

# Enhancing Personalized Ranking Quality through Multidimensional Modeling of Inter-Item Competition

Qinyuan Feng <sup>†§</sup>, Ling Liu <sup>§</sup>, Yan (Lindsay) Sun <sup>‡</sup>, Ting Yu <sup>\*</sup>, Yafei Dai <sup>†</sup>

<sup>§</sup>Georgia Institute of Technology, Atlanta, GA, USA  
lingliu@cc.gatech.edu

<sup>†</sup>Peking University, Beijing, China  
fqy@pku.edu.cn, dyf@pku.edu.cn

<sup>‡</sup>University of Rhode Island, Kingston, RI, USA  
yansun@ele.uri.edu

<sup>\*</sup>North Carolina State University, Raleigh, NC, USA  
tyu@ncsu.edu

**Abstract**—This paper presents MAPS—a personalized Multi-Attribute Probabilistic Selection framework—to estimate the probability of an item being a user’s best choice and rank the items accordingly. The MAPS framework makes three original contributions in this paper. First, we capture the inter-attribute tradeoff by a visual angle model which maps multi-attribute items into points (stars) in a multidimensional space (sky). Second, we model the inter-item competition using the dominating areas of the stars. Third, we capture the user’s personal preferences by a density function learned from his/her history. The MAPS framework carefully combines all three factors to estimate the probability of an item being a user’s best choice, and produces a personalized ranking accordingly. We evaluate the accuracy of MAPS through extensive simulations. The results show that MAPS significantly outperforms existing multi-attribute ranking algorithms.

## I. INTRODUCTION

Social networks and online communities are one of the most successful collaborative computing platforms in the computing and communication history. Personalized ranking capability is fundamental for search, question answering, and recommendation in eCommerce and social networks related applications. Such applications require ranking a set of multi-attribute items to help a user find his/her best choice among all items. For example, Alice wants to buy an iPhone 8G using eBay. There are many sellers selling the iPhones 8G with different prices and reputations. Alice expects eBay to provide her a personalized ranking of all sellers with high ranking accuracy. By high ranking accuracy, we mean that given a user and a set of multi-attribute items from which the user needs to select one as his/her best choice, the best choice should be ranked as high as possible in the ranking list. Concretely, the *best choice* of Alice refers to the seller from which Alice will purchase an iPhone after an exhaustive search over all available sellers.

In this paper, we argue that there are three key factors in designing a personalized multi-attribute ranking algorithm with high accuracy: inter-attribute tradeoff, inter-item competition, and personalized user preferences.

The first factor is **inter-attribute tradeoff**. Different items often have different attribute values. When the items with multiple attributes can be compared, it is straightforward to rank them. In eBay, if one seller has higher reputation and lower price than another, this seller is clearly a better choice. However, if one seller has higher reputation and higher price than another seller, it is hard to determine which one is

better. In this situation, the inter-attribute tradeoff needs to be considered. Existing approaches concentrate on balancing weights of multiple attributes for each item. In this paper we argue that only considering inter-attribute tradeoff may not be sufficient. We observe that when it is hard to compare items with contradicting inter-attribute tradeoff, a user also makes his/her best choice decision based on other factors, especially the competition between sellers with similar prices and reputations in the eBay case. This motivates us to introduce the next factor.

The second factor is **inter-item competition**. The probability of an item being a user’s best choice depends on not only its own attribute values, such as price and reputation of the seller, but also other similar items that are in competing value ranges. For example, the existence of an iPhone seller will reduce the probabilities of other iPhone sellers, who have prices and reputations similar to this seller, being the best choice. In this paper, we show that inter-item competition not only plays an important role in determining the probability of an item being the best choice, but also helps in making inter-attribute tradeoff in the situation where the multiple attributes of items being ranked do not agree with one another. Thus, we argue that a ranking algorithm should incorporate inter-item competition by jointly considering other similar items when calculating an item’s ranking score.

The third critical factor is **personalized user preferences**. We argue that both inter-attribute tradeoff and inter-item competition can vary significantly for different users or for the same user under different contexts. In eBay, we observe that some users prefer sellers with low price and reasonable reputation, some users prefer high reputation and reasonable price, and some extreme users always choose the items with the lowest price. Furthermore, a user may prefer sellers offering low price when purchasing a cheap product, and prefer sellers with high reputation when purchasing an expensive product. Thus, a ranking algorithm should capture personalized user preferences with respect to inter-attribute tradeoff and inter-item competition.

Unfortunately, most existing multi-attribute ranking algorithms rank a set of items based solely on inter-attribute tradeoff and personalized user preferences on how such inter-attribute tradeoff is handled. However, they fail to address inter-item competition. Concretely, existing multi-attribute ranking algorithms fail to capture the background knowledge of a user about how he/she has handled the inter-item compe-

tion in the past when items being ranked have contradicting attribute values. We show in this paper that existing algorithms work well only in simplistic scenarios and tend to fail drastically for users with slightly more sophisticated preferences in terms of inter-item competition. As a result, existing ranking algorithms produce low ranking accuracy, i.e., placing a user’s best choice at low position on the ranking list.

With these challenges in mind, we present MAPS – a **personalized Multi-Attribute Probabilistic Selection framework**. The MAPS is unique in three aspects. First, MAPS presents a visual angle model, a novel approach to modeling items such that inter-attribute tradeoff and inter-item competition can be elegantly captured using the same underlying model of items. Second, MAPS presents a methodical scheme to modeling personalized user preferences by capturing the past behaviors of a user in terms of how the user makes the best choice selection. Third but not the least, MAPS develops a probability-based ranking algorithm. It estimates the probability of each item being a user’s best choice as the ranking score to rank the items that match to a user’s query.

We evaluate MAPS through extensive simulations. We show that MAPS offers higher ranking accuracy compared to existing multi-attribute ranking algorithms. Furthermore, MAPS requires a short learning curve and can scale to a large number of items. To the best of our knowledge, MAPS is the first multi-attribute ranking algorithm to date that identifies and incorporates inter-item competition into both the ranking score computation and the personalized user preference profile construction processes.

The rest of the paper is organized as follows. Section II describes the problem formulation and illustrates the limitations in existing ranking algorithms. We introduce MAPS in detail in Section III. We evaluate the performance of MAPS in Section IV and conclude in Section VI.

## II. PROBLEM DEFINITION

### A. Problem formulation

The problem of personalized multi-attribute ranking focuses on providing a ranking list of items in response to a given user query. The goal is to ensure high accuracy in the sense that the user’s best-choice item, which the user will choose after an exhaustive search over all items, should be ranked as high as possible on the ranking list. Existing multi-attribute ranking algorithms address the problem solely based on inter-attribute tradeoff through weight function design and differ from one another mainly in terms of concrete weight functions. In this paper we argue that the multi-attribute item ranking should be based upon a user’s personal preferences on both inter-attribute tradeoff and inter-item competition.

We formulate the multi-attribute ranking problem using the running example that Alice wants to buy an iPhone 8G, and the e-market has  $n$  sellers selling it. These sellers form the **seller set** or the **item set** in general, denoted by  $S^{set} = \{S_1, S_2, \dots, S_n\}$ . Different products often have different seller sets. We refer to the different products as the different **contexts** or queries of our ranking problem.

A seller has many attributes, such as name, location, specialty, product name, unit price, shipping cost, and reputa-

tion. Typically only a subset of the attributes is used for ranking (e.g., unit price, shipping cost, and reputation). We call this subset of attributes the ranking attributes. These  $m$  ranking attributes form the **attribute set**, denoted by  $A^{set} = \{A_1, \dots, A_m\}$ . For example, in eBay, Alice selects the best iPhone seller based on two ranking attributes: price (sum of unit price and shipping cost) and reputation.

In general, a user considers the tradeoff between the ranking attributes ( $A^{set}$ ) of the items ( $S^{set}$ ) based on his/her personal preferences, and chooses the best one. We refer to this chosen one as the user’s **best choice**, denoted by  $S_B$ . The **goal** of the multi-attribute ranking algorithm is to rank the item set of a context for a given user, such that the user’s best choice ( $S_B$ ) is ranked as high as possible in the top-down ranking list of items. For presentation convenience, we may simply use attributes to refer to ranking attributes in the rest of the paper when no confusion occurs. Similarly items and sellers may be used interchangeably.

In the rest of the paper we will focus the discussion more on the original contributions of MAPS. Thus, we assume that the users considered in this paper are rational in the sense that they only choose the sellers that are not worse than any other one in the seller set. For example, if a seller offers higher price and lower reputation compared to another seller, none of the rational users will choose this obviously “worse” seller since none of its attributes is competitive compared to the other seller. Therefore, we assume that all sellers in  $S^{set}$  are **skyline sellers** [1], so none of the sellers considered in our seller set will be worse than another seller.

### B. Ranking quality metric

Let  $V_i$  denote the ranking score of item  $S_i$  and  $V_B$  denote the ranking score of the best choice. Since the items are ranked according to their ranking scores, which represent their likelihood of being the best choice within the item set  $S^{set}$ , we evaluate multi-attribute ranking algorithms with **ranking quality**, which is the percentage of the items whose ranking scores are smaller than the ranking score of the best choice.

$$\text{Ranking quality} = \frac{|\{S_i \in S^{set} | V_i < V_B\}|}{|S^{set}| - 1}. \quad (1)$$

We use  $R_i$  to denote the ranking position of the item  $S_i$ , which means the position of this item on the ranking list of the  $n$  items in  $S^{set}$ . We use  $R_B$  to denote the ranking position of the best choice. The highest rank is 1 and the lowest rank is  $n$ . We can also formulate ranking quality as follows.

$$\text{Ranking quality} = \frac{n - R_B}{n - 1}. \quad (2)$$

This ranking quality metric amounts to say that if the best choice has ranking position 1, the ranking quality is 1, so the ranking score of the best choice is the largest among all the items. When the best choice has ranking position  $n$ , the ranking quality is 0, so the ranking score of the best choice is the smallest among all the items. Thus, the higher is the ranking quality, the better accuracy is the ranking algorithm.

### C. Example

To clearly state our formulation, we give an example used throughout this paper. In eBay, a user sends a query searching

Seller ID	Price	Reputation
$S_1$	\$480	49
$S_2$	\$667	352
$S_3$	\$685	1560
$S_4$	\$778	5885

TABLE I  
SKYLINE SELLERS OF IPHONE 8G IN EBAY

for ‘‘iPhone 8G’’ sellers. This user judges the sellers based on two attributes: price and reputation. The reputation of a seller is calculated as the number of the seller’s previous good transactions subtracted by the number of the seller’s previous bad transactions. In general, a user prefers high reputation and low price. Among the seller set returned for this query, there are four skyline sellers, denoted by  $S_1$ ,  $S_2$ ,  $S_3$ , and  $S_4$ . The seller IDs ( $S_i$ ), price ( $p_i$ ), and reputation ( $r_i$ ) are shown in Table I. Note that the data in Table I are real data collected from eBay.

We explain the meaning of *ranking quality* with an example. Assume that Alice will choose  $S_3$  as her best choice from the four sellers. *Algorithm*<sub>1</sub> rank the items as  $S_1 > S_2 > S_3 > S_4$  which means  $R_B = 3$ , so the ranking quality is  $(4 - 3)/(4 - 1) = 0.33$ . *Algorithm*<sub>2</sub> rank the items as  $S_3 > S_4 > S_2 > S_1$  which means  $R_B = 1$ , so the ranking quality is  $(4 - 1)/(4 - 1) = 1$ . *Algorithm*<sub>3</sub> rank the items as  $S_2 > S_1 > S_4 > S_3$  which means  $R_B = 4$ , so the ranking quality is  $(4 - 4)/(4 - 1) = 0$ . We can conclude that the ranking quality of *Algorithm*<sub>2</sub> is the best and the ranking quality of *Algorithm*<sub>3</sub> is the worst.

#### D. Limitations in weight-based ranking approach

The problem of ranking items with multiple attributes has been addressed in literature [2]–[4]. Different methods simply provide different weight functions in terms of how to combine the attributes of items for ranking. This weight-based multi-attribute ranking is currently used in some commercial systems, such as eBay [5].

In the weight-based multi-attribute ranking approach, utility scores are calculated by utility functions that combine multiple attributes with different weight values. The most widely used utility function is a linear combination of the transformed attribute values, as shown in equation (3).

$$U(S) = \sum_{i=1}^m \alpha_i \cdot F_i(a_i) \quad \text{with} \quad \sum_{i=1}^m \alpha_i = 1. \quad (3)$$

$U(S)$  is the utility score of seller  $S$ ,  $a_i$  is the attribute value of  $A_i$  for seller  $S$ ,  $\alpha_i$  is the weight value assigned to attribute  $A_i$ , and  $F_i(x)$  is the transformation function for attribute  $A_i$ . The most widely used transformation functions are  $F_i(x) = x$  and  $F_i(x) = \log(x)$ . The sellers are ranked according to their utility scores.

However, we find two limitations in the weight-based multi-attribute ranking approach. **First**, by solely using weight to combine attributes, some skyline sellers have no chance to be ranked as the best choice. **Second**, existing weight-based multi-attribute ranking algorithms fail to capture the concept of inter-item competition.

We demonstrate the first limitation through a case study. Let us use the following utility function:

$$U(S_i) = \alpha \cdot r_i + (1 - \alpha) \cdot (-p_i). \quad (4)$$

With the sellers in Table I, the utility scores are calculated as  $U(S_1) = 529\alpha - 480$ ,  $U(S_2) = 1019\alpha - 667$ ,  $U(S_3) = 2245\alpha - 685$ ,  $U(S_4) = 6663\alpha - 778$ .

From a survey with 30 real users, we find that most of them choose  $S_3$  as their best choice. So we examine what the parameter  $\alpha$  should be to make  $S_3$  have the largest utility score among the four sellers. This means that  $U(S_3)$  should be larger than  $U(S_1)$ ,  $U(S_2)$ , and  $U(S_4)$ , as shown in equation (5), (6), and (7).

$$U(S_3) \geq U(S_1) \Rightarrow \alpha \geq 0.119. \quad (5)$$

$$U(S_3) \geq U(S_2) \Rightarrow \alpha \geq 0.015. \quad (6)$$

$$U(S_3) \geq U(S_4) \Rightarrow \alpha \leq 0.021. \quad (7)$$

However, the condition on  $\alpha$  in equation (5) conflicts with that in equation (7). It means that there is no valid choice of  $\alpha$  for those users who choose  $S_3$  as their best choice.

It is important to point out that this problem is the consequence of the linear combination of multiple attributes in the utility functions. It does not depend on the specific form of the utility functions. We can prove that if a point (item) is a skyline point but not a convex hull point, it will never get the highest utility score among the point set. The proof is omitted due to page constraint. Therefore, some users’ best choices cannot be described or captured by simply applying weight-based multi-attribute ranking approach. This is the first limitation of personalized ranking of items that only considers inter-attribute tradeoff using weight-based multi-attribute ranking approach.

Now we demonstrate the second limitation of weight-based multi-attribute ranking using the same real-user study. We asked 30 real users to rank the items according to the items’ probabilities to be their best choice in two scenarios. In the first scenario, there are only three sellers in the seller set:  $\{S_1, S_2, S_4\}$ ; In the second scenario, there are four sellers in the seller set:  $\{S_1, S_2, S_3, S_4\}$ .

For the first scenario, 22 of the users ranked the items as  $S_2 > S_1 > S_4$  or  $S_2 > S_4 > S_1$ . However for the second scenario, all of these 22 users switched their ranking list to  $S_3 > S_1 > S_4 > S_2$  or  $S_3 > S_4 > S_1 > S_2$ . It means that when  $S_3$  is not in the seller set, the probability of  $S_2$  is larger than the probability of  $S_1$ ; when  $S_3$  is added into the seller set, the probability of  $S_2$  is smaller than the probability of  $S_1$ . So on the ranking lists produced for these users, the order of  $S_2$  and  $S_1$  is affected by whether  $S_3$  exists.

The reason behind this phenomenon is that after  $S_3$  is added into the seller set, it competes with the other sellers to be the best choice. Since the attribute values of  $S_3$  is the closest to the attribute values of  $S_2$  (see Table I), it will greatly reduce the probability of  $S_2$  and make the probability of  $S_2$  smaller than all the other items. From further interviews with these users, we find that inter-item competition mostly happens between the sellers with similar attribute values. There is significant competition between  $S_3$  and  $S_2$ , some competition between  $S_3$  and  $S_4$ , but nearly no competition between  $S_3$  and  $S_1$ .

The above case study demonstrates that the ranking score of a multi-attribute item should depend on not only its own attributes (inter-attribute tradeoff), but also the attributes of other similar items (inter-item competition). In fact, inter-item competition is the reason of why adding or removing an item may change the ranking order of other items. In contrast, weight-based ranking algorithms consider only inter-attribute tradeoff and compute the utility score of an item only based on its own attributes. Thus, they fail to take into account of inter-item competition introduced by those items with similar attributes. As a result, whenever the inter-item competition plays a critical role or inter-item competition changes, such as adding or removing an item, the weight-based multi-attribute ranking algorithms will fail miserably.

The above limitations in weight-based approach to personalized ranking of items motivate us to design MAPS, a Multi-Attribute Probabilistic Selection framework for personalized multi-attribute ranking. A unique feature of MAPS is its capability to produce a ranking of items by carefully combining three critical factors: inter-attribute tradeoff, inter-item competition, and personalized user preferences.

### III. THE MAPS FRAMEWORK

In this section we describe the MAPS framework by focusing on the design and development of the three key components of MAPS: the visual angle model, the dominating area, and the density function. We will illustrate these concepts with the example shown in Section II-C. Further detail on the MAPS framework and its high dimensional model can be found in our technical report [6].

#### A. Visual angle model

In MAPS, a visual angle model is designed for the eBay case in two steps. In the **first step**, given a set of items with two attributes, we map all items into points (stars) in a two dimensional space with each dimension representing one of the attributes. The goal of such mapping is two folds. First, we want to utilize the two dimensional space to devise a visual angle model to capture the intrinsic relationship among the two attributes of each item (inter-attribute tradeoff). Second, we want to build a foundation for capturing inter-item competition by comparing items in terms of the relative angle distance between their multiple attributes. With these objectives in mind, we need to normalize all attributes into the same value range, say the range of  $[0,1]$ , and make sure that a larger normalized attribute value indicates a higher preference of a user.

In general, we can divide all types of attributes into two classes. The first class of attributes carries the semantics of the-larger-the-better within the range of  $[0, +\infty)$ , such as the reputation attribute of eBay sellers. For an attribute value in the first class, say  $a_i$ , function (8) is used to perform the normalization:

$$F(a_i) = \frac{a_i}{\sqrt{a_i \cdot a_i + \beta}}. \quad (8)$$

The second class of attributes carries the ordering semantics of the-smaller-the-better within the range of  $[0, +\infty)$ , such as the price attribute of eBay sellers. Function (9) is used

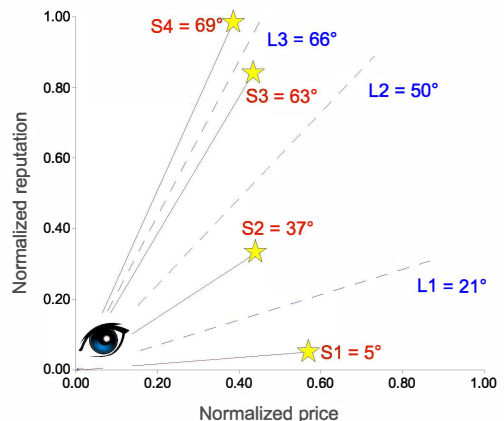


Fig. 1. Starry sky of iPhone 8G sellers

to perform the normalization of an attribute value  $a_j$  in the second class:

$$G(a_j) = 1 - \frac{a_j}{\sqrt{a_j \cdot a_j + \beta}}. \quad (9)$$

There are three remarks on the normalization step. First, in both normalization function (8) and function (9),  $\beta$  is the only parameter that is system-defined. Once it is set, all the users will use the same value. We will evaluate the effect of different settings of  $\beta$  in Section IV-C. Second, we have evaluated a number of popular normalization functions and found that the performance of MAPS is not sensitive to the choice of the normalization function. Third, for the attributes with other properties, we can also design normalization functions accordingly [7].

After normalization, an seller  $S_i$  with two attributes can be normalized into a point in a two dimensional space, denoted as  $(x_i, y_i)$ , where  $x_i$  and  $y_i$  represent the normalized price and the normalized reputation of seller  $S_i$ .

Based on the eBay example in Section II-C, Figure 1 shows the four sellers in Table I marked as four stars, each representing one of them. We set  $\beta$  as  $10^6$  in this example.

In the **second step**, a preference space is constructed by a visual angle model with two objectives: to capture the inter-attribute tradeoff and to establish the foundation for modeling inter-item competition. First, we represent each item as the angle of the ray from the origin to the point of the item in space.

In our eBay example, the sellers are mapped into 2D space with normalized price as x-axis and normalized reputation as y-axis (Figure 1). For each seller  $S_i$ , we choose the angle between the ray—from the origin to the corresponding star of  $S_i$  in the 2D space—and the x-axis to represent this seller. This angle is calculated as:

$$z_i = \arctan(y_i/x_i). \quad (10)$$

In Figure 1, the solid lines show the rays from the origin marked by an eye symbol to the four stars. Using the visual angle model to represent the items gives two useful properties.

First, the visual angle can uniquely represent an item in the item set. If two items in the item set have the same visual angle, they will be directly comparable. One of them will be clearly worse and thus never be chosen by any rational user.

As we discussed in Section II-A, to simplify the presentation, we assume that all items in the item set are skyline items such that none of the items is worse than another. Thus we focus on the challenging case to compare and rank the items with different visual angles.

Second, the visual angle representation of an item can describe the inter-attribute tradeoff. For example, in Figure 1, a seller with a large visual angle (e.g., seller  $S_4$ ) means that the seller has high reputation but relatively worse price. In contrast, a seller with a small visual angle (e.g., seller  $S_1$ ) means that it has low reputation but relatively better price.

In addition, the angle value could be used to capture a user's personal preferences. For example, a user, say Alice, is looking at the starry sky of iPhone sellers (Figure 1). If Alice prefers the sellers with high reputation and moderate price, she is more likely to look at the sky with a large visual angle. So she may find that  $S_4$  has the smallest angle distance to her own visual angle, and choose  $S_4$  as her best choice. However, if Alice prefers the sellers with low price and moderate reputation, she is more likely to look at the sky with a small visual angle. This time,  $S_1$  may have the smallest angle distance to Alice's visual angle and be chosen as Alice's best choice. This motivates us to define a **preference space** based on the angle model for our eBay case: The whole preference space is the angle value of  $[0^\circ, 90^\circ]$ . A user's preferences are described by the user's taste on visual angles over the preference space.

### B. Dominating areas of items

In MAPS, dominating area is introduced to capture inter-item competition. The motivation of defining the dominating area of an item comes from the following observation. When a user looks at the multidimensional space with a specific visual angle, say  $z_u$ , this user will select the item with the smallest angle distance to his/her visual angle as the best choice. We refer to this selected item as the dominating item of this special visual angle.

Let  $z_i$  denote the angle of item  $S_i$  and  $z_u$  denote the visual angle of a specific user  $U$ . The angle distance between  $z_u$  and  $z_i$  is calculated as:

$$\text{AngleDist}(z_u, z_i) = |z_u - z_i| \quad (11)$$

We define the **dominating area** for item  $S_i$  as the angle range that satisfies the following condition: If a user looks at the sky within the dominating area of item  $S_i$ , the angle distance between  $z_u$  and  $z_i$  is the smallest compared to the angle distance from any other items to  $z_u$ . Formally, the dominating item  $S_i$  for  $z_u$  should satisfy the follows:

$$\text{AngleDist}(z_u, z_i) \leq \text{AngleDist}(z_u, z_j), \forall 1 \leq j \leq n \quad (12)$$

This property ensures that increasing the dominating area of an item for a given user will increase its probability of being the user's best choice. Since a user can only look at the sky with the visual angle within the preference space, the inter-item competition can be captured by the competition among different items in partitioning the preference space into their dominating areas.

For our eBay case, we introduce a concrete approach to define the dominating area of item  $S_i$  ( $1 \leq i \leq n$ ). Recall

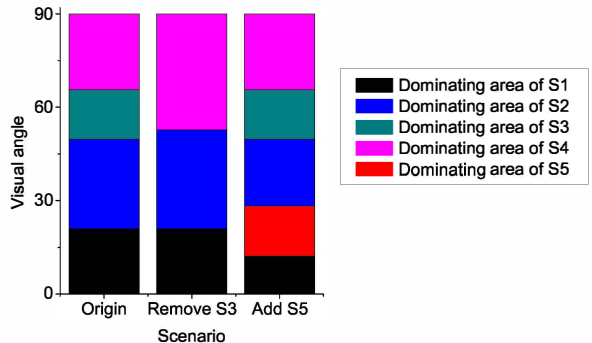


Fig. 2. The change of dominating areas when adding or removing an item

that each seller  $S_i$  is represented by an angle value  $z_i$  in the eBay example. Assume that all  $n$  items are ordered according to their visual angles from low to high. Let  $\mathfrak{R}(S_i) = [lower_i, upper_i]$  denote the dominating area of item  $S_i$ , where  $lower_i$  and  $upper_i$  denote the lower boundary and the upper boundary of the angle range of the dominating area  $\mathfrak{R}(S_i)$ . Then  $\mathfrak{R}(S_i)$  is defined as follows:

$$\mathfrak{R}(S_i) = \begin{cases} [0, \frac{z_1+z_2}{2}] & \text{if } i = 1, \\ [\frac{z_{i-1}+z_i}{2}, \frac{z_i+z_{i+1}}{2}] & \text{if } 1 < i < n, \\ [\frac{z_{n-1}+z_n}{2}, 90] & \text{if } i = n. \end{cases} \quad (13)$$

We use the sellers in Table I to compute their dominating areas. Recall Figure 1, the visual angles of the four sellers are  $z_1 = 5^\circ$ ,  $z_2 = 37^\circ$ ,  $z_3 = 63^\circ$ , and  $z_4 = 69^\circ$ . Their dominating areas are calculated as  $\mathfrak{R}(S_1) = [0^\circ, 21^\circ]$ ,  $\mathfrak{R}(S_2) = [21^\circ, 50^\circ]$ ,  $\mathfrak{R}(S_3) = [50^\circ, 66^\circ]$ , and  $\mathfrak{R}(S_4) = [66^\circ, 90^\circ]$ . In fact, we can see from Figure 1 that the entire preference space from  $0^\circ$  to  $90^\circ$  is divided by three angle bisectors,  $L_1 = \frac{z_1+z_2}{2} = 21^\circ$ ,  $L_2 = \frac{z_2+z_3}{2} = 50^\circ$ , and  $L_3 = \frac{z_3+z_4}{2} = 66^\circ$ , into four dominating areas  $\mathfrak{R}(S_1)$ ,  $\mathfrak{R}(S_2)$ ,  $\mathfrak{R}(S_3)$ , and  $\mathfrak{R}(S_4)$ .

Both equation (13) and the example above show that during the calculation of the dominating area of an item, we consider not only the attributes of this seller but also the competing neighborhood sellers. **Inter-item competition** happens when two items have adjacent dominating areas. Thus we define **the neighbors of an item** as the items whose dominating areas are adjacent to the dominating area of this item. For example, the neighbors of  $S_2$  are  $S_1$  and  $S_3$ .

Obviously, the dominating areas will change if an item is removed from or added to the item set. Figure 2 shows the dominating areas of three scenarios for the eBay example; the x-axis means different scenarios and the y-axis is the whole preference space. In the first scenario (column 1), there are four sellers as shown in Figure 1. In the second scenario (column 2),  $S_3$  is removed from Figure 1. Thus the dominating areas of its neighbors,  $S_2$  and  $S_4$ , will increase. In the third scenario (column 3), a new seller  $S_5$  with price \$500 and reputation 200 is added; this new seller will compete with its neighbors,  $S_1$  and  $S_2$ , and their dominating areas will reduce.

### C. Density functions of users

Beside inter-attribute tradeoff and inter-item competition, the third key factor that plays a critical role in achieving high accuracy for multi-attribute ranking is to capture the diversity and uncertainty in users' preferences. Naturally, different users

may have different preferences over the same set of multi-attribute items. For instance, Alice and Bob may not choose the same iPhone seller as their best choices. Such preferences may depend on the life styles and income levels of the users and may change over time. Second, even the same user may have different preferences under different contexts. For example, by analyzing the past selection behaviors of a user, we observe that this user prefers the sellers with high reputations and moderate prices sometimes, but prefers the sellers with low prices and moderate reputations at some other times.

In order to capture such uncertainty and diversity, we propose to use a probability density function to capture a user's personal preferences based on his/her past item selection behaviors. Concretely, Let  $D(z)$  denote the probability that this user looks at the space with the visual angle  $z$ . As  $z$  varies,  $D(z)$  changes. Different probabilities on different visual angles reflect this user's personal preferences. In the eBay example, the density function of Alice could look like the one shown in Figure 3(a). In this figure, the x-axis means the preference space ranging from  $0^\circ$  to  $90^\circ$ . The y-axis means the probability of Alice to look at the sky with a specific visual angle.

In the remainder of this section, we will answer two questions: (1) how to estimate the probability of an item being the best choice given a user's density function, and (2) how to estimate a user's density function from this user's past selection behaviors.

First, we will discuss **how to estimate the probability of an item being the best choice**. Given the dominating area of item  $S_i$  and the density function of a user, we can calculate the probability that the user chooses this item as his/her best choice by accumulating the density function within this item's dominating area:

$$V_i = \int_{z \in \mathfrak{R}(S_i)} D(z) dz. \quad (14)$$

Recall that  $V_i$  is the ranking score of  $S_i$ . We can then rank all items  $\{S_1, S_2, \dots, S_n\}$  according to  $V_i$  ( $1 \leq i \leq n$ ) from high to low.

In the eBay example, based on Alice's density function shown in Figure 3(a) and the dominating areas shown in Figure 2 (the first scenario), we calculate  $V_1 = 39\%$ ,  $V_2 = 11\%$ ,  $V_3 = 31\%$ , and  $V_4 = 19\%$ . (The dashed lines in Figure 3(a) divide the density function for the four sellers based on their dominating areas.) Thus, for Alice, the four sellers will be ranked in the order of  $S_1 > S_3 > S_4 > S_2$ . However, in the second scenario where  $S_3$  is removed from Figure 2, the probabilities for sellers  $S_1, S_2, S_4$  become  $V_1 = 39\%$ ,  $V_2 = 14\%$ , and  $V_3 = 47\%$ , which yields the ranking order of  $S_4 > S_1 > S_2$ . Similarly, in the third scenario where  $S_5$  is added to Figure 2, the ranking order becomes  $S_3 > S_5 > S_1 > S_4 > S_2$ .

The equation (14) and the above discussion on the running eBay example show that both a user's density function and the inter-item competition between neighboring items influence the probability of an item being the best choice of the user. *To the best of our knowledge, MAPS is the first work that addresses inter-item competition.*

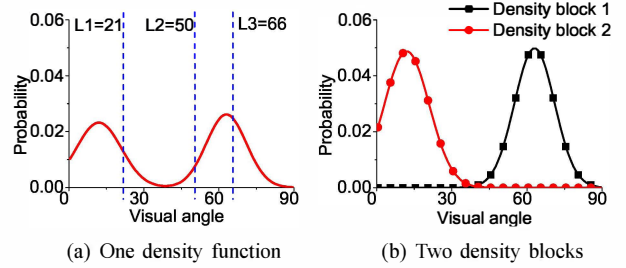


Fig. 3. Example of density function and density blocks

Now we answer the second question of **how to infer a preference density function from a user's past selection behaviors**. Assume that we have a set of past selection behaviors, denoted by  $H^{set} = \{H_1, H_2, \dots, H_L\}$ , for a given user and the size of the past selection behaviors is  $L$ . Each past selection behavior ( $H_j \in H^{set}$ ) records an item set ( $S_j^{set}$ ), and the best choice ( $S_{B_j}$ ) of this user within this item set.

Recall our eBay case, we construct one density function over the preference space from the past selection behaviors in two steps: (1) We represent each past selection behavior as a density block. (2) We accumulate these density blocks into a density function  $D(z)$  for the user.

First, we use kernel density estimation [8] to construct one density block from the  $j^{th}$  past selection behavior. We use  $d_j(z)$  to denote the density block for the  $j^{th}$  past selection behavior. Let  $S_{B_j}$  denote the user's best choice and  $S_j^{set}$  denote the item set in this past selection behavior. Let  $z_{B_j}$  denote the angle value of  $S_{B_j}$ . Note that this angle describes the user's preference in the  $j^{th}$  past selection behavior.

The density block  $d_j(z)$  is a shape of Gaussian distribution. We choose Gaussian distribution since its mathematics foundation in kernel density estimation [8]. However, MAPS framework can use other probability distributions to generate density blocks. Formally, we construct  $d_j(z)$  in our eBay case as follows.

$$d_j(z) = \mathcal{N}(\mu_j, \delta_j), \quad (15)$$

The *mean* of  $d_j(z)$ , denoted by  $\mu_j$ , is the visual angle of the user's best choice in the  $j^{th}$  past selection behavior.

$$\mu_j = z_{B_j}. \quad (16)$$

The *variance* of  $d_j(z)$ , denoted by  $\delta_j$ , is the average angle distance from  $S_{B_j}$  to its neighbors. If we assume that all the items in the item set are ranked by their angles from small to large, we know the two neighbors of  $S_{B_j}$  is  $S_{B_{j-1}}$  and  $S_{B_{j+1}}$ . The exception happens only when  $S_{B_j}$  is the first item or the last item in the item set since it will only have one neighbor. Therefore, we can calculate the variance as follows.

$$\delta_j = \begin{cases} z_{B_{j+1}} - z_{B_{j-1}} & \text{if } B_j = 1, \\ \frac{(z_{B_{j+1}} - z_{B_j})^2 + (z_{B_j} - z_{B_{j-1}})^2}{2} & \text{if } 1 < B_j < |S_j^{set}|, \\ z_{B_j} - z_{B_{j-1}} & \text{if } B_j = |S_j^{set}|. \end{cases} \quad (17)$$

The choice of the mean and the variance ensures two properties. First, the density block constructed from the  $j^{th}$  past selection behavior mainly resides in the dominating area of the best choice in that selection. Second, there is non-zero probability that the density block resides in other items'

dominating areas to capture the uncertainty and diversity inherent in the user's selection behaviors.

Figure 3(b) shows the example of two density blocks for the eBay scenario. One is marked by the red circle curve and the other is marked by the black rectangle curve. If Alice chose  $S_3$  as her best choice previously and we know that the angle value of  $S_3$  is  $63^\circ$  from Figure 1, then this past selection behavior introduces a Gaussian shape density block with  $\mu = 63^\circ$ . In the 2D space of Figure 1, we calculate the variance as the average angle distance from the user's best choice to its two neighbors. For  $S_3$ , its neighbors are  $S_2$  with visual angle  $37^\circ$  and  $S_4$  with visual angle  $69^\circ$ . The variance is calculated as  $\delta = \frac{(69-63)+(63-37)}{2} = 16^\circ$ . This density block is represented as the black rectangle curve in Figure 3(b).

In MAPS, when the system knows a user's best choices in the past  $L$  selection behaviors, the system constructs  $L$  density blocks as  $d_j(z)$  for  $1 \leq j \leq L$ . The overall density function of a given user is constructed by normalizing the corresponding  $L$  density blocks as:

$$D(z) = \frac{\sum_{j=1}^L d_j(z)}{\int_0^{90} \sum_{j=1}^L d_j(z) dz} \quad (18)$$

The density function in Figure 3(a) is normalized from the two density blocks in Figure 3(b).

We now discuss how to estimate the probability of the item being the best choice of the user based on the density function constructed from function (18). In the first prototype implementation of MAPS, we use function (19) to cumulate the dense function of Gaussian distribution with mean  $h_\mu$  and variance  $h_\delta$  from  $-\infty$  to  $x$ .

$$F(x, h_\mu, h_\delta) = (1 + \operatorname{erf}(\frac{x - h_\mu}{\sqrt{2} \cdot h_\delta}))/2, \quad (19)$$

where  $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$  is the well-known error function encountered in integrating Gaussian distribution [8].

We can prove that the computation complexity of MAPS for the eBay case is  $O(n \cdot L)$ . For storage, MAPS only needs to store the mean and variance values (i.e.,  $\mu_i$  and  $\delta_i$ ) for each past selection behavior. This yields low storage complexity since the storage cost depends only on the length of the history ( $L$ ). From our later experiment, we know that  $L = 32$  is enough for accurate prediction.

#### IV. EXPERIMENT

We evaluate MAPS through both synthetic simulations and real-user experiments. In synthetic simulations, we concentrate on the comparison between MAPS and existing weight-based multi-attribute ranking algorithms. The factors that affect the accuracy of MAPS are also examined. In real-user experiments, we evaluate MAPS using the first prototype system of MAPS [9] with 50 real users. The experiments reported in this section concentrate on the eBay scenarios used throughout the paper. The experiments on high dimensions and with real users are given in our technical report [6].

##### A. Simulation configuration

We simulate the scenario that a user selects his/her favorite seller for a particular product in e-market, such as Amazon and eBay. The simulation environment is composed by five parts: (1) generating seller sets, (2) simulating a user's selection

behaviors, (3) implementation of MAPS, (4) implementation of weight-based multi-attribute ranking algorithms, and (5) performance evaluation criteria.

**Generating seller sets.** To generate the seller set for a query, we need to determine the number of sellers ( $n$ ) as well as their prices and reputations. Based on real data collected from Amazon and eBay, we obtain the following observations. The price is mostly within  $[10, 1000]$ ; the reputation is mostly within  $[0, 10^6]$ ; the number of sellers for a query is mostly within  $[20, 100]$ ; and both price and reputation follow power-law distributions. Based on these observations, we generate the seller set for each query as follows.

First, the size of seller set, denoted by  $n$ , is randomly chosen within  $[20, 100]$ . We will evaluate larger item set in Section IV-C.

Second, the minimum price and maximum price are randomly chosen within  $[10, 1000]$ . Then,  $n$  different price values are generated according to the power-law distribution, within the range of minimum price and maximum price. Let  $\{p_1, p_2, \dots, p_n\}$  denote these price values, ordered from low to high.

Third, the minimum reputation and maximum reputation are randomly chosen within  $[0, 10^6]$ . Then,  $n$  different reputation values are generated according to the power-law distribution, within the range of minimum reputation and maximum reputation. Let  $\{r_1, r_2, \dots, r_n\}$  denote these reputation values, ordered from low to high.

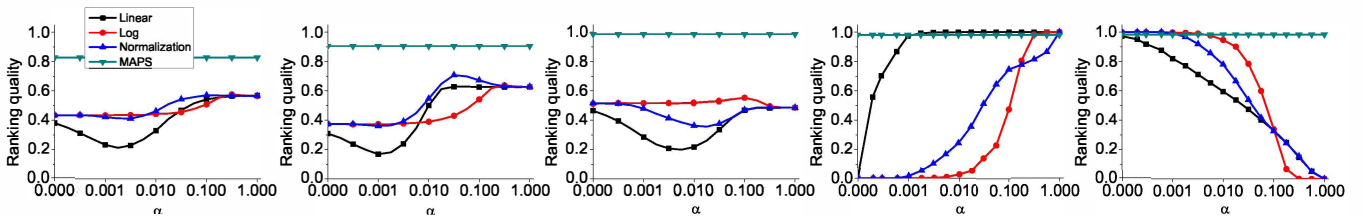
Finally, combine reputations and prices to generate  $n$  items where item  $S_i$  has reputation  $r_i$  and price  $p_i$ . By doing so, all items are skyline items.

**Simulating users' selection behaviors** To our best knowledge, none of the existing work provides usable models to simulate the users' uncertain and diverse selection behaviors. Therefore, we interviewed 30 people to understand their selection principles when choosing eBay sellers, and summarized their behaviors into four categories. Although this approach may not cover all possible user behaviors, it provides good guidance to generate synthetic but representative users in our simulations. The categories of synthetic users are summarized as follows:

**Behavior category I: price threshold.** Users in this category first filter out the sellers whose price is larger than a *price threshold*. Their best choice is the remaining seller with the highest reputation. The price thresholds are often highly correlated to the price of the items in the seller set. In this category, we construct one synthetic user, denoted as  $U_1$ , whose price threshold is the average price of the sellers in the seller set.

**Behavior category II: dynamic reputation threshold.** Users in this category first filter out the sellers whose reputation is lower than a *reputation threshold*. Their best choice is the remaining seller with the lowest price. The reputation threshold is related to the price which means that the more expensive of the item, the higher of the reputation threshold. In this category, we construct one synthetic user, denoted as  $U_2$ , whose reputation threshold is  $50 \times$  *average price*.

**Behavior category III: fixed reputation threshold.** Users



(a) Ranking quality for  $U_1$  (b) Ranking quality for  $U_2$  (c) Ranking quality for  $U_3$  (d) Ranking quality for  $U_4$  (e) Ranking quality for  $U_5$   
 Fig. 4. Ranking quality of different algorithms for the users in four categories

in this category have similar behaviors as the users in category II, except that the reputation threshold is fixed. We construct one synthetic user, denoted as  $U_3$ , whose reputation threshold is 1000.

**Behavior category IV: extreme selection.** Users in this category consider only one attribute and neglect other attributes. We construct two synthetic users.  $U_4$  selects the seller with the highest reputation and  $U_5$  selects the seller with the lowest price.

We would like to point out that after conducting simulations for many synthetic users with different threshold values, we observe that the performance of MAPS is **insensitive** to the threshold settings in the four behavior categories. In this section we show the results for the five representative users.

**Implementation of MAPS** We set the MAPS parameters as  $\beta = 10^8$  and  $L = 32$  in Section IV-B, and evaluate the performance of MAPS when varying these parameters in Section IV-C.

**Implementation of weight-based multi-attribute ranking algorithms** For these algorithms, we implement three typical utility functions. The first is a linear combination of price and reputation as

$$U(S_i) = \alpha \cdot r_i + (1 - \alpha) \cdot (-p_i). \quad (20)$$

Recall that  $U(S_i)$ ,  $r_i$ , and  $p_i$  denote the utility score, reputation, and price for seller  $S_i$ , respectively.

The second utility function adopts the log function [10], [11].

$$U(S_i) = \alpha \cdot \log(1 + r_i) + (1 - \alpha) \cdot (-\log(1 + p_i)) \quad (21)$$

The constant value 1 is added to avoid negative logarithm values when reputation or price is smaller than 1.

The third utility function adopts the normalization functions (8) and (9) used in MAPS.

$$U(S_i) = \alpha \cdot \frac{r_i}{\sqrt{r_i \cdot r_i + \beta}} + (1 - \alpha) \cdot \left(1 - \frac{p_i}{\sqrt{p_i \cdot p_i + \beta}}\right). \quad (22)$$

**Performance evaluation criteria** We use *ranking quality* defined in equation (1) in our evaluation. The example used to explain it is given in Section II-C.

For each configuration, we run simulations for 1,000 times to obtain the average result.

### B. Comparing MAPS with weight-based multi-attribute ranking algorithms

To facilitate the understanding of our experiments, we discuss the evaluation method used to compare MAPS and weight-based multi-attribute ranking algorithms before presenting the results.

**Evaluation method** The performance of weight-based multi-attribute algorithms is very sensitive to the selection of the weight value  $\alpha$ . It requires to learn the best  $\alpha$  value setting for each individual user in a given context, which is considered one of the difficult tuning parameter for them. Instead of investigating specific ways to obtain the best settings of  $\alpha$  value in our experiments, we compare MAPS with the upper bound performance of weight-based multi-attribute algorithms by measuring the performance with varying  $\alpha$  values within the  $\alpha$  range. The upper bound performance of weight-based multi-attribute algorithm is represented by the highest points on the curves of the algorithm. Since the curves for MAPS are constant as it does not depend on  $\alpha$ , we will compare the MAPS line with the highest points of the curves for weight-based multi-attribute algorithms.

We would like to point out that this comparison method is fair since it uses the best possible settings of  $\alpha$  to compare the weight-based multi-attribute ranking algorithms with MAPS. In fact none of the existing weight-based multi-attribute ranking algorithms can yield the best choice of  $\alpha$  for different types of users in a given context, especially when there is diversity and uncertainty in users' selection behaviors.

**Comparison in terms of ranking quality** Figure 4 shows the comparison between MAPS and the weight-based multi-attribute ranking algorithms with three most popular utility functions (recall Section IV-A). The x-axis represents the various settings of  $\alpha$  for the weight-based multi-attribute ranking algorithms and the y-axis shows the measured ranking quality of MAPS and the measured ranking quality of the weight-based multi-attribute algorithms.

Figure 4(a) shows the result for synthetic users (e.g.,  $U_1$ ) in Category I. The ranking quality of MAPS for this category of users is 0.83. This means that the best choice for users of  $U_1$  type is ranked higher than 83% of items in the item set. The performance upper bounds of weight-based multi-attribute approach with linear, log, and normalization utility functions are at best 0.57 when  $\alpha$  is around 1. This experiment shows that MAPS improves ranking quality by 26% over the weight-based multi-attribute approach, no matter which utility function is used and what  $\alpha$  value is set. Clearly, this is a *significant performance improvement* for users of  $U_1$  type.

Figure 4(b) measures the ranking quality for the synthetic users (e.g.,  $U_2$ ) in Category II with varying  $\alpha$  values. Similarly, MAPS improves the ranking quality over the weight-based multi-attribute algorithms by 20% comparing to the highest ranking quality of the weight-based multi-attribute algorithm with normalization based utility function. Similar observation is shown in Figure 4(c) for the synthetic users (e.g.,  $U_3$ ) in

Category III. MAPS improves the ranking quality of weight-based multi-attribute approach by 43% over log based utility function, 46% over normalization based utility function, and 50% over linear based utility function no matter what  $\alpha$  value is used.

Finally, we run the performance comparison for Category IV users with simplified selection behaviors (i.e.,  $U_4$  and  $U_5$ ) in Figure 4(d) and Figure 4(e). In these extreme cases users simply prefer those items (sellers) based only on one attribute. That is,  $U_4$  always chooses the highest reputation and  $U_5$  always chooses the lowest price. It is obvious that both MAPS and weight-based multi-attribute algorithms can achieve the best ranking quality. It is worth to point out that the weight-based multi-attribute approach can only achieve good results for Category IV users when  $\alpha$  is properly chosen, which is known to be a hard problem for weight-based multi-attribute algorithms.

The group of experiments in Figure 4 also shows that the performance of weight-based multi-attribute approach not only depends on the choice of  $\alpha$  but also depends on the choice of utility function. For instance, the weight-based multi-attribute with normalization function can achieve the best results for users of type  $U_2$  (Figure 4(b)), but the log function is the best for users of type  $U_3$  (Figure 4(c)). In comparison, we evaluate the performance of MAPS for different normalization functions and the results are similar, which shows that the choice of normalization function does not have any significant impact on the MAPS performance.

This group of experiments shows that MAPS achieves much better performance than the upper bound performance of weight-based multi-attribute ranking algorithms. The advantage of MAPS comes from its unique features: the visual angle representation of items and user preferences, the computation of best-choice probability based on inter-attribute tradeoff and inter-item competition, and its modeling of diversity and uncertainty of users' selection behaviors through density functions. Also MAPS does not rely on setting certain specific parameters for different users (see the next section for detail), whereas existing weight-based multi-attribute ranking algorithms are sensitive to the choice of utility function and the proper setting of weight value  $\alpha$ .

### C. Factors affecting MAPS performance

In this section, we investigate how the performance of MAPS is affected by three factors: the seller set size ( $n$ ), the history length ( $L$ ), and the  $\beta$  value in equation (8) and (9). Since the ranking quality for extreme users in Category IV will not change much with respect to  $n, L, \beta$ , we only show the results for users in the first three categories, represented by  $U_1, U_2$ , and  $U_3$ .

**Effect of seller set size** In the previous simulations, the size of seller set ( $n$ ) is randomly chosen between 20 and 100. Now we run different tests by changing  $n$  from 50 to 500. In Figure 5(a), the x-axis represents the various values of  $n$  and the y-axis measures the ranking quality for each  $n$  value. We can see that even in such a wide range of  $n$ , the ranking quality of MAPS is still larger than 0.8. Namely, the ranking score of the best-choice item is larger than 80% of the other

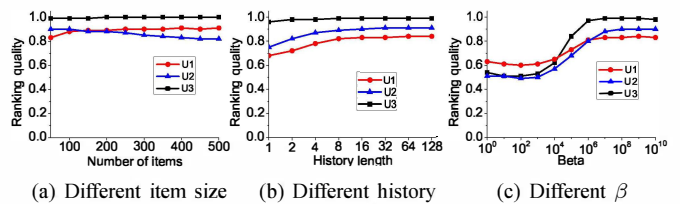


Fig. 5. Performances of MAPS with different parameters

items. This indicates that MAPS is not sensitive to the size of item set being ranked.

**Effect of history length** In the previous simulations, the density functions are estimated from 32 past selection behaviors, i.e.,  $L = 32$ . In Figure 5(b), we vary the history length  $L$  from 1 to 128 (x-axis), and measure the ranking quality (y-axis). We see that increasing the length of the history can increase the ranking quality. However, after  $L$  reaches 32, the ranking quality does not change much. Therefore, MAPS in most cases would need no more than 32 past selection behaviors to accurately predict a user's best choice. More importantly, even when only a couple of past selection behaviors are known (when  $L = 1, 2, 4$ ), MAPS achieves good results (about 0.7~0.78 for Category I users, 0.78~0.85 for Category II users, and 0.98~1.0 for Category III users). This shows that MAPS can reach high ranking quality with a very short learning curve.

**Effect of parameter  $\beta$**  The parameter  $\beta$  is used in the normalization equation (8) and (9). In Figure 5(c), the x-axis is the  $\beta$  value varying from  $10^0$  to  $10^{10}$ , and the y-axis is the ranking quality. We can see that (1) the ranking quality increases with the increase of  $\beta$  and (2) the ranking quality does not change much after  $\beta$  reaches  $10^8$ . This is because  $\beta$  is simply a system-level parameter used in the normalization, and the setting of its value only depends on the value range of the attributes. For the experimental datasets, MAPS can achieve the best results when the  $\beta$  value is comparable to the square of reputation and price. Since most of the reputation and price values is within  $[0, 10^4]$  due to power-law distributions, we only need to set  $\beta = 10^8$  in our experiments. In MAPS,  $\beta$  is a system-defined parameter applied to all users once it is set.

## V. RELATED WORK

The personalized multi-attribute ranking problem and the proposed solution are related to many research topics, including recommender systems [12], web search [13], and database queries [14]. In this section, we review related work according to the challenges in our problem: (1) modeling inter-attribute tradeoff, (2) modeling inter-item competition, (3) inferring a user's personal preferences, and (4) fundamental ranking methodology.

**Modeling inter-attribute tradeoff:** There are two types of existing approaches that address the tradeoff among multiple attributes. The first type focuses on identifying the attributes that are important for ranking. The goal is to provide a personalized set of attributes to determine skyline points [15]. Some work further organize these attributes into an importance hierarchy [16], [17]. The work may reduce the number of skyline items but they cannot rank them. In the second type,

weight values are used to describe the relative importance of multiple attributes. The representative schemes [2], [3] are solely based on attribute-weighting [18], whose limitations have been discussed in Sec II-D.

*Modeling inter-item competition:* To our best knowledge, this paper is the first work that formally addresses inter-item competition. Previously, some researchers realized the *consequence* of inter-item competition from different views, such as increasing the diversity of top-k set to improve the quality of recommendation [14], [19]. However no solid study on the cause, i.e. inter-item competition, is available.

*Inferring user preferences:* The user preferences can be obtained through either explicit or implicit ways. Many existing systems use explicit methods, such as asking the users to input their preferences directly [4] or through answering a set of interactive questions [3], [20]. Explicit methods obviously add burden to the user side, and the implicit methods are more desirable. However, the implicit methods in the current literature [18] cannot be used to solve the problem in this paper for two reasons. First, they highly depend on the specific representation of user preferences. Second, when there is uncertainty in a user's behaviors, they would need a lot of historical data to construct the user preference model. But MAPS works when only a few historical data are available.

*Fundamental ranking methodology:* In most of the existing ranking algorithms, the ranking score of an item describes this item's relevance to the query [14], importance [13], match to a user's taste [3], [12], and so on. Similar to the weight-based multi-attribute approach, they do not address inter-item competition, which is a critical factor in personalized multi-attribute ranking problem. In addition, their ranking scores often do not have clear physical meanings. In MAPS, however, the ranking score is the probability of an item being a user's best choice. This is another advantage of MAPS. With a clear physical meaning, the ranking scores in MAPS can be used by other algorithms that would need to know the probabilities of users selecting certain items.

## VI. CONCLUSION

Social computing and social networking are one of the emerging forms of collaborative computing. Personalized ranking is a fundamental capability of collaborative computing in social networks and eCommerce today. We have presented MAPS, a novel multi-attribute probabilistic selection framework for personalized multi-attribute ranking. MAPS presents a number of unique features: the invention of visual angle model to depict inter-attribute tradeoff, the introduction of dominating area to model inter-item competition, and the utilization of density function to capture uncertainty and diversity in a user's preferences. In addition, MAPS computes the ranking of an item using the probability of this item being the best choice for a given user in terms of inter-attribute tradeoff, inter-item competition, and personalized user preferences. The effectiveness of MAPS is evaluated with extensive simulations through fair comparisons with existing multi-attribute ranking algorithms. We show that MAPS significantly outperforms them in terms of ranking quality.

**Acknowledgement:** The first author performed this work while he was a visiting PhD student at Georgia Institute of Technology, funded under China Education Scholarship. Ling Liu is partially supported by grants from NSF NetSE and NSF CyberTrust, an IBM SUR grant, an IBM faculty award, and an Intel research council grant. Yan Sun is partially supported by NSF (0643532). Ting Yu is partially supported by NSF (CNS-0747247 and IIS-0914946). Yafei Dai is partially supported by 973 Program (2011CB302305).

## REFERENCES

- [1] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *ICDE*, 2001.
- [2] U. Junker, "Preference-based search and multi-criteria optimization," in *AAAI*, 2002.
- [3] P. Viappiani, B. Faltings, and P. Pu, "The lookahead principle for preference elicitation: Experimental results," *Lecture Notes in Computer Science*, vol. 4027, p. 378, 2006.
- [4] J. Butler, J. Dyer, J. Jia, and K. Tomak, "Enabling e-transactions with multi-attribute preference models," *European Journal of Operational Research*, vol. 186, no. 2, pp. 748–765, 2008.
- [5] Q. Feng, K. Hwang, and Y. Dai, "Rainbow: Multi-attribute product ranking to advance qos in on-line shopping," *IEEE Internet Computing*, vol. 13, no. 5, pp. 72–80, 2009.
- [6] Q. Feng, L. Liu, Y. Sun, T. Yu, and Y. Dai, "Find your favorite star with maps: a personalized multi-attribute ranking algorithm," in *Technical Report*, 2010. [Online]. Available: <http://fengqinyuan.com/wp-content/uploads/2010/06/TR2010.pdf>
- [7] K. P. Yoon and C.-L. Hwang, *Multiple Attribute Decision Making: An Introduction*. Sage Publications, Inc, 1995.
- [8] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [9] "The maps demo homepage," 2010. [Online]. Available: <http://mapsprojectapp.appspot.com>
- [10] N. Archak, A. Ghose, and P. G. Ipeirotis, "Show me the money!: deriving the pricing power of product features by mining consumer reviews," in *KDD*, 2007.
- [11] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews," in *ICEC*, 2007.
- [12] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: a Survey of the State-of-the-Art and Possible Extensions," *IEEE TKDE*, vol. 17, no. 6, pp. 734–749, June 2005.
- [13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford, Tech. Rep., 1998.
- [14] Z. Chen and T. Li, "Addressing diverse user preferences in sql-query-result navigation," in *SIGMOD*, 2007.
- [15] D. Xin and J. Han, "P-cube: Answering preference queries in multi-dimensional space," in *ICDE*, 2008.
- [16] W. Kießling, "Foundations of preferences in database systems," in *VLDB*, 2002.
- [17] D. Mindolin and J. Chomicki, "Discovering relative importance of skyline attributes," in *VLDB*, 2009.
- [18] J. Wallenius, J. Dyer, P. Fishburn, R. Steuer, S. Zions, and K. Deb, "Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead," *Management Science*, vol. 54, no. 7, p. 1336, 2008.
- [19] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia, "Recommendation diversification using explanations," in *ICDE*, 2009.
- [20] J. Zhang and P. Pu, "Refining preference-based search results through bayesian filtering," in *IUI*, 2007.