Rankr: A Mobile System for Crowdsourcing Opinions

Yarun Luon, Christina Aperjis, and Bernardo A. Huberman

HP Labs,
Palo Alto, CA 94304, USA
{yarun.luon,christina.aperjis,bernardo.huberman}@hp.com

Abstract. Evaluating large sets of items, such as business ideas, is a difficult task. While no one person has time to evaluate all the items, many people can contribute by each evaluating a few. Moreover, given the mobility of people, it is useful to allow them to evaluate items from their mobile devices. We present the design and implementation of a mobile service, *Rankr*, which provides a lightweight and efficient way to crowdsource the relative ranking of ideas, photos, or priorities through a series of pairwise comparisons. We discover that users prefer viewing two items simultaneously versus viewing one image at a time with better fidelity. Additionally, we developed an algorithm that determines the next most useful pair of candidates a user can evaluate to maximize the information gained while minimizing the number of votes required. Voters do not need to compare and manually rank all of the candidates.

Keywords: ranking, mobility, crowdsourcing, incentives, user interfaces.

1 Introduction

Increasingly, businesses and organizations want to encourage innovation by soliciting new ideas from employees and customers. But that is the easy part—figuring out which ideas are the best is a time-consuming process, often handled by a single person or a panel of reviewers. While some campaigns invite large populations to help vote for the best ideas, it is less likely that many people will have time to thoughtfully consider a large number of ideas. Similarly, developers building a product or service face a deluge of feature requests. More and more are adopting so-called agile methodologies, which requires them to regularly prioritize open issues. And individual people also face dilemmas of selection, as for example a photographer trying to choose which out of dozens of shots to enlarge.

Given the gargantuan number of choices that firms and consumers face when dealing with information, it is necessary to develop efficient mechanisms for filtering and ranking the set of possibilities. And since consumers and members of organizations are often mobile, it is important to be able to gather their opinions/votes in a mobile setting. In other words, people should not be required to vote from their desktops.

We propose a mobile service, Rankr, which provides a lightweight and efficient way to crowdsource the relative ranking of ideas, photos, or priorities. Rankr is a service we created and deployed for deriving a rank ordering of multiple objects, suggestions or

J.Y. Zhang et al. (Eds.): MobiCASE 2011, LNICST 95, pp. 20–31, 2012.

websites that uses pairwise comparisons among them. Unlike typical rank voting methods, voters do not need to compare and manually rank all of the candidates. Moreover, pairwise comparisons are well-suited for devices with smaller screens.

Given the votes that others have already cast, Rankr automatically determines the next most useful pair of candidates a user can evaluate to maximize the information gained while minimizing the number of votes required. This ensures that the mechanism scales beyond traditional voting schemes while enabling the crowdsourced ranking of many items–many more than any one user could be expected to evaluate.

We implemented our solution as a mobile service that systematically displays a pair of items side by side. A user (which could be an enterprise) interested in having people select the best among a collection of items, begins by organizing the items into a poll. The poll may be comprised of text descriptions (e.g., business ideas, slogans, feature requests) or media (e.g., photos). Each poll is easily accessible through a range of devices, such as a desktop browser and any mobile device.

The wide accessibility of Rankr together with the fact that each user is not required to perform a large number of comparisons make our solution very effective for crowdsourcing the ranking of items. Users who have idle time can easily get into the system, perform as many comparisons as they wish, and have their work recorded by the system. Rankr scales well not only to users who spend a lot of time using the mobile service, but also to users who only spend a small amount of time (e.g., while waiting in line at the grocery store or while waiting for the train). These design decisions lend themselves to the philosophy that a user can contribute his or her opinion with very little effort and time.

2 Related Work

A variety of prior solutions exist to crowdsource opinions. Most of these approaches evaluate each item on its own without comparing it to other items. For instance, Starbucks allows consumers to suggest ideas and vote "up" or "down" on ideas of others at "My Starbucks Idea".[1] Dell offers a similar service called IdeaStorm on its webpage, where people can promote or demote ideas.[2] However, when the goal of the evaluation is to get a ranking of the items, it is important to see how different items compare to each other. The simplest way to achieve this is through pairwise comparisons.

The method of paired comparisons [2] presents items in pairs for comparative judgment to one or more judges. The outcomes of the pairwise comparisons are then used to assign a score to each item. The estimated scores can in turn be used to rank the items. The Elo method addresses the problem of estimating the scores using data from pairwise comparisons under specific assumptions about the underlying distribution [3].

The Elo method has been used in a variety of settings, most notably for ranking chess players, and has recently been applied for ranking in twitter-like forums [8]. It has also been used for eliciting preferences from pairwise comparisons of pictures in a matching game [5]. Pairwise comparisons are also used by FotoVotr[3] for its weekly photo contests

[1] `http://mystarbucksidea.force.com/`
[2] `http://www.ideastorm.com/`
[3] `http://www.fotovotr.com/`

and by the dating site OkCupid[4] to obtain a ranking for potential profile pictures of a user; however, we do not know what algorithms are used by these websites.

In this paper, we use the Elo method to rank items of various types (such as business ideas, priorities or agile feature requests) in a mobile setting. We use ordinal pairwise comparisons, but note that the problem of obtaining aggregate rankings from cardinal pairwise comparisons has also been studied [6].

The problem of obtaining an aggregate ranking that represents the preferences of a group of people accurately and fairly is a central question in social choice theory. The difficulties in accomplishing this task are illustrated in Arrow's impossibility theorem [1], which states that when voters have three or more distinct alternatives, no voting system can convert the ranked preferences of individuals into a community-wide ranking that meets a certain set of criteria. On the other hand, it is possible to get a satisfactory aggregate ranking under certain assumptions on individual preference [7]. However, here we do not consider these concerns of social choice theory. Moreover, we do not require knowledge of the complete preferences of each user. Instead, each user may only perform a few pairwise comparisons which are used to obtain an aggregate ranking with the Elo method.

Our approach is also related to conjoint analysis, a statistical technique used in market research to determine how people value different features that make up an individual product or service [4]. Rankr, however, focuses on ranking items and does not consider how individual features of an item affect people's preferences.

3 Design

Rankr is a mobile service that displays a pair of items side by side. A user interested in having people select the best among a collection of items begins by creating a set of them which Rankr can then access. The collection may be comprised of text descriptions (e.g., business ideas, slogans, feature requests) or media (e.g., photos). We refer to such a collection as *a poll*. In the case of photos, they can be uploaded to the desktop version of Rankr. Polls that are text based can be created via the desktop or mobile version.

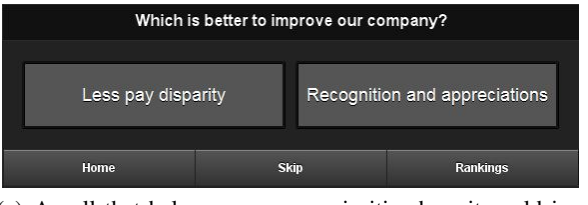
The creator of the poll then sends a link to a list of people, asking them for help in evaluating the items. When a recipient clicks on the Rankr link, s/he is presented with a pair of items and is asked to select the item that is better with respect to some criterion. For instance, in the case of photos, the question could be: "Which photo would look better in a desk frame?" Some other examples are shown in Figure 1. After the user chooses one of the items, the page performs an AJAX reload with another pair so that s/he can continue voting on pairs if s/he wants to. In order to avoid biases, because the order in which items are presented/loaded can vary due to the asynchronous nature of the Internet, in the Rankr prototype, we insured that both images were loaded and then presented simultaneously to the user.

Given that we are interested in aggregating opinions from a large and heterogeneous crowd, we allow users to enter and leave the system at any time, having completed an

[4] `http://www.okcupid.com/`

(a) A poll that ranks pictures.

(b) A poll that measures brand awareness.



(c) A poll that helps a company prioritize how it could improve employee satisfaction.

Fig. 1. Examples of polls

arbitrary number of votes. In order to obtain a ranking of the items, we associate a score with each item. Once a user compares two items, the scores of those items are updated.

Users are able to see the relative rankings of the items they vote on at any time during the voting process (Figure 2). The rankings show the current rank of the item as well as its points. Points are a simplified representation of the difference between the items with the highest and lowest scores.

In Section 3.1, we discuss the user interface. In Section 3.2, we describe the algorithm that updates the items' scores in order to produce a ranking. In Section 3.3, we present the results of an internal pilot study. In Section 3.4, we discuss incentives for participation.

3.1 Interface

We produced a design that is intuitive for the user when s/he has to choose between two items. As a starting point, we chose to focus on the best way to display two images. To help solve this problem, usability tools, in particular a cognitive walkthrough, paper

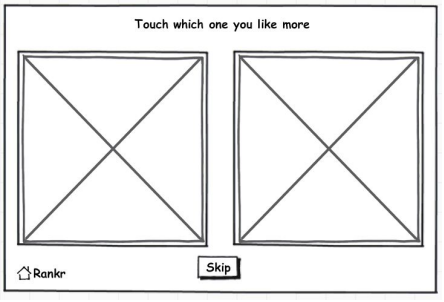
Fig. 2. The ranking of a list of items from a poll

prototypes, and a task analysis were employed. The cognitive walkthrough explored different ways to display two images side by side and yielded four possible interaction models. These models were translated into paper prototypes using Balsamiq[5], printed, and then cut. A usability study was conducted and the favored paper prototype was implemented. Next, a task analysis was done on the hi-fi prototype to insure the system was operating as intended and to seek further guidance on the next iteration of interface design. We will discuss in further detail the usability studies using paper prototypes and the task analysis.

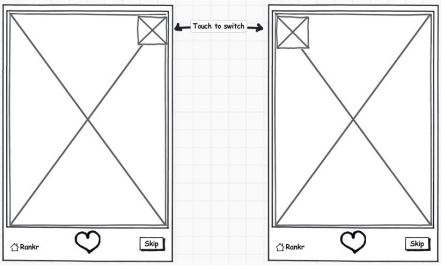
Paper Prototype. The four prototypes had a different interaction model for how the images would be displayed (Figure 3). One of the prototypes simply displayed both images side by side (Figure 3(a)). The remaining prototypes predominately displayed one of the images while giving cues to the second image (Figures 3(b), 3(c), 3(d)).

Six participants, recruited through availability sampling, between the ages of 20 and 50 with varying experience with touchscreen smartphones interacted with the lo-fi prototypes. The study revealed that participants unanimously *preferred viewing both images at the same time rather than viewing the images separately at higher fidelity*.

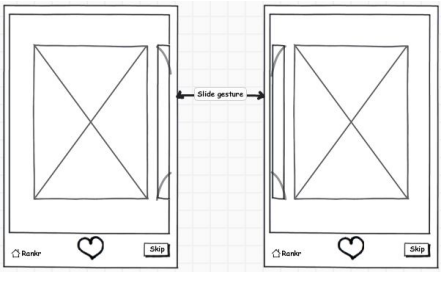
[5] `http://www.balsamiq.com`



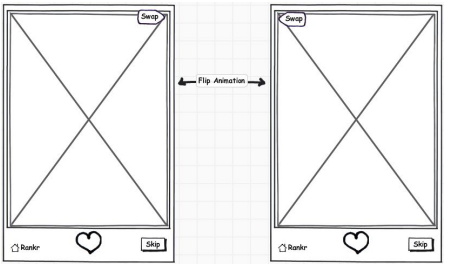
(a) **Pair Prototype:** Pictures are side by side.



(b) **Thumbnail Prototype:** One of the images takes up the majority of the screen. A thumbnail version of the other image is displayed in the corner.



(c) **Gallery Prototype:** One of the images takes up the majority of the screen. The second image is partially visible to either side of the image in focus. This interface suggests the user is viewing the images in a gallery.



(d) **Button Prototype:** One of the images takes up the majority of the screen. Also displayed on the screen is a labeled button (e.g., "swap"). Clicking the button switches the image to the other item.

Fig. 3. Paper Prototypes

It is interesting to note that all participants acknowledged the multi-touch reverse pinch gesture as a method to zoom in on the image to increase its fidelity.

A secondary goal of the paper prototypes was to determine an intuitive interaction model for the user to select the image they prefer. Possible methods included an icon by each picture the user would select to indicate the preferred item or a direct touch to select the image. Possible choices for the icon included a heart, a thumbs-up, or a smiley face (Figure 4). Out of the three, users preferred the heart icon. However most participants felt that the most intuitive interaction was to directly touch the image they preferred.

Task Analysis. The results of the task analysis were used to inform the next stage of design as well as provide a sanity check on the current system. Four participants, recruited through availability sampling, with familiarity of touchscreen smartphones between the ages of 20 and 40 were asked to do simple tasks using a first generation Nexus phone running Android 2.3. The four tasks were:

Fig. 4. Users were asked which icon best represented the action of preferring one image over another. Although users liked the heart icon the best, most users wanted to directly touch the image to indicate preference.

1. Choose a poll
2. Make pairwise comparisons
3. Skip a pair
4. View the rankings of the poll

Even though there were no cognitive mismatches between the system design and the user's conceptual model of how the system should operate, participants felt the interface was too information scarce and wanted to be more informed about their decisions.

The word "points" was confusing to users. They did not understand how points were being calculated nor how the points should be interpreted. To address this issue, we could look into a more intuitive way to display the scores and potentially explain to users how points are calculated.

3.2 The Algorithm

We now describe how the results of comparisons are used to (1) rank the items and (2) select the next pair to be compared.

Comparisons are translated into a rank ordering through the Elo method [3], which is used for calculating the relative skill levels of players in two-player games such as chess. A score is associated with every item, which increases whenever the item "wins" in a pairwise comparison and decreases whenever the item "loses" in a pairwise comparison. The magnitude of the increase (resp., decrease) depends on the difference between the score of the item and the score of the item that it defeated (resp., was defeated by). Winning against a high score item results in a larger increase than winning against a low score item. With the goal of achieving more accuracy at the top, the algorithm is more likely to select items with higher scores for the next comparison.

Let s_i be the score of item i. Suppose that item i is compared with item j. Let y be the indicator of whether item i is selected in the pairwise comparison between i and j (that is, $y = 1$ if item i is selected and $y = 0$ if item j is selected). After the comparison, the score of item i is updated according to:

$$s_i \leftarrow s_i + K\left(y - \frac{1}{1 + e^{s_j - s_i}}\right), \tag{1}$$

where K is a scaling parameter, often referred to as the "K-factor."

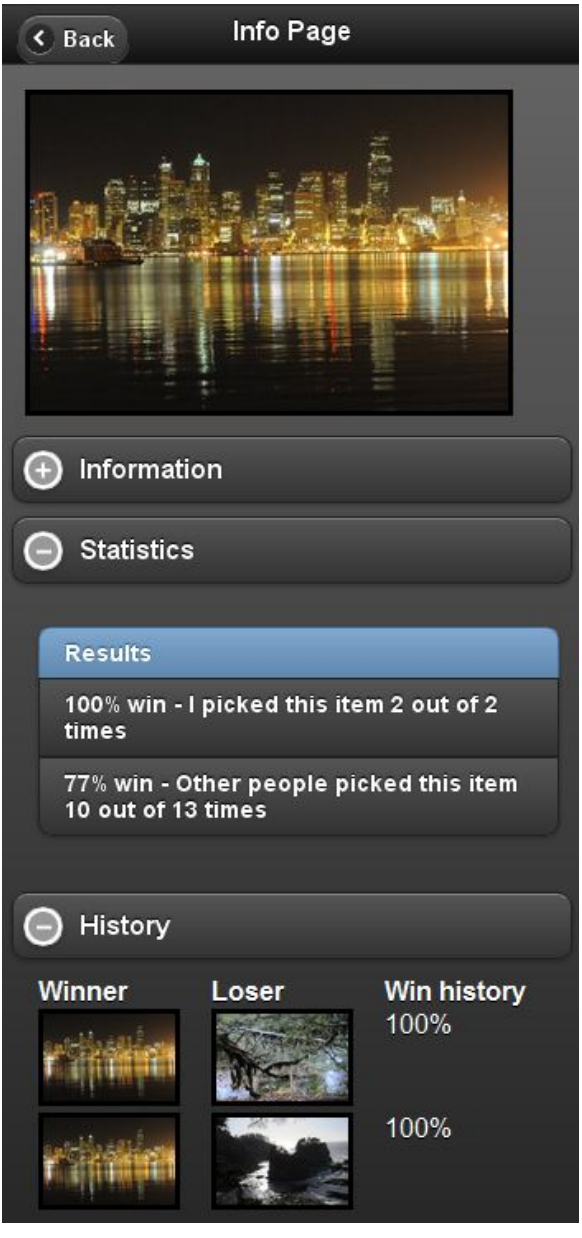


Fig. 5. Users can see the personal history of comparisons of a particular item from the results screen. This feature was heavily requested by participants in the task analysis.

The idea behind the Elo method is to model the probability of the possible outcomes of a pairwise comparison between item i and item j as a function of the scores of the two items. For instance, if the score of item i is greater than the score of item j, then item i should have a larger probability of being selected in a pairwise comparison between

the two items. The update rule given by (1) is derived under the assumption that an item with score s_i is selected against an item with score s_j with probability

$$\frac{e^{s_i}}{e^{s_i} + e^{s_j}} = \frac{1}{1 + e^{s_j - s_i}}.$$

While Elo is a method to calculate relative scores of items, it is not a ranking algorithm per se, in that it does not specify which two items should be in the next ballot. The easiest method would be to pick randomly these two items each time a user is willing to perform a comparison; however, this does not give preference to the top items. Since we are interested in identifying accurately the top items, rather than obtaining a reliable complete ranking, for the next comparison we preferentially select items which have high scores at the time the ballot is requested. This selection algorithm does not alter the Elo method, but rather considers the scores returned from the Elo method in determining the next ballot.

3.3 Pilot Test

The objective of the internal pilot was to gather perception of the best ping pong players out of 12 colleagues in addition to discovering any problems in the Rankr service. The poll was setup as a 12-item poll with each item being a ping pong player. A total of 278 comparisons were made from 8 participants who completed on average 35 comparisons or 53% of the total available ballots[6]. We discovered through the pilot that people were uncertain when to stop voting due to lack of a progress bar, and participants were confused when the lowest ranked item had 0 points when they knowingly voted for the item. The feedback from the pilot will help inform the next iteration of design.

3.4 Incentives for Participation

In certain cases, it may be useful to introduce an incentive mechanism that will induce people to volunteer their pairwise comparisons of whatever set of items or ideas they are presented with, and ideally convey a truthful opinion of what they think is best.

One such mechanism gives a reward to whoever ranks highest the item that all other participants also consider the most desirable. So all participants are not only choosing what they think is most desirable but also what they think others will find attractive and desirable. A variant of this mechanism would have the participants paying a small amount of money to enter into this "game" and receive the total amount collected as a prize. The introduction of a monetary reward could encourage the participants to think carefully about their choices. Alternately, the evaluation of these items can be done quickly and independently so that these comparisons could be farmed out to workers on a service like Amazon's Mechanical Turk.[7] Users could also be incentivized to do comparisons in exchange for having others evaluate their own items, in a tit-for-tat scheme. These could naturally be combined: users could rank friends' content just to be helpful, or rank strangers' content to earn credits for getting their own content evaluated, or pay to have other users evaluate their content.

[6] A 12-item poll will have $\binom{12}{2}$ or 66 combinations.

[7] `https://www.mturk.com/mturk/welcome`

4 Implementation

Rankr is implemented as a mobile web service using the jQueryMobile[8] framework, an extension of the jQuery framework. The jQueryMobile framework is designed to create a unified user interface across popular mobile browsers. The decision to develop Rankr as a mobile web service versus a mobile application was intentional. We wanted Rankr to reach as many mobile users as possible while minimizing the development time for each mobile platform. Additionally, because Rankr only requires minimal user input for pairwise comparisons to be made (i.e., screen taps from the user), the benefits of developing Rankr as a mobile application, such as access to the mobile devices' hardware instruments (e.g., accelerometer) are not needed.

The server handles authentication and, for general purposes, serves a single page containing "mini mobile pages" to the mobile device with accompanying CSS and JavaScript files. Aside from the first page load, further communication between the mobile device and the server is done with AJAX requests using JSON.

Rankr has been verified to look and work the same on Android 2.0+, BlackBerry 6.0+, iOS 3.0+, and WebOs 1.4.1+ which confirms jQueryMobile's success in creating a unified user interface independent of mobile device.

The Rankr server runs on a combination of Ubuntu, Apache, Python, and Django. These specific software packages were chosen out of convenience and familiarity.

5 Future Work

Having completed the initial stages of development, we now intend to open Rankr to a larger user base to stress test the system. This would also involve running additional usability studies with a larger sample size.

In terms of specific improvements to the interface, we plan to integrate a more intelligent method to display images to maximize the real estate of the small screen (Figure 6, 7). Moreover, we are considering to incorporate sound and video in order to have multimedia polls where users could potentially compare a text item versus an audio item.

In the Rankr prototype, we insured that both images were loaded before presenting the images to the user in order to eliminate biases that could arise if one image is loaded before the other. Research into measuring other potential user biases because of how items are presented is needed. For instance, a user may be more or less likely to select an image because of its size, or users may behave differently when images are top-aligned (as in the current implementation) versus middle-aligned.

Focusing on the mechanism of pairwise comparisons, we could potentially use Rankr as a tool for profiling. As an example, consider a firm that wants to cluster consumers with respect to their preferences. The firm can ask consumers to do a few pairwise comparisons between specific products or more general categories. Using the results, the firm can cluster consumers into different types, and suggest different products to consumers of different types.

[8] `http://jquerymobile.com`

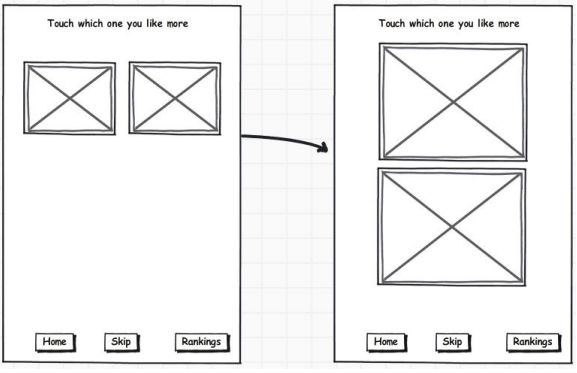


Fig. 6. In portrait mode, items will be stacked vertically

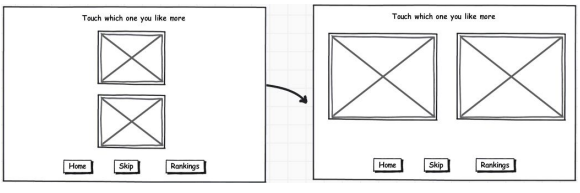


Fig. 7. In landscape mode, items will be stacked horizontally

6 Conclusions

In this paper, we have presented Rankr, a mobile service that ranks items through pairwise comparisons. Our service has a number of advantages. First, it is accessible from any smartphone. Second, it has an intuitive interface. Third, it scales well irrespectively of how many comparisons each individual user performs, as long as all users in aggregate complete a sufficiently large number of comparisons. These properties make Rankr very effective for crowdsourcing the ranking of collections of items.

Acknowledgements. We would like to thank the following people for their contributions: Mike Brzozowski and Alex Vorbau whose early development work helped make Rankr a reality; Hang Ung for his efforts in the algorithm and implementation; and Thomas Sandholm for his thoughtful insights.

References

1. Arrow, K.J.: Social Choice and Individual Value. Wiley (1963)
2. David, H.A.: The Method of Paired Comparisons. Oxford University Press, New York (1988)
3. Elo, A.E.: The rating of chess players: Past and present. Arco Publishing, New York (1978)
4. Green, P.E., Srinivasan, V.: Conjoint analysis in marketing: New developments with implications for research and practice. The Journal of Marketing 54(4), 3–19 (1990)

5. Hacker, S., Ahn, L.V.: Matchin: Eliciting user preferences with an online game. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. ACM (2009)
6. Hochbaum, D.S.: The separation, and separation-deviation methodology for group decision making and aggregate ranking. In: Hasenbein, J.J. (ed.) TutORials in Operations Research, INFORMS, Hanover, MD, vol. 7, pp. 116–141 (2010)
7. Mas-Collel, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press (1995)
8. Sarma, A.D., Sarma, A.D., Gollapudi, S., Panigrahy, R.: Ranking mechanisms in twitter-like forums. In: Proc. Conf. on Web Search and Data Mining (2010)