sidebar — see Figure 1.1). Those bidding agents whose recommendations are shortlisted pay the auctioneer agent according to how much they bid. Those bidding agents whose recommendations are not shortlisted do not pay anything. The user may then take up a number of these shortlisted recommendations in which case the agent that supplied them is rewarded.

The protocol for each auction round is formally defined in Figure 3.1. It should be noted that: *(i)* function $GenerateBid\ (A_{bi}, rec_j, price_j)$ relates to the bidding strategy and will be discussed in section 3.4; *(ii)* function $UserSelectsRecs(SU)$ concerns the user's behaviour of making choices among the shortlisted recommendations and will be discussed in section 4.1.3; and *(iii)* function $ComputeReward(b_h)$ concerns the reward mechanism and will be discussed in section 3.3.

3.3 The Reward Mechanism

With the auction protocol in place, we now turn to the reward mechanism. According to our protocol, the user may select multiple recommendations from the shortlist. For

The Variables:

- S: the number of recommender agents ($S > 1$);
- $A_{b1}, A_{b2}, ..., A_{bS}$: S bidding agents;
- A_B: complete set of bidding agents, i.e., $A_{b1}, A_{b2}, ..., A_{bS}$;
- T_b: duration of the auction;
- M: number of recommendations that the auctioneer agent requests;
- $b_{ij} = \langle A_{bi}, rec_j, price_j \rangle$: bid provided by A_{bi}, containing the j^{th} recommendation with bidding price $price_j$ ($i \in [1..S], j \in [1..M]$);
- B^{ALL}: a set of bids which represents all bids submitted to the auctioneer agent;
- B^M: a set of bids which represents the shortlisted bids that will be recommended to the user;
- B^R: a set of bids which represents those selected by the user (and will be rewarded by the auctioneer agent);
- SU: a set of recommendations displayed in the user's sidebar (i.e. B^M ignoring the prices);
- SU^R: a set of recommendations that are selected by the user (i.e. B^R ignoring the prices);
- N: number of user-selected recommendations;
- b_l, b_h: two bids for temporary use ($l, h \in [1..M]$);
- R_h: reward to h^{th} user-selected recommendation.

The Algorithm:

```
B^ALL = φ;
B^M = φ;
B^R = φ;                                        // system initialization
CallForBids(A_B, M, T_b);                        // system calls for bids
repeat during the duration of auction T_b
{
    b_ij = GenerateBid(A_bi, url_j, price_j);
    B^ALL = B^ALL ∪ {b_ij};
}
for l = 1 to M do                               // shortlist M highest bids
{
    b_l = FindBidWithLthTopPrice(B^ALL, l);
    B^M = B^M ∪ {b_l};
}
SU = { ⟨A_bi, url_j⟩ | ⟨A_bi, url_j, price_j⟩ ∈ B^M };    // the set of shortlisted URLs
SU^R = UserSelectsURLs(SU);                      // user makes selection (SU^R ⊆ SU)
B^R = { ⟨A_bi, url_j, price_j⟩ | ⟨A_bi, url_j⟩ ∈ SU^R and ⟨A_bi, url_j, price_j⟩ ∈ B^M };
N = |B^R|;                                       // the number of user selected items
for h = 1 to N do
{                                                // reward the user selected items
    b_h = FindHthBid(B^R, h);
    R_h = ComputeReward(b_h);
}
```

FIGURE 3.1: The Auction Protocol

each such user-selected recommendation, the suggesting agent is given a reward. A given agent may have multiple recommendations selected in a given auction in which case it receives multiple rewards. In defining the $ComputeReward$ function, our aim is to ensure that it is both Pareto efficient and social welfare maximizing (as per section 3.1). To this end, this section addresses the following issues: *(i)* How is one reward mechanism judged to be better than another? *(ii)* Does there exist a reward mechanism that is the best amongst all possible mechanisms? First, however, a complete set of reward mechanisms is introduced.

3.3.1 The Complete Set of Reward Mechanisms

Let us assume we have N (defined in section 3.2) user-selected recommendations to be rewarded and the auctioneer has an amount of payoff, R^T, to be distributed to the relevant agents. The problem is then how to best split R^T into parts and distribute them to each of the rewarded recommender agents such that we cannot find any other more optimal allocation solutions.

To this end, we define the complete set of reward mechanisms as follows: Suppose the h^{th} ($h \in [1..N]$) user-selected recommendation receives an amount of payoff R_h. Then, all possible reward mechanisms are such that the sum of each payoff is less than or equal to R^T. That is, $\sum_{h=1}^{N} R_h \leq R^T$. Therefore, we have a complete set of reward mechanisms, $\hat{\Re}$, such that:

$$\hat{\Re} = \{ \ (R_1, R_2, \cdots, R_N) \ | \ \sum_{h=1}^{N} R_h \leq R^T \}$$

Now each element of $\hat{\Re}$ is a possible allocation of R^T and $\hat{\Re}$ can be split into two complementary subsets: $\hat{\Re}_1$ that does not completely allocate all of R^T (called a *With Surplus Mechanism* (WSM)) and $\hat{\Re}_2$ that does allocate all of R^T (called a *No Surplus Mechanism* (NSM)):

$$\hat{\Re}_1 = \{ \ (R_1, R_2, \cdots, R_N) \ | \ \sum_{h=1}^{N} R_h < R^T \} \quad \textbf{(WSM)}$$
$$\hat{\Re}_2 = \{ \ (R_1, R_2, \cdots, R_N) \ | \ \sum_{h=1}^{N} R_h = R^T \} \quad \textbf{(NSM)}$$

From these two subsets, we want to identify those that are both Pareto efficient and social welfare maximizing.

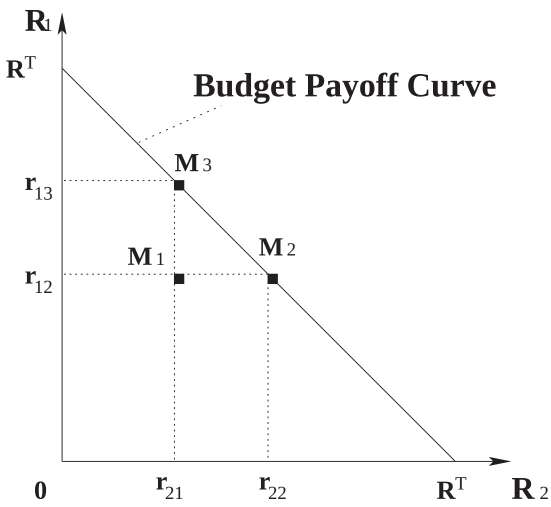


FIGURE 3.2: Pareto Optimization

3.3.2 Pareto Optimal Reward Mechanisms

If there is only one recommendation to be rewarded, it is trivially true that awarding all of R^T to this recommendation is the Pareto optimal solution. However, when there is more than one recommendation to be rewarded, the allocation is more complicated. To simplify the presentation, we discuss the case where two recommendations are rewarded (i.e., $N = 2$). This is chosen since it can easily be depicted and it gives us a direct impression of allocation. The general case with multiple recommendations rewarded ($N \geq 2$) can be analyzed in the same way.

To this end, Figure 3.2 depicts the case where there are two recommendations to be rewarded (R_1 and R_2). The axes represent the payoff allocated to each recommendation. We define the budget payoff curve as the line joining $(0, R^T)$ and $(R^T, 0)$ and it represents the payoffs whose sum is R^T. The triangle formed by the budget payoff curve and the axes, including the edges, contains all possible allocations of R^T. The outcome of each possible reward mechanism corresponds to a point within this area[1]. Those points on the budget payoff curve represent the elements of NSM since each of these points allocates the total amount of R^T. For example, for point M_2, $R_1 = r_{12}, R_2 = r_{22}$ and $R_1 + R_2 = r_{12} + r_{22} = R^T$. Generally speaking, therefore, any mechanism in the set

[1]One point in this area may represent multiple reward mechanisms since different mechanisms may result in the same outcome. In this case, our concern is how much the reward to one recommendation is related to the reward of another. Hence, we are concerned only with the outcome and ignore what specific reward mechanism the outcome comes from.

NSM maximizes the total payoff and it is impossible to distinguish between any of these points. A mechanism that produces a reward in the triangle, but not on the budget payoff curve, is by definition in the WSM set. For example, for point M_1, $R_1 = r_{12}$, $R_2 = r_{21}$ and $R_1 + R_2 = r_{12} + r_{21} < R^T$.

In terms of Pareto efficiency, for any point representing a WSM outcome, at least one Pareto optimal point can be found representing a related NSM. For example, in Figure 3.2, point M_1 (in WSM) can straightforwardly be transformed into M_2 (in NSM) by giving R_2 the extra amount of reward $(r_{22} - r_{21})$. However, those points on the budget curve cannot be improved upon since giving extra reward to either recommendation necessarily results in a loss to the other. Therefore, all NSM outcomes are Pareto efficient.

3.3.3 Social Welfare Maximizing Reward Mechanisms

Pareto efficiency has nothing to say about the distribution of welfare across agents. Thus, given two mechanisms that produce outcomes that are both Pareto efficient, it is not possible to say which is better. Thus, we need a further means of differentiation. To this end, we seek to define a social welfare function that is able to assign a ranking to all Pareto efficient mechanisms. This ranking specifies the "social preference" [Varian, 2003] p590 of a distribution of overall welfare to different rewarded recommendations and should ensure that recommendations are rewarded according to how good they are. However, in our system, there are two different views on the quality of a recommendation: internal quality and user perceived quality (see section 1.2). The INQ is used to compute the agents' bid price — the higher the quality, the higher its bid price. The UPQ indicates how a recommendation satisfies the user. Since the recommender system's objective is to satisfy the user, *the marketplace's objective is to shortlist the most valuable recommendations in decreasing order of user perceived quality*. Therefore, we decide to reward the user-selected recommendations based on the UPQ and this quality can be defined as $Q_h \in [1..100]$ $(h \in [1..N])$. To do this, we can segment our set of potential reward mechanisms that are Pareto efficient (i.e. $\hat{\Re}_2$) into two complementary subsets — those that allocate reward in a manner proportional to the UPQ and those that do not:

Proportional Reward Mechanism (PRM)

$$\hat{\Re}_P = \{ \ (R_1, R_2, \cdots, R_N) \ | \ R_h = \frac{Q_h}{\sum_{i=1}^{N} Q_i} \times R^T \ , \ \ where \ \ h \in [1..N]\}$$

Non-Proportional Reward Mechanism (NPM) [2]

$$\hat{\Re}_N = \hat{\Re}_2 - \hat{\Re}_P$$

Given these two sets, we can now define our social welfare function in terms of utility. As noted above, we want the system to prefer a reward mechanism that distributes the welfare to the user-selected recommendations according to how well they satisfy the user. Therefore, a standard Cobb-Douglas utility function [Varian, 2003] is introduced. This function shows preferences of the inputs in a manner proportional to the value of their powers:

$$U(R_1, R_2, \cdots, R_N) = {R_1}^{Q_1} \cdot {R_2}^{Q_2} \cdot \ \cdots \ \cdot {R_N}^{Q_N} \tag{3.1}$$

In this function, the powers, $Q_1, Q_2, \cdots, Q_N$, describe how important each rewarded agent's utility is to the overall social welfare. Specifically, a reward mechanism, $\mathbf{m}_i = (R_{1,i},\ R_{2,i},\ \cdots,\ R_{N,i})$ is better than (or more socially-preferred to) $\mathbf{m}_j = (R_{1,j},\ R_{2,j},\ \cdots,\ R_{N,j})$, if $U(\mathbf{m}_i) > U(\mathbf{m}_j)$.

Our objective now is to find if there exists a best mechanism within $\hat{\Re}_2$. Thus, we need to determine if there is a mechanism that has the maximum utility value, given a total amount of reward R^T. That is:

Proposition:

There exists an $\mathbf{m}' \in \hat{\Re}_2$*, such that* $\forall \mathbf{m} \in \hat{\Re}_2$*, if* $\mathbf{m} \neq \mathbf{m}'$, $U(\mathbf{m}') > U(\mathbf{m})$.

Conditions:

$$N \text{ is a natural number} \tag{3.2}$$

$$Q_i > 0 \ \ \text{and is constant} \quad (i \in [1..N]) \tag{3.3}$$

$$R^T > 0 \text{ and is constant} \tag{3.4}$$

$$R_1 + R_2 + \cdots + R_N = R^T \tag{3.5}$$

$$R_i > 0 \ ^{3} \quad (i \in [1..N]) \tag{3.6}$$

[2]Note that $\hat{\Re}_P$ contains only one element (given an R^T and a set of Q_i, whose values are fixed, there is only one solution for $\hat{\Re}_P$), while $\hat{\Re}_N$ contains multiple elements.

[3]We do not consider the case of $R_i = 0,\ i \in [1..N]$, since this case must result in $U = 0$ and any mechanism with a positive utility is better than this solution.

Proof:

In the case of $N = 1$, $R_1 = R^T$ ensures the maximal value of U and this is the solution that we want. We now turn to the case of $N > 1$. Based on the given conditions, the utility function always results in a positive value within the whole input range. Therefore, a monotonic transformation of the original utility function, $V = \ln U$, simplifies the problem.

$$V(R_1, R_2, \cdots, R_N) = \sum_{i=1}^{N} Q_i \ln R_i \tag{3.7}$$

Hence, finding the maximum value of U is equivalent to finding that of V. We will first determine whether there exists extrema in function V, and then verify how many there are, and then, whether they are maxima or minima.

Function (3.7) has one constraint (condition (3.5)) on the N input variables. Thus, only $N-1$ variables remain independent. Let us consider that R_1 is dependent of the other $N-1$ variables

$$R_1 = R^T - (R_2 + R_3 + \cdots + R_N) \tag{3.8}$$

Substituting equation (3.8) for R_1 in function (3.7), we get:

$$V = Q_1 \ln[R^T - (R_2 + \cdots + R_N)] + Q_2 \ln R_2 + \cdots + Q_N \ln R_N \tag{3.9}$$

Therefore, in (3.9), $R_2 \cdots R_N$ are independent of each other. The necessary condition for V reaching extrema is:

$$\begin{cases} \frac{\partial V}{\partial R_2} = \frac{-Q_1}{R^T-(R_2+R_3+\cdots+R_N)} + \frac{Q_2}{R_2} = 0 \\ \frac{\partial V}{\partial R_3} = \frac{-Q_1}{R^T-(R_2+R_3+\cdots+R_N)} + \frac{Q_3}{R_3} = 0 \\ \qquad\vdots \\ \frac{\partial V}{\partial R_N} = \frac{-Q_1}{R^T-(R_2+R_3+\cdots+R_N)} + \frac{Q_N}{R_N} = 0 \end{cases} \tag{3.10}$$

Now (3.10) has $N-1$ equations and $N-1$ variables and is non-simplified. Its unique series of solutions is $R_j = R^T \frac{Q_j}{\sum_{i=1}^{N} Q_i}, (j \in [2 \cdot \cdot N])$. Substituting this for R_2 to R_N in equation (3.8), we get:

$$R_h = R^T \frac{Q_h}{\sum_{i=1}^{N} Q_i}, \; \textit{where } h \in [1 \cdot \cdot N]. \tag{3.11}$$

We record this extremum, (3.11), as M_{PRM} and note that it represents the PRM by its definition.

We now need to verify whether point M_{PRM} is a maximum or a minimum. This depends on the second derivative of V at this point, $d^2V_{M_{PRM}}$. $V_{M_{PRM}}$ (value of V at point M_{PRM}) is sufficiently maximum if $d^2V_{M_{PRM}} < 0$; $V_{M_{PRM}}$ is sufficiently minimum if $d^2V_{M_{PRM}} > 0$; $V_{M_{PRM}}$ is unclear if $d^2V_{M_{PRM}} = 0$.

From (3.11), we know that $\frac{Q_1}{R_1} = \frac{Q_2}{R_2} = \cdots = \frac{Q_N}{R_N}$. We assume $K = \frac{Q_1}{R_1}$. From condition (3.3) and (3.6), we know $K > 0$. The first derivative of V is,

$$dV = \frac{Q_1}{R_1}dR_1 + \frac{Q_2}{R_2}dR_2 + \cdots + \frac{Q_N}{R_N}dR_N \tag{3.12}$$

The second derivative of V is:

$$\begin{aligned} d^2V &= d(dV) \\ &= d(\frac{Q_1}{R_1})dR_1 + d(\frac{Q_2}{R_2})dR_2 + \cdots + d(\frac{Q_N}{R_N})dR_N \\ &= -(\frac{Q_1}{R_1^2}dR_1^2 + \frac{Q_2}{R_2^2}dR_2^2 + \cdots + \frac{Q_N}{R_N^2}dR_N^2). \end{aligned} \tag{3.13}$$

At point M_{PRM}, there is a constraint on $dR_1, dR_2, \cdots, dR_N$. This is, by differentiating condition (3.5) on both sides,

$$dR_1 + dR_2 + \cdots + dR_N = 0. \tag{3.14}$$

Therefore,

$$dR_1 = -(dR_2 + \cdots + dR_N). \tag{3.15}$$

The second derivative of V at point M_{PRM} is,

$$d^2V_{M_{PRM}} = -K^2[\frac{(dR_2 + \cdots + dR_N)^2}{Q_1} + \frac{dR_2^2}{Q_2} + \cdots + \frac{dR_N^2}{Q_N}]. \tag{3.16}$$

Since $Q_i > 0$ ($i \in [1..N]$), $d^2V_{PRM} < 0$. Therefore, V and U get maximum value at solution (3.11). Hence, given a specific R^T, M_{PRM} is the unique maximum point (i.e. $\mathbf{m}'$ in the proposition) and $\hat{\Re}_P$, represented by M_{PRM}, is the best mechanism in $\hat{\Re}_2$. ■

We now illustrate this outcome with an example with two recommendations being rewarded (Figure 3.3). Here, the axes represent the payoffs allocated to the two recommendations. M_{PRM} and M_{NPM} represent the PRM and an element of the NPM set. The utility curves defined by $U(R_1, R_2) = {R_1}^{Q_1} \cdot {R_2}^{Q_2}$ (as per function (3.1)) are depicted in Figure 3.3 and they give us a direct impression of the comparison of the different

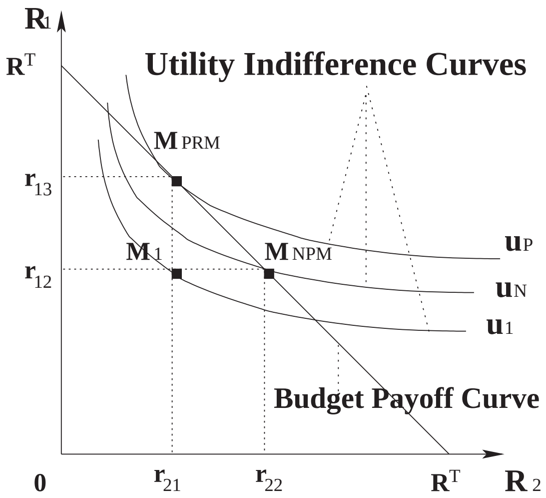


FIGURE 3.3: Utility Curve for Reward Mechanisms

mechanisms. In Figure 3.3, the mechanisms represented by points on the same utility indifference curve are as good as each other since they produce the same utility. However, given an arbitrary pair of utility indifference curves, mechanisms represented by points on the curve that is far from the origin (compared to the other curve) are better (or more preferred) than those on the other. This is because the curves bear higher utility the further away from the origin they are. Thus, in Figure 3.3, $u_P > u_N > u_1$. So, the mechanism represented by M_{PRM} is better than the one represented by M_{NPM}, which, in turn, is better than M_1. This discussion tells us that, by providing utility function (3.1) for the reward mechanism, the unique element of $\hat{\Re}_P$ represents the best possible mechanism. Therefore, this is the one we should use.

3.3.4 Designing the Practical Reward Mechanism

Having identified $\hat{\Re}_P$ as the best reward mechanism for our protocol, we now need to transform it into an applicable form that we can use in practice for our recommender system. What the analysis in section 3.3.3 essentially tells us is how a reward to one recommendation should be related to that of another. Thus, for example, if the user likes recommendation rec_A twice as much as recommendation rec_B, the reward to the former should be twice that of the latter. In addition, it is difficult to determine the actual value of R^T without delving into the specifics of a particular marketplace. To this end, we adjust the reward mechanism of (3.11) to an equivalent one that does not rely on R^T and is, therefore, easier to compute. In our revised mechanism, all user-selected recommendations are ordered in decreasing rank of UPQ (such that $Q_1 > Q_2 > \cdots > Q_N$) and each reward is based on the $(M+1)^{th}$ price P_{M+1} (the highest not shortlisted bid) instead of R^T:

$$R_h = \delta \cdot Q_h \cdot P_{M+1} \tag{3.17}$$

where $h \in [1..N]$, δ is the reward coefficient and $\delta > 0$. This new mechanism also ensures recommendations are rewarded proportionally to their UPQs and is therefore also ideal from the perspective of maximizing social welfare. This is because as each R_h ($h \in [1..N]$) is known, the value of the total payoff $\sum_{h=1}^{N} R_h$ is also known. Among all possible allocations for this total payoff, mechanism (3.17) ensures the maximal social welfare according to section 3.3.3. Given any value of δ, with respect to each specific auction round, P_{M+1} is fixed. So, R^T is fixed and thus solution (3.17) is equivalent to (3.11). We base the reward on P_{M+1} (whose value is not known by the bidding agents) so that the market cannot be manipulable by the participants [Varian, 2003] p289. If the reward is based on the prices from the rewarded recommendations, the rewarded agents might be able to affect the market through their prices since they are aware of the history of both rewards and bid prices. Our approach also reduces the possibility of bidding collusions because the reward is based on something that the rewarded agents are unaware of and cannot control.

However, the reward mechanism (3.17), as it currently stands, does not satisfy the system objective of shortlisting the most valuable recommendations in decreasing order of UPQ. This is because all individually rational agents will bid the same price (marginally higher than P_{M+1}) to maximize their revenue. This is because a bidder's revenue is the reward obtained minus the bidding price that has been paid (section 3.2) and, hence, a rational bidder should bid as low as possible to be shortlisted. When all shortlisted recommendations have the same bidding price, the system cannot differentiate and rank them by price. Therefore, we need a mechanism that can relate and regulate the bidding price according to the UPQ (i.e. higher quality means a higher price).

To achieve this, we involve two other variables: P_h ($h \in [1..N]$) and P_m^* ($m \in [1..M]$). P_h is the bidding price of *the h^{th} rewarded recommendation* (user-selected recommendation with the h^{th} highest UPQ). P_m^* is the historical average bidding price of the m^{th} shortlisted recommendation during the system's lifetime (note the bidding agents do not actually know this value). By this definition, P^*_m indicates the price for the m^{th} advertisement displayed in the user's browser sidebar which is decided by the invisible hand (namely the market) [Samuelson and Nordhaus, 2001].

With this additional information, we can now fine-tune the reward mechanism towards the system objective. We expect the fine-tuned mechanism to incentivise the recommender agents to bid their recommendations in a way that is proportional to the UPQs of the items (so as to shortlist the recommendations in decreasing order of the UPQ). And, on the other hand, the shortlisted agents can only maximize their revenue by bidding in this manner. To achieve this, instead of (3.17), we adjust the reward to the h^{th} rewarded recommendation to:

$$R_h = \delta \cdot Q_h \cdot P_{M+1} - \alpha \cdot |P^*_h - P_h| \tag{3.18}$$

where α is another system coefficient and $\alpha > 1$. The specific values of δ and α are not yet defined and their values will depend upon the specifics of the application (see Chapter 4 for details).

The reward mechanism in (3.18), compared with (3.17), gives recommender agents the incentive to adjust their bids to different levels according to their belief about the corresponding UPQ. With (3.18), the market can differentiate shortlisted recommendations by price so that the marketplace can shortlist good recommendations in decreasing order of UPQ. Moreover, under certain conditions, mechanism (3.18) will tend to be the same as (3.17) (to be discussed in section 3.5). Hence, mechanism (3.18) satisfies all the requirements listed in section 3.1.

3.4 Designing the Agents' Bidding Strategies

Rational bidders seek to maximize their revenue and they do this by bidding sensibly for recommendations that they believe are valuable to the user. If they provide poor recommendations they will either be not-shortlisted (revenue unchanged) or shortlisted but not rewarded (revenue decreases); if they provide recommendations that are relevant to the user's interest but either bid too high or too low (with respect to P^*_h —

section 3.3.4) they will also lose revenue[4]. The outcome of such bids is that the corresponding recommendation is: not shortlisted, shortlisted but not rewarded, or rewarded. Depending on what happened to its previous bid *for the given recommendation*, a rational bidder should base the bidding price of its next bid (P^{next}) for that recommendation on *(i)* the INQ, *(ii)* the last bid price (P^{last}) and *(iii)* the previous rewards to this recommendation. Assuming the INQ for next recommendation is unchanged, we need only consider the bidding strategies with respect to price and reward. To this end, we make one assumption and we will ensure this holds in developing and simulating the practical bidding strategies discussed in sections 4.1.2 and 4.2.3:

Assumption: *With respect to a given internal quality level, there are recommendations in the next bid that have the same quality level.*

With this in place, we now consider the three potential outcomes from bidding.

3.4.1 Bid Not Shortlisted

This leaves the agent's revenue unchanged since it neither has to pay for its advertising, nor does it receive a reward. The only way to increase revenue is to get the recommendation shortlisted (since this might bring a reward). Therefore, the agent will increase its bidding price for the same recommendation:

$$P^{next} = Y \cdot P^{last} \qquad (Y > 1)$$

This is the dominant strategy (something the agent is best off by using no matter what strategies the other agents use — section 2.4) in this case since being shortlisted and rewarded is the only way of increasing revenue.

3.4.2 Bid Shortlisted But Not Rewarded

These agents lose revenue since they pay for the advertising but receive no reward. This means the agent overrated its INQ with respect to the UPQ and so the agent should

[4]This is because the second term of equation (3.18) will not be zero, which results in a decrease of revenue. This point will be proved in detail in the end of this subsection.

decrease its price in subsequent rounds so as to lose less:

$$P^{next} = Z \cdot P^{last} \quad (0 < Z < 1)$$

This is the dominant strategy in this case since keeping the same price or even raising it will result in further losses.

3.4.3 Bid Rewarded

These agents have a good correlation between their INQ for a recommendation and that of the UPQ. Therefore, these agents have a chance of increasing their revenue. The profit made by the h^{th} rewarded recommendation is:

$$\xi_h = \delta \cdot Q_h \cdot P_{M+1} - \alpha \cdot |P_h^* - P_h| - P_h$$

However, since the agent is unaware of P_h^*, it does not know whether ξ_h has been maximized. Hence, it must minimize $(\alpha \cdot |P_h^* - P_h| + P_h)$ so as to maximize ξ_h. Furthermore, the agent does not know whether P_h is higher or lower than P_h^*. In either case, however, the agent will definitely make a loss if P_h is not close to P_h^* (proof below).

Assume the set of recommending agents remains unchanged between successive auctions and they produce recommendations of the same quality level (we discuss what happens when this situation does not hold in section 3.5). The UPQ for the h^{th} rewarded recommendation will remain in the h^{th} place in subsequent auctions. Given this, P_h^* is related to Q_h, such that the agent with the h^{th} rewarded recommendation is able to estimate the value of P_h^*. Now consider the design of the strategy for the h^{th} rewarded recommendation. We find that the h^{th} rewarded agent can always be aware of whether its price is closer to or farther from the h^{th} historical average market price, P_h^*, by adjusting its bidding prices. In this way, the agent can minimize its loss. The proof is given below.

Assumptions [*static marketplace*]:

(i) The h^{th} rewarded recommendation remains the h^{th} highest UPQ in subsequent bids. *(ii)* P_h^* remains stable in subsequent bids. *(iii)* There are sufficient bidders in the market with not-shortlisted increasing prices and shortlisted but not rewarded decreasing prices

to ensure P_{M+1} remains stable. *(iv)* $\Delta P > 0$, which represents an increment or a decrement of bidding prices.

Proposition:

When adjusting the bidding price of the h^{th} rewarded recommendation, an agent makes a profit increment if the price approaches the historical average price of the h^{th} advertisement slot; otherwise, it makes a profit decrement.

Proof:

According to assumptions *(i)* and *(ii)*, P_h^* is unchanged with respect to the h^{th} rewarded recommendation. So the corresponding agent can estimate the value of P_h^*.

When $P_h < P_h^*$, its profit in the current bid is:

$$\begin{aligned} \xi_{hl} &= \delta \cdot Q_h \cdot P_{M+1} - \alpha|P_h^* - P_h| - P_h \\ &= \delta \cdot Q_h \cdot P_{M+1} - \alpha P_h^* + (\alpha - 1)P_h \end{aligned}$$

Given that P_{M+1} is stable (assumption *(iii)*), if the agent raises the price by ΔP in the next bid, its profit in the next bid will be:

$$\begin{aligned} \xi_{hli} &= \delta \cdot Q_h \cdot P_{M+1} - \alpha|P_h^* - (P_h + \Delta P)| - (P_h + \Delta P) \\ &= \delta \cdot Q_h \cdot P_{M+1} - \alpha P_h^* + (\alpha - 1)P_h + (\alpha - 1)\Delta P \end{aligned}$$

Since $\alpha > 1$ and $\Delta P > 0$, $\xi_{hli} - \xi_{hl} = (\alpha - 1)\Delta P > 0$.

When $P_h < P_h^*$, if the agent decreases the price by ΔP in the next bid, its profit will be:

$$\begin{aligned} \xi_{hld} &= \delta \cdot Q_h \cdot P_{M+1} - \alpha|P_h^* - (P_h - \Delta P)| - (P_h - \Delta P) \\ &= \delta \cdot Q_h \cdot P_{M+1} - \alpha P_h^* + (\alpha - 1)P_h - (\alpha - 1)\Delta P \end{aligned}$$

Therefore, $\xi_{hld} - \xi_{hl} = -(\alpha - 1)\Delta P < 0$.

The case when $P_h > P_h^*$ can be proven in the same way.

Thus, when changing the bid price, an agent makes a profit increment if the price approaches the average price; otherwise, it makes a profit loss. In case of the bid price crossing over the average price, for example the current price is below the average price and by making an increment the next bid is above the average price, a rational agent will keep increasing its price to make more profit since it does not know its price has crossed over the average price. However, from this point, its price is going away from the average and so the agent will be aware of the loss in profit in the following bids. The case when the price crosses down over the average price can be proven in the same way.

■

This proof tells us that a rational rewarded bidder will adjust its price to the corresponding average market price to maximize its profit. The proof also indicates that, whatever its current price is with respect to the historical average, when adjusting the bid price, if the adjustment results in making less profit, it indicates the action is wrong and $(P_h \pm \Delta P)$ is farther from P_h^*; if it results in making more profit, it indicates the action is right and $(P_h \pm \Delta P)$ is closer to P_h^*. This phenomenon is listed in Table 3.1 ($\Delta\xi$ represents the possible profit of the next bid compared to that of the current bid). Table 3.1 also specifies the strategy for the rewarded agents. This strategy (to bid closer to the corresponding historical average market price) is the dominant strategy for the rewarded agents since otherwise they will definitely incur a loss of revenue. The actual value of ΔP will be defined in an application-specific manner.

TABLE 3.1: Price Adjustment and Results

Current Price	**Adjustment**	$\mathbf{\lvert P_h^* - P_h \rvert}$	$\boldsymbol{\Delta\xi}$
$P_h < P_h^*$	$+\Delta P$	$\searrow$	> 0
	$-\Delta P$	$\nearrow$	< 0
$P_h > P_h^*$	$+\Delta P$	$\nearrow$	< 0
	$-\Delta P$	$\searrow$	> 0

3.5 Market Equilibrium

According to the strategy for rewarded bidders (section 3.4), such bidders must bid in a manner that aligns their internal view of quality with that of the user. Thus, over time, each individual recommender agent improves its correspondence between its bid price and the user's preferences for recommendations. Only by achieving this, can an agent

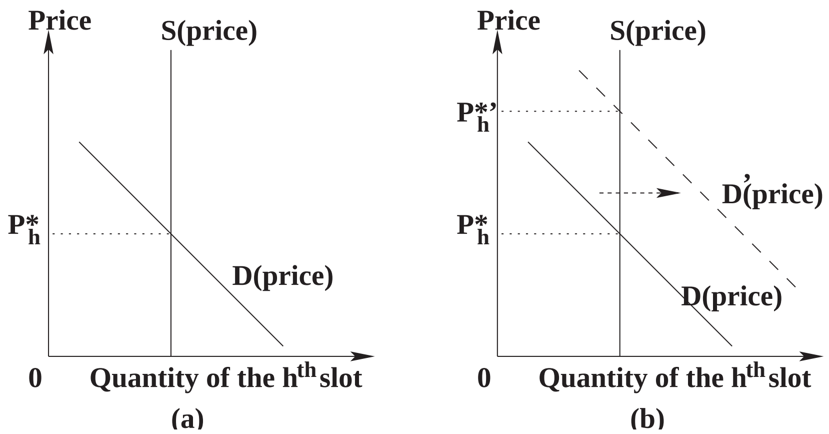


FIGURE 3.4: Market Equilibrium and Its Change
(a) The supply curve **S** is vertical indicating that whatever the deal is, the supply of the h^{th} advertisement slot is constant. The demand curve **D** has a slope indicating that more agents are willing to pay a low price and few agents are willing to pay a high price for the same slot. The cross indicates that at a certain price level the quantity of demand equals that of supply. This cross point represents the market equilibrium. **(b)** At each price level, more recommendations become available and the demand curve shifts to the right.

maximize its profit. How quickly this convergence occurs depends on the adjustment of price ΔP.

Under the assumption of a static marketplace (section 3.4.3), the market reaches an equilibrium. The h^{th} historical average market price reflects the market equilibrium price: thus, at a certain price, the quantity of demand of the h^{th} advertisement slot equals the quantity of the supply (see Figure 3.4(a) [5]).

In the long run, however, these assumptions will not hold and the equilibrium will tend to be broken. However, this new market situation will gradually tend towards another equilibrium and will reach it as long as the changes in the recommender agents are not too frequent with respect to convergence times (see Figure 3.4(b)). If, for example, there is more demand in the system, the demand curve will shift right compared to (a). This means at each price level, there are more bidders willing to pay for the same advertisement slot (because, for example, more better recommendations are being produced).

[5]Strictly speaking, the demand curve should be discrete in this case. And the quantity of supply is 1 since we differentiate between each of the M slots and there is only one h^{th} slot. To simplify the discussion, however, we use a continuous demand curve in this context.

At equilibrium, since the bidding prices are aligned with the UPQ, the system can produce a shortlist of recommendations in decreasing order of UPQ which is precisely the objective of the recommender system.

3.6 Economical Evaluations of the Marketplace

This section evaluates the market mechanism design with respect to the first three desiderata of section 3.1.

- *Pareto Efficiency:* With the reward mechanism defined in (3.18), the historical average market price, P_h^*, reflects how the majority of bidders value a given advertisement slot and this price becomes the expected equilibrium price. With such a reward mechanism, each bidder iterates itself to the corresponding expected equilibrium price. Therefore, the market has a tendency to converge to the equilibrium. With the market tending to equilibrium, the second term in reward mechanism (3.18) tends to zero. Therefore, this mechanism tends to be the same as mechanism (3.17), which is the ideal Pareto efficient mechanism.
- *Social Welfare Maximization:* With the market tending to equilibrium, reward mechanism (3.18) tends to be the same as mechanism (3.17). Thus, this reward mechanism tends to reward all user selected recommendations in a manner that is proportional to their UPQ. Therefore, (3.18) maximizes social welfare and is the most socially-preferred.
- *Individual Rationality:* According to the analysis in section 3.4, the market mechanism produces individually rational dominant strategies for the cases in which the bids are not shortlisted, shortlisted but not rewarded and rewarded. Thus, all the agents bid rationally by taking the dominant strategy accordingly.

3.7 Summary

Based on the above discussions, the generalized first-price sealed-bid auction protocol was adopted and the PRM reward mechanism was found to be appropriate for our context. Specifically, this reward mechanism is capable of incentivising the recommender agents to bid in a manner that aligns their recommendations with those that are likely to satisfy the user. And, in turn, based on the reward received and the price paid, the agents can build up their bidding strategies in terms of individual rationality and stability so as to maximise their revenue. As the marketplace dynamically chases the equilibrium, the frequently shortlisted recommendations adjust their prices to some suitable levels to maximise the revenue and thus the higher the UPQ a recommendation gains the higher the price is. Therefore, the frequently shortlisted recommendations tend to bid in the decreasing order of the UPQ when the market tends to equilibrium. In this way, the marketplace works as a black box in controlling a recommender system that has a large number of raw input items and a small number of desirable output items. Moreover, the marketplace has been shown to be capable of correlating the UPQs with the recommenders' INQs of the recommendations.

This chapter contributes to the thesis in the following manner: an auction protocol, reward mechanism and a set of bidding strategies are developed to assist the market-based recommendations in this chapter. The reward mechanism is proved to be Pareto optimal and social welfare maximizing. It is able to give clear incentives of user preferences to recommender agents, relate the recommenders' bids to the UPQs of their recommendations and shortlist recommendations in decreasing order of their UPQs. Compared to other work in the literature of information filtering and recommender systems (content-based, collaborative, demographic and hybrid filtering techniques), our approach is the first attempt to automate the coordination of multiple different techniques.

With the auction mechanism design in place, a number of experiments are needed to evaluate the operational characteristics of the marketplace so that its practicality as a controller for a recommender system can be ascertained. This is the subject of Chapter 4.

Chapter 4

Simulating and Evaluating the Marketplace

This chapter reports on the simulation experiments to evaluate the market mechanisms designed for our recommender system in Chapter 3 with respect to the last five criteria described in section 3.1.

This chapter contributes to the thesis in the following manner. The simulations demonstrate that *(i)* the theoretical design of the market mechanism in Chapter 3 is viable to coordinate recommendations; *(ii)* the marketplace does indeed work as a black box that gives clear incentives of users' interests to the recommender agents; *(iii)* the marketplace is able to correlate the UPQs of recommendations to the recommender agents' INQs; *(iv)* the recommender agents are able to maximize their revenue by bidding with the feedback from the marketplace that reflects a user's interests; and *(v)* the marketplace is stable in that it can avoid aggressive bidders dominating the marketplace and it is fair to all agents that bid their recommendations.

In more detail, the experimental settings are discussed in section 4.1. The evaluations of the marketplace are then presented in section 4.2. Section 4.3 evaluates the market properties and the correlation between the UPQ and the INQ in more general cases when multiple features of recommendations are considered. Section 4.4 evaluates the system's ability to seek out the recommendation with the highest UPQ value from all bids and recommend it to the user.

4.1 Experimental Settings

As per Figure 1.2, our system is composed of three kinds of agents: the auctioneer agent, the recommender agents and the user agent, which will be dealt with in sections 4.1.1, 4.1.2 and 4.1.3 respectively. Before we discuss these agents, however, an important system variable, the number of bids called for, M (defined in section 3.2), needs to be decided. Here we use the value of ten (because our previous study showed this is the number of items that can be managed efficiently in the browser's sidebar [Moreau et al., 2002]).

4.1.1 Configuring the Auctioneer Agent

The auctioneer agent determines the reward paid to the agents who make recommendations selected by the user. Given that the rewarded mechanism is defined in formula (3.18), two system variables control the auctioneer agent: δ and α (defined in section 3.3.4). From the reward mechanism, we can see that δ affects the volume of the credit paid to a particular user-selected recommendation. The bigger δ is, the more the recommendation is paid. We can also see that α affects the sensitivity of the incentives the marketplace delivers to the recommender agents to make them aware of the equilibrium (because the recommender agents need large alterations to chase the equilibrium price if α is big). In our experiment, we set $\delta = 1.5$ and $\alpha = 1.5$ (based on our experience that these values enable the recommender agents to both increase their revenue by making good recommendations over the long term and chase the equilibrium quickly [Wei et al., 2003a]).

4.1.2 Configuring the Recommender Agents

In this subsection, we discuss how a recommender agent generates a bid and how it relates the bidding price to its INQ for a recommendation. Before delving into this discussion, however, the number of recommender agents contained in our system needs to be defined. We assign this system variable (see S defined in Figure 3.1) a value of nine (to ensure there is a sufficient number of input recommendations and sufficient competition in the marketplace). This value is chosen for experimental expediency and, in practice, it would depend on how many actual recommender agents participate in the marketplace.

Each agent has a set of recommendations available to suggest (typically ordered according to their INQs). Each such agent needs to compute the relation between its local perception of relevance and the user's preference. Having done this, it can then bid an appropriate price to maximize its revenue. Thus, the agent will relate its bidding price to its knowledge about the UPQ (reflected by the rewards it has received) with respect to different INQ levels. We term this relationship between the bidding price and the INQ an agent's *strategy profile*. This profile is on a per agent basis. It records an agent's bidding price for different INQ levels and indicates how an agent should relate its bid to its INQ.

Simulating Recommendation Methods. To assess the broad feasibility of our market-based approach, we want our representation of the INQs to be capable of corresponding to as many recommendation techniques as possible. Moreover, we do not want our results to be skewed by any innate bias in the recommendation methods themselves. Therefore, we take an abstract view on the recommender methods and view them simply as being able to learn a user's interests based on their internal belief about certain recommendation properties (features or attributes) that the user's context focuses on. We believe this is a reasonable abstraction because a recommendation method's ability to adaptively match certain recommendation properties to the user's actual preferences has been shown to be crucial to making high quality recommendations [Claypool et al., 1999] (see Chapter 6 for three real recommendation methods developed for user evaluations). Given these observations, we define the INQ of a specific recommendation method to be the sum of the weighted evaluation scores made of different techniques on different properties of a recommendation (see equation (4.1)). This is consistent with the observation that effective recommendation methods need to combine filtering techniques based on different recommendation properties to achieve peak performance [Burke, 2002]. To this end, we simulate the recommendation methods' INQs on a linear basis[1]. More formally,

$$q(Rec) = k_1 \cdot \Phi_1(Rec) + k_2 \cdot \Phi_2(Rec) + \cdots + k_I \cdot \Phi_I(Rec) \quad (I > 0) \tag{4.1}$$

where $q(Rec)$ represents the INQ of item Rec based on a specific method. This method evaluates an item from I perspectives (i.e. properties, features or attributes). The value of I is on a per method basis because different methods evaluate different numbers of properties of an item. Here, each $\Phi_i(Rec)$ ($i \in [1..I]$) represents the evaluation function based on a specific property of Rec ($\Phi_i(Rec) \in [0, 1.0]$)[2]. Such properties can be either objective (such as the TFIDF [Salton, 1989] of a document), subjective (such as customers' opinions of the tastes of the foods in a restaurant) or a mixture of the two (such as users' opinions of the textual and graphical descriptions of the products of a store). Note that different evaluation functions might evaluate the same property of a recommendation but from different perspectives. For example, Φ_i and Φ_j are two functions that evaluate the same property ($\mathbf{x}$) of a recommendation. However, $\Phi_i = \sin(\mathbf{x})$ whereas $\Phi_j = \cos(\mathbf{x})$. Variable k_i ($k_i > 0$) specifies the weight of $\Phi_i(Rec)$ and $k_1 + k_2 + \cdots + k_I = 1.0$ in order to ensure $0 \leqslant q \leqslant 1.0$.

For example, consider the case where the user's browsing context is local restaurants. In this situation, an individual recommendation method might base its INQ on the TFIDF of an online restaurant menu with a value between 0 to 1.0, other people's opinions of the food on the restaurant's website with an integer voting value of 1 .. 5 (normalization will be used), whether the user has ever consumed the service of the current restaurant with a binary value of 0 or 1, or any other possible properties of the item. In our case, each of these corresponds to a specific $\Phi_i(Rec)$ and if a particular method uses a combination of these base terms then appropriate values for the respective k_i's would be set.

The next step is to determine how to simulate Φ. Based on our previous studies in this area, by randomly collecting 400 different Web pages on the subject of "world news", we

[1]This linear combination is used by several hybrid recommender systems in combining results from different recommendation methods (see section 2.1.4). Through combining different weighted properties or features, it is believed that a recommendation method can improve its precision in predicting the user's preference and improve its quality of recommendations [Pazzani, 1999, Yu et al., 2003].

[2]When evaluating different recommendation methods, we perform a normalization on the results to fix them into a range of $[0, 1.0]$. This is because different recommendation methods have different quality (or rating) ranges [Pennock et al., 2000]. This can be achieved in practice by adaptively matching a method's min and max INQ value onto 0 and 1.0 respectively. This makes the values from different methods meaningful in our market based recommender system in terms of INQ and UPQ.

find that the keyword similarity [Moreau et al., 2002] of the 400 documents compared to CNN's frontpage (`http://www.cnn.com`) follows a Gaussian normal distribution (see the contour of the distribution in Figure 4.1(a)). Hence, we decide to use some Gaussian normal distributions to model the valuations of properties (Φ) of recommendations in predicting user's preferences on a probabilistic basis [Popescul et al., 2001, Sharma and Poole, 2001]. Specifically, in our experiments, we simulate different document properties of one method by different random variables that follow different normal distributions. The probability density function of the normal distribution is defined as[3]:

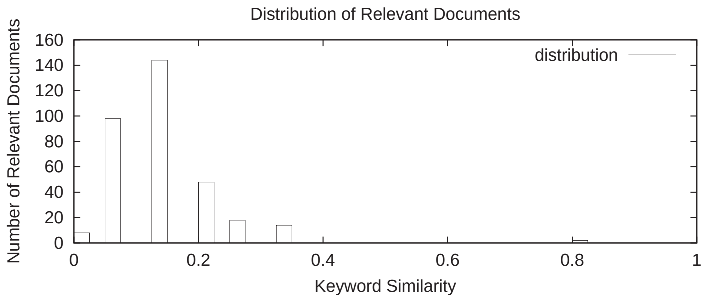
$$N(\mu, \sigma^2): \quad f(q) = \frac{1}{\sqrt{2\pi}\ \sigma}\ e^{-\frac{(q-\mu)^2}{2\sigma^2}}, \qquad q \in [0, 1.0] \tag{4.2}$$

where μ and σ are the mean value and the standard deviation of the random samples (see Figure 4.1(b)). The mean of the distribution represents the average value of the INQs of all samples generated by the corresponding method. The middle range (between one unit of standard deviation on both sides of the mean) of the distribution contains the majority of the samples (about 68 percent of its total).

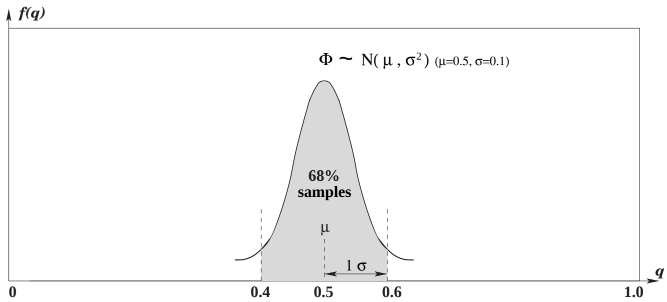
One of the key objectives of the recommender agents is to build up their strategy profiles so that they can relate their bidding price to their INQs based on their knowledge about the reward (which, in turn, reflects the UPQ of the recommendations). In order to learn such characteristics for all INQ levels, each agent divides its strategy profile into 20 continuous segments. In each auction, a recommender agent needs to compute the INQs of ten recommendations and make ten corresponding bids. In the early auction rounds, all the agents' strategy profiles are empty. With an empty strategy profile, an agent will bid proportionally (because it can only expect a high INQ recommendation to receive a high UPQ and, consequently, more reward than a low INQ recommendation) to the INQ of ten (value of M defined in section 3.2) recommendations based on an initial seeding price. We set different initial seeding price values (randomly generated from the range [128, 256][4]) for different recommender agents (because different agents value their recommendations differently with their empty strategy profiles). After each auction, all strategy profile segments record and update information about: the last bid

[3] We fix the sample into the range [0, 1.0] (rather than ($-\infty$, $+\infty$)) since we have manipulated the INQ into this range (see equation (4.1)).

[4] The exact values of the boundary of the range are not important. What matters is whether such a randomly given range can make the market converge and exhibit the other properties specified in section 3.1.



(a) The Contour of the Relevant Documents Distribution



(b) Distribution of a Hypothetical Property of a Set of Recommendations

FIGURE 4.1: Simulating Evaluation Technique

status (not shortlisted, shortlisted but not rewarded, or rewarded), the last bid price, the last rewarded price and the last rewarded profit. Based on such information about each segment, and using the appropriate bidding strategy, an agent can compute its bids in subsequent auctions if there are recommendations that belong to this segment. After a number of iterations, those segments that cover the majority of samples will have sufficient information to reach the equilibrium price and form a stable strategy profile.

4.1.3 Configuring the User Agent

Again in seeking to evaluate the principle of a market-based approach to recommendation, we want to work in a well controlled environment. Thus we simulate the users of our recommender system (as others have done when seeking to validate the principle of a new method [Billsus and Pazzani, 2000, Bohte et al., 2004, Gonzalez et al., 2004]). Specifically, when a user is faced with a set of shortlisted recommendations, he will visit some of the recommendations and will then have a valuation of each visited item. Thus, a user assigns a number, Q_i ($Q_i \in [0..100]$, $i \in [1..M]$), to each visited item according

TABLE 4.1: User's Decision of Different Models

Shortlisted Recommendations	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
User Perceived Quality	70	50	75	30	60	82	90	85	65	55
Decision of Independent Selection	1	0	1	0	1	1	1	1	1	0
Decision of Search-Till-Satisfied	1	0	1	0	1	1	0	0	0	0

Both models have the same AT of 60. Search-till-satisfied model has a ST of 80. "1" means the recommendation is selected to be rewarded, while "0" means not selected.

to his valuation of the recommendation. This number Q_i is the UPQ value. To simulate the choices of a user in selecting recommendations, we deploy a user model inside the user agent. Building on the user simulation of Bohte *et al.* (see section 2.5), we adopt the following models:

- *Independent Selection:* The selection of one recommendation is independent of the others. Once the UPQ of a recommendation is greater than or equal to a particular *acceptance threshold* (AT), the recommendation is accepted and rewarded. Those recommendations with UPQ less than AT will not be selected and therefore receive no reward.

- *Search-Till-Satisfied Behaviour:* The selection of one recommendation is dependent on other recommendations that are ranked above it in the list. In this case, the user stops searching once he discovers a recommendation that has a UPQ greater than or equal to a particular *satisfaction threshold* (ST).

By means of an illustration, Table 4.1 is an example of a user's decision under the two different models. All recommendations with UPQ above the AT (60) are selected to be rewarded in the case of independent selection. However, Q_7, Q_8 and Q_9 are not selected to be rewarded by the search-till-satisfied behaviour though their UPQs are above the AT. Indeed, the user stops searching since a document with a quality of 82 (Q_6) has been found above the ST (80).

We simulate the user by a user agent which knows its valuation for each recommendation and assigns the UPQ based on this valuation correspondingly. Thus, when a real user considers I' ($I' > 0$) evaluation perspectives of a recommendation (Rec), the UPQ of Rec

is defined as:

$$Q(Rec) = k'_1 \cdot \Phi'_1(Rec) + k'_2 \cdot \Phi'_2(Rec) + \cdots + k'_{I'} \cdot \Phi'_{I'}(Rec) \tag{4.3}$$

where $\Phi'_i(Rec)$ (whose definition is equivalent to that of $\Phi_i(Rec)$ in equation (4.1), $\Phi'_i(Rec) \in [0, 1.0]$, $i \in [1..I']$) is the evaluation function based on a specific property of *Rec*. k'_i ($k'_i > 0$, $i \in [1..I']$) is the weight of $\Phi'_i(Rec)$ contributing to $Q(Rec)$. We set $k'_1 + k'_2 + \cdots + k'_{I'} = 100$ to ensure $0 \leqslant Q \leqslant 100$.

4.1.4 Correlating the UPQ to the INQ

From the formal specifications of the UPQ and the INQ of a given item, as given in equations (4.1) and (4.3), it can be seen whether the evaluation functions of the document that the user considers overlap with those that a recommendation method considers. Here, we define the set of evaluation functions $\{\Phi'_1, \Phi'_2, \cdots, \Phi'_{I'}\}$ that the user evaluates as φ_Q. Likewise, we define the set of functions $\{\Phi_1, \Phi_2, \cdots, \Phi_I\}$ that a recommendation method evaluates as φ_q. We define $\varphi = \varphi_Q \cap \varphi_q$ as the recommendation method's *effective factors* in terms of the UPQ. We define $\overline{\varphi} = \varphi_Q - \varphi_q$ as the recommendation method's *ineffective factors*. The variable φ is important, because if $\varphi \neq \phi$ (ϕ stands for "empty set") the method will have some correlation with the UPQ since their evaluations of the recommendation items share some of the same evaluation functions. Otherwise, if $\varphi = \phi$, a recommendation method cannot correlate its INQ to the UPQ since they evaluate the items from totally different perspectives. These issues will be discussed in detail in sections 4.2 and 4.3.

By abstracting all recommendation methods as independent learners that predict user's preferences, all predictions can be seen as composed of effective data and noisy data on a probabilistic basis [Popescul et al., 2001, Sharma and Poole, 2001]. This, in turn, simplifies modeling the market-based constituent recommenders on a high abstraction level. Specifically, by defining a recommendation method's effective and ineffective factors, given a recommendation item *Rec*, its UPQ can be represented in terms of a method's INQ as follows:

$$Q(Rec) = \Gamma(\varphi(Rec)) + \overline{\Gamma}(\overline{\varphi}(Rec)) \tag{4.4}$$

where Γ and $\overline{\Gamma}$ are two mapping functions that align the coefficients of the elements of φ and $\overline{\varphi}$ with the weightings (k'_i) of the evaluation functions (Φ'_i) of Q (see equation (4.3)). For example, assume a user evaluates an item *Rec* from perspectives of Φ_a, Φ_b and Φ_c and the importance of these functions are k'_a, k'_b and k'_c respectively ($k'_a + k'_b + k'_c = 100$ and $k'_a, k'_b, k'_c > 0$), the UPQ will be $Q = k'_a\Phi_a + k'_b\Phi_b + k'_c\Phi_c$. Assuming a recommendation method evaluates the item from perspectives of Φ_a, Φ_b and Φ_d and their relative importance is k_a, k_b and k_d respectively ($k_a, k_b, k_d > 0$ and $k_a + k_b + k_d = 1.0$), its INQ is $q = k_a\Phi_a + k_b\Phi_b + k_d\Phi_d$. Thus, the INQ's effective factors are $\varphi = \{\Phi_a, \Phi_b\}$ and its ineffective factor is $\overline{\varphi} = \{\Phi_c\}$. Therefore, $\Gamma(\Phi_a, \Phi_b) = (\frac{k'_a}{k_a}\ \frac{k'_b}{k_b}) \times \begin{pmatrix} k_a\Phi_a \\ k_b\Phi_b \end{pmatrix}$ and $\overline{\Gamma}(\Phi_c) = (k'_c) \times (\Phi_c)$. We find that when a recommendation method's effective factors form a major weighting of both its INQ and the UPQ (*e.g.*, in the above example, Φ_a and Φ_b contribute $\frac{k_a+k_b}{k_a+k_b+k_d}$ of the weighting of the INQ and $\frac{k'_a+k'_b}{k'_a+k'_b+k'_c}$ weighting of the UPQ), this method can easily correlate its INQ to the UPQ (see section 4.3 for more details), and can continuously produce good recommendations and make profits. Otherwise, if a method has only ineffective factors, the method cannot correlate its INQ to the UPQ and therefore makes poor recommendations most of the time and will go bankrupt. These properties will be discussed in more detail in section 4.3.

4.2 Evaluation of the Marketplace

Having outlined the setup of the three kinds of agents specified in section 4.1, this section will focus on evaluating the system properties. In our case, the market is the key to coordinating the various recommendation methods. If it does not work effectively, the system will not be able to make good recommendations. Among the five properties we want our market to exhibit, convergence is the most important because it forms the basis of the other four. Therefore, we will start with experiments on market convergence.

4.2.1 Market Convergence

We endow our system with 100 documents ready to be recommended to the user[5]. Every time the user visits a specific recommendation, the UPQ of this recommendation is assigned by the user and this value is independent of the various methods' INQs. To simplify the experiments on evaluating the properties (Φ_i) of a recommendation item, we assume each recommendation method evaluates items from only one perspective (but two different methods may use different perspectives). The more general case with more than one Φ_i involved in each method is dealt with in section 4.3. We further assume the user considers two different properties of the recommendations (Φ_0 and Φ_1). Thus, the effective and ineffective factors of the recommendation methods can be easily controlled[6]. Assuming the weighting of the two properties are k_0 and k_1, respectively, the UPQ can be represented formally as:

$$Q(Rec) = k_0 \cdot \Phi_0(Rec) + k_1 \cdot \Phi_1(Rec) \tag{4.5}$$

To generalize the experiments, nine constituent recommender agents are placed in our marketplace and each of them is based on one of three different properties (Φ_1, Φ_2, Φ_3) of recommendations (note here Φ_1 is the same as in equation (4.5)), meaning that some of the recommendation methods contain the effective factors in terms of the UPQ and some of them do not. We will use three Gaussian normal distribution functions (see equation (4.2)) to simulate the valuations of the three properties. Each property relates to one of the three distributions: $N(0.35, 0.1^2)$, $N(0.5, 0.1^2)$ and $N(0.65, 0.1^2)$ (see Figure 4.2). We set the standard deviation to a value of 0.1, meaning the three different properties share only a small intersection (so as to easily differentiate the different methods). Thus, those methods' INQs based on Φ_1 can be presented formally as:

$$q_i(Rec) = \Phi_1(Rec) \qquad (i \in [1..3]) \tag{4.6}$$

[5]The working scenario and the configurations of the UPQ and the INQ in this section will be used for all experiments in section 4.2.

[6]We can exemplify this case in a scenario where the user is browsing the local restaurants on the Web. We assume the user evaluates the recommended restaurant websites from two perspectives: whether the restaurant sells some specific foods (Φ_0) and other customers' opinions of the foods in the restaurant (Φ_1). If a recommendation method also computes Φ_1, then Φ_1 is its effective factor and Φ_0 is its ineffective factor in terms of the UPQ (mutatis mutandis if the method computes Φ_0). If a recommendation method evaluates the recommendations by Φ_x (which is different from Φ_1 and Φ_0), then it has no effective factors.

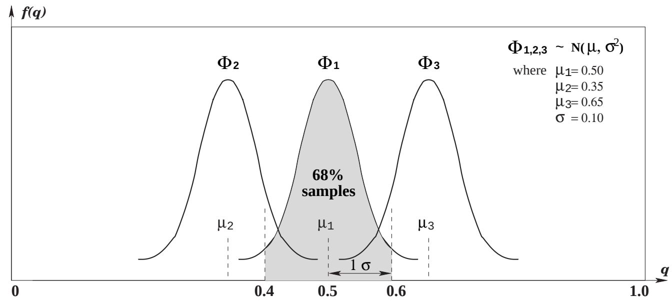


FIGURE 4.2: Distributions of Three Properties of a set of Recommendations

In this case, the UPQ (equation (4.5)) can be represented in terms of the INQ which contains the effective factors:

$$Q(Rec) = k_1 \cdot q_i(Rec) + k_0 \cdot \Phi_0(Rec) \qquad (i \in [1..3]) \tag{4.7}$$

Having further configured the experimental settings, we are going to examine the system property from the perspective of market convergence. In section 3.5, we showed that the marketplace can reach an equilibrium such that the shortlist prices converge at different levels with respect to different UPQ levels. To evaluate this, we arranged 300 auctions with 10 shortlisted recommendations using the independent selection user model ($AT = 66$) and ($k_1 = 75$, $k_0 = 25$)[7] for equations (4.5) and (4.7) to see if the marketplace does indeed have such a convergence property. We organized three groups of experiments, each of which contains a different number of agents having the effective factors, to see whether the market converges in various cases. The configurations are shown in Table 4.2.

TABLE 4.2: Configurations of the Three Groups of Experiments

Experiments	**Configurations**
Experiment 1	$q_i(Rec) = \Phi_1(Rec)$ $(i \in [1..3])$ and $q_j(Rec) = \Phi_2(Rec)$ $(j \in [4..6])$ and $q_k(Rec) = \Phi_3(Rec)$ $(k \in [7..9])$
Experiment 2	$q_i(Rec) = \Phi_1(Rec)$ $(i \in [1..9])$
Experiment 3	$q_1(Rec) = \Phi_1(Rec)$ and $q_j(Rec) = \Phi_2(Rec)$ $(j \in [2..5])$ and $q_k(Rec) = \Phi_3(Rec)$ $(k \in [6..9])$

[7] k_1 and k_0 can be set to any other combinations in these experiments. 75 and 25 are chosen to exhibit the higher importance of Φ_1 than Φ_0.

In the first experiment, each of the three properties is shared by three agents; thus only the first three agents contain the effective factor, whereas the other six do not[8]. From Figure 4.3(a), we can see that the shortlisted prices converge (for example, the 4^{th} and 10^{th} bid oscillate around 150 and 130 respectively, which indicate P_4^* and P_{10}^* respectively) after about 100 auctions. We find that, with the search-till-satisfied user model (with $ST = 60$ and $AT = 45$), the market also converges (for which we do not provide a figure), but only after a longer time (more auction rounds) compared to the independent selection. This takes longer because fewer agents are rewarded in this case and they need more bids to chase the equilibrium price.

In the second experiment, all nine agents evaluate recommendations by property Φ_1. In this case, the market converges very quickly (after about 30 auctions, see Figure 4.3(b)), because all agents' INQs are actually the effective factors in terms of the UPQ. Thus they have a good correlation with the user's valuation of the recommendations. Therefore, more recommendations at each quality level can be related to the UPQ and the agents receive more signals of the user's interests. This, in turn, means agents get sufficient chances to alter their price effectively to chase the equilibrium price with respect to each UPQ level. This results in a market that converges quickly.

In the third experiment, only the first agent evaluates Φ_1 and the other eight agents evaluate Φ_2 or Φ_3. The market still converges but very slowly (after about 600 auctions, see Figure 4.3(c)) with the first bid price oscillating around 125. This slow speed can be accounted for by the fact that only one agent can relate its good recommendations' bidding price to its INQ with respect to each UPQ level and there are insufficient good recommendations. Therefore, the agent needs a longer time to get a sufficient number of high quality recommendations to be rewarded and to chase the equilibrium price. In this experiment with very few agents taking the effective factor in terms of UPQ, it is interesting to see that the 10^{th} bid price decreases till it reaches zero (see Figure 4.3(c))[9].

[8]In this case, the first three agents can relate their bidding price to their INQs, since their INQs have a relationship with the UPQ (contributing 75% of its total weighting, see equation (4.7)). Also the rewards they received reflect the UPQ with respect to a specific recommendation. The remaining six agents cannot relate their bids to their INQs because their INQs have no relationship with the UPQ and their rewards. This subject will be discussed further in section 4.2.3.

[9]Actually only the first and second bid prices converge in this experiment. The second bid is not plotted in Figure 4.3(c) because it is close to the first bid and we want to clearly display the convergence. The other eight bids, the $3^{rd} \sim 10^{th}$, do not converge to a positive level and decrease continuously till reaching zero (for the same reason only the 10^{th} bid is plotted).

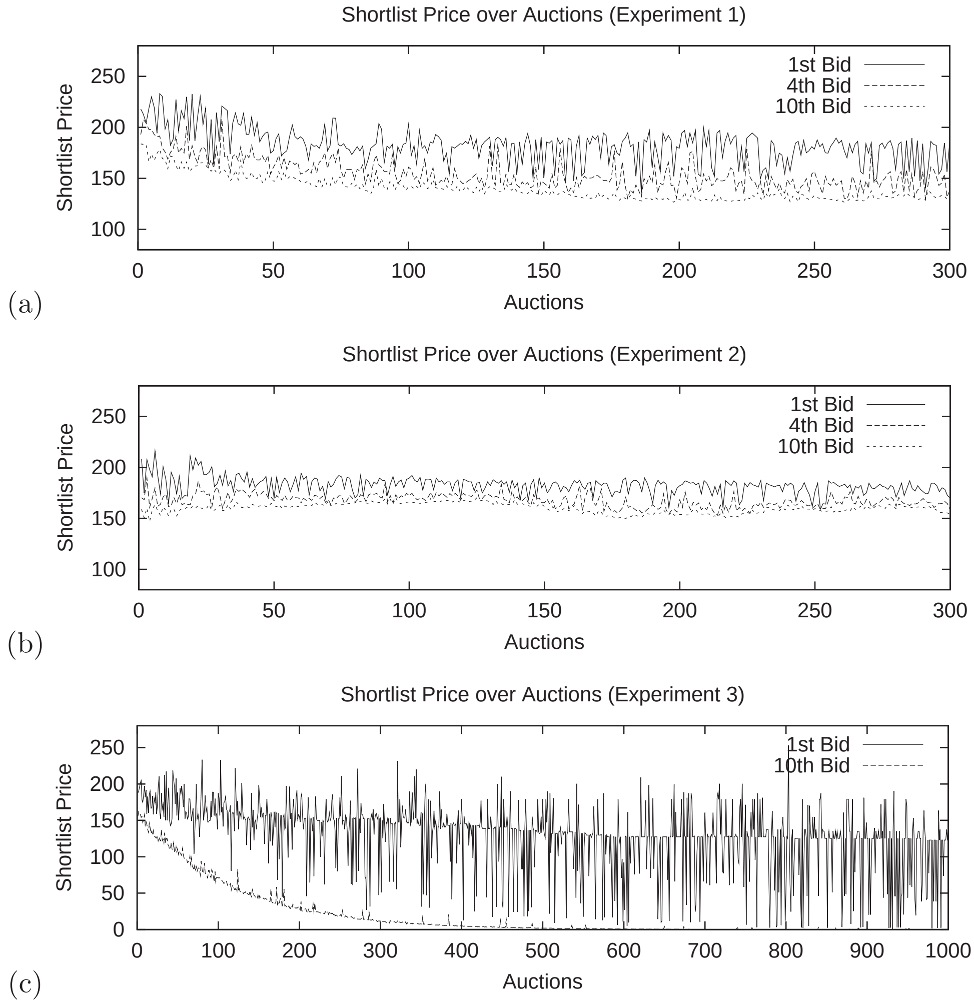


FIGURE 4.3: Convergence of Shortlist Prices

The explanation is that most of the recommendations, from the eight agents with only ineffective factors as their INQs, cannot relate their bidding price to their INQs. Thus, these agents cannot reason about the relationship between the rewards and the INQs of the rewarded recommendations (since the rewards are based on the UPQ, *not* on the INQ). Therefore, the equilibrium price for such bids (if it exists) has no relationship with the INQ. Such a recommender agent cannot chase the equilibrium price based on the INQ. Such shortlisted (both rewarded and not-rewarded) recommendations will make loss most of the time. Hence, most of the recommendations will bid as low as possible to reduce their loss (this phenomenon continues till the bid price reaches zero meaning paying nothing). The exception to this is the small number of bids from the only agent with the effective factors. Overall, this experiment demonstrates that the marketplace

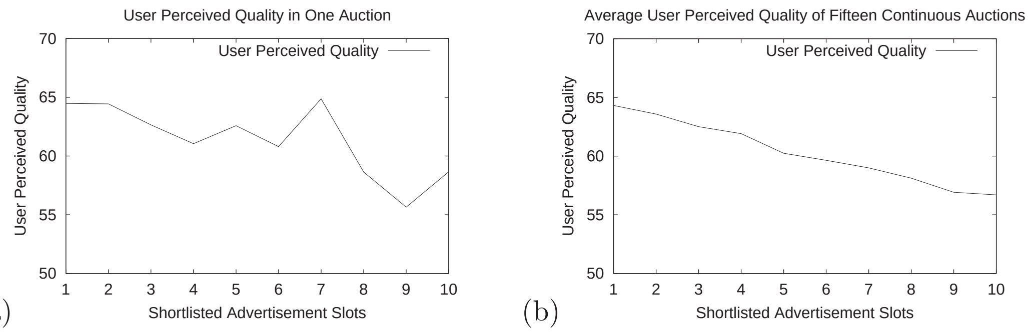


Figure 4.4: The UPQ of Shortlisted Recommendations (Experiment 1)

deters bad recommendations and only good recommendations can pass through.

When all the experiments are taken together, we find that the shortlisted prices always converge after a number of iterations as long as there is at least one agent that has effective factors. The speed of the convergence depends on the setting of the parameters α, AT, ST, Y and Z. Since these variables are not our main concern here, we only overview their effects. Broadly speaking, AT and ST affect the number of recommendations being rewarded (because more agents are rewarded if their values are small). By being rewarded more times, an agent receives more information and therefore can chase the equilibrium faster. The variables Y and Z also affect the speed with which the agent can chase the equilibrium. Specifically, with high values of these variables, an agent alters its price quickly to reach the equilibrium price.

4.2.2 Efficient Shortlists

The most important feature of our system is its capability to shortlist the best recommendations in decreasing order of UPQ when the market converges. To this end, Figure 4.4(a) shows the UPQ of the shortlisted recommendations during the 100^{th} auction (which is after convergence) in the first experiment introduced in section 4.2.1. Here, we can see that the quality of the ten shortlisted recommendations has an overall tendency to decrease in most cases (although there are some exceptions). Figure 4.4(b) shows the average UPQ of fifteen continuous auctions after convergence (from the 101^{st} to the 115^{th} auction). By averaging over these auctions, we can see that the UPQ decreases monotonically. Thus, Figure 4.4 tells us that our market mechanism is indeed capable of shortlisting the best recommendations in decreasing order of UPQ. Through various experiments stated in section 4.2.1, we find that our market can always do so and our results hold more broadly than just for this specific experiment.

4.2.3 Clear Incentives

The next step is to see if the recommender agents can relate their bids to the INQs of their recommendations (meaning an agent can generate a steady strategy profile). In this case, each recommender agent builds up its strategy profile from its knowledge about the bids with respect to its 20 INQ segments. Specifically, Figure 4.5(a) shows the bidding prices for different segments of the first recommender agent with the effective factors Φ_1 as its INQ. From Figure 4.5(a), we can see that this agent's bidding prices for different INQ segments oscillate around certain levels after the market reaches the equilibrium (after about 100 auctions). Figure 4.5(b) shows the agent's strategy profile (equilibrium bidding price versus the INQ segments) and that higher INQ does indeed relate to higher bidding price. Indeed, this agent evaluates its INQ on the effective factors, in particular, on those that have a high weighting in the UPQ (see equations (4.5) and (4.7)). Thus, the agent can relate its bidding price to its INQ in such a way that the higher the INQ, the higher the corresponding UPQ, and the higher the bidding price. In this way, the agent maximizes its revenue. Figure 4.5(c) shows the bidding prices for different segments of the seventh agent with the ineffective factor Φ_3 as its INQ and Figure 4.5(d) depicts this agent's strategy profile (which shows there is no relationship between the bidding price and the INQ). From figures 4.5(c) and (d), we can see that this agent's bidding prices do not reach equilibrium (because the agent has only ineffective factors as its INQ). Therefore, it cannot relate its bids to its INQ, because it cannot reason about the relationship between the occasional rewards and the INQs of the rewarded recommendations. Since high INQ does not indicate high UPQ in this case, the UPQ with respect to a specific INQ segment can vary dramatically. Therefore, based on the UPQ, the rewards with respect to a specific INQ level do not converge (meaning that the agent can learn nothing from the marketplace). Hence, based on the rewards (see the relationship between the reward and the bidding price in equation (3.18)), the bidding prices with respect to this INQ level do not converge. Thus, the agent cannot build up a practical strategy profile after the market converges. Agents with ineffective factor Φ_2

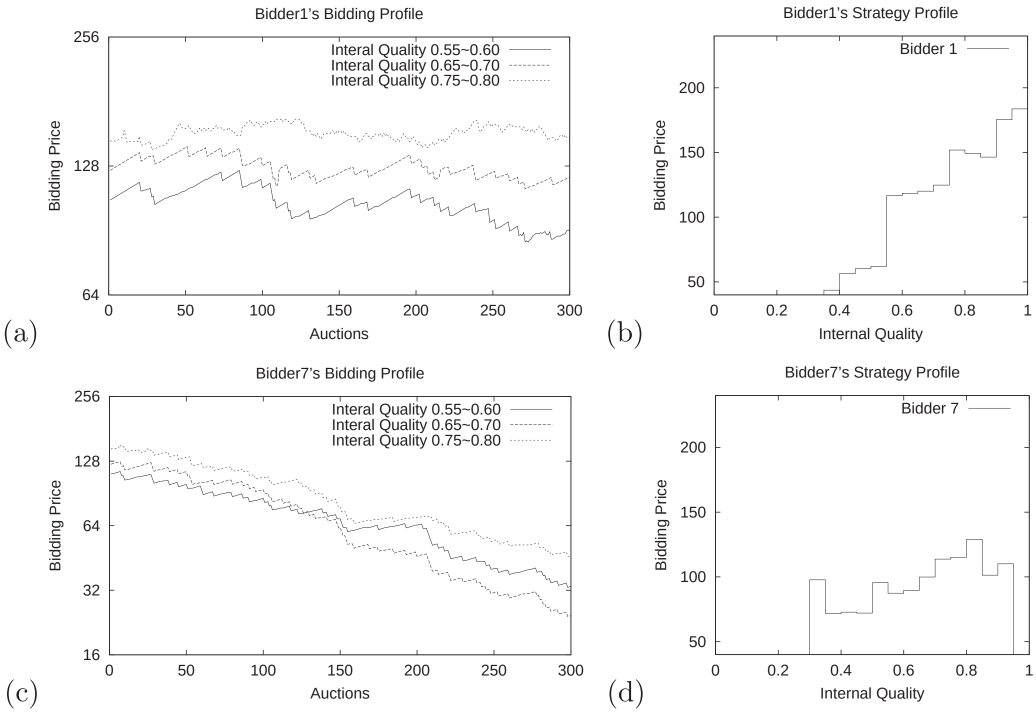


FIGURE 4.5: Bidding Profile and Strategy Profile of Bidders with Effective and Ineffective Factors
(Experiment 1)

exhibit the same properties as those agents with Φ_3 and we do not comment further on them.

In addition to the bidding strategy profile, we also examined the revenue and the number of times these agents won in the auctions. From Figure 4.6(a), we can see that the first three agents, with the effective factors, win more times than the remaining six agents (that have ineffective factors). Figure 4.6(b) shows that the first three agents can make profits whereas the other six make a loss over time. Indeed, the agents with ineffective factors always bid high enough to be shortlisted (see section 4.2.4 for more information about equal opportunities of being shortlisted), but they are not able to learn anything from the few occasional rewards that they receive. Thus, these agents pay more when shortlisted than they earn when rewarded[10].

When taken together, figures 4.5 and 4.6 indicate that the agents with effective factors in terms of UPQ are capable of "learning" from the marketplace to alter their bids to certain

[10] The rational bidding strategy for those agents who cannot learn anything from the market is to bid as low as possible to lose less money, see Figure 4.3(c) and its explanation.

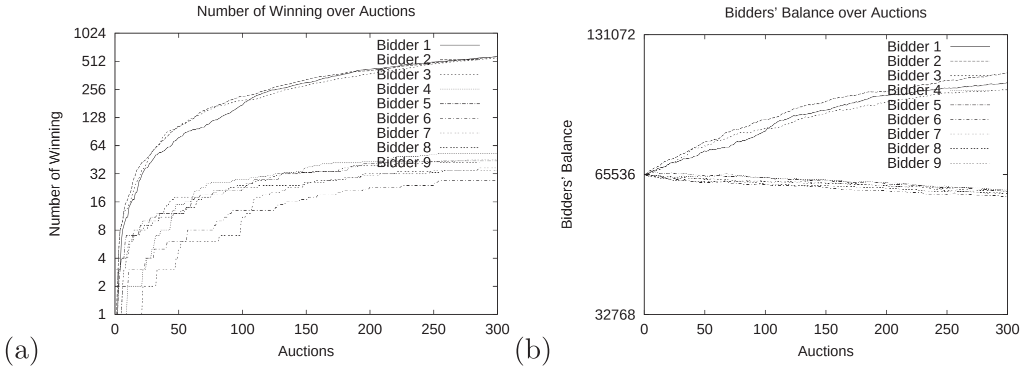


FIGURE 4.6: Number of Winning and Bidders' Balance of Bidders with Effective and Ineffective Factors
(Experiment 1)

levels in order to chase the equilibrium price. This, in turn, results in a maximization of their revenue (see Chapter 5 for more details about learning). In contrast, agents with ineffective factors are not capable of learning from the market. From our observation of the various simulations, with good correlations to the UPQ, a recommender agent's strategy profile changes quickly before the market converges and then becomes relatively stable after convergence.

4.2.4 Fairness

We expect the market to be fair to all recommender agents irrespective of the recommendation method they use. To see this, we use the first experiment configuration introduced in section 4.2.1. From Figure 4.7, it can be seen that the curves that represent the number of recommendations being shortlisted (including both rewarded and not rewarded) for each agent are closer to each other compared to the number of recommendations being rewarded as shown in Figure 4.6(a). This means that all agents have equal opportunities of suggesting their recommendations. Thus, the market is fair to all agents whatever methods they use.

However, different methods do not necessarily have an equal opportunity of being rewarded as shown in Figure 4.6(a). This, in turn, highlights the fact that a fair market does not mean that all agents are equally likely to receive reward. Rather, the opportunity of being rewarded depends on the UPQ of the recommendations.

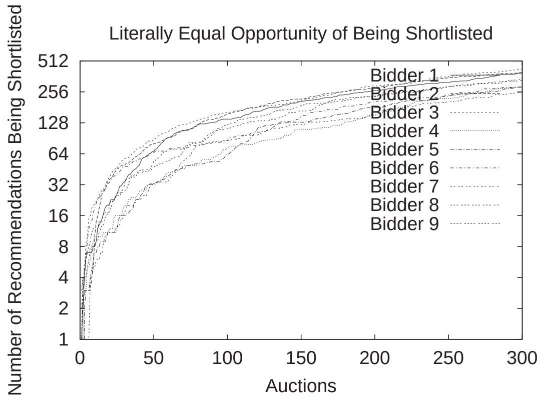


FIGURE 4.7: Opportunity of Being Shortlisted (Experiment 1)

4.2.5 Stability

To evaluate the stability of the market with respect to bidding strategies, we now consider what happens if some of the agents no longer follow the dominant strategies of section 3.4. Here we assume the agents adopt a greedy strategy meaning they bid as much as possible on every round to outbid others. To this end, we use the setting of the second experiment introduced in section 4.2.1 with all nine agents taking the effective factors as their INQs. However, we select one recommender agent (say the first one) as the greedy bidder and the other agents still take the dominant strategy. Here, all recommender agents are endowed with an initial credit of 65535. The greedy bidder always bids much higher than the others to get its recommendations shortlisted with the hope of making profit. However, this greedy bidder does not receive any more rewards from its recommendations when compared with the rewarded recommendations provided by the other non-greedy bidders. Indeed, the reward is not based on the bid price, but rather on the UPQ (for exactly this reason). With the same amount of reward with respect to the same level of UPQ, however, the greedy bidder pays much more for each of its shortlisted recommendations. Therefore, the greedy bidder goes bankrupt over time (see "Bidder 1" in Figure 4.8(a)), while the other non-greedy bidders keep increasing their balance steadily. In comparison, when no greedy bidders participate, all recommender agents keep increasing their balance as shown in Figure 4.8(b).

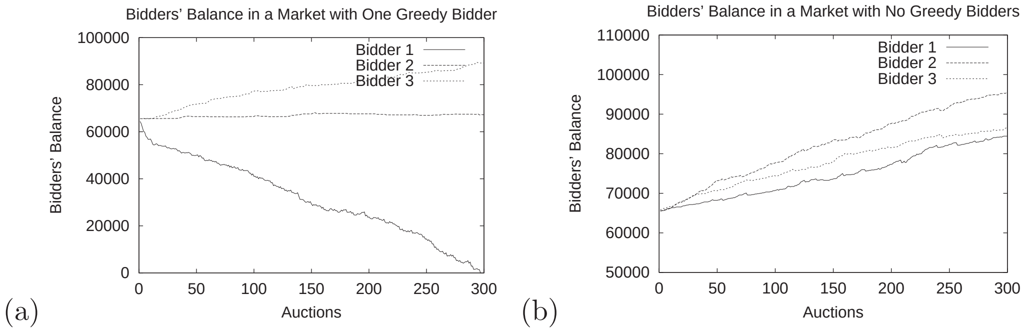


FIGURE 4.8: Balance of Bidders with Effective and Ineffective Factors

4.3 Dealing with Multiple Recommendation Properties

Having evaluated the system properties with respect to the metrics stated in section 3.1, this section considers the case where more than one recommendation property (Φ introduced in section 4.1.2) is evaluated by both the user and the recommender agents. This is important because many real recommendation methods evaluate more than one property (or feature) of recommendations [Burke, 2002, Littlestone and Warmuth, 1994] and it is important that our market-based recommender system performs well in such cases.

However, first, we need to establish the configurations of the three kinds of agents in our marketplace. Since the auctioneer agent simply acts as the organizer of the marketplace, rewarding the user-selected recommendations based on the UPQ, this agent remains the same as in section 4.1.1. We still use the independent selection user model with $AT = 66$. Since it is not practical to gather up every possible case that contains an arbitrary number of properties (Φ) in one formula (for either UPQ or INQ) and to exemplify the correlations between these two qualities in a simple set of experiments, we begin the analysis with two properties involved for each quality function (both the user and the recommender agents). The more general cases in which each quality function evaluates more than two properties can be analyzed in the same way. To this end, the configuration of the user agent also remains unchanged, $Q(Rec) = 75\Phi_1(Rec) + 25\Phi_0(Rec)$. The recommender agents are each configured to evaluate two properties: some agents share both properties, some share only one property, and some share no property with the user's valuation of the recommendations. In this section, we consider eight recommender agents and their INQs are configured as follows:

$$\begin{aligned} q_1(Rec) = q_5(Rec) &= 0.75\Phi_1(Rec) + 0.25\Phi_0(Rec) \\ q_2(Rec) = q_6(Rec) &= 0.75\Phi_1(Rec) + 0.25\Phi_3(Rec) \\ q_3(Rec) = q_7(Rec) &= 0.75\Phi_3(Rec) + 0.25\Phi_0(Rec) \\ q_4(Rec) = q_8(Rec) &= 0.75\Phi_2(Rec) + 0.25\Phi_3(Rec) \end{aligned} \tag{4.8}$$

Φ_0, Φ_1, Φ_2 and Φ_3 are configured as per section 4.2.1. With these settings, we can see that q_1 and q_5 fully contain the effective factors and they match the user's valuation of recommendations accurately. Likewise, q_2, q_6, q_3 and q_7 partially match the user's valuation, whereas q_4 and q_8 have no match. More formally, using a transformation of the UPQ, $Q(Rec) = (75\Phi_1(Rec) + 25\Phi_0(Rec))/100$, to subtract each item in equation array (4.8), we can expect the four methods to exhibit the following correlations to the UPQ (where"$\ncong$" stands for "has no relationship to"):

$$\begin{aligned} & q_1(Rec) = q_5(Rec) = 0.01 \cdot Q(Rec) \\ & q_2(Rec) = q_6(Rec) = 0.01 \cdot Q(Rec) + 0.25 \cdot (\,\Phi_3(Rec) - \Phi_0(Rec)\,) \\ & q_3(Rec) = q_7(Rec) = 0.01 \cdot Q(Rec) + 0.75 \cdot (\,\Phi_3(Rec) - \Phi_1(Rec)\,) \\ & q_4(Rec) = q_8(Rec) \ncong Q(Rec) \end{aligned} \tag{4.8$'$}$$

Having configured the three kinds of agents, we are going to evaluate the market properties and validate that the correlations in equation array (4.8′) do effect the agents' bidding and learning behaviour. Again, the evaluation begins with the most important system property — market convergence. Figure 4.9 again demonstrates that the market converges (after about 80 auctions) with at least one agent capable of relating its INQ to the UPQ (the first and the fifth agents in this experiment).

Using similar simulations to the ones of section 4.2, we find that the market exhibits the same properties: namely efficient shortlists, clear incentives for agents to bid, stability and fairness. Thus, we do not further discuss these issues in this section. Instead, we will focus on how the different recommendation methods correlate their INQs to the UPQ. This problem can be decomposed into two subproblems:

(i) Can the agents relate their bids to their internal quality?

(ii) To what extent does each individual agent relate its INQ *to the* UPQ*?*

FIGURE 4.9: Convergence of Shortlisted Prices

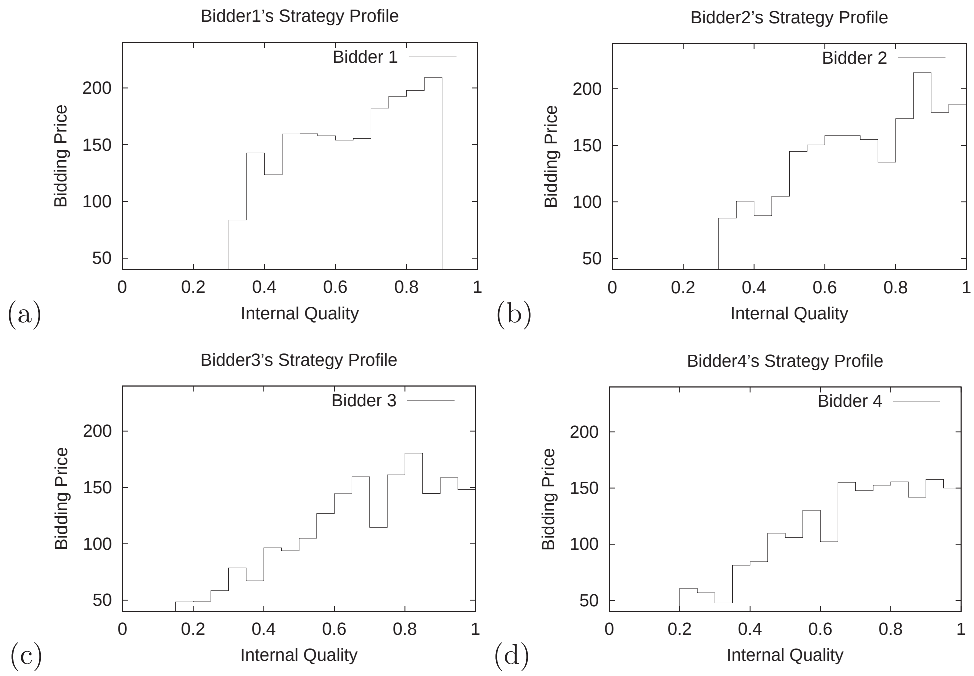


FIGURE 4.10: Strategy Profiles of Bidders with Effective and Ineffective Factors

To this end, the strategy profiles for four agents (q_1, q_2, q_3 and q_4) at the point when the market reaches equilibrium are plotted in Figure 4.10. From Figure 4.10(a), we can see that the first agent bids its recommendations from INQ segments that are above the level of 0.65 at a level that is much higher than 160, which is actually the equilibrium price of the tenth bid (see Figure 4.9 after 80 auctions). Since the equilibrium price of the tenth bid represents the lowest price to be shortlisted, we refer to it as the *market access price*. For the first agent, both evaluation properties (Φ_1 and Φ_0) are the effective factors and their weightings both match those in the UPQ. Thus, its INQ fully matches the UPQ. Being capable of relating its INQ to the UPQ, this agent can establish from which specific INQ segments its recommendations can be rewarded. From Figure 4.10(a), we can also see that bids from INQ segments that are below the level of 0.65 are lower than the market access price. Indeed, the first agent learns from the marketplace that these recommendations will not be rewarded and so it decreases their price so as not to shortlist these items and avoid paying for them when they are unlikely to produce a return.

From Figure 4.10(b), we can see that the second agent bids its recommendations from very high INQ segments (higher than the level of 0.80) at a level that is higher than the market access price. The second agent has one of its two evaluating properties (Φ_1) as the effective factor and this contributes significantly to both the INQ and the UPQ (both with a weighting of 0.75). In this case, only a very high value of Φ_1 can give a high value of q_2 since Φ_1's weighting is much bigger than Φ_3's. Thus, very high INQs indicate high values of UPQ, and, therefore, such recommendations have good correlations to the user's preferences. Therefore, the agent only bids on very high INQ recommendations that are highly likely to be shortlisted. It does this to make profit without incurring a high risk of losing credits (i.e. shortlisted but not rewarded).

From Figure 4.10(c), we can see that the third agent has few segments with bids higher than the market access price (compared to the first and second agents). The explanation is that, even though one of its two evaluating properties (Φ_0) is the effective factor, it contributes too little to its INQ (coefficient value 0.25). Therefore, its INQ cannot easily be related to the UPQ. With fewer concrete signals from the rewards received, it is difficult for the agent to relate its bids to its INQs. Thus the agent is not confident enough to bid for certain items at a very high price (since it has a high risk of losing credits without earning).

Figure 4.10(d) demonstrates that the fourth agent, having no effective factors, does not dare to bid high enough for any items from any segments to be shortlisted. It behaves in this way because it does not want to incur the risk of being shortlisted without receiving any reward. This uncertainty comes from the fact that the agent cannot effectively relate its INQ to the UPQ. Thus it does not know what items from which segments match the user's preference.

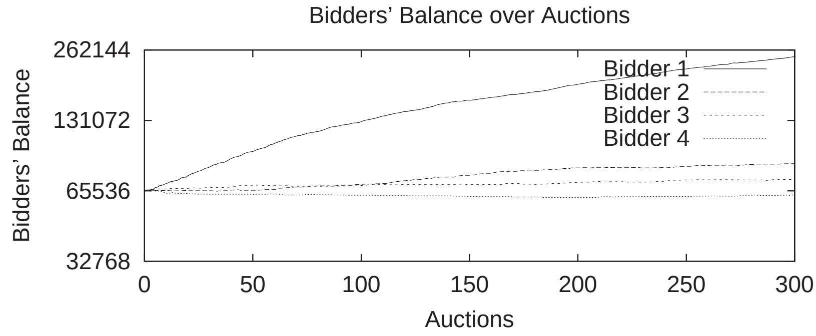


FIGURE 4.11: Balance of Bidders with Effective and Ineffective Factors

When taken together, these experiments show that the agents' confidence to relate their bids to the INQ decreases from the first agent to the fourth. Theoretically, this point can be shown in their INQ functions with respect to the UPQ (see equation (4.8′)). Thus, the noise between the four agents' INQs and the UPQs is, respectively, 0, $0.25(\Phi_4(Rec) - \Phi_0(Rec))$, $0.75(\Phi_5(Rec) - \Phi_1(Rec))$ and full noise. Therefore, the agents' ability to relate their INQs to the UPQ is in decreasing order. On the other hand, since the agents' bids are based on rewards and rewards are based on the UPQ, the bids can be related to the UPQ. Thus, the agents' ability to relate their INQs to their bids is in decreasing order. This, in turn, effects their balance. Specifically, Figure 4.11 demonstrates that the more strongly an agent can relate its INQs to its bids, the more profit it will make.

4.4 Validating the System's Ability to Seek Out the Best Recommendation

Having evaluated the market with respect to the metrics listed in section 3.1 and the correlation between the INQ and the UPQ of the recommendations, this section evaluates the system's ability to seek out the best item from all the source recommendations. This is clearly an important feature from the user's viewpoint, since if the system cannot recommend the best items, the user will not use it.

To evaluate this aspect of the system, we use the first experiment discussed in section 4.2.1 and trace the bidding price of the recommendation with the highest UPQ value selected by the first agent (see Figure 4.12, in which the cross points represent the bidding price of this particular recommendation). From this, we can see that this

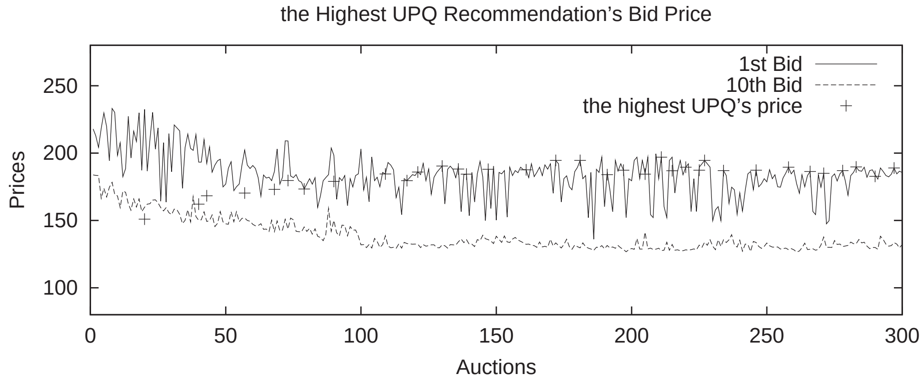


FIGURE 4.12: The Best Recommendation's Bidding Price

recommendation's bidding price keeps increasing till it converges to the first bid price of the shortlisted items. This means that as long as the first agent chooses the highest UPQ recommendation to bid in an auction round (after the market converges), this item is always displayed in the first slot of the sidebar of the user's browser. Therefore, in case of either user model (independent selection or search-till-satisfaction), this recommendation will be selected by the user, since the first shortlisted recommendation has the highest UPQ. This result shows that the system is capable of seeking out the best recommendation and presenting it to the user.

4.5 Summary

Based on these simulations, the auction mechanism designed in Chapter 3 is shown to be effective. Specifically, we organized a set of consecutive auctions with nine recommender agents to offer their recommendations. We simulate the user with two kinds of user models: independent selection of recommendations and search till satisfied. We find that our market can always converge after a number of auctions with either user model. After convergence, the marketplace is able to give incentives of users' preferences to the recommender agents and to shortlist the best recommendations in decreasing order of their UPQs. Additionally, the marketplace gives either effective or ineffective recommender agents equal opportunity to bid their recommendations and is stable meaning that it is able to stop greedy bidders aggressively bidding their recommendations. By simulation, our marketplace is shown to be capable of successfully correlating the two perspectives of recommendation quality (internal and user perceived) and is able to identify the highest UPQ item to be shortlisted at the top position of the recommendation sidebar.

This chapter has proved that the auction mechanism design developed in Chapter 3 is feasible to coordinate multiple different recommendation methods in one single system and is able to relate their good recommendations to the user's interests. However, how a recommender agent learns the user's interests is left to be addressed and this is the subject of Chapter 5.

Chapter 5

Learning Users' Interests

Having shown the effectiveness of our market mechanism as a means of coordinating different recommendation methods, an open problem from the point of view of the individual recommender agents remains: *given a set of recommendations with different* INQ *levels, in what order should an agent try to advertise them so that it can learn the user's interests as quickly as possible, while still maximizing its revenue?* Thus, for example, the agent could bid the items that have never been advertised to the user, which would allow it to learn the user's interests quickly but could also result in it losing money. Conversely, the agent could always bid those items that have been highly rewarded, so ensuring a good return, but it would take a very long time to learn the extent of the user's interests. To this end, this chapter reports a quality classification mechanism and a reinforcement learning strategy we built for the recommender agents to learn the user's interests.

This chapter contributes to the thesis in the way that a marketplace with learning recommender agents converges quicker and seeks out the best recommendations quicker and more frequently than that with non-learning agents. Moreover, with a learning capability, a recommender agent is able to make a larger amount of profit, while still making good recommendations.

Specifically, section 5.1 outlines the metrics over which we can evaluate our learning strategy, section 5.2 details the learning algorithm and the exploration strategy, and section 5.3 evaluates the learning strategy against the metrics defined in section 5.1.

5.1 Evaluation Metrics

To evaluate the learning strategy we use the following evaluation metrics (the first two are concerned with an individual learner's performance and the second two with the performance of the collective of learners):

❶ *Convergence to Optimality:* Many learning algorithms come with a provable guarantee of asymptotic convergence to optimal behaviour [Mitchell, 1997]. This criterion is included here to evaluate the quality of learning itself; it is important because if an algorithm does not converge, the agent will have no incentive to follow its behaviour.

❷ *Individual Rationality:* See the third metric defined in section 3.1.

❸ *Quick Market Convergence:* See the definition of "market convergence" in the fourth metric in section 3.1. In the analysis of our recommender system in Chapter 4, we proved that convergence is necessary to ensure only the best items are displayed and that they are shortlisted in decreasing order of UPQ. Therefore, a market that converges quickly means that it starts satisfying the user quickly. This is clearly important since a user will stop using a recommender if it takes too long to produce good suggestions.

❹ *Best Recommendation's Identification:* A good recommender system should be able to identify the best recommendation (the one with the highest UPQ) quickly and suggest it frequently [Konstan et al., 1997, Bohte et al., 2004]. This is important because, otherwise, if the best recommendations cannot be identified and displayed, the user will stop using the system.

The first metric is new compared to those metrics listed in section 3.1. This metric is needed specifically to evaluate the learning performance which is why it has not been considered until now. The second and the third metrics are chosen from the same perspective of those listed in section 3.1. The second metric is used to evaluate the individual agents' bidding behavior and the third is used to evaluate the overall performance of the marketplace as the coordinator of multiple recommenders. The fourth metric is also new because with this we can compare the system performance in

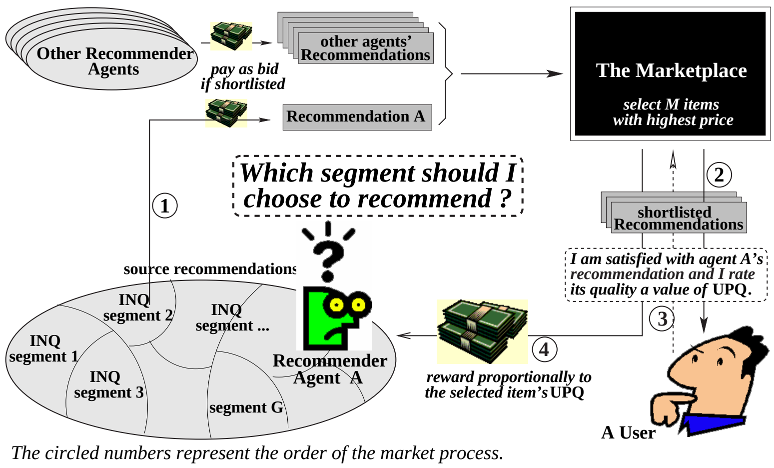


FIGURE 5.1: An Individual Agent's Quality Classification Problem

terms of high quality recommendations between one with learning capability and one without.

With these metrics in place, we are going to design our learning strategy.

5.2 The Learning Strategy

This section details a recommender agent's learning algorithm (section 5.2.2), exploration strategy (section 5.2.3) and the overall strategy (section 5.2.4). Before we go into the details of the strategy, however, section 5.2.1 briefly discusses an agent's learning problem in our marketplace.

5.2.1 The Quality Classification Problem

From the point of the view of an individual constituent recommender agent in our system, its aim in advertising recommendations is to maximize its revenue through satisfying the user. Thus, an agent needs to learn which recommendations the user prefers and align its bidding behaviour with the interests of the user. To do this, agents classify their recommendations into a predetermined number (G) of categories (or segments, e.g., $G = 20$ in section 4.1.2 on page 56) based on their INQs (e.g. in the simplest case, where $G = 2$, an agent could classify "bad" recommendations as those with an INQ of less than 0.5 and those with an INQ between 0.5 and 1.0 as "good") and then they relate these INQs to the UPQs (see Figure 5.1). Intuitively, the more the user is satisfied with a recommendation, the more reward the corresponding agent receives. Thus, 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) they receive [Wei et al., 2003a]. This, in turn, 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 recommendation displayed to the user in one auction round) as the reward it received minus the price it has paid for the advertisement[1]. With this, what an agent needs to do is to learn how much immediate reward, on average, it can expect for items in each category (i.e. each INQ segment). We term this average immediate reward for each INQ segment an agent's *expected revenue*. Thus, a self-interested agent can maximize its revenue by frequently bidding recommendations from the segments with high expected revenue. Therefore, an agent's recommending task can be seen as a quality classification problem and it needs to align the user's preferences with its INQ segments (reflected by expected revenue) and, meanwhile, make maximal revenue [Wei et al., 2004b,a, To appeara].

However, when an agent starts bidding in the marketplace, it has no information about how much revenue it can expect for each segment. Therefore, the agent needs to interact in the marketplace by taking actions over its G segments to learn this information (as per Figure 5.1). In this way, an agent can produce a profile of such information from which it can form an optimal strategy to maximize its overall revenue. In this context, the agent's learning behaviour is on a "trial-and-error" basis. The agent bids its recommendations and receives the corresponding feedback in a manner that 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* [Mitchell, 1997]. Thus, to be more concrete, an agent needs an algorithm to learn the expected revenue over each segment. In addition, it also needs an exploration strategy to make trials on its G segments such

[1] Agents pay nothing for items they put forward that are not displayed to the user (this occurs when other agents are willing to pay more to advertise their recommendations). By definition, an immediate reward may either be positive or negative. If a displayed recommendation is not selected by the user or if it has paid too much to display an item, the corresponding agent's immediate reward is negative since it has paid for the display and received less reward.

that it strikes a balance between learning as quickly as possible, while still maximizing revenue.

5.2.2 The Q-Learning Algorithm

In chapters 3 and 4, we have proved (theoretically and empirically) that our marketplace enables an agent to relate the rewards it received to its G INQ segments. Building on this basis, the contribution of this chapter is in how to effectively learn the expected revenue that is likely to accrue over its G segments. Such a strategy is desirable because high expected revenue on a specific segment implies that more rewards can be expected if it repeats bidding on that segment in future. Therefore, this subsection aims to address the problem of producing the expected revenue profile over an agent's G segments, while still trading profitably in the marketplace.

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 (depending on whether its recommendation satisfies the user). 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:

$$E[R(a_i)] = \lim_{t \to \infty} (\frac{1}{t} \sum_t r_{i,t}) \ . \tag{5.1}$$

In this context, the Q-learning technique provides a well established way of estimating the optimality [Mitchell, 1997]. In particular, we use a standard Q-learning algorithm to estimate $R(a_i)$ by learning the mean value of the immediate rewards:

$$\hat{Q}_i := (1 - \frac{1}{t}) \cdot \hat{Q}_i + \frac{1}{t} \cdot r_{i,t} \ , \tag{5.2}$$

where $\hat{Q}_i$ is the current estimation of $R(a_i)$, and $\frac{1}{t}$ is the learning rate that controls how much weight is given to the immediate reward (as opposed to the old estimation). As $\frac{1}{t}$ decreases, $\hat{Q}_i$ builds up an average of all experiences, and the odd new unusual experience, $r_{i,t}$, does not significantly affect the established $\hat{Q}_i$. As t approaches infinity, the learning rate tends to zero which means that learning is no longer taking place. This, in turn, makes $\hat{Q}_i$ converge to a unique set of values that define the expected revenue of each segment.

Proposition:

As $t \longrightarrow \infty$, $\hat{Q}_i$ *converges to* $E[R(a_i)]$.

Proof:

We use $Q_{i,0}$ to represent the initial value of $\hat{Q}_i$, and $\hat{Q}_{i,t}$ to represent the local estimation to $R(a_i)$ when a_i has been experienced t times. $\hat{Q}_i$'s updates go:

$\hat{Q}_{i,1} = 0 \cdot \hat{Q}_{i,0} + 1 \cdot r_{i,1} = r_{i,1}$

$\hat{Q}_{i,2} = \frac{1}{2} \cdot r_{i,1} + \frac{1}{2} \cdot r_{i,2} = \frac{1}{2}(r_{i,1} + r_{i,2})$

$\hat{Q}_{i,3} = \frac{2}{3} \cdot \frac{1}{2}(r_{i,1} + r_{i,2}) + \frac{1}{3} \cdot r_{i,3} = \frac{1}{3}(r_{i,1} + r_{i,2} + r_{i,3})$

$\vdots$

$\hat{Q}_{i,t} = \frac{1}{t}(r_{i,1} + r_{i,2} + \cdots + r_{i,t}) = \frac{1}{t}\sum_{j=1}^{t} r_{i,j}$

As $t \rightarrow \infty$, $\lim_{t\rightarrow\infty}(\frac{1}{t}\sum_{j=1}^{t} r_{i,j})$ statistically defines $E[R(a_i)]$. ■

This proof exemplifies how newly experienced immediate rewards, combined with the learning rate, produce convergence. With the Q-learning algorithm in place, an agent needs an exploration strategy to execute actions to build up its $\hat{Q}$ profile.

5.2.3 The Exploration Strategy

We assume all agents are self-interested and want to gain maximal revenue as they bid. However, before $\hat{Q}_i$ converges, it is difficult for an agent to know how much can be expected through each action and, therefore, which action it should choose. It is faced with the classic dilemma of choosing actions that have a well-known reward or choosing new ones that have uncertain rewards (which may be higher or lower than the well-known actions). To this end, the agent needs an exploration strategy over its G segments to build up its $\hat{Q}_i$ in an effective way so that it can know how much return can be expected from each segment.

In general, there is a fairly well developed formal theory for exploration strategies for problems similar to that faced by our agents [Kaelbling et al., 1996]. However, the standard methods require very specific conditions (detailed in section 2.6) that do not hold in our context[2]. Specifically, the number of times that an agent can interact with the marketplace is not limited. Thus, the agent can gather as much information as it wants in order to form its expected revenue profile. 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 [Thorndike, 1898]: *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 [Kaelbling et al., 1996]. The probability of taking action a_i is formally defined as:

$$P_{a_i} = \frac{\mathbf{e}^{\hat{Q}_i/T}}{\sum_{j=1}^{G} \mathbf{e}^{\hat{Q}_j/T}} \qquad (T > 0), \tag{5.3}$$

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).

In practice, however, we have observed that the learning algorithm of equation (5.2) accompanied with the exploration strategy of equation (5.3) has a problem of producing bias from the optimal and very little work has been done to address this. This problem occurs when an agent obtains a very small negative $\hat{Q}_i$ value for a particular action in its first few trials[3]. If this happens, a bias from the true expected revenue of this action may occur (since the action may in general produce positive $R(a_i)$) and the agent will seldom choose it. This kind of bias is a particular problem in our system. This is because a user may not always visit all displayed items in the sidebar and, thus, some good recommendations may be skipped and, therefore, be deemed bad ones. To avoid such bias, T needs to be assigned a very large value in the beginning of learning to limit the exploration priority given to those actions with very large $\hat{Q}$ values. However, controlling

[2] In fact, it is hard to find the absolutely best strategy for most complex problems. In reinforcement learning practice, therefore, 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 [Kaelbling et al., 1996].

[3] A negative immediate reward means punishment and an erroneous action. A reward of zero means that the action has received no feedback. Thus, actions with negative, zero and positive feedback are differentiated and exploration priority should be given to the latter two.

T in terms of producing the unbiased optimal strategy is hard to achieve, since different actions' $\hat{Q}$s converge with different speeds and their convergence is difficult to detect. Even with other exploration strategies, such biases still exist since no exploration can avoid such unlucky trials at the beginning of learning. To this end, we developed an algorithm that takes positive initial $\hat{Q}_i$ values into account to overcome this problem. We detail this in the next section.

5.2.4 The Overall Strategy

To overcome the impact of bias in the beginning of learning, we use positive initial $\hat{Q}$ values (i.e. $\hat{Q}_{i,0}$) and make them affect the learning. Thus, instead of algorithm (5.2), we use the following learning algorithm:

$$\hat{Q}_i := (1 - \frac{1}{t_0 + t}) \cdot \hat{Q}_i + \frac{1}{t_0 + t} \cdot r_{i,t} \ . \tag{5.4}$$

The difference between (5.2) and (5.4) is that the former does not take $\hat{Q}_{i,0}$ into account, whereas the latter does. Specifically, algorithm (5.4) assumes that each action has been experienced t_0 (t_0 is positive and finite) times and each time with a feedback of $\hat{Q}_{i,0}$ ($\hat{Q}_{i,0} \gg 0$) before the agent starts learning. This, in turn, removes the problem discussed in section 5.2.3. Indeed, if an action causes a negative immediate reward in the beginning, it does not force its $\hat{Q}_i$ to become negative. In this way, all actions will still be allocated a relatively equal opportunity of being explored as an agent begins learning. As the agent continues to interact with the marketplace, its $\hat{Q}_i$s update gradually to different levels and these levels still make its exploration follow the law of effect. Thus, the agent's exploitation tends to optimality with its $\hat{Q}$ values tending to converge. Additionally, by initializing $\hat{Q}$ with positive values, the exploration does not need a sophisticated control on T, since a relatively small positive value is sufficient and is easier to control. Moreover, the change from (5.2) to (5.4) does not affect the convergence (as proved below)[4].

Proposition:

Given $\hat{Q}_i$'s definition by algorithm (5.4), its convergence to $E[R(a_i)]$ is independent of its initial value $\hat{Q}_{i,0}$ and initial time t_0 .

[4]However the time it takes to converge is extended slightly depending on the values of $\hat{Q}_0$ and t_0 (the larger their values are, the longer it takes to converge).

```
The Main Strategy:
  for  i = 1 to G do {
      Q̂_{i,0} = Q_{init};                                  // Initialize Q̂_i and Q_{init} ≫ 0
      t_i = 0;                                             // Initialize t_i
  }
  do {
      for i = 1 to G do
          P_{a_i} = ExploreProbability( i, Q̂_1, Q̂_2,···, Q̂_G );   // Equation (5.3)
      a_k = ActionSelection( P_{a_1}, P_{a_2}, ···, P_{a_G} ) ★;   // k ∈ [1..G]
      t_k = t_k + 1;                                       // a_k has been experienced t_k times
      r_{k,t_k} = ImmediateReward( a_k );                  // compute immediate reward
      Q̂_k = UpdateQ( Q̂_k, t_k, r_{k,t_k} );                // Equation (5.4)
  } while (true)

★ Method ActionSelection:
  ActionSelection( P_{a_1}, P_{a_2}, ···, P_{a_G} ){
      double boundary[0..G];                               // probability boundary for G segments
      for i = 0 to G do
          boundary[i] = 0;
      for i = 1 to G do                                    // compute the G actions' probability boundary
          for j = 1 to i do
              boundary[i] = boundary[i] + P_{a_j};
      double Rand = UniformRandom0to1() ♠;                 // generate a probability
      for k = 1 to G do
          if ( boundary[k − 1] ⩽ Rand < boundary[k] )
              return a_k;                                  // select a random action based on its probability
  }

♠ UniformRandom0to1() returns a random value that follows a uniform distribution within the range [0, 1.0).
```

FIGURE 5.2: The Learning Strategy for an Individual Agent

Proof:

$\hat{Q}_i$'s updates go:

$$\hat{Q}_{i,1} = \frac{t_0}{t_0+1} \cdot \hat{Q}_{i,0} + \frac{1}{t_0+1} \cdot r_{i,1}$$

$$\begin{aligned}\hat{Q}_{i,2} &= \left(1 - \frac{1}{t_0+2}\right)\left(\frac{t_0}{t_0+1} \cdot \hat{Q}_{i,0} + \frac{1}{t_0+1} \cdot r_{i,1}\right) + \frac{1}{t_0+2} \cdot r_{i,2} \\ &= \frac{t_0}{t_0+2} \cdot \hat{Q}_{i,0} + \frac{1}{t_0+2} \cdot (r_{i,1} + r_{i,2})\end{aligned}$$

$$\begin{aligned}\hat{Q}_{i,3} &= \left(1 - \frac{1}{t_0+3}\right)\left(\frac{t_0}{t_0+2} \cdot \hat{Q}_{i,0} + \frac{1}{t_0+2} \cdot (r_{i,1} + r_{i,2})\right) + \frac{1}{t_0+3} \cdot r_{i,3} \\ &= \frac{t_0}{t_0+3} \cdot \hat{Q}_{i,0} + \frac{1}{t_0+3} \cdot (r_{i,1} + r_{i,2} + r_{i,3}) \\ &\vdots\end{aligned}$$

$$\hat{Q}_{i,t} = \frac{t_0}{t_0+t} \cdot \hat{Q}_{i,0} + \frac{t}{t_0+t} \cdot \frac{1}{t} \cdot \sum_{j=1}^{t} r_{i,j}$$

Since t_0 is finite, $\lim_{t\to\infty} \frac{t_0}{t_0+t} \longrightarrow 0$ and $\lim_{t\to\infty} \frac{t}{t_0+t} \longrightarrow 1$.

Thus, $\lim_{t\to\infty} \hat{Q}_{i,t} \longrightarrow \lim_{t\to\infty}(\frac{1}{t}\sum_{j=1}^{t} r_{i,j}) = E[R(a_i)]$. ■

This proof shows that algorithm (5.4) also produces unbiased learning. Thus, we will use (5.4) and (5.3) for our agents and the overall strategy is detailed in Fig. 5.2.

5.3 Evaluation

This section reports on the experiments to evaluate the learning strategy we have developed. Section 5.3.1 outlines the experimental settings, whereas section 5.3.2 evaluates the learning strategy against the metrics defined in section 5.1.

5.3.1 Experimental Settings

We assume that there are four good recommendation methods (able to correlate their INQs to the UPQ) and four poor ones (unable to do so). Given a specific recommendation (Rec), the correlations of its UPQ to a good method's INQ (INQ_g) and to a poor one's (INQ_p) are described in equations (5.5) and (5.6) respectively (again "$\ncong$" means "has no relation to"):

$$UPQ(Rec) = INQ_g(Rec) \pm 0.1 \cdot random() \tag{5.5}$$

$$UPQ(Rec) \ncong INQ_p(Rec) \tag{5.6}$$

where $random()$ returns a random value that follows a uniform distribution within the range [0, 1.0). This random value can be seen as the noise between the INQ and the UPQ. All UPQ and INQ values are fixed within [0, 1.0). In each auction round, the marketplace calls for ten bids. Again we use an independent-selection user model to decide which recommendations displayed to the user will be rewarded (see section 4.1.3). In this model, selecting one item is independent of selecting another and all recommendations with a UPQ higher than a particular threshold will be rewarded. Here, we set this threshold to 0.75. To correlate their INQs to the UPQs, all agents divide their INQ range into $G = 20$ equal segments. We assume that all agents share the same set of recommendations and each agent has at least ten items in each segment. Before starting to bid, Q_{init} is set to 250, $T = 200$ and $t_0 = 1$ for all agents. All agents are initially endowed with same amount of credit (65536). At the beginning, each agent will bid the same (128) for items from any segment, since it does not know which segments are more valuable than others.

5.3.2 Simulating and Evaluating the Strategy

Having outlined the configuration of the agents, this section details the evaluations. The results shown in this section are for a single simulation run. However, to ensure these results are typical for our system, we repeated the experiments for two hundred simulation trials. Thus, the results we will show and discuss are representative of the outcomes. Specifically, over the two hundred simulations, we found that: in 78.1% of the trials the good recommendation methods' $\hat{Q}$s converge; all good recommendation methods with converged $\hat{Q}$ profiles make a significantly greater amount of credit (38.3% on average); a marketplace with learning agents takes 59.4% less time to converge than one without; and the number of best recommendations that a learning market is able to identify is, on average, 2.73 times that of a market without learning capability.

Among all the properties that we want the learning strategy to exhibit, $\hat{Q}$ convergence is the most important. Indeed, in its absence, an agent loses its basis to reason (see section 5.1). Thus, we will start with experiments on the convergence of $\hat{Q}$ values.

5.3.2.1 Convergence to Optimality

To evaluate an agent's $\hat{Q}$ value convergence, we arranged 300 consecutive auctions. Among the eight agents, the first four employ the good recommendation method and the last four employ the poor one. We find that, with a good method, an agent's $\hat{Q}$ values always converge such that high INQ segments' $\hat{Q}$s (corresponding to high UPQ because of equation (5.5)) converge to high values and low INQ segments' $\hat{Q}$s converge to low values (see Fig. 5.3(a)). Specifically, the $\hat{Q}$ values of those INQ segments corresponding to the UPQs above the user's satisfaction threshold (0.75) converge proportionally to their corresponding UPQs. The higher the corresponding UPQ, the higher the $\hat{Q}_i$'s convergence value, because the recommendations from a segment corresponding to higher UPQs receive more immediate reward than those corresponding to lower UPQs. The $\hat{Q}$ values of those segments that correspond to the UPQs below 0.75 converge to negative values, since they do not receive rewards if their recommendations are displayed. Moreover, the convergence is independent of the specific form of equation (5.5). Specifically, once there is a unique UPQ level corresponding to each INQ level (even high INQ corresponding to low UPQ), the $\hat{Q}$ value of an INQ segment corresponding to a high UPQ will

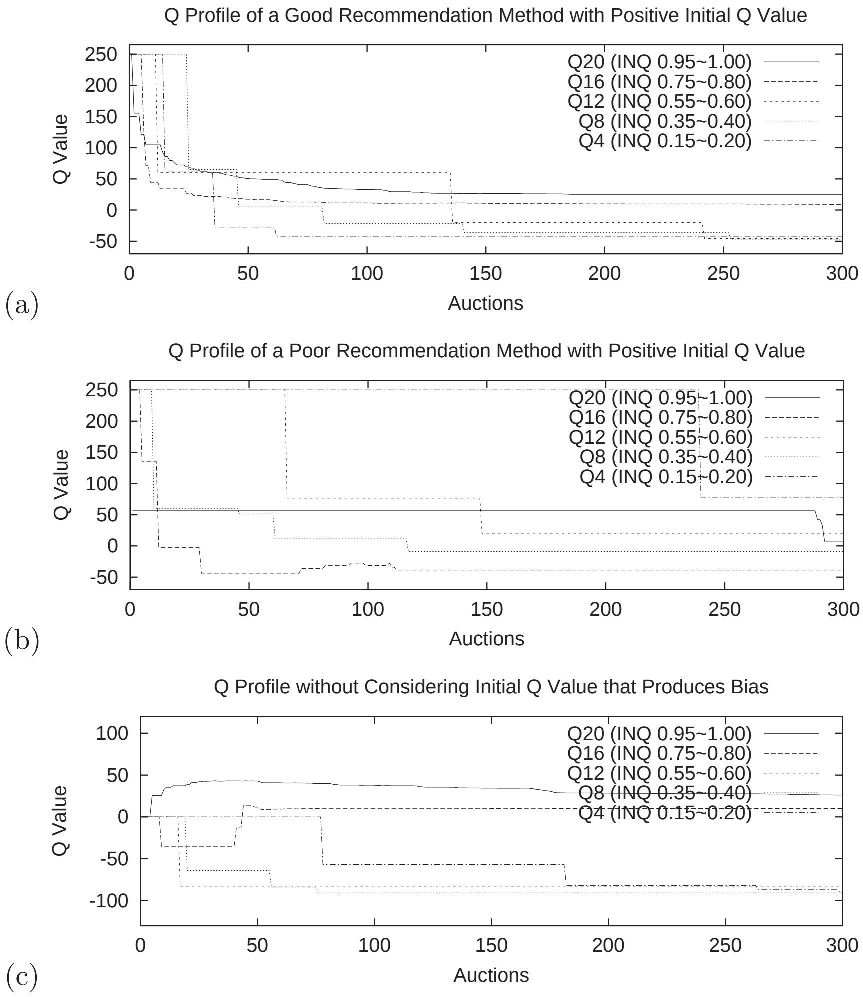


FIGURE 5.3: Q-Learning Convergence

always converge to a high level (since it induces high immediate rewards). However, with a poor method, an agent's $\hat{Q}$ values cannot converge such that high INQ segments' $\hat{Q}$s converge to high values (see Fig. 5.3(b)). This is because a specific INQ corresponds to very different UPQs (and very different immediate rewards) at different times because of equation (5.6).

To exemplify that our learning algorithm (5.4) overcomes the bias problem that may occur in (5.2), we organized another set of experiments with all agents taking zero initial $\hat{Q}_i$ values and all other settings remained unchanged (see Fig. 5.3(c)). From Fig. 5.3(c), we can see that $\hat{Q}_{12}$ is updated only once and with a large negative value of -82 (this gives the corresponding action virtually no chance of being selected in future). $\hat{Q}_{16}$ also produces a bias in the beginning. In even worse cases, $\hat{Q}_{16}$ can never update itself

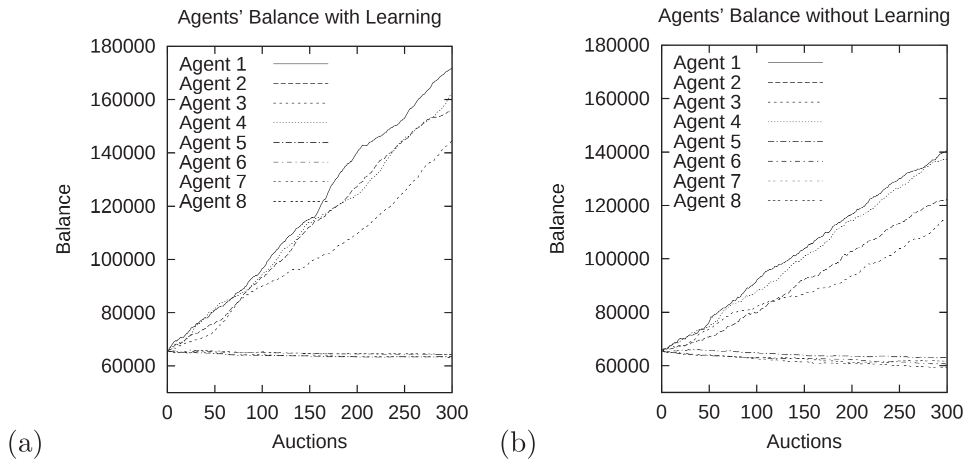


FIGURE 5.4: Recommenders' Balance

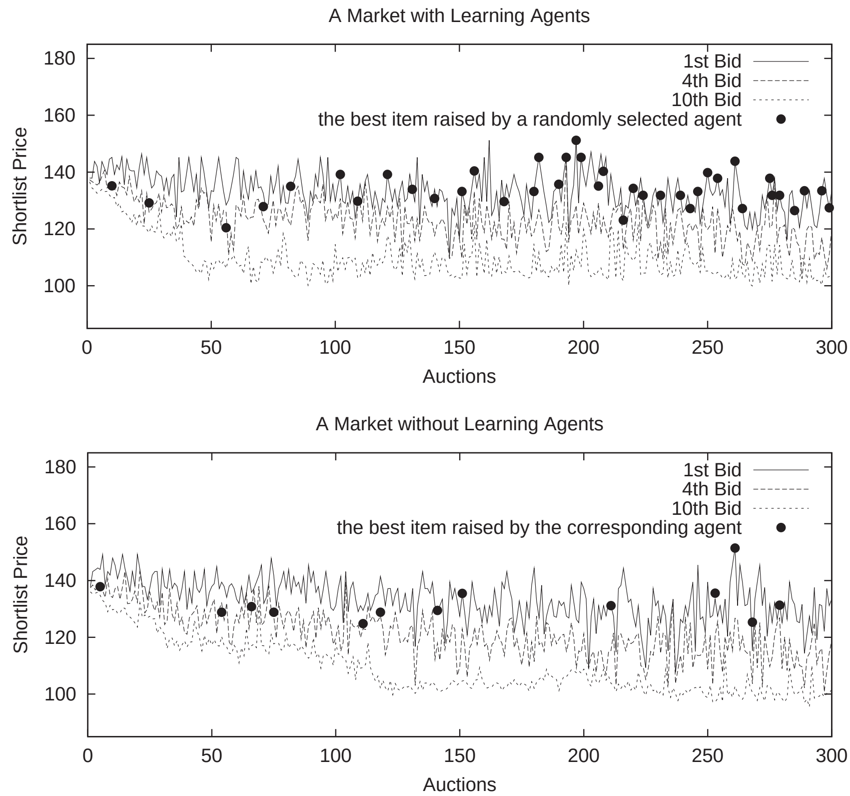
like $\hat{Q}_{12}$ (however, it should actually have a positive expected revenue). However, with positive initial $\hat{Q}_i$ values, such biases do not occur (see Fig. 5.3(a)).

5.3.2.2 Individual Rationality

The agents with good methods are able to know what recommendations better satisfy the user. Therefore, they can achieve more immediate rewards. Thus, good recommendations are raised more frequently by a learning agent than by a non-learning one. This, in turn, means learning agents can maximize their revenue by selecting good recommendations. In particular, Fig. 5.4 shows that good recommendation methods with learning capability (the first four agents in Fig. 5.4(a)) make, on average, significantly greater amounts (about 43%) of credit than those without (the first four agents in Fig. 5.4(b)). With a poor method, the agents cannot relate their bids to the user's interest and therefore bid randomly. Thus, they cannot consistently achieve positive immediate rewards and their revenue is low (the last four agents in Fig. 5.4 (a) and (b)).

5.3.2.3 Quick Market Convergence

We have shown that market convergence enables the agents to know what prices to bid for recommendations relating to certain UPQs so as to gain maximal revenue in Chapter 4. Thus, quick market convergence lets agents reach this state quickly. To evaluate this, we



(a)

(b)

FIGURE 5.5: Market Convergence

organized two sets of experiments (using the same settings as the experiments assessing the convergence). The first one contains all learning agents and the other contains none. We find that a marketplace with learning agents always converges quicker than the one without. From Fig. 5.5, we can see that a marketplace with learning agents (Fig. 5.5(a)) converges after about 40 auctions, whereas one without (Fig. 5.5(b)) converges after about 120 auctions. Indeed, as the learning agents' $\hat{Q}$ profiles converge, more high quality recommendations are consistently suggested (since their high $\hat{Q}$ values induce high probability for the agent to bid these items because of equation (5.3)) and low quality ones are deterred. This, in turn, accelerates effective price iterations to chase the market equilibrium. It takes approximately one third of the time for a market with learning agents to chase the equilibrium compared to one without.

5.3.2.4 Best Recommendation's Identification

To evaluate the learning strategy's ability to identify the best recommendation (from the viewpoint of the user, i.e. the top UPQ item) quickly and bid it consistently, we use the same set of experiments that were used to assess market convergence. We then trace the top UPQ item highlighted by a randomly selected learning agent with a good recommendation method and a corresponding one from a non-learning agent in Fig. 5.5 (a) and (b) respectively. We do this by plotting this top UPQ items' bidding prices with circle points in the figures. To clearly display the points of the trace and not to damage the quality of lines (representing the three displayed bids), we do not display the points when this item is raised by other agents. From Fig. 5.5(a), we can see that this item's bidding price keeps increasing till it converges to the first bid price of the displayed items. This means that as long as the randomly selected agent chooses this particular item to bid in an auction (after the market converges), it is always displayed in the top position displayed to the user. However, in contrast, this phenomenon in a market without learning agents proceeds slowly (see Fig. 5.5(b)). This means that a learning market can satisfy the user quicker than a non-learning one. Additionally, a learning market raises the best recommendation more frequently (39 times by the selected learning agent, see Fig. 5.5(a)) than a market without learning capability (13 times by the corresponding non-learning agent, see Fig. 5.5(b)).

5.4 Summary

This chapter presents the learning problem that a recommender agent faces in our marketplace. Specifically, the agent needs to classify its recommendations into different INQ categories and to quickly identify and frequently suggest those categories that highly interest a user so as to maximize its revenue, while still satisfying the user. By simulating and evaluating our Q-learning strategy, we show that the strategy can always come to the optimal solution, is able to quickly identify the effective INQ categories and frequently suggest items from these categories, and enables the agents to make more revenue than those without a learning capability.

In sum, this chapter has developed a reinforcement learning algorithm and a Boltzmann exploration strategy for the recommender agents to learn the users' interests. This chapter has also proved the effectiveness of recommender agents using the learning strategy in our marketplace. With this learning capability, our marketplace converges quicker and suggests the best items more quickly and frequently than without it.

Chapter 6

User Evaluations of the Recommender System

With the marketplace designed, simulated and formally analyzed, we now need to evaluate the feasibility and the efficiency with real users of our market-based approach to recommender systems. To do this, we implemented a market-based recommender system that incorporates three typically-used recommendation methods (content-based, collaborative and demographic). We then arranged for a number of people (thirty-one in this case) to use our system so that we could record various aspects of their interactions and the system's outputs. These records were then analyzed in order to provide a user evaluation of the efficiency of our system.

With the user evaluations of our system, we have shown that *(i)* multiple constituent recommenders contribute to the recommendations that are placed in front of users through the marketplace, *(ii)* the marketplace converges with respect to most of the users, *(iii)* the market-based recommender's top recommendation is the best item of those suggested by whatever constituent recommenders for most of the users most of the time, and *(iv)* the marketplace is able to seek out the best recommendations for a given user most of the time and place these among the top positions in the recommendation sidebar most of the time. By undertaking these user trials, this chapter contributes to the thesis by showing that the market-based approach is capable in practice (as well as in theory) of coordinating multiple recommendation methods and effectively identifying the best recommendations quickly and frequently.

Specifically, section 6.1 defines the metrics that are used to evaluate our system. Section 6.2 outlines a user's task in terms of using the recommender system. Section 6.3 details the system configurations in terms of the three recommendation methods. Consequently, section 6.4 discusses the evaluation results and section 6.5 summarizes our findings from this aspect of the work.

6.1 Evaluation Metrics

In seeking to evaluate our system with real users, the first step is to identify the properties that we would like our market-based recommender system to exhibit. This then gives us the requirements against which we perform our evaluation. In particular, we are interested in the following metrics (the first, second and fourth metrics are the most important system properties selected from those discussed in sections 4.2 and 4.4; the third metric is defined according to the essential purpose of the market-based approach to recommender systems discussed in section 1.1):

❶ *Balanced Output Contribution*

There are three constituent recommenders incorporated in our marketplace (each of which exploits a different recommendation method, see section 6.3 for details). Here we term the recommendations suggested by one constituent recommender and eventually displayed (shortlisted) to users as that recommender's *output contributions*. For a given user, it might be the case that one recommender makes the significant majority of output contributions and the others make very few output contributions. In this case, we say that the recommender that contributes the majority of outputs *dominates* the marketplace. Such domination with respect to a specific user is not necessarily a bad thing (because it means the dominating recommender has learnt this user's interests more efficiently and therefore contributes more good recommendations than the remaining recommenders). However, it would be a problem if the same method dominates the user population across all their various interests. Indeed, if the users' interests literally follow a uniform distribution among a number of potential interesting browsing topics (meaning that different users have different interests and no one interesting topic dominates the majority population of the users), if one constituent recommender dominates the marketplace for most users most of the time, the marketplace essentially degenerates to the single dominant method. To capture the fact that multiple methods actively work simultaneously, generally speaking, we expect the different constituent recommenders to make balanced (broadly similar) output contributions with respect to a number of users with various interests. This metric is important because, on the one hand, compared to the equal opportunity of bidding that the marketplace gives to different constituent recommenders (discussed in simulations in section 4.2.4), this metric further evaluates the fairness of the marketplace in terms of output contribution in a real environment (with real users and real recommendations) and this cannot be done in the design and simulation stages. On the other hand, the balanced output contribution metric eventually verifies the fact that the marketplace works as a means of coordinating multiple different recommendation methods and ensures the marketplace does not degenerate to a single method.

❷ *Market Convergence*

As highlighted in sections 4.2 and 4.4, market convergence is a key desirable characteristic of our system. Such convergence is important because it ensures that the system makes an effective shortlist of recommendations, gives the appropriate incentives to the constituent recommenders, gives equal opportunity of bidding to different constituent recommenders, makes the marketplace stable, and seeks out the best recommendations frequently. Now section 4.2.1 showed that convergence happened with our simulated users, but here we want to ensure that it does also happen with real ones.

In more detail, to demonstrate the market convergence for each user, we evaluate whether the bidding prices for all UPQ levels[1] have an overall tendency to converge to their corresponding equilibrium prices (meaning that the bidding prices for items with a UPQ value of 1 tend to converge to the equilibrium price for UPQ of 1, prices for items with a UPQ value of 2 tend to converge to the equilibrium price for UPQ of 2, and so on). In our previous simulations (section 4.2), we evaluated the convergence by validating whether the bidding prices for each shortlisted

[1]The UPQ of a recommendation is identical to the user's rating throughout this chapter. This is because by definition a real user's rating for an item reflects the user's perceived quality of that item.

advertisement slot converged to a small oscillation around a constant level after a number of auction rounds. Thus this approach directly observes the economic market equilibrium price (the point where the demand meets the supply) for each advertisement slot. However, in practice, we need quick market convergence so as to quickly suggest high quality recommendations to a real user without spending a vast amount of time (i.e. hundreds of auction rounds as per section 4.2) before good recommendations come out. Therefore, instead of evaluating how prices deviate from corresponding equilibria of different advertisement slots, we seek to evaluate the tendency towards market convergence by evaluating how the bidding prices for each UPQ level deviate from their corresponding equilibria prices. Indeed, we have already demonstrated that the convergence of prices for different UPQs is consistent with the convergence of prices for different advertisement slots in section 3.5. This is because as the bidding prices for different advertisement slots converge, prices of recommendations of a specific UPQ also converge (otherwise, with recommendations of at least one specific UPQ level not converging, prices with respect to advertisement slots do not converge). More formally, with respect to a specific UPQ level $\bar{Q}$, in a specific auction round $\bar{a}$, there are $N_{\bar{Q}}$ recommendations (from whatever constituent recommenders) with bidding prices P_1, P_2, $\cdots$, $P_{N_{\bar{Q}}}$ being rated at $\bar{Q}$ level by a user and the corresponding equilibrium price is $\bar{P}$ (see section 6.3.1 for the definition of equilibrium price for one specific UPQ level). The deviation from equilibrium for $\bar{Q}$ in auction round $\bar{a}$ is then calculated as:

$$D_{\bar{Q},\bar{a}} = \frac{1}{N_{\bar{Q}}} \sum_{i=1}^{N_{\bar{Q}}} |P_i - \bar{P}|. \tag{6.1}$$

With this definition, the ideal market convergence would be that $D_{\bar{Q},\bar{a}}$ converges to zero with $\bar{a}$ increasing for all different $\bar{Q}$s. However, rewards to high UPQ recommendations give more confident incentives to recommenders than those to low UPQ ones. Thus, it takes more time to converge for low UPQ levels than high ones. In some cases, recommenders fail to learn users' interests with respect to very low UPQ recommendations and the market cannot converge on these UPQ levels. Therefore, in practice, we expect $D_{\bar{Q},\bar{a}}$ to converge for most $\bar{Q}$s, especially the high ones.

❸ *Effective Peak Performance*

As discussed in section 1.1, we ideally want the market-based recommender to always perform as well as the best of the constituent recommenders (whatever that is for the given user in the given context). Thus, we view our market-based system as a meta-recommender whose recommendations are those shortlisted items that are displayed to the users. Specifically, we expect to see that the first displayed recommendation (in the first slot of the recommendation sidebar, see Figure 1.1) suggested by our market-based recommender at any auction round is as good, from the user's viewpoint, as the best of the first bid items suggested by all constituent recommenders. This is important because a good recommender system is one that makes the best recommendations. To do this, we define a metric, called *peak performance*. A constituent recommender's peak performance at a given auction round is defined as the UPQ of its first bid item, whereas the market-based recommender's is defined as the UPQ of its first displayed item. Note that in the case of a constituent recommender that has no items shortlisted at an auction round, its local peak performance is zero. Therefore, we expect the market-based recommender's peak performance to be as high as that of the best of the constituent recommenders' for most auction rounds for most users. To do this, we define the *effective peak performance point* as an auction round in which the market-based recommender's peak performance is as high as that of the best of the constituent recommenders'. With this, we statistically evaluate how many times the marketplace performs as well as the best of all constituent recommenders for all users over all auction rounds.

❹ *Best Recommendation Identification*

The previous evaluation metric evaluates the qualities of recommendations from the perspective of comparison between the market-based recommender and its constituent recommenders. However, whether a recommender is able to satisfy a user is eventually decided by the user. Thus, we also need to evaluate the qualities of recommendations from the users' point of view. This is the most important property of any kind of recommender systems. Indeed, whether a user likes a recommender system eventually depends on its ability to identify the best recommendations to him. Here, we define the *best recommendations* as the items of the highest two UPQ levels (i.e. "4" and "5", see section 6.2 for the configuration of the rating value range of recommendations). As discussed in section 4.4, we want our system to be able to identify and frequently suggest the best recommendations to users. To evaluate this, we define two measurements: *qualified recommending round* and *satisfied recommending round*. Specifically, with respect to a particular user, a qualified recommending round means an auction round with at least one best recommendation displayed in any advertisement slot of the recommendation sidebar, whereas a satisfied recommending round means an auction round with at least one best recommendation displayed in any of the first two advertisement slots. Thus, a satisfied recommending round must be a qualified recommending round, but a qualified recommending round may not be a satisfied recommending round. With these two measurements, we evaluate how the numbers of qualified and satisfied recommending rounds, with respect to a given user, are compared to the total number of recommending rounds.

With these metrics in place, we now outline the user trial process.

6.2 User Trials

In this section, we outline the process of the trial from the perspective of a user of our system. In using our system, a user is required to choose a specific browsing topic, browse a set of recommended web documents on that chosen topic, and then rate each document according to how it satisfies him. The user's task can be segmented into two stages: building an interest profile and then browsing and rating the recommendations. The flowchart in Figure 6.1 depicts these two stages and the remainder of this subsection elaborates on the details involved.

Our evaluation involves thirty-one effective user trials from the School of Electronics and Computer Science at the University of Southampton. The user population covers PhD students, post-doctoral researchers and academic staff and they are all researchers in the Intelligence Agents Multimedia Group (IAM Group). In particular, most of them are researchers in the areas of "agents", "artificial intelligence", "machine learning", "knowledge technologies", "automated negotiation", "auctions", "game theory" and "hypermedia".

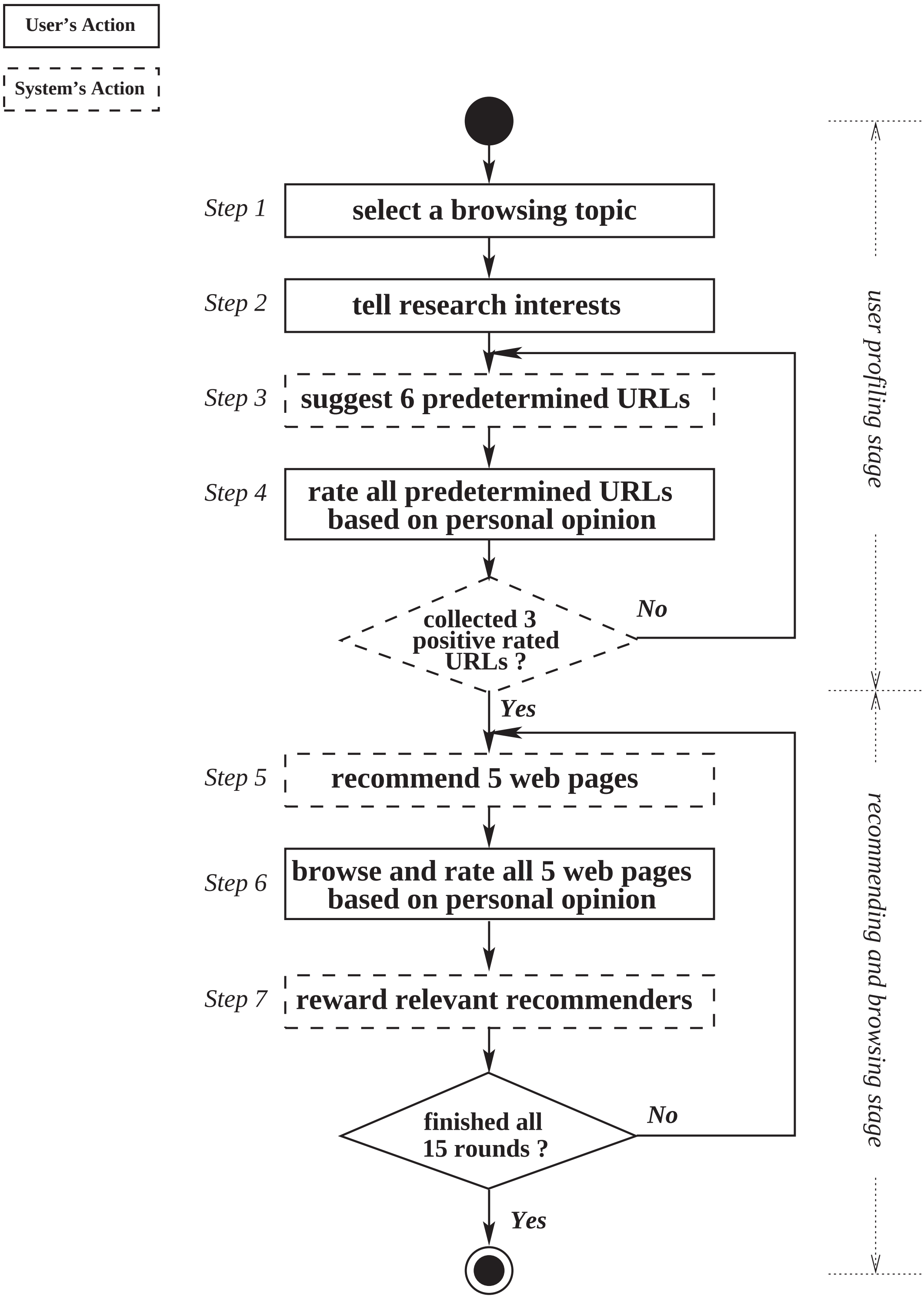


Figure 6.1: A User's Task

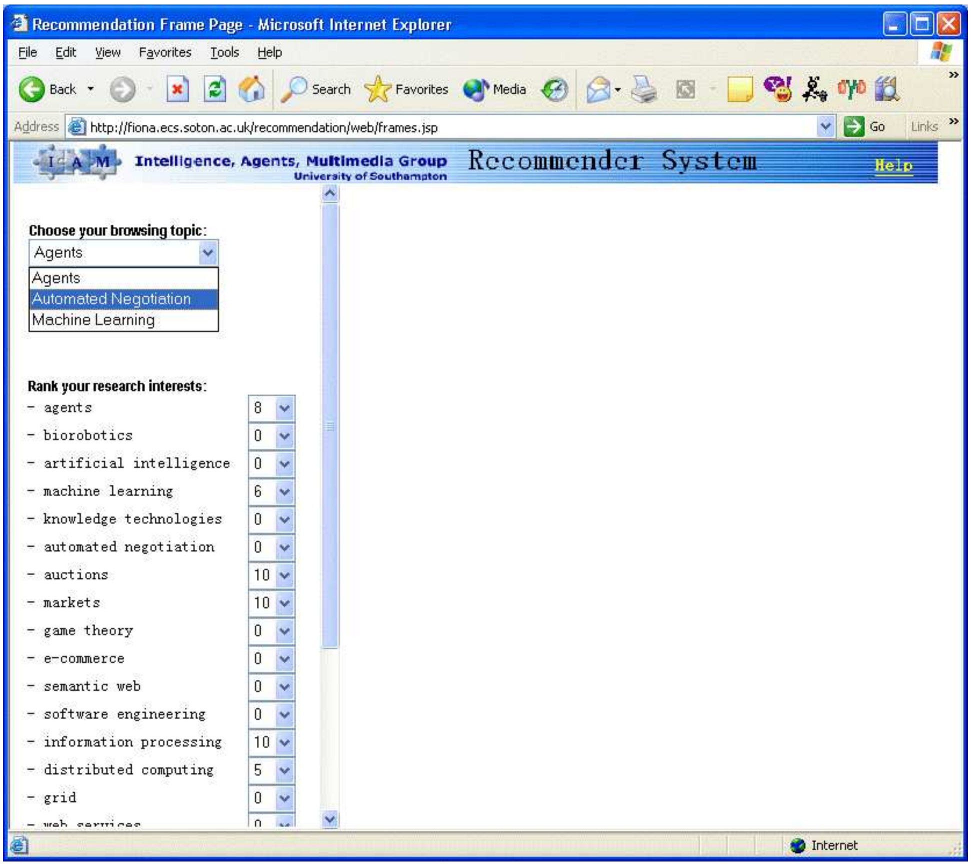


FIGURE 6.2: Selecting Browsing Topic and Telling Research Interests

To start using our recommender system, each user needs to build up a user profile of his interests, based on which, our system suggests its recommendations. There are four steps in the user profiling stage (as per the upper part of Figure 6.1). In the first step, a user needs to select a browsing topic. Here a *browsing topic* is an interesting topic that a user chooses as his browsing context throughout his browsing and rating task (i.e. the whole trial). Figure 6.2 shows a screenshot of an example of choosing browsing topics when a user faces our system. As shown in the upper corner of the sidebar of Figure 6.2, a dropdown box comprises three predetermined topics for a user to choose. They are: *Agents, Automated Negotiation and Machine Learning*. The three topics are chosen according to the most popular research topics within the IAM Group based on an email survey of the thirty-one people.

In order to recommend good documents, the system needs to learn the users' interests.

Thus, each constituent recommender needs to build a user profile as the basis to compute its recommendations[2]. Since it is a difficult and complex process to precisely and automatically profile a user's interests [Middleton et al., 2004] and because it is not the main focus of this work, we decided to profile users' interests in a straightforward way as follows. From step 2 to step 4 in Figure 6.1, three kinds of user interest profiles are built (one for each of the three recommendation methods). In the remaining phase of the trial described as the "recommending and browsing stage" in Figure 6.1, a user is required to rate a set of keywords that may be relevant to his research interests in the field of computer science. These words are[3]: *agents, biorobotics, artificial intelligence, machine learning, knowledge technologies, automated negotiation, auctions, markets, game theory, e-commerce, semantics, software engineering, information processing, distributed computing, grid, web services, networks, security, trust, mobility, ontologies and hypermedia*. The user is required to rate all these topics according to how important they are to his research interests (see the dropdown boxes in the lower part of the sidebar in Figure 6.2 for part of the twenty-two topics). A rating number is limited to the range between "0" and "10": "0" indicates totally irrelevant, "1" indicates weakly relevant and "10" indicates perfectly relevant. Based on the ratings of these topics, two user profiles for collaborative and demographic recommendation methods are built respectively (see sections 6.3.3 and 6.3.4 for more details about these two methods). To produce profile for the content-based method, the system randomly recommends six web documents on the user's chosen browsing topic and displays their corresponding URLs in the browser sidebar (see Figure 6.3). We term these recommended URLs in the profiling stage the

[2]With respect to a specific user, because different constituent recommenders compute their recommendations independently and use their user profiles in their own ways, we need to build each recommendation method a separate user profile.

[3]This list was produced by performing an email survey on the most popular research topics within the IAM Group of the University of Southampton. Specifically, we ask people to provide a list of topics that represent their main research interests. We received thirty-four responses. Among all these responses, twenty-two topics appear in at least three responses, whereas the remaining ones rarely appear. Thus, the twenty-two topics are used in our experiment to profile a user's interests. These interest topics are related to the three predetermined browsing topics in the following manner. Only three topics are insufficient to define good correlations between users. To accurately compute different users' correlations, more interest topics define more accuracy. Thus, we use a larger number of topics to specify a user's interests. Therefore, these topics can be seen as an extension of the three predetermined browsing topics. Actually the three browsing topics can be extended to the same set of interest topics. However, in so doing, we need to prepare a huge number of source recommendation web documents. But it is not our concern in this research to recommend as many interest topics and prepare as many source recommendations as possible. Thus, we select only a few of most popular ones as the browsing topics to do our experiment.

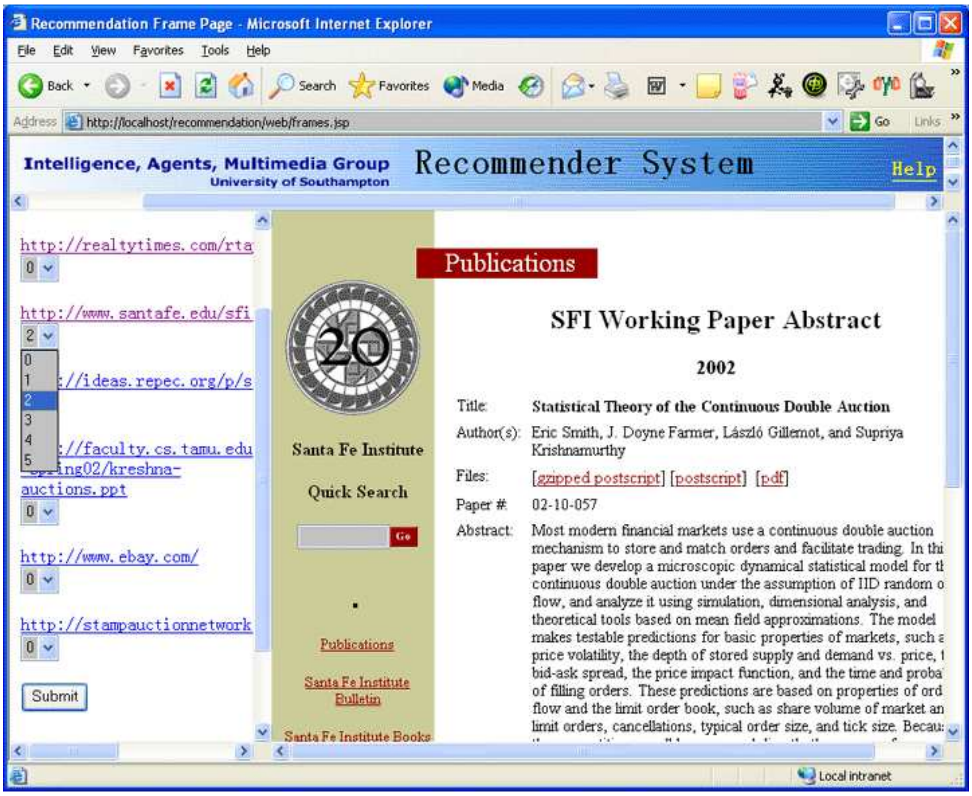


FIGURE 6.3: Rating Predetermined URLS

predetermined URLs[4]. The user is then required to browse all these predetermined URLs and give each a rating according to his personal opinion (how a user rates recommendations is detailed in the end of this section). With these ratings to recommendations, the content-based recommender collects a couple of the most interesting documents and analyzes their contents to produce its user profile. To capture the user's actual interests, three positive-rated URLs are needed. If less than three positive-rated URLs collected in the third and fourth step, steps three and four are repeated until three have been collected. Based on these most interested Web documents, a user profile for content-based recommendation method is produced (see section 6.3.2 for more details about how the content-based method helps recommendations).

After the profiling stage, a user starts his main browsing and rating stage (the last three steps in Figure 6.1). In this stage, the user is each time suggested five Web documents

[4]The predetermined URLs are randomly selected from a separate recommendation pool, whereas the three constituent recommenders have their own recommendation pools. These four recommendation pools do not share any public items.

(step 5 in Figure 6.1). The user is required to browse and give each a rating according to how relevant it is to his research interests (step 6 in Figure 6.1).

For example, a user has a list of five interest topics `(Agents, 6), (machine learning, 4), (auctions, 7), (markets, 9) and (information processing, 10)` (numbers represent their relevance) and seventeen other topics with zero relevance. The user chooses "agents" as his browsing topic and is recommended two Web documents in this broad area. Specifically, one document is on a topic of "using market-based mechanism to coordinate information agents", whereas the other is on "mobile agent's security over the Internet". Thus, the user should rate the former higher than the latter. This is because, besides agents, the former is related to markets and information processing which are also the user's interests, whereas the latter relates to mobility and security which are not. For another example with respect to the same user, a third Web document is suggested on a topic of "agents and machine learning". In this case, the user should prefer the first recommendation to this one because machine learning is less relevant than markets and information processing.

Therefore, a rating to a recommendation Web document is a user's personal opinion about how well it relates to his research interests. A rating number is limited to "0" to "5" (see the dropdown box below each predetermined URL in Figure 6.3), in which, "0" means totally irrelevant, "1" means weakly relevant and "5" means perfectly relevant. We use five positive levels to specify recommendation quality because this number has previously proved to be sufficient and effective in differentiating users' preferences [Resnick et al., 1994, Shardanand and Maes, 1995, Pazzani, 1999][5]. Actually, these rating numbers represent the UPQs of the recommendations by definition. A user's rating number of each recommendation is an absolute value throughout his trial. Thus, if a recommendation is rated by the user in an earlier time, he is required not to change its rating value. Therefore, a user's evaluation criterion of the qualities of recommendations should be consistent throughout his task. With these ratings to recommendations, the system rewards the relevant constituent recommenders to assist their learning about the user's interests (step 7 in Figure 6.1). We term the subroutine of recommending,

[5]We use more levels to define the importance of interest topics because more levels define more accuracy of users' correlations. To avoid too much computation, ten levels are used.

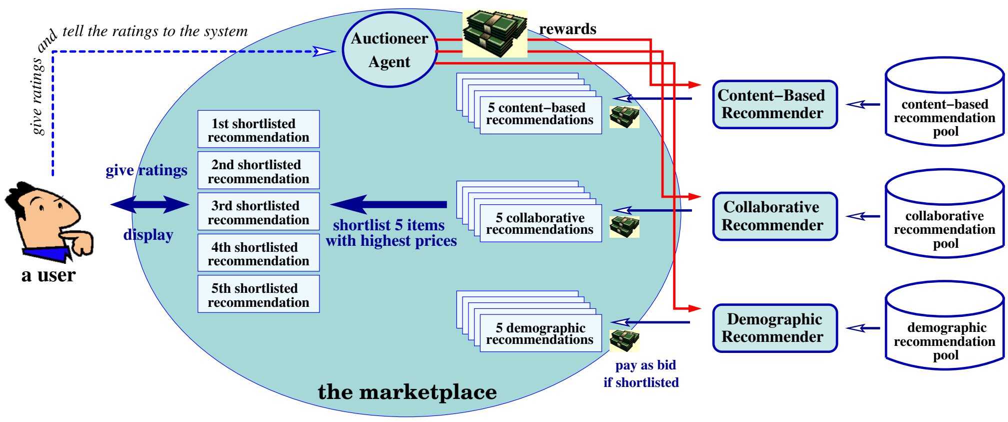


FIGURE 6.4: Configurations of the Market-Based Recommender System

browsing/rating and rewarding a *recommending round* (step 5 to 7 of Figure 6.1). Each user is required to do fifteen rounds to complete his trial.

Knowing what the users are required to do, we then outline how the system suggests recommendations in terms of three component recommendation methods.

6.3 System Configuration

This section describes the configuration of our market-based recommender system (Figure 6.4 depicts its architecture). This section is organized into four main parts: configuration of the marketplace (section 6.3.1), and the details of the three different component recommendation methods (section 6.3.2 to 6.3.4). We choose a number of different recommendation methods since one of the goals of our system is to be able to incorporate and coordinate various recommendation methods and seek out the best items from whichever source is most appropriate (see section 1.1). Specifically, we use three different methods that compute the INQs of their recommendations based on different similarity measures:

- one based on similarity between the current document and those the user has previously indicated as being of interest (i.e. a typical content-based method);
- one based on the correlations between the user's interests and those of other users' (i.e. a typical collaborative method);

- one based on the similarity between the available documents and the user's profile as represented by their keyword topics of interest (i.e. a variation of the typical demographic method[6]).

Therefore, our three component recommendation methods are based on three different similarity measures: document-to-document, user-to-user, and document-to-user. Before we detail the configurations of these component methods, however, we briefly outline the settings of the marketplace that coordinates these methods.

6.3.1 Marketplace Configuration

This section outlines the configurations of all system variables with respect to the marketplace defined in Chapters 3 and 5. We discuss them in the same order as they appeared in the two chapters.

The following four variables are defined in section 3.2. They are briefly described and configured as below.

- S represents the number of constituent recommenders that are incorporated in our marketplace. Thus, $S = 3$ because we have three constituent recommenders.
- T_b represents the duration of marketplace calling for bids. We set $T_b = 5$ *seconds* because, in practice, we find that five seconds allows sufficient time for the three constituent recommenders to compute their recommendations in most cases.
- M represents the number of recommendations that the marketplace calls for in each auction round. We set $M = 5$ because five recommendations do not overburden the users and five is a practical number in terms of the trials.

[6]A typical demographic method analyzes the characteristics of people (such as age, gender and occupation) and groups people with similar characteristics. Then, it analyzes the attributes of recommendations (such as textual descriptions or contents of books, colour of material of clothes and price of products), and, finally, matches people with certain characteristics to recommendations with suitable attributes. We do not analyze people's characteristics by age, gender and the like, but by their research interests since what we recommend are only Web documents that are relevant to a particular set of interesting topics. Thus, we group people by characteristics of their interesting topics and match people to documents with relevant topics. We do not consider this method a content-based one though it also analyzes the textual contents of documents. Rather, we consider it a demographic method. This is because content-based method compute similarities between documents, whereas this method computes similarity between the characteristics of people and the attributes of recommendations. For example, a group of people share the interest topic of "machine learning", thus, all of them would probably be interested in documents related to reinforcement learning.

- N represents the number recommendations that are rated by a user in one recommending round. We set $N = 5$ because we require each user to rate all five recommendations in each recommending round. Thus, having rated all displayed recommendations, the marketplace gives as much information about the user: interests to the recommenders as possible.

The following variables are defined in Equation 3.18 in section 3.3.4. They are configured as below.

- Q_h represents the UPQ of a shortlisted recommendation. We set $Q_h \in \{0, 1, 2, 3, 4, 5\}$ because, in section 6.2, we have argued that five positive levels are sufficient to specify the qualities of relevant recommendations and, additionally, a level of zero indicates irrelevant ones.

- δ controls the amount of reward to good recommendations and α controls the signal of deviation from equilibria that are delivered to the recommenders. We keep $\delta = 1.5$ and $\alpha = 1.5$ as per section 4.1.1 because they have been proved to be effective and we want our user evaluation to also be effective based on our empirical studies in Chapter 4.

- $\bar{P}_h$ represents the market equilibrium price for recommendations with UPQ level of h (as we stated in section 6.1❷, we do not use P_h^* defined in section 3.2 to evaluate market convergence). However, to kick-start the process, so that we do not need lots of recommending rounds to get good recommendations, we set some constant values rather than let the marketplace find them all out for itself. Based on our empirical study in section 4.2, we set these constant values to $\bar{P}_1 = 110$, $\bar{P}_2 = 120$, $\bar{P}_3 = 130$, $\bar{P}_4 = 140$ and $\bar{P}_5 = 150$ to differentiate recommendations with different UPQs. The system gives clearer incentives to recommenders by fixing these equilibrium prices to constant levels rather than by using the historical average bidding prices for different UPQ levels. This is because the historical averaging prices change from one auction round to another and this makes the rewards to the same UPQ recommendations in different auction rounds differently. Consequently, this makes it difficult for a recommender to learn the user's preferences (reflected by rewards) to its recommendations. Moreover, since the constituent recommenders do not know these fixed equilibrium price values, a marketplace with a set of fixed equilibrium prices does not affect their bidding strategies.

- P_{M+1} represents the highest bidding price that is not shortlisted in one auction round. It is the basic unit reward and it controls the amount of actual reward to an agent together with δ. Like setting $\bar{P}_h$ with constant values, we set $P_{M+1} = 100$ in order to give quick and clear incentives to recommenders (because constant unit reward makes it easier for recommenders to learn the bidding price deviations from the equilibria).

The last system variable we need to discuss was defined in section 5.2 for agents to learn users' interests.

- G represents the number of INQ segments of a constituent recommender. In practice, based on our computation of the INQs of the recommendations of our three constituent recommenders (detailed in sections 6.3.2, 6.3.3 and 6.3.4), we find that six segments are sufficient to effectively differentiate recommendations in terms of INQ. Thus, we set $G = 6$ for our three constituent recommenders.

6.3.2 The Content-Based Method's Configuration

A content-based method suggests recommendations based on the contents of a user's previously top rated documents. Therefore, before it can start recommending, this method needs to learn something about documents that the user thinks are valuable. In our case this is the purpose of the system collecting at least three positively rated predetermined URLs in the user-profiling stage (see Figure 6.1). To this end, the top rated document of the three is chosen as the initial basis for this method since this is the one the user currently likes best. Thus this method recommends web documents based on their similarity to the top rated predetermined URL. In the recommending and browsing stage (see Figure 6.1), if subsequent Web pages with even higher ratings are uncovered, then these become the basis for recommending based on content. If there is more than one page with the same highest rating value, all of them are used[7].

To be more precise, let P_1, P_2, $\cdots$, P_{N_c} be the N_c ($N_c \leqslant 3$) previously rated different Web pages with the same top rating value R_v. Then let P_P be a potential Web document to be recommended. Now, the internal quality of this potential Web page is computed by the similarity between the N_c top rated pages and itself:

$$INQ_{con}(P_P) = \frac{1}{N_c} \sum_i Similarity(P_P, P_i) * R_v \tag{6.2}$$

In Equation 6.2, the subscript of the function is used to differentiate it from the other two methods introduced in sections 6.3.3 and 6.3.4. The similarity measure will be formally discussed in the end of this subsection. Thus, the content-based method compares the source recommendation web pages to previously top rated pages and recommends those with high similarity values. To compute the similarity value, we extract fifteen keywords with the highest term frequency (TF) from each document[8]. Actually, the more keywords extracted the more accurate they are able to stand for a document. However, extracting a large number of keywords induces much computation and affects the efficiency of recommending. In practice, therefore, we find that fifteen frequently occurring keywords are able to cover the meaning that a document delivers in most cases. Thus, a source web page is represented as a fifteen-dimensional term vector (for reasons of computational simplicity we do not use more keywords). The similarity measure of two web pages is then computed using the standard technique of considering the cosine between two vectors with a result value between 0 and 1.0, where 0 indicates not strongly related and 1.0 indicates very strongly related [Salton, 1989]. These fifteen most frequently occurring words are then stored in a Content-Based Recommendation Table (see Table 6.1 for part of the actual table we use). Likewise, the predetermined URLs are prepared with the fifteen most frequently occurring terms and their TFs in a

[7] We seek to use a common one for each of the three typical kinds of recommendation method (i.e. content-based, collaborative and demographic) described in Chapter 2. However, we do not aim to refine each method to a perfect one because we are not aiming to build perfect information filtering methods and this is not our main concern in this work.

[8] To extract the most frequently occurring keywords from a web document, a lookup table is used to filter out unimportant words that do not make sense in our context and need to be ignored (such as "a", "the", "in", "that" and "and"). This look up table is constructed according to Middleton's work [Middleton, 2003]. Meanwhile, a stop-list technique also taken from Middleton's work is used to match different words with the same meaning. For example, "negotiation", "negotiations", "negotiating" and "negotiated" are tokenized into "negotiat" and are all deemed the same word.

TABLE 6.1: The Content-Based Recommendation Table

RecID	URLs	K1	W1	K2	W2	K3	W3	K...	W...	K15	W15
160	http://agents.umbc.edu/	agent	33	web	13	robot	13	... ...	... ...	rdf	5
161	http://www.agentlink.org/resources/clearing-house-view.html	agent	25	system	17	control	9	... ...	... ...	develop	4
162	http://www.msagentring.org/	ring	26	soft	25	agent	24	... ...	... ...	copi	4
163	http://www.agentland.com/	agent	18	intellig	7	internet	5	... ...	... ...	site	2
164	http://www.w3.org/TR/WAI-USERAGENT/	docum	28	user	24	access	17	... ...	... ...	version	5
165	http://www.ai.sri.com/~oaa/main.html	agent	70	oaa	28	languag	16	... ...	... ...	facilit	6
166	http://www.agentbuilder.com/Documentation/whyAgents.htm	agent	107	softwar	43	system	35	... ...	... ...	implem	8
167	http://www.agentbuilder.com/Documentation/product.html	agent	40	develop	24	agentbuild	16	... ...	... ...	java	6
168	http://www.objs.com/agent/	agent	83	workshop	36	intern	35	... ...	... ...	group	11
169	http://www.cs.toronto.edu/km/aometh/methodologies.htm	agent	14	system	11	methodologi	9	... ...	... ...	editor	3

TABLE 6.2: The Predetermined URL Table

RecID	Topics	URLs	K1	W1	K2	W2	K...	W...
1	Agent	http://people.cs.uchicago.edu/~firby/aap/	project	15	agent	13	... ...	... ...
2	Agent	http://www-2.cs.cmu.edu/~softagents/	agent	33	sycara	12	... ...	... ...
3	Agent	http://www.davidreilly.com/topics/software_agents/	agent	23	softwar	11	... ...	... ...
4	Agent	http://www.engr.uconn.edu/masa/	nbsp	78	comput	15	... ...	... ...
5	Agent	http://www.ercim.org/publication/Ercim_News/enw53/franklin.html	ida	17	agent	17	... ...	... ...
6	Agent	http://www-2.cs.cmu.edu/~softagents/project_grants_NSF.html	nbsp	75	robot	9	... ...	... ...
7	Agent	http://www.irishscientist.ie/2002/contents.asp?contentxml=02p120.xml&content	agent	34	commun	17	... ...	... ...
8	Agent	http://www.ercim.org/publication/Ercim_News/enw30/gagliardi.html	electron	11	commerc	10	... ...	... ...
9	Agent	http://www.rpi.edu/dept/llc/hci/web/Agents.html	agent	5	softwar	3	... ...	... ...

Predetermined URL Table in the similar style (see Table 6.2 for part of the actual table we use). As stated in section 6.2, the contents of the Predetermined URL Table do not overlap those in the Content-base Recommendation Table, nor do they with the source recommendation tables presented in the subsequent two recommendation methods (see sections 6.3.3 and 6.3.4).

From the Content-Based Recommendation Table, we can see that each potential Web document is represented by a record of that table. Each record contains a vector of fifteen dimensions decided by the fifteen keywords and each dimension has a value of the keyword's TF. With respect to a specific record, k_i represents the i^{th} most frequently occuring keyword in a document and w_i represents the times it occurs (i.e. the TF). With this representation, we are going to formally discuss the similarity measure of two Web documents. Assuming P_x and P_y are two different documents and they each have fifteen keywords that are the same $(k_1, k_2, \cdots, k_{15})$. Thus, the two documents can be represented by two vectors in the same fifteen-dimensional Euclidean space: $\mathbf{x} = (w_1, w_2, \cdots, w_{15})$ and $\mathbf{y} = (w'_1, w'_2, \cdots, w'_{15})$ where w_i and w'_i are their keyword k_i's TFs respectively. The INQ of P_x is defined as the cosine of the two term vectors of P_x and

P_y [Salton, 1989]:

$$
\begin{aligned}
Similarity(P_x, P_y) &= \cos(\mathbf{x}, \mathbf{y}) \\
&= \frac{\mathbf{x} \times \mathbf{y}}{|\mathbf{x}| \cdot |\mathbf{y}|} \\
&= \frac{\sum_{i=1}^{15}(w_i \cdot w_i')}{\sqrt{\sum_{i=1}^{15}(w_i)^2} \cdot \sqrt{\sum_{i=1}^{15}(w_i')^2}},
\end{aligned} \tag{6.3}
$$

where $\mathbf{x} \times \mathbf{y}$ is the inner product of $\mathbf{x}$ and $\mathbf{y}$, and $|\mathbf{x}| = (\mathbf{x} \times \mathbf{x})^{\frac{1}{2}}$ is the Euclidean norm of $\mathbf{x}$. However, in many cases, two vectors share only a few keywords (less than fifteen or even none). In these cases, the inner product of the two vectors cannot be computed directly, since they are not in the same Euclidean space. Therefore, both vectors will be converted into the same space such that their cosine similarity is computable. For example, a different Web page $P_{y'}$ contains fifteen most frequently appeared keywords $(k_1, k_2, k_3', \cdots, k_{15}')$, where $k_i \neq k_i', i \in [3..15]$ compared to that of P_x. $P_{y'}$ can be represented by its TF vector as $\mathbf{y'} = (w_1^\star, w_2^\star, \cdots, w_{15}^\star)$ (no weighting value is zero). Thus, $P_{y'}$ shares only the first two keywords with P_x. In this case, the similarity between P_x and $P_{y'}$ is simplified as:

$$
\begin{aligned}
Similarity(P_x, P_{y'}) &= \cos(\mathbf{x}, \mathbf{y'}) \\
&= \frac{w_1 \cdot w_1^\star + w_2 \cdot w_2^\star}{\sqrt{\sum_{i=1}^{15}(w_i)^2} \cdot \sqrt{\sum_{i=1}^{15}(w_i^\star)^2}}.
\end{aligned} \tag{6.4}
$$

6.3.3 The Collaborative Method's Configuration

We use a standard collaborative method based upon Pazzani's model (see section 2.1.4). This method computes recommendations based on other users' ratings to the current item. Assuming N_s users $(\mathbf{u_i}, i \in [1..N_s])$ have similar interests with the active user $(\mathbf{u})$ and their ratings for a potential recommendation web page (P_p) are $R_i, i \in [1..N_s]$, the prediction of the INQ of P_p is defined as

$$
INQ_{col}(P_p) = \frac{\sum_{i=1}^{N_s}(\gamma_{u,u_i} \times R_i)}{\sum_{i=1}^{N_s} \gamma_{u,u_i}}, \tag{6.5}
$$

where γ_{u,u_i} represents the Pearson-γ collaborative correlation [Shardanand and Maes, 1995] between the active user and the i^{th} user with similar interests.

However, before discussing how to compute Pearson-γ correlation between two users, we need a way of representing users' interests. With the users' interests profile based on ratings for twenty-two interest topics (see section 6.2), a user can be represented as a 22-dimensional vector of these topics. For example (the sequence of these 22 fields is in the same order among all users), a vector of (8, 0, 10, 0, $\cdots$) represents a user whose interests are related to those topics with positive field values (such as agents, artificial intelligence, $\cdots$ in this case) and not related to any topics with a field value of "0". Thus, the positive field values represent the extent to which their corresponding topics are relevant. With this in place, we can illustrate the correlation between two different users as follows. Given two users represented as two user vectors $\mathbf{u} = (u_1, u_2, \cdots, u_{22})$ and $\mathbf{v} = (v_1, v_2, \cdots, v_{22})$, their Pearson-$\gamma$ correlation is defined as

$$\gamma_{u,v} = \frac{\sum_{i=1}^{22}(u_i - \bar{u}) \cdot (v_i - \bar{v})}{\sqrt{\sum_{i=1}^{22}(u_i - \bar{u})^2 \cdot \sum_{i=1}^{22}(v_i - \bar{v})^2}}, \tag{6.6}$$

where $\bar{u} = \frac{1}{22}\sum_{i=1}^{22} u_i$ and $\bar{v} = \frac{1}{22}\sum_{i=1}^{22} v_i$. The user vectors are stored in the User Profile Table (see Table 6.3 for an example of the actual table we use). The recommendation Web pages are stored in the Collaborative Recommendation Table. This table has the same structure as that of the Content-Based Recommendation Table which we do not show in a separate table here. In the Collaborative Recommendation Table, the fifteen most frequently occurring keywords and their TFs are used to solve the cold start problem that bedevils this approach (see the subsequent paragraph). The users' ratings for each recommendation Web document are stored in the Rating Table (see Table 6.4 for part of the actual table we use).

TABLE 6.3: The User Profile Table

UserID	u1	u2	u3	u...	u21	u22
0	10	0	6	... ...	4	10
1	8	0	0	... ...	0	6
2	1	1	0	... ...	6	3
3	9	2	8	... ...	0	7
4	2	6	7	... ...	0	0
5	8	0	7	... ...	0	7
6	9	1	8	... ...	5	7
7	8	0	0	... ...	0	6
8	10	0	6	... ...	0	5
9	0	0	0	... ...	0	0

TABLE 6.4: The Rating Table

recID	userID	rating
104	22	3
112	22	4
118	22	0
116	22	3
88	23	3
75	23	4
81	23	3
67	23	2
72	23	4

Column "userID" in this table corresponds to column "ID" in the User Profile Table, whereas Column "recID" corresponds to column "RecID" in the Collaborative Recommendation Table. For example, the first record in this table represents the fact that user "22" gives a rating "3" to a Web document that stored in the Collaborative Recommendation Table with a RecID of "104".

To make the collaborative recommendation method work effectively in our system, we need to overcome the cold start problem. As discussed in section 2.1, when the first few users start to use the system, the system is unlikely to have any other users with similar interests. Thus, it will have no users' ratings for any of the source recommendation web pages from which to predict their INQs for the active user. To solve this problem, a "collaboration via contents" technique is used to predict the INQ of the source recommendations (see section 2.1.4). Thus, for each potential web page selected for recommendation, a rating value within the range of [0..5] is assigned by the filtering agent through computing the number of keywords shared between the document and the user's interests (see the demographic method in Section 6.3.4 for more details). Thus, when there are an insufficient number of similar users, it is still possible to predict their INQs using this method.

6.3.4 The Demographic Method's Configuration

The third kind of recommendation method we use is introduced based on document-user correlation. This method directly measures how a recommended Web page's content is related to a user's interests [Moreau et al., 2002]. Specifically, this method computes the number of keywords in a user's interest profile (see Table 6.3) that overlap with a document's fifteen most frequently occuring keywords (as per section 6.3.2). In this case, all recommended Web documents are stored in the Demographic Recommendation Table (this table is not shown separately because it is structured in the same way as the Content-Based Recommendation Table). Then assuming that a user's interest profile contains a set of keywords $K_u = \{k_1, k_2, \cdots, k_{22}\}$ and that a potential recommendation Web document (P_p) is represented by its set of keywords $K_p = \{k_1, k_2, \cdots, k_{15}\}$, we can compute the set intersection as $K = K_u \cap K_p$. Given this, the INQ of this recommendation with respect to this user is defined as:

$$INQ_{dem}(P_p) = \begin{cases} |K|, & \text{if } |K| \leqslant 4; \\ 5, & \text{if } |K| > 4, \end{cases} \tag{6.7}$$

where $|K|$ represents the number of elements contained in set K. Equation 6.7 indicates that the more keywords a potential Web document shares with a user's interest profile, the higher its INQ is. However, based on our experiments, there are few documents that share more than six keywords with a user's interest profile and, thus, we make all such documents' INQs equal to 5 in this case.

6.4 Evaluation and Discussion

Having configured the marketplace and the three constituent recommenders, we then implemented a system that helps users to find web documents relevant to their research interests. Based on our thirty-one successful user trials, the rest of this section reports on the evaluations we performed with respect to the metrics outlined in section 6.1.

6.4.1 Balanced Output Contributions

To evaluate the different constituent recommenders' eventual contributions to the users, we record each constituent recommender's output contribution to each user trial (as per section 6.1❶). We then compute the percentage of each constituent recommender's output contribution to each user over the complete trial. This information is recorded in Table 6.5 along with the standard deviation of three methods' contributions with respect to each individual user. Given n statistical samples (x_1, x_2, $\cdots$, x_n), their standard deviation is defined by:

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}},$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. We are interested in the standard deviation in this context because it literally indicates the differences among the three methods' contributions (the bigger it is, the more likely a method is to dominate in the marketplace). In this case, we choose the second deviation (15.28 with respect to user "2") as the criterion to differentiate whether or not domination occurs. This is because, with respect to a specific user, if the deviation is bigger than or equal to this value, there must be one constituent recommender that contributes 2.5 times (see the second item in Table 6.5) more output contributions than (or as many as) another. This, we feel, is a quantified view of dominance.

In Table 6.5, the first column shows the (anonymized) identity of the users. The second, third and fourth columns show, in percentage terms, the different constituent recommender's output contributions to each user. The last column shows the standard deviation of the three recommenders' contributions. From this, we can see that there are twenty-four user trials where no one method dominates, three trials dominated by the content-based recommender, and two trials dominated by the collaborative and the demographic recommenders respectively (visually depicted in Figure 6.5). This means that in most cases (77.42%) all three constituent recommenders make significant output contributions. From this, we conclude that the auction and reward mechanisms we have designed do not encourage domination in the marketplace.

The above analysis is based on individual users. However, we can also evaluate the overall contributions of the different recommenders to all users. This is important because it gives us an insight into the difference among the overall contributions of different recommenders. When we add up each individual recommender's output contributions to all users, they contribute 35.1% (content-based), 30.8% (collaborative) and 34.1% (demographic) of the recommendations displayed to the users respectively (see Figure 6.6). This indicates that, broadly speaking, each of the three constituent recommenders contribute about the same number of output contributions to the users. Again, this result shows that the marketplace is not biased towards any specific method.

TABLE 6.5: Different constituent recommenders' Output Contributions

User ID	**Content-Based Recommender's Output Contribution**	**Collaborative Recommender's Output Contribution**	**Demographic Recommender's Output Contribution**	**Standard Deviation of Three Contributions**
1	72 ★	20	8	34.02
2	50 ★	20	30	15.28
3	37.14	14.29	48.57 ★	17.46
4	28	53.33 ★	18.67	17.93
5	32	26.67	41.33	7.42
6	25.34	29.33	45.33	10.58
7	36	26.67	37.33	5.81
8	20	33.33	46.67	13.34
9	32	48	20	14.05
10	41.33	30.67	28	7.05
11	40.69	28.28	31.03	6.52
12	32	25.33	42.67	8.75
13	23.81	35.24	40.95	8.73
14	33.33	29.34	37.33	4.0
15	45.33	30.67	24	10.91
16	28	28	44	9.24
17	44	28	28	9.24
18	33.33	29.34	37.33	4.0
19	40	29.33	30.67	5.81
20	20	62.67 ★	17.33	25.44
21	32	33.33	34.67	1.34
22	22.67	30.67	46.66	12.22
23	22.67	40	37.33	9.33
24	40	22.67	37.33	9.33
25	54.67 ★	17.33	28	19.23
26	41.33	32	26.67	7.42
27	22.67	38.67	38.67	9.24
28	29.33	38.67	32	4.81
29	42.67	21.33	36	10.91
30	37.33	8	54.67 ★	23.59
31	29.33	44	26.67	9.33

A contribution with a ★ indicates its domination in the corresponding user trial.

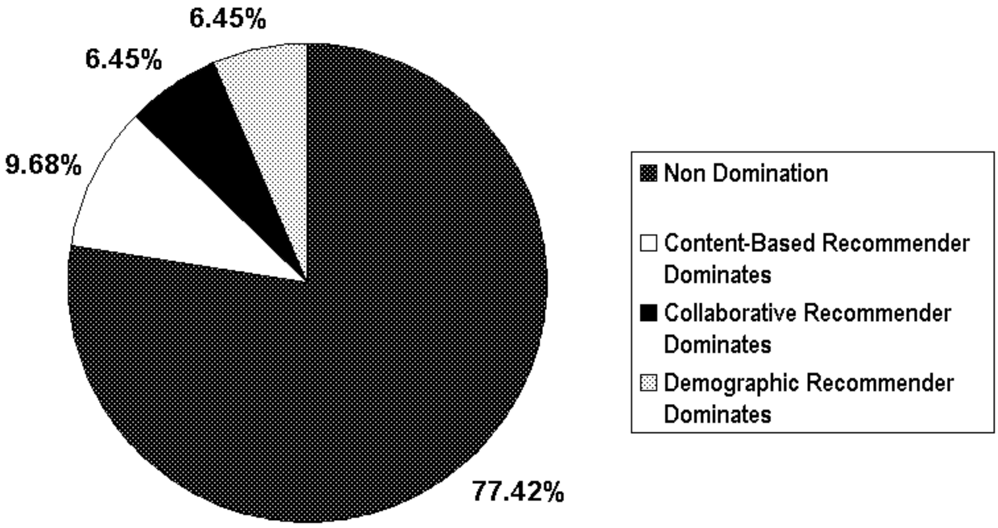


Figure 6.5: Domination in the Marketplace

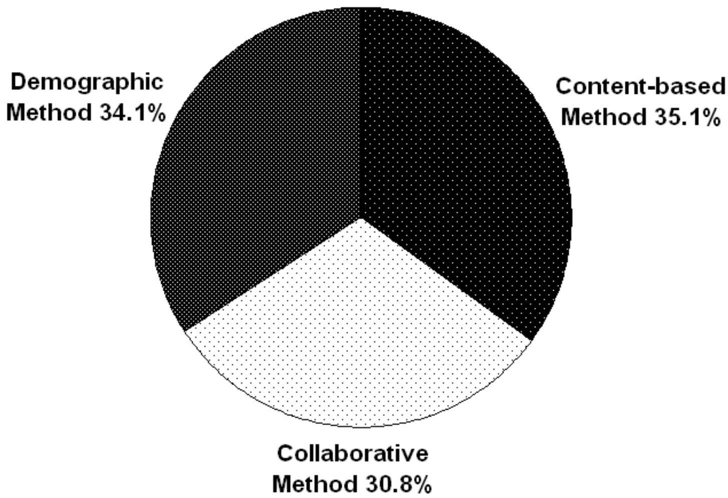


Figure 6.6: Different Constituent Recommenders' Overall Output Contributions

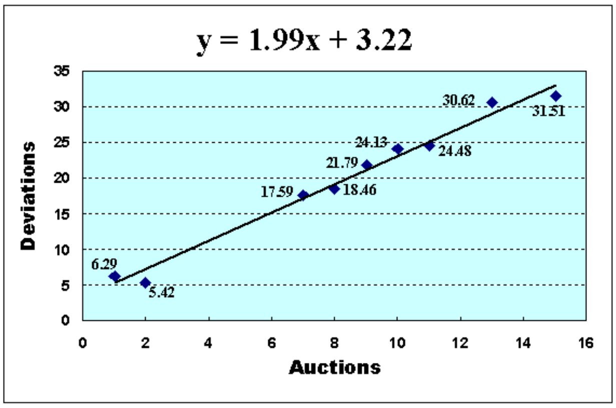
6.4.2 Market Convergence

In this subsection, we are going to validate market convergence by evaluating the bidding price deviations from the equilibria defined in section 6.1 (Equation 6.1). To this end, we collected the results of the thirty-one user trials and plotted all price deviation points over auction rounds for each UPQ level with respect to each user. For example, Figure 6.7(a) depicts the price deviation points over fifteen auctions for the UPQ level of "1" with respect to user "22". In this figure, the x-axis represents the fifteen consecutive auctions and the y-axis represents the value of the price deviation from equilibrium. As can be seen, there are nine auctions with recommendations rated at level "1" by this user and their price deviations from the corresponding equilibrium $\bar{P}_1$ (110) are comparatively small. Then, when we overlay a linear trend line over these deviation points: we get a line of $y = 1.99x + 3.22$ (all such coefficients for each UPQ level with respect to each user are summarized in Table 6.6). From the figure, we can see that the linear trend line with a large positive coefficient tends to increase as the auction rounds increase. For another example, Figure 6.7(b) depicts the price deviation points for the UPQ level of "4" for the same user. As can be seen, its linear trend line with a big negative coefficient tends to decrease as the auction rounds increase. For yet another example, Figure 6.7(c) depicts the price deviation points for the UPQ level of "3" for user "3". In this case, the linear trend line has a small negative coefficient (-0.58) which means that it is hard to judge the tendency to increase or decrease for this UPQ level for this user.

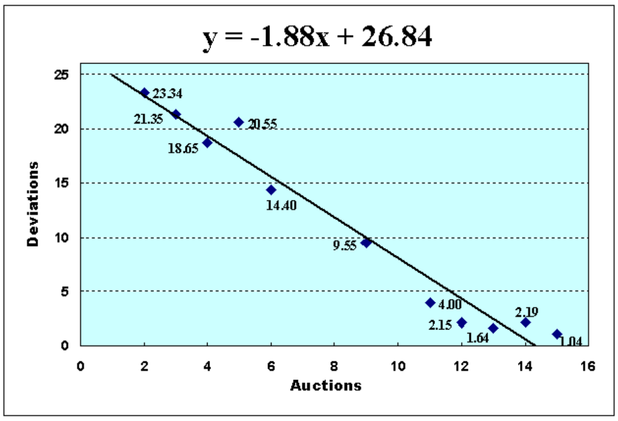
Based on the thirty-one user-trials, we find that all trend lines with a big positive deviation coefficient (larger than 0.58) tend to increase as the auction rounds increase, all those with a big negative deviation coefficient (smaller than -0.58) tend to decrease, and all those with a very small deviation coefficient's absolute value (between -0.58 and 0.58) do not have a clear tendency to increase or decrease. Based on this, we define that *any trend line that has a positive deviation coefficient larger than 0.58 diverges; that has a negative coefficient smaller than -0.58 converges, and that has a coefficient's absolute value smaller than 0.58 is unclear of convergence or divergence*. With this definition in place, Table 6.7 details the convergence for each UPQ level for each user (based on Table 6.6).

Based on this individual data, we are now in a position to evaluate how the system performs in terms of convergence for the user trial population. From Table 6.7, it can be seen that the price deviations of the top effective rating levels[9] for most of the users tend to converge and the price deviations of the lowest effective rating levels for most of the users tend to diverge. Specifically, if we exclude the rating levels where there are an insufficient number of deviation points, there are twenty-eight users that have their top rating level's deviation points converge and three users where it is unclear (see Figure 6.8). This shows that in almost all cases the recommender agents can learn

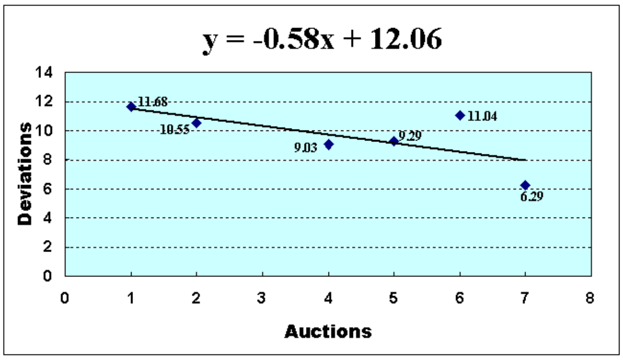
[9]For example, in Table 6.7, UPQ of 4 is considered the top rating of user "3" since there are insufficient data in his rating level of 5. Likewise, UPQ of 2 is the lowest effective rating level for this user since there are insufficient data with UPQ of 1.



(a) Divergent Example (Deviations from Equilibrium of UPQ level "1" for User "22")



(b) Convergent Example (Deviations from Equilibrium of UPQ level "4" for User "22")



(c) Unclear Example (Deviations from Equilibrium of UPQ level "3" for User "3")

FIGURE 6.7: Examples of Linear Trend Line of Deviations from Equilibria

Table 6.6: Table of Deviations from Equilibria

User ID	**Deviation Coefficient of UPQ=1**	**Deviation Coefficient of UPQ=2**	**Deviation Coefficient of UPQ=3**	**Deviation Coefficient of UPQ=4**	**Deviation Coefficient of UPQ=5**
1	i *	2.2431	1.4725	1.16	-0.0758
2	1.1846	i	-0.482	1.0163	-0.1799
3	i	-0.7188	-0.5793	-0.8951	i
4	1.3584	0.6709	-1.0621	-1.1711	-1.2193
5	i	1.8647	0.0251	-1.4178	i
6	2.4336	1.429	-0.3855	-1.8994	i
7	1.9331	1.8553	0.0322	-2.6029	-2.162
8	-0.8877	i	i	i	i
9	2.3799	2.0587	0.1701	-1.3394	-2.0863
10	1.9146	1.4524	-0.2918	i	i
11	-0.8575	0.0711	i	i	-1.1789
12	2.0133	1.4123	0.4087	-1.9071	-2.2316
13	1.5394	1.2418	-0.9607	-1.0426	i
14	2.4343	1.2828	0.6611	-1.1131	-2.1542
15	1.466	0.6606	-1.6457	-1.0571	-1.8371
16	1.0512	-0.0168	-0.8276	i	i
17	1.6033	-1.6873	-1.6588	-3.4476	i
18	2.1997	1.1581	0.359	-1.6291	-2.1904
19	2.0186	1.814	0.5971	-1.2874	i
20	1.9116	1.9783	-0.8311	-1.2845	i
21	2.8543	1.6986	-0.9827	-1.5299	-2.5208
22	1.9912	1.296	-0.2344	-1.8763	-1.8136
23	i	1.5569	i	i	-2.1535
24	1.4248	1.0002	-0.9642	-0.8143	-1.2109
25	1.3664	1.0361	-1.0033	-1.5027	-1.6487
26	i	2.321	-0.3399	-1.695	-2.3186
27	2.2501	1.8482	0.3385	-2.4102	-2.0388
28	i	1.4562	-0.5451	-1.9798	-1.5968
29	i	1.7507	0.1352	-1.5527	-2.9792
30	2.2777	1.3224	-0.0797	-1.5096	-1.9212
31	1.5925	0.9022	-0.6588	-2.1787	-2.1907

* "i" indicates that there are insufficient deviation points (less than four) to judge converge/diverge.

TABLE 6.7: Table of Convergence

User ID	**Convergence of UPQ=1**	**Convergence of UPQ=2**	**Convergence of UPQ=3**	**Convergence of UPQ=4**	**Convergence of UPQ=5**
1	i	d	d	d	u
2	d	i	u	d	u
3	i	c	u	c	i
4	d	d	c	c	c
5	i	d	u	c	i
6	d	d	u	c	i
7	d	d	u	c	c
8	c	i	i	i	i
9	d	d	u	c	c
10	d	d	u	i	i
11	c	u	i	i	c
12	d	d	u	c	c
13	d	d	c	c	i
14	d	d	d	c	c
15	d	d	c	c	c
16	d	u	c	i	i
17	d	c	c	c	i
18	d	d	u	c	c
19	d	d	d	c	i
20	d	d	c	c	i
21	d	d	c	c	c
22	d	d	u	c	c
23	i	d	i	i	c
24	d	d	c	c	c
25	d	d	c	c	c
26	i	d	u	c	c
27	d	d	u	c	c
28	i	d	u	c	c
29	i	d	u	c	c
30	d	d	u	c	c
31	d	d	c	c	c

"i" indicates that there are insufficient deviation points to judge converge/diverge, "c" represents convergence, "d" represents divergence and "u" indicates it is unclear whether there is convergence/divergence.

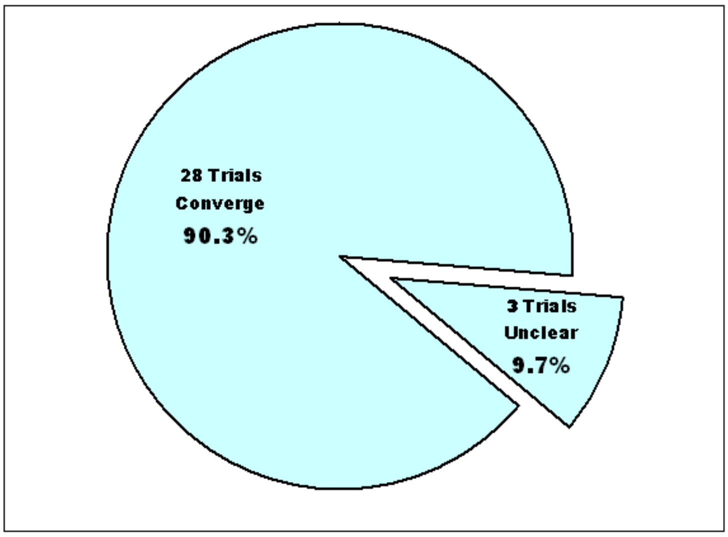


FIGURE 6.8: Convergence of Highest UPQ (excluding unclear levels) Price Deviations

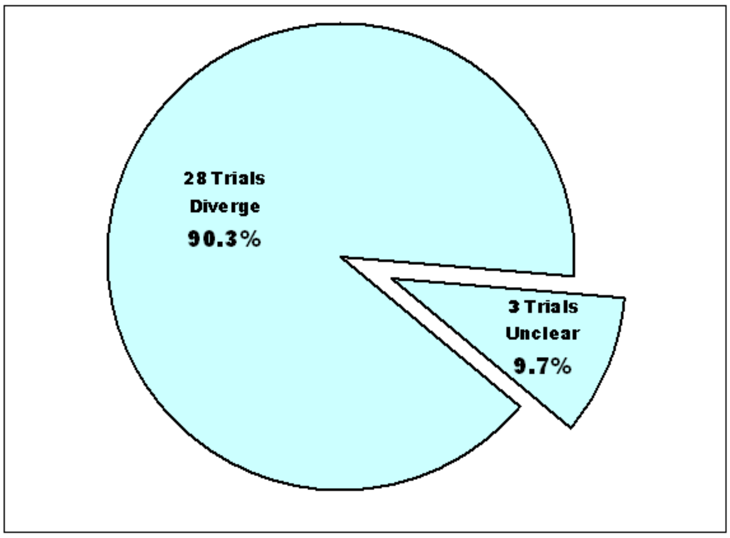


FIGURE 6.9: Convergence of Lowest UPQ (excluding unclear levels) Price Deviations

users' interests quickly when there are recommendations that are highly rated by users. In such cases, the agents receive large rewards and are able to be confident about their recommendations. When evaluating the lowest rating level, again excluding those cases where there are insufficient deviation points, we find that there are twenty-eight users that have their lowest rating level's deviation points diverge and three users where it is unclear (see Figure 6.9). This pattern is observed because the recommender agents learn users' interests slowly when the recommendations are lowly rated by users (since they are more uncertain about such recommendations because they receive a small amount of reward). This divergence is symptomatic of the fact that at these low rating levels the recommender agents are still iterating their bids to chase the corresponding equilibrium (according to the bidding strategy introduced in Table 3.1). However, the deviation cannot diverge to very large values in this case. This is because, on the one hand, recommendations with very high bid prices compared to the corresponding equilibria will receive no rewards and the corresponding greedy bidders will go bankrupt (see section 4.2.5). While, on the other hand, recommendations with very low bid prices compared to the corresponding equilibria cannot be shortlisted and they will try to increase their bid prices in later auctions (see section 3.4). This dual effect makes the price deviations of low UPQ levels uncertain in a small number of recommending rounds: they might converge after some further auctions or they might not (if the recommender agents cannot learn the users' interests with respect to the moderately interesting recommendations).

In sum then, these experiments show the marketplace tends to converge quickly over the recommendations with high user ratings because of the clear incentives associated with the users' interests. However, the marketplace is uncertain about convergence over the recommendations with low user ratings because of weak or unclear incentives. This does not mean that our marketplace does not work in this case. However, to effectively correlate the constituent recommenders' bids to the qualities of the recommendations with low user ratings is subject to a few other issues, such as:

- long learning time. A long learning period enables the constituent recommenders to have more information to learn users' interests so as to be certain about the moderately interesting items;
- good learning effectiveness. A good learning capability enables the constituent recommenders to achieve learning convergence quickly and this affects their bidding; and
- long training process. All kinds of recommender system need a training process to learn users' interests [Middleton et al., 2004]. However, our system is somewhat weak in such process and the constituent recommenders are uncertain about the moderately interesting items.

However, these issues are beyond the scope of this work (see the discussions in Chapter 7 where some thoughts on addressing these issues are given).

6.4.3 Effective Peak Performance

To evaluate whether our market-based recommender's peak performance is indeed above that of all the constituent recommenders', we recorded their peak performance points for all users over all auction rounds. For example, Figure 6.10 shows the marketplace's effective peak performance points versus those of the three constituent recommenders with respect to user "5". From Figure 6.10, we can see that the market-based recommender's effective peak performance points are at the first, third, fourth, fifth, sixth, seventh, ninth, tenth, eleventh, twelfth, fourteenth and fifteenth auction rounds. From this, it is apparent that the marketplace's peak performance is, in most cases, above or equal to the best of the three constituent recommenders'.

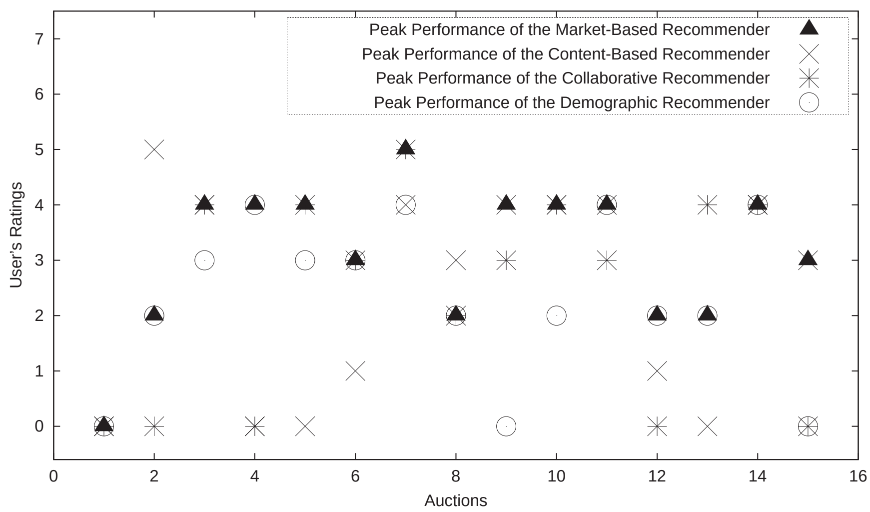


FIGURE 6.10: Different Recommenders' Peak Performances

Now we need to determine whether this happens for most of the users. To do this, we added up all the effective peak performance points for all thirty-one user trials. Among all the auctions for all users, 66.4% of them have their market-based recommender's peak performance as high as the best of the three constituent recommenders' (see Figure 6.11).

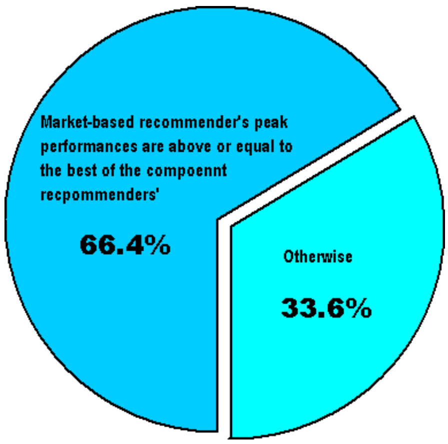


Figure 6.11: Marketplace's Overall Effective Peak Performance

6.4.4 Best Recommendations Identification

After evaluating the above three market related properties of our system, we now seek to determine whether our system is indeed suggesting those documents that the user values most. To do this, we randomly selected a few users to evaluate this property[10]. With respect to a given user, a number (Num) of recommendations are made available for recommendation. Here, Num is bigger than 75 (5 recommendations per auction multiplied by 15 auctions, for a typical user's task) so that the system can make its recommendations from a big pool of items from all quality levels. With respect to a specific user, Num recommendations can be divided into two sets: one with items suggested by the system to the user and the other with items omitted by the system during the user's task (i.e. not presented in any auction round in the trial for that user). Then, to identify the best items for this user, we require him to perform a trial (as we have outline previously) and, additionally, to rate all the omitted items. This means, we can manually identify the best recommendations for the user (defined in the fourth metric in section 6.1). From this, we are able to validate how many of the best recommendations our system is able to identify.

In more detail, Figure 6.12 shows a typical example of these experiments. Here, the horizontal axis represents the different rating levels and the vertical axis represents

[10]This is a time-consuming task since it involves examining a huge number of Web documents from the source recommendation pool and rating each of them. Therefore, we performed this with a subset of our user trial population. Specifically, we did it for one randomly selected user.

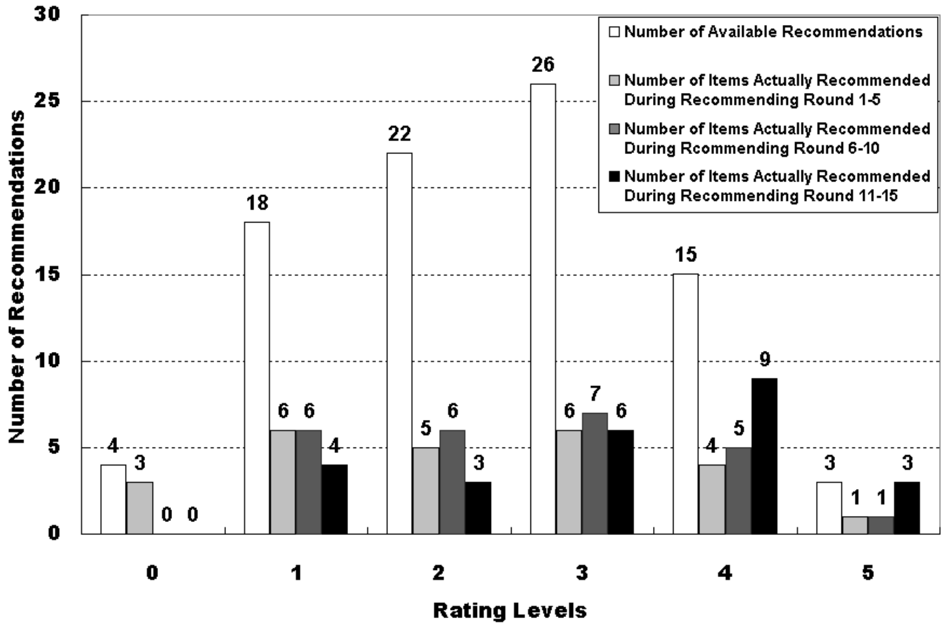


Figure 6.12: Available Recommendations vs Actual Recommended Items

the number of recommendations. The white bars represent the numbers of available recommendations (made available to recommend before the user performs the task) at each of the different rating levels. The light gray bars represent the numbers of items actually suggested by our system from the first to the fifth recommending round of the user's task. The dark gray bars represent the numbers of items actually suggested from the sixth to the tenth recommending round. The black bars represent those suggested from the eleventh to the fifteenth round. The white bars in Figure 6.12 show that there are 18 recommendations (fifteen items with rating "4" and three with "5") that this user considers best. From Figure 6.12, we can see that the numbers of the best recommendations have an overall tendency to increase over the recommending rounds. This indicates that our marketplace is able to effectively learn the user's interests and identify the best recommendations more frequently over time. From the numbers of recommendations made at rating levels "0" and "1", we can see that our marketplace is able to deliberately deter such bad and weakly positive recommendations because the numbers of such recommendations have an overall tendency to decrease over time.

During the user's trial from the first to the fifth recommending round, we find that there are four qualified recommending rounds and one of them is a satisfied recommending

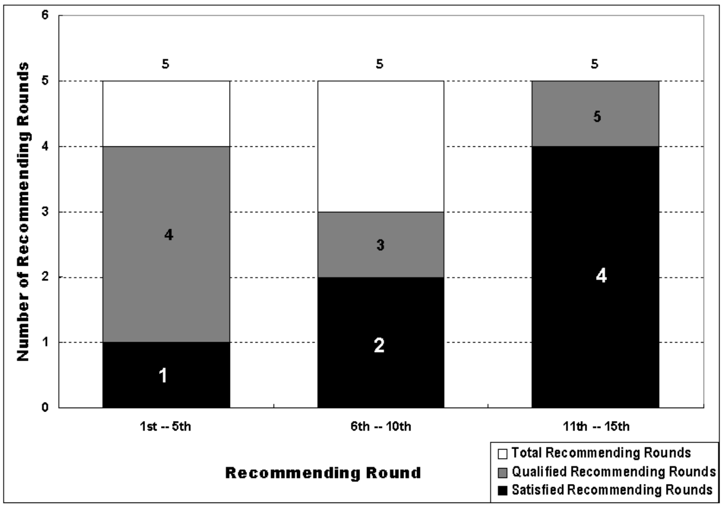


FIGURE 6.13: Best Recommendations Identification for a Given User

round (see the definitions in the fourth metric in section 6.1); from the sixth to the tenth round, there are three qualified rounds and two of them are satisfied; from the eleventh to the fifteenth round, there are five qualified rounds and four of them are satisfied. These results are displayed in Figure 6.13. This means that 80% of the first five rounds, 66.7% of the second five rounds and 100% of the last five rounds are qualified, whereas 20% of the first five rounds, 40% of the second five rounds and 80% of the last five rounds are satisfied. Therefore, both the qualified and the satisfied recommending rounds have the overall tendency of increasing. When taken together, these results show that our marketplace is indeed able to identify the best recommendations and display them in the top positions of the recommendation sidebar quickly and frequently.

6.5 Summary

This chapter presents the user evaluation of our market-based recommender system. Specifically, we validated and verified the feasibility and efficiency of our market-based approach to recommender systems through thirty-one user trials. Based on our experiments with these real users, we conclude that our market-based recommender system has the following properties:

- Our marketplace is able to coordinate multiple recommenders and guarantee they have equal opportunities to display their recommendations to users. Compared to the fairness defined in section 3.1 and evaluated in section 4.2.4, the identical output contribution again reinforces the fairness property of the marketplace. With the simulations in section 4.2.4, it is hard to assert fairness with respect to different users' interests and browsing contexts. However, the evaluations on identical output contribution demonstrate that our marketplace is able to act fairly to different constituent recommenders across a user population of different interests and browsing contexts in a real environment.

- The market converges quickly with respect to highly rated recommendations since there are clear incentives associated with these items. The evaluations on market convergence directly evaluate the bidding price deviations from corresponding equilibria of different UPQ levels. This is based on the fact that we have demonstrated, in section 4.2.1, that as prices for different advertisement slots converge, prices for recommendations of different UPQ levels also converge. In this chapter, we successfully demonstrated the convergence of recommendations with high UPQs, while we observe a lack of convergence of low UPQ recommendations.

  Additionally, the price convergence for different UPQ levels also indicates the marketplace's ability to correlate the recommendations' UPQs to the constituent recommenders' INQs. This is because after the high UPQs prices convergence, the individual recommenders have built up their internal correlations between the prices and the corresponding UPQs, as well as to their INQs (this relationship has been demonstrated in section 4.2.3). Thus, the correlations between these UPQs and their corresponding INQs have been built up.

- After convergence, the market-based recommender's peak performance is above or equal to the best of all three constituent recommenders' most of the time. This means that our marketplace is able to work as a coordinator of multiple different recommenders and always output the best items from whatever of these constituent recommenders. This is the original objective of our market-based approach to recommendations as stated in section 1.1.

- Our market-based recommender is able to identify the best items most of the time and, in general, these are displayed in the top positions of the recommendation side bar. This is again precisely the objective of our market-based approach that is stated in section 1.1.

To conclude, this chapter contributes to the thesis in the following ways: *(i)* this chapter built up a real market-based recommender system and evaluated our market mechanism design from chapters 3, 4 and 5 with real users; *(ii)* the evaluation results show that our marketplace works as an effective coordinator of multiple recommendation methods and gives fairness to all constituent recommenders without any one dominating the marketplace across the user population; *(iii)* the evaluation results show that the marketplace always converges with respect to the recommendations that highly interest the users. Additionally, this indicates the marketplace is able to correlate such recommendations' UPQs and INQs; and *(iv)* the marketplace works effectively and is able to quickly and frequently seek out the best items from all constituent recommenders and display them at the top positions of the recommendation side bar to the users in most cases. In summary, the user evaluations of the real market-based recommender demonstrates that our market-based approach is an effective means of coordinating multiple different recommendation methods in one single system and that it is an effective way of dealing with the problem of information overload.

Chapter 7

Conclusions and Future Work

Information overload on the Web is an ever-growing problem and recommender systems have been widely advocated as a tool that can help in this context. Given this, there are many recommendation methods being developed to assist recommendations. However, there is no one that is able to effectively make recommendations for all users with various interests. Thus, we believe that the way to go in this area is to develop an overarching system that incorporates multiple recommendation methods and that lets only the best items from whatever methods pass through to the user.

Following this philosophy, in this work, a market-based approach is used to coordinate multiple recommendation methods. Here a market-based recommender is a system that uses an economic marketplace to let multiple recommenders compete with each other to offer their recommendations. By so doing, the marketplace automates the coordination of this competition and the process of selecting the best items from the viewpoint of users. Such a system is designed to be used in conjunction with traditional recommendation methods (such as content-based, collaborative, demographic and hybrid filtering techniques). In particular, the market-based recommender system is effectively a meta-recommender that selects only the best items from whatever sources to display to users.

7.1 Conclusions

The *central hypotheses* of this thesis is that a market-based approach is a good means of coordinating multiple recommendation methods and letting only the best items pass through to the user. In order to prove this hypotheses, three sub-hypotheses are established and evaluated.

The *first sub-hypotheses* is that all the constituent recommenders incorporated in the marketplace work in a broadly similar manner and that in gross terms they contribute a similar number of recommendations across various user interests. This property ensures there is sufficient competition between the various recommendation methods and that the market-based recommender does not degenerate to a single constituent recommendation method. To this end, simulation results in section 4.2.4 show that different methods do indeed contribute a similar number of recommendations to users without considering users' interests and browsing contexts. This is the first attempt to prove the first sub-hypotheses. Building on this, section 6.4.1 then uses real users' evaluations to support this sub-hypotheses and the evaluation results demonstrate that our marketplace does give different constituent recommenders an equal opportunity of displaying their recommendations across a user population with various interests and browsing contexts.

The *second sub-hypotheses* is that the market is always able to converge so that it is able to give clear incentives of the UPQs of recommendations to the recommender agents. To this end, sections 4.2.1 and 4.2.3 demonstrate that the market converges to different equilibria with respect to different advertisement side bar slots. In addition, to evaluate the convergence of economical equilibria (a point where supply crosses demand) of advertisement slots, section 6.4.2 looks into the bidding price convergence with respect to different UPQs of recommendations and shows that the market does indeed converge with respect to high UPQ recommendations. With the market convergence in place, a constituent recommender agent is able to reason about the amount of reward and the bidding price it is likely to receive and should bid over the recommendations with certain INQs. After the market converges, the rewards an agent receives reflect the UPQs of its recommendations, whereas the optimally adjusted bidding prices reflect the value of advertising these items. In this way, the market correlates the recommenders' INQs to the UPQs of their recommendations.

The *third sub-hypotheses* is that the marketplace quickly and frequently identifies the best recommendations to the users. To this end, section 4.4 attempts to validate the marketplace's ability to identify the best recommendations. To improve the efficiency of this identification ability, chapter 5 describes a reinforcement learning strategy to assist the agent to make the best recommendations quickly and frequently. The simulations in chapter 5 do indeed show that a marketplace with learning agents is able to find out the best items more quickly and more frequently than one without. Moreover, armed with three kinds of traditional filtering agents with our learning capability, the user evaluations of the real market-based recommender system in sections 6.4.3 and 6.4.4 help demonstrate this sub-hypotheses from two perspectives. The evaluations in section 6.4.3 show from the marketplace's internal point of view, the way in which the market-based recommender is always able to perform as good as the best constituent recommender. In contrast, the evaluations in section 6.4.4 show, from the users' point of view, that the marketplace is indeed able to quickly and frequently identify the top UPQ recommendations and display them in the top position of the recommendation side bar.

Taking the evidence shown by the experiments we carried out in chapters 4, 5 and 6 against the market-based mechanism designed in chapter 3, we are now in a position to conclude that the market-based approach to recommender systems is indeed a very effective means of coordinating multiple different recommendation methods in one single system and that it is a very effective way of dealing with the problem of information overload by selecting only the best items from whatever methods to be displayed to users. In other words, the market-based approach to recommender systems is neither a new recommendation algorithm, nor a new filtering technique, but rather it is a new paradigm for economic-oriented approaches to recommendations.

7.2 Future Work

Despite the success of the system we have developed, there are a number of ways in which it can be improved still further. Given this, we outline the main direction of future work.

7.2.1 Improving the Speed of Market Convergence

As can be seen from the experiments demonstrated in chapters 4, 5 and 6, market convergence is the backbone that makes the marketplace work in terms of recommending effectively. If a marketplace fails to converge, it is unable to give incentives of UPQs of recommendations to the constituent recommenders, the constituent recommenders are unable to learn the users' interests, nor are they able to correlate their bidding prices to the UPQs and the INQs of their recommendations.

Therefore, it is important to develop methods and techniques that can be developed in the marketplace in order to speed up convergence. Such methods enable a market-based recommender to quickly give clear incentives to the constituent recommender agents and enables the agents to learn the users' interests quickly. However, except for validating the suitability and effectiveness of using market-based mechanisms as means of coordinating multiple recommendation methods, this thesis did little on developing techniques to improve the speed of market convergence. In more detail, our simulations in chapter 4 aim to assert the suitability of our system for an environment with full competition (meaning with a large number of constituent recommenders and source recommendations). However, the comparatively large number of price iterations (because of a large degree of competition) make the convergence slow. To kick start the convergence, the experiments in chapter 6 use constant equilibria instead of finding the equilibria of historical average prices by the marketplace itself. This approach makes convergence quicker and quickly differentiates recommendations in terms of UPQs by their corresponding prices. However, it is rigid to equilibria prices since with different system configurations the equilibria tend to change and we do not know this change.

Therefore, a future direction of work in this area is to improve the degree of automation of finding the market convergence across different system configurations such as the number of constituent recommenders, the number of UPQ levels and the number of shortlisted recommendations to be displayed to the users at a time. Relevant methods that could be considered in this context include the principles of supply and demand (i.e. the market equilibrium) [Varian, 2003].

7.2.2 Dealing with Dynamically Changing User Interests

This thesis deals with information overload in a static user environment (meaning that the user does not change his browsing interest and context while interacting with the system). However, in more general cases, a user may have multiple, different topics of interest such as music, fashion and food. Furthermore, a user's interest topic within a broad area may change from one sub-area to another, such as from classical music to pop music, from spring fashion to summer fashion, and from Italian to French food. In some even more complex environments, a user may have multiple parallel browsing sessions on different topics. Now all these issues undoubtedly make it more difficult for the market-based mechanism (and all other mechanisms) to suggest as good recommendations as those it makes in a static user environment. To deal with this, techniques such as user interest profile management [Case et al., 2003, 2001] and Bayesian decision theory [Duda et al., 2000] are potential candidates because the former provides a means of managing users interests when they have multiple interesting topics and the latter provides the confidence for recommenders to detect the status of the changing user interest.

7.2.3 Improving the Sharing of Information

When learning the users' interests and making recommendations, each constituent recommender needs to collect a large amount of data such as the users' topics of interest and the Web documents' keywords. From the overall system's point of view, this collection process can be highly repetitive since the different recommenders cover large amounts of collected data. Such duplication also unavoidably damages the computational efficiency of the individual recommenders and the overall system. Therefore, providing a means of sharing such information between the different constituent recommenders will undoubtedly save computational resources and improve computational efficiency. However, the kind of information that can be should be shared among the different recommenders and different users has privacy implications. This is because users may not be willing to share their private information with others and, on the other hand, the self-interested recommender agents may not be willing to share their knowledge with others since they work in a competitive manner and their knowledge affects their profits directly. Therefore, providing a means of information sharing to improve the system's computational efficiency, while not destroying privacy among the users and among the constituent recommenders is an interesting potential future research topic. Here promising work can be found in [Lau et al., 1999].

7.2.4 Improving the Degree of Personalization

Personalization plays an important role in dealing with information overload. Generally speaking, in our context, personalization is the adoption and arrangement of information that is tailored to a single user or a group of users. This is, therefore, undoubtedly a research issue for market-based recommender systems. However, while we described the big picture of the general issues of the market-based approach, we did little in this aspect. In this context, there are two aspects of personalization research that come to the fore: *(i)* the individual constituent recommendation methods' personalization and *(ii)* the marketplace's personalization. The former has been extensively studied in traditional recommender system research (e.g. [Sheth and Maes, 1993, Mladenic, 1996, Zuno, 1997, Lynch, 2001]) and is an ongoing line of enquiry. However, the latter needs further investigation and should focus on issues related to improving the speed of market convergence, information sharing and security issues.

Bibliography

Autonomy. Agentware i3. Technical report, Autonomy, Inc., 301 University Avenue, Suite 200 Palo Alto, CA 94301, USA, 1997.

Paul E. Baclace. Personal information intake filtering. In *Proceedings of the Bellcore Information Filtering Workshop*, Nov. 1991.

Marko Balabanovic and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992. ISSN 0001-0782.

T. Berners-Lee, R. Cailliau, T.-F. Groff, and B. Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications, and Policy*, 1 (2):52–58, 1992.

D. A. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments.* Chapman and Hall, London, 1985.

Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–54. Morgan Kaufmann, 1998.

Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000. ISSN 0924-1868.

Sander M. Bohte, Enrico Gerding, and Han La Poutré. Competitive market-based allocation of consumer attention space. In *Proceedings of the Thirdrd ACM Conference on Electronic Commerce*, pages 202–205, Tampa, USA, 2001. ISBN 1-58113-387-1.

Sander M. Bohte, Enrico Gerding, and Han La Poutré. Market-based recommendation: Agents that compete for consumer attention. *ACM Transactions on Internet Technology*, 4(4):420–448, 2004.

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Wisconsin, USA, 1998.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002. ISSN 0924-1868.

Simon Case, Nader Azarmi, Marcus Thint, and Takeshi Ohtani. Enhancing e-communities with agent-based systems. *IEEE Computer*, 34(7):64–69, 2001. ISSN 0018-9162. doi: http://dx.doi.org/10.1109/2.933505.

Simon Case, Marcus Thint, Takeshi Ohtani, and Stephen Hare. Personalisation and web communities. *BT Technology Journal*, 21(1):91–97, Jan 2003.

Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *ACM SIGIR Workshop on Recommender Systems*, Berkeley, US, 1999.

Scott H. Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Publishing, 1996.

R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

David C. DeRoure, Wendy Hall, Siegfried Reich, Aggelos Pikrakis, Gary J. Hill, and Mark Stairmand. Memoir – an open framework for enhanced navigation of distributed information. *Information Processing and Management*, 37:53–74, 2001.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2000. ISBN 0471056693.

Samhaa R. El-Beltagy, Wendy Hall, David DeRoure, and Leslie Carr. Linking in context. In *Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia*, pages 151–160, Denmark, 2001. ISBN 1-59113-420-7.

J. C. Gittins. *Multi-Armed Bandit Allocation Indices.* Wiley, 1989.

D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

Gustavo Gonzalez, Beatriz Lopez, and Josep Lluis de la Rosa. Managing emotions in smart user models for recommender systems. In *Seventeenth of Sixth International Conference on Enterprise Information Systems*, pages 187–194, 2004.

Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of ACM Conference on Computer supported cooperative work*, pages 241–250, Philadelphia, US, 2000. ISBN 1-58113-222-0. doi: http://doi.acm.org/10.1145/358916.358995.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information System*, 22(1):5–53, 2004. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/963770.963772.

Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95*, pages 194–201, 1995.

Adele E. Howe and Daniel Dreilinger. Savvysearch: A meta-search engine that learns which search engines to query. *AI Magazine*, 18(2):19–25, 1997.

N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

Nicholas R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

Nicholas R. Jennings and Michael J. Wooldridge, editors. *Agent Technology: Foundations, Applications, and Markets.* Springer Verlag, Berlin, 1998. ISBN 3-540-63591-2.

Leslie Pack Kaelbling. *Learning in Embedded Systems.* MIT Press, Cambridge, 1993.

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

John H. Kagel and Alvin E. Roth, editors. *The Hand Book of Experimental Economics*. Princeton University Press, 1995.

B. Kahle and B. Gilliat. Alexa - navigate the web smarter, faster, easier. Technical report, Alexa Internet, Presidio of San Fransisco, CA, 1997.

Grigoris J. Karakoulas and Innes A. Ferguson. Sigma: Integrating learning techniques in computational markets for information filtering. In *AAAI 1996 Spring Symposium on Machine Learning in Information Access*, Stanford, CA, 1996.

Paul Klemperer. Auction theory: A guide to literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997. ISSN 0001-0782.

Ivan Koychev. Gradual forgetting for adaptation to concept drift. In *Proceedings of ECAI2000 Workshop Current Issues in Spatio-Temporal Reasoning*, pages 101–106, Berlin, 2000.

B. Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.

Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, San Mateo, US, 1995. Morgan Kaufmann publishers.

Tessa Lau, Oren Etzioni, and Daniel S. Weld. Privacy interfaces for information management. *Communications of ACM*, 42(10):88–94, 1999. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/317665.317680.

Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the Twenty-second International Conference on Very Large Databases*, pages 251–262, Bombay, India, 1996. VLDB Endowment, Saratoga, Calif.

Henry Lieberman. Letizia: An agent that assists web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Canada, 1995. ISBN 1-55860-363-8.

Nick Littlestone and Manfred Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Shoshana Loeb and Douglas Terry. Information filtering. *Communications of the ACM*, 35(12):26–28, Dec 1992.

Robert M. Losee. Minimizing information overload: The ranking of electronic messages. *Journal of Information Science*, 15(3):179–189, 1989.

Clifford A. Lynch. Personalization and recommender systems in the larger context: New directions and research questions. In *Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries*, Dublin, Ireland, 2001.

Pattie Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–40, July 1994.

David Maltz. Distributing information for collaborative filtering on usenet net news. Thesis Proposal MIT/LCS/TR-603, MIT, Nov 1993.

David Maltz and Kate Ehrlich. Pointing the way: Active collaborative filtering. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95)*, pages 202–209, Denver, USA, 1995. ISBN 0-201-84705-1.

R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699–738, June 1987.

Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22 (1):54–88, 2004. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/963770.963773.

Stuart Edward Middleton. *Capturing Knowledge of User Preferences with Recommender Systems*. PhD thesis, University of Southampton, May 2003.

Paul Milgrom. Auctions and bidding: A primer. *The Journal of Economic Perspectives*, 3(3):3–22, Summer 1989.

Paul R. Milgrom and Robert J. Weber. A theory of auctions and competitive bidding. *Econometrica*, 50(5):1089–1122, Sep. 1982.

Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the Eightth International Conference on Intelligent User Interfaces*, pages 263–266. ACM Press, 2003. ISBN 1-58113-586-6. doi: http://doi.acm.org/10.1145/604045.604094.

Tom Mitchell. *Machine Learning*. McGraw Hill, 1997. ISBN 0070428077.

D. Mladenic. Personal webwatcher: Implementation and design. Technical report ijs-dp-7472, Department of Intelligent Systems. Slovenia: J. Stefan Institute, 1996.

Miquel Montaner, Beatriz Lopez, and Josep Lluis Dela. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19:285–330, 2003.

Luc Moreau, Norliza Zaini, Jing Zhou, Nicholas R. Jennings, Yan Zheng Wei, Wendy Hall, David De Roure, Ian Gilchrist, Mark O'Dell, Sigi Reich, Tobias Berka, and Claudia Di Napoli. A market-based recommender system. In *Proceedings of the Fourth International Workshop on Agent-Oriented Information Systems (AOIS-2002)*, pages 50–67, Bologna, July 2002.

Tracy Mullen and Michael P. Wellman. A simple computational market for network information services. In *Proceedings of the First International Conference on Multiagent Systems*, pages 283–289, San Francisco, 1995. AAAI Press / MIT Press.

Michael Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999. ISSN 0269-2821.

Michael Pazzani, J. Muramatsu, and D. Billsus. Syskill & webert: Indentifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 54–61, 1996.

David M. Pennock, Eric Horvitz, and C. Lee Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and*

Twelfth Conference on Innovative Applications of Artificial Intelligence, pages 729–734. AAAI Press / The MIT Press, 2000. ISBN 0-262-51112-6.

Brian Pinkerton. *WebCrawler: Finding what people want*. PhD thesis, University of Washington, U.S., 2000.

Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 437–444, Seattle, US, 2001.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994.

Paul Resnick and Hal R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.

E. Rich. User modeling via stereotyps. *Cognitive Science*, 3:329–354, 1979.

Alvin E. Roth. The economist as engineer: Game theory, experimental economics and computation as tools of design economics. *Econometrica*, 70(4):1341–1378, 2002.

Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989. ISBN 0-201-12227-8.

Gerard Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Publishing Company, New York, 1983.

Paul A. Samuelson and William D. Nordhaus. *Economics*. McGraw-Hill/Irwin, 17th edition, 2001.

Tuomas W. Sandholm. Distributed rational decision making. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT Press, Cambridge, 1999.

Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, 2000.

Badrul M. Sarwar, Joseph A. Konstan, Al Borchers, John Herlocker, Brad Miller, and John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM Conference on Computer supported cooperative work*, pages 345–354, Seattle, US, 1998. ISBN 1-58113-009-0.

Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating "word of mout". In *Proceedings of Conference on Human factors in computing systems*, pages 210–217. ACM Press, 1995. ISBN 0-201-84705-1.

Rita Sharma and David Poole. Symmetric collaborative filtering using the noisy sensor model. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 488–495, Seattle, US, 2001.

Beerud Sheth and Pattie Maes. Evolving agents for personalized information filtering. In *Proceedings of the nineth Conference on Artificial Intelligence for Applications (CAIA'93)*, pages 345–352, Orlando, 1993.

L. Terveen and W. Hill. Beyond recommender systems: Helping people help each other. In J. Carroll, editor, *HCI in the New Millennium*. Addison Wesley, 2001.

Loren Terveen, Will Hill, Brian Amento, David McDonald, and Josh Creter. PHOAKS: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62, 1997.

Edward L. Thorndike. Animal intelligence: An experimental study of the associative processes in animals. *Psychological Review, Monograph Supplements*, (8), 1898. New York: MacMillan.

S. B. Thrun. The role of exploration in learning control. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control*. Van Nostrand, New York, 1992.

Efraim Turban, Jae Lee, David King, and H. Michael Chung. *Electronic Commerce: A Managerial Perspective*. Prentice Hall, 2000.

Hal R. Varian. *Intermediate Microeconomics: A Modern Approach*. Norton, 6th edition, 2003.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, Mar 1961.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Market-based recommendations: Design, simulation and evaluation. In *Proceedings of the Fifth International Workshop on Agent-Oriented Information Systems (AOIS-2003)*, pages 63–78, Melbourne, Australia, 2003a. Springer LNAI 3030.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Recommender systems: A market-based design. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03)*, pages 600–607, Melbourne, Australia, 2003b. ACM Press.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Learning users' interests in a market-based recommender system. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04)*, pages 833–840, Exeter, UK, 2004a. Springer LNCS 3177.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Market-based recommender systems: Learning users' interests by quality classification. In *Proceedings of the Sixth International Workshop on Agent-Oriented Information Systems (AOIS-2004)*, pages 119–133, New York, US, 2004b.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. Learning users' interests by quality classification in market-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, To appeara.

Yan Zheng Wei, Luc Moreau, and Nicholas R. Jennings. A market-based approach to recommender systems. *ACM Transactions on Information Systems*, To appearb.

Michael P. Wellman and Peter R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24:115–125, 1998.

Michael P. Wellman, Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Shou-De Lin, Daniel Reeves, and William E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, March-April 2001. ISSN 1089-7801. INSPEC Accession Number: 6934076.

Michael J. Wooldridge and Nicholas R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

T. Yan and H. Garcia-Molina. SIFT—A tool for wide-area information dissemination. In *Proc. 1995 USENIX Technical Conference*, pages 177–186, New Orleans, 1995.

Kai Yu, Xiaowei Xu, Martin Ester, and Hans-Peter Kriegel. Selecting relevant instances for efficient and accurate collaborative filtering. In *Proceedings of the tenth International Conference on Information and Knowledge Management*, pages 239–246, Atlanta, Georgia, USA, 2001. ACM Press. ISBN 1-58113-436-3.

Kai Yu, Xiaowei Xu, Martin Ester, and Hans-Peter Kriegel. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach. *Knowledge and Information Systems*, 5(2), April 2003.

Gustavo Zamboni. Search tools. Technical report, University of Cordoba, 1998.

Zuno. Vrisko - personal knowledge manager. Technical report, Zuno Limited, 1997. URL URL300 Third Avenue, Waltham, MA 02154.