Evaluating Preference Collection Methods for Interactive Ranking Analytics

Caitlin Kuhlman, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Paul-Henry Schoenhagen, MaryAnn VanValkenburg, Elke Rundensteiner, Lane Harrison
cakuhlman|ddoherty2|mnurbekova|godeva|zphyo|pmschoenhagen|mevanvalkenburg|rundenst|ltharrison@wpi.edu
Worcester Polytechnic Institute

ABSTRACT

Rankings distill a large number of factors into simple comparative models to facilitate complex decision making. Yet key questions remain in the design of mixed-initiative systems for ranking, in particular how best to collect users' preferences to produce high-quality rankings that users trust and employ in the real world. To address this challenge we evaluate the relative merits of three preference collection methods for ranking in a crowdsourced study. We find that with a categorical binning technique, users interact with a large amount of data quickly, organizing information using broad strokes. Alternative interaction modes using pairwise comparisons or sub-lists result in smaller, targeted input from users. We consider how well each interaction mode addresses design goals for interactive ranking systems. Our study indicates that the categorical approach provides the best value-added benefit to users, requiring minimal effort to create sufficient training data for the underlying ranking algorithm.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; • **Information systems** → *Content ranking*;

KEYWORDS

Preference Elicitation, Interactive Ranking, User Study

ACM Reference Format:
Caitlin Kuhlman, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo,, Paul-Henry Schoenhagen, MaryAnn VanValkenburg, Elke Rundensteiner, Lane Harrison. 2019. Evaluating Preference Collection Methods for Interactive Ranking Analytics. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3290605.3300742

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300742

1 INTRODUCTION

Ranking is a commonly employed tool which simplifies decision making when the number of factors impacting choice is large. A ranking of objects distills high dimensional information into a simple ordered list, helping people to quickly grasp the relative merit of objects or choices. People rely on rankings published by companies, consumer groups, and government agencies for guidance across many domains - from evaluating the quality of institutions like colleges and hospitals [25, 48], assessing potential employees [47], to evaluating regional economics [13, 40]. These rankings are typically consumed "off the shelf", lacking personalization to reflect the priorities of individual users for decision making.

At the same time, personalized ranking is commonly performed as a subtask for web-based search and recommendation engines to provide top search results. Recently, interactive systems have been proposed [30, 31, 43] which support personalized ranking to empower users to leverage learning-to-rank algorithms [33] for their own decision making. Rather than relying on the preferences of similar users as is common in recommender systems, these systems allow a single user to fully control the ranking process by explicitly specifying their personal preferences. For instance, with a personalized college ranker, a student may specify her preferences over colleges she has visited so far based on her own goals and interests. The system then automatically generates a global ranking over a larger set of universities that the student is not able to visit in person. This process may provide insight by revealing the data attributes used by the system to create the ranking – thus allowing the student to better understand their own priorities.

Such mixed initiative systems [24] rely on user-machine collaboration to facilitate sense-making that would otherwise not be possible. Machine automation provides computational power, while humans provide domain expertise, understanding of the task at hand, and personalized preferences. A key challenge in the design of ranking systems thus lies in

the elicitation of this information from users. While preference elicitation techniques are well studied [2, 10, 22, 28, 29], their use in interactive ranking systems has not been formally evaluated. The impact of different preference collection mechanisms on user behavior and level of satisfaction with the ranking system is thus not well understood.

Further, the specification of enough information so that the learning engine can reliably infer a useful ranking represents an arduous task for humans. Yet the availability of a large enough training dataset over which to learn a ranking is imperative in determining a meaningful ranking [46]. As discussed by Crouser *et al.* [11], for the design of appropriate interactive systems we should evaluate and quantify both the computational complexity of the processes used as well as the complexity of the human effort required.

Our Approach. In this work, we consider the impact of preference collection methods for interactive ranking systems. Our aim is to understand the impact on both user behavior and system performance. To evaluate this, we consider three primary modes of interaction which allow users to directly express relative preferences among items: sublist ranking, categorical binning, and pairwise comparisons. We evaluate the effect of these three modes on interactive ranking using a between-participants crowdsourced user study on the Mechanical Turk platform with $n = 144$ participants. The study results reveal that categorical binning leads to significantly more user interactions, without increasing the time spent. As a result, this mode provides the largest amount of training data to the ranking engine, providing the best tradeoff between user effort and model quality. Our discussion of these results covers implications of these findings for the design of mixed-initiative ranking systems.

The contributions of this work include:

(1) We conduct a large scale (144 participants) crowdsourced user study on the effect of preference collection methods on interactive ranking. We evaluate the complexity of human effort in interaction, rate of information extracted from interactions, and the impact on user satisfaction.
(2) We design three alternative interfaces that embody three distinct modes of preference specification embedded into an interactive ranking tool to provide the study participants with an end-to-end experience.
(3) Our study finds that a categorical binning approach provides the best value-added benefit, requiring minimal effort while producing more training data. This in turn positively impacts the quality of the ranking created by the underlying machine learning algorithm.
(4) Our findings raise interesting questions that point at future investigation into the composition of interaction modes and alternative means for increased user engagement in ranking systems.

2 RELATED WORK

Multi-Attribute Ranking Systems

In recent years, several multi-attribute ranking systems have been developed to help users interact with rankings [8, 19, 30, 31, 38, 41, 43]. Much focus has been on aiding users in manually *adjusting the data attribute weights* of a multi-criteria ranking and on visualizing the resulting impact across attribute subsets [38], alternative rankings of the same items [19], and rankings over time [41].

Lately, work has begun exploring the incorporation of user *preferences over objects* into mixed-initiative systems [24] to better capture users' intuitive sense of the relative value of the objects to be ranked. A learning-to-rank algorithm [33] is then employed to infer a global ranking over the rest of the objects in the dataset based on these interactions. As detailed in Section 3, the RankSVM algorithm [27] features properties that naturally fit the interactive ranking task.

To date, the mechanisms by which users specify their preferences in these interactive ranking systems have not been formally evaluated. The Podium [43] system applies a semantic interaction approach to preference collection [16]. Preferences are inferred from users' interactions re-ordering items in a list, rather than being directly specified by the user. This lead to cases where the user would place an item at a certain position, but that position would not be maintained in the final ranking. One user reported feeling that they were "arguing with the model" since their intentions were not well-captured. Other systems allow users to directly judge the relative merits of pairs of items [30] or organize a subset of items from the dataset according to their preference [31].

Preference Elicitation in Recommender Systems

For insight into the problem of collecting user preferences, we can look to the wealth of research around HCI for recommender systems [7, 22, 28, 29], which rely on ranking according to user preferences as a subtask. However, key differences between recommendation and ranking systems exist. Recommender systems aim to automatically find interesting items in a dataset, while interactive ranking systems help users gain a global understanding of a dataset. In the latter process, users may come to understand their own intuition and preferences better, similar to how systems for personal informatics encourage self-reflection [32].

For recommendation, user preferences are often collected *implicitly*, based on interactions such as search queries or click-through logs. Preferences of *multiple users* are typically aggregated using collaborative filtering [2]. In our setting, a subset of data is manipulated by a *single user* to deliberately train a ranking model. The model is then applied to help the user create a global ordering *over the same dataset* in a semi-supervised manner [42].

For recommendation systems, it has been observed that user satisfaction is positively impacted by a sense of control over the recommendation process [7, 29]. As discussed in a survey by He et al. [22], explicit interaction and visualization have been incorporated to improve qualitative aspects of the recommendation process. User preferences may be used to match similar users or address "cold start" problems. The most prevalent way recommender systems collect this information is to have users rate items (such as giving 1 to 5 stars). However, studies have demonstrated that user ratings can be inconsistent or inaccurate [3], and that humans are more cognitively adept at making *relative judgments* [9, 10]. Some interactive recommender systems accomplish this by allowing users to group together items they consider similar [22]. One recommender system [35] evaluated the impact of collecting pairwise preferences between subsets of items, finding that it improved user satisfaction.

3 METHOD OF THE STUDY

In this work we compare the impact of alternative preference collection interfaces which allow users to *explicitly* specify their preferences using *relative judgments*. The chosen interaction modes cover a broad spectrum of core methods employed in interactive ranking and recommendation systems to date, following from our literature review above. Chosen methods to collect preferences over items in the dataset are *sub-list ranking*, *categorical binning* and *pairwise comparisons*, Pair comparison has been popular for preference elicitation [9, 10, 35]. A list is included given its use in a recent visual analytics system, Podium [43]. Finally, to allow users to group similar items as in previous ranking and recommendation systems [22, 31], we develop an interface where users group items into categories: high, medium, or low. Our study used a **between-participants design** in which each participant was randomly assigned to one of three preference collection modes.

Research questions investigated in the study are (1) do users behave differently depending on the interaction mode, (2) does the mode of interaction impact user satisfaction, and (3) what kind of trade-off does each mode offer between user effort versus the training requirements of the underlying ranking engine. Drawing on analytic approaches from several recent studies that examine user behavior [5, 14, 17, 21, 44], and analysis of the computational complexity of system processes [11], we frame our research questions as follows:

- **interaction behavior:** does the collection mode impact measures of behavior such as total time spent entering preferences or the number of items added?
- **self-reported user experience:** does the collection mode affect the perceived ease of use of the ranking tool? Does it impact their anticipated adoption of the tool for ranking tasks?
- **system performance:** does the collection mode affect the size of the training data generated from the given user preferences?

Design of Alternate Preference Collection Interfaces

We implement each interaction mode as part of a mixed-initiative ranking system to compare the effect on user interactions of each mode as a study condition. The resulting interfaces are depicted in Figure 1. A *pairwise* learning-to-rank algorithm powers the ranking engine [27]. For this, pairs of items are extracted from the user interactions and used to train the ranking model.

Sub-list Ranking. The sub-list preference collection mode (Figure 1a) closely matches a typical ranking activity. Preferences are specified by sorting a subset of items into a completely ordered list. Items at the top of the list are preferred to those placed below. At a minimum, two items must be placed in the list so as to form one pair. The user can add any number of additional items up to specifying a complete ordering over all items.

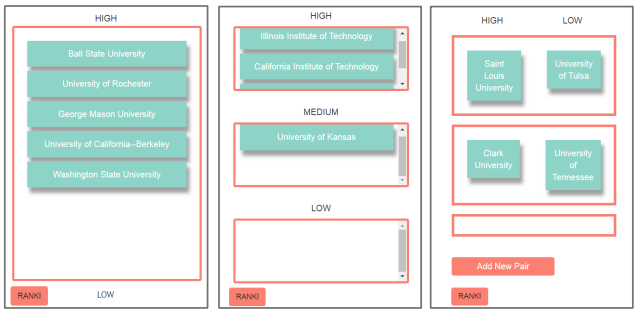
Categorical Binning. The second preference collection method (Figure 1b) uses a categorical approach. Users express their preference by binning a subset of items into three categories: high, medium, low. Items within each category are not compared. However, items in the high category are preferred to all items in both the medium and low categories, and items in the medium category are preferred to those in the low category. At a minimum, two items must be placed in separate categories in order to derive a ranking. The user may choose to organize objects in any two out of three categories, or use them all, with any number of objects in each.

Pairwise Comparison. The last ranking method we consider is the pairwise preference collection mode (Figure 1c). Here, users directly express their preferences as binary relations between pairs of items. Users place two items in an ordered pair, with the one on the left preferred to the one on the right. Unlike the other comparison modes, in the pairwise interface the same object can be entered multiple times if it is preferred to multiple other items.

College Ranker Interactive Ranking Scenario

These alternative preference collection modes are incorporated into an interactive College Ranking system. The US News and World Report Best Colleges dataset[1] is used. The dataset contains both numeric and categorical attributes of colleges in the United States. The system is composed of two views, a "Build" view where users enter their preferences, and an "Explore" view where they access the global

[1] https://www.usnews.com/best-colleges/rankings/national-universities



(a) Sub-list ranking mode. **(b) Categorical binning mode.** **(c) Pairwise comparison mode.**

Figure 1: Alternative preference elicitation interfaces.

ranking generated based on their input. Users can iterate between these views to continually refine their ranking. Each component of the system is described in detail below.

Build View. The Build view has two main components - the data is displayed on the left side of the screen and the preference collection interface on the right. To avoid biasing the user toward any pre-ranked numeric or lexicographic ordering [39], we display the colleges from our target dataset in a "data pool" where they are represented only by name arranged in a grid format. As a user may want to further learn about each college and its properties, we provide the attribute values for each item in a tooltip accessible on hover. On page load, the college dataset is randomly shuffled and displayed in the data pool. Multiple navigation modes are offered: users can search for a college by name, sort the data alphabetically, or use the"Shuffle" button to randomly permute the data. All three build modes employ the same basic interaction; to enter their preferences, the user drags colleges from the data pool into the preference collection interface (Figure 1). Users can move as many objects as they want, swapping their order and moving them between the pool and the different fields within the Build view.

Explore: Global Ranking Interface. Once the user hits the "Rank!" button, they are redirected to the Explore view. A thinking step displays a spinner and message "we are computing your global ranking ... " to communicate the conceptual division between the data the users have manipulated to train the underlying model, and the learned global ranking. The Explore view visualizes the learned college ranking in a tabular display. To easily identify and evaluate their input from the previous view, colleges that the user manipulated are highlighted with bold text.

Ranking Engine. Internally, a mixed-initiative system for ranking leverages a machine learning algorithm to generate a global ranking of the dataset from the partial input collected from the user. Existing systems [30, 31, 43] adopt a *pairwise* formulation of the learning-to-rank problem [33] which allows a ranking to be learned by applying a binary classifier to ordered pairs of data instances. Building on this result, many classification models have been employed for learning-to-rank [6, 18, 27, 37]. Here we briefly review the Rank SVM algorithm [27] employed in our college ranker system. We aim to provide the reader with the intuition behind these algorithms as foundation for our subsequent analysis on the effect of preference collection mode on training dataset size and system performance (See Section 4). In particular, we describe the type of training data required and the amount of information needed to infer a meaningful resulting ranking.

The RankSVM algorithm [27] uses a Support Vector Machine (SVM) algorithm to distinguish between correctly ordered and incorrect pairs of data instances. This rests on the assumption of a linear function $U(x) = \vec{w}^T x$ where $\vec{w}$ is a d-dimensional weight vector mapping each object in the dataset x_i to a value corresponding to its rank. Then the following holds true:

$$\vec{w}^T x_i > \vec{w}^T x_j \implies \vec{w}^T (x_i - x_j) > 0$$

Therefore, instead of learning the ranking from the individual data instances x_i in the training dataset, the function can be learned over the combined feature vector $(x_i - x_j)$ of **each ordered pair of objects**. Each training pair is assigned a binary class label $c \in \{-1, 1\}$, where a label of 1 indicates a correctly ordered pair, and -1 indicates an inverted pair. The weight vector $\vec{w}$ is a hyperplane decision boundary which distinguishes between these two classes while maximizing the space between them. Once the boundary has been learned from the training data, a global ranking over all unseen data can be extracted. For each object x_i, $\vec{w}x_i$ gives a score $\hat{y}_i$ which determines its rank position.

In a typical *supervised learning* problem formulation, the true ranking over all n training data points is known. For interactive ranking, the problem is *semi-supervised* [42], in that labels are given only for a subset of $m < n$ data points. A key consideration in the performance of the ranking model in a semi-supervised setting is sample complexity analysis to evaluate the number of training pairs required to effectively learn a model. Wauthier et al. [46] consider the sample complexity of the RankSVM algorithm. They observe that if pairs are selected at random and labeled, then the RankSVM algorithm performs optimally and requires $O(n)$ pairs to produce a better than random expected result.

This complexity analysis is crucial to understanding whether interactive ranking can be effective given a small number of examples from a user. Clearly, for dataset of $n = 100$ items, having to manually specify preferences over 100 pairs puts a non-trivial burden on the user. To reduce this, elicitation techniques should generate as much information as possible

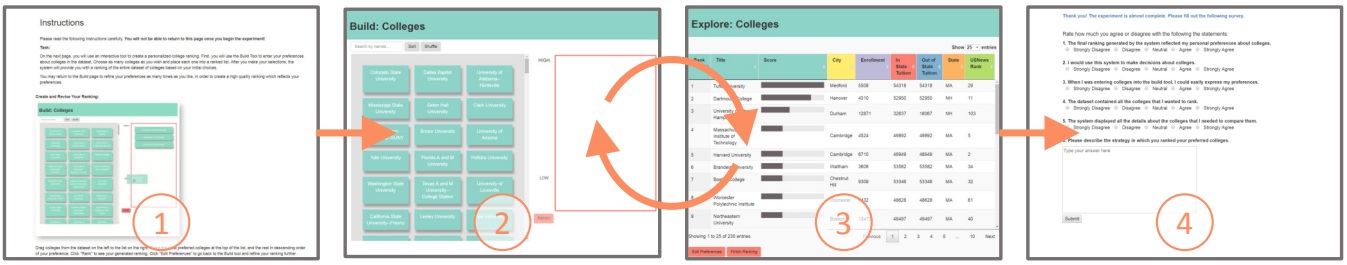


Figure 2: Experiment phases: (1) training, (2) rank building, (3) rank exploration, (4) post-test survey. Participants can iterate between build and explore unlimited times.

for the least amount of user effort. The mode of preference collection employed impacts the number of pairs that can be extracted from the user input. We examine this effect in depth in Section 4.

Procedure and Tasks

Pilot studies were used to refine experimental instructions and pacing, and to estimate the required sample size for the final study. The final study was conducted on Amazon's Mechanical Turk (AMT) crowdwork platform, which has been validated as a platform for studies in human-computer interaction [36] and visualization [23]. In particular, Mason and Watts find that financial incentives on AMT influence the quantity, but not the quality of work [36]. Following Hara *et al.* [20], our workers were paid $1.25 based on the average completion time of 5-8 minutes in our pilot studies. Payments were structured as $1.00 base rate with a bonus of $0.25 offered for creating a satisfactory ranking, to incentivize engagement with the tool. Unbeknownst to the workers, they all were paid the bonus. Average hourly wage for the full study was $10.42, exceeding US federal minimum wage of $7.25. Each participant was randomly assigned to one of the three interaction modes and rewards were consistent throughout the three methods. All participants viewed an IRB-approved consent form.

Our procedure consisted of four phases: *Training, Rank Building, Rank Exploration*, and *Post-test Survey*. Views of each phase are shown in Figure 2.

Training: We provided participants with an instruction page that described their task and the interaction mechanisms in the ranking tool. For example, in the sub-list collection mode, the instructions stated:

> *You will use an interactive tool to create a personalized college ranking. First, you use the Build Tool to enter your preferences about colleges in the dataset. Choose as many colleges as you wish and place each into a ranked list. After you make your selections, the system will provide you with a ranking of the entire dataset of colleges based on your initial choices.*

Each specific interaction mode was described, with animated gifs illustrating the preference collection process. For the Sub-list mode users were instructed:

> *Drag colleges from the dataset on the left to the list on the right. Place the most preferred colleges at the top of the list, and the rest in descending order of your preference.*

Rank Building: After viewing the instructions, participants proceeded to the College Ranker tool. The Build view contained the randomly assigned preference collection interface. Participants then interacted with the colleges in the dataset, entering as many preferences as they desired without any time limit. When participants were satisfied with their preferences, they could click a "Rank" button to advance to the Rank Exploration phase.

Rank Exploration: The Explore view displayed the generated ranking over the entire dataset in a tabular format. On this page participants could explore the ranking by scrolling or paging through results, examining the order of items and the scores assigned by the ranking engine. From here, clicking the "Edit Preferences" button navigated back to the previous Build view. Participants could amend or refine their preferences, iterating between the Build and Explore views as many times as they wished. To complete the ranking task, users participants clicked "Finish Ranking". A modal window prompted them "Would you like to revise your ranking by returning to the Build page?" to ensure users were aware of the option to return to Build. Users could then click "Yes - Return to edit preferences" or "No - This is my final ranking", which advanced them to the final phase of the study.

Post-test Survey: Participants were provided with a short set of statements and asked to indicate their agreement to provide qualitative feedback.

Measures

We collect quantitative measures by logging user interactions during the *rank building* and *rank exploration* phases of the

study. We also record the time spent in each phase of the study. Interactions logged include:

- **Additions:** The number of items participants entered into the preference collection interface by dragging them from the data pool.
- **Removals:** The number of items participants removed from the preference collection interface and returned to the data pool.
- **Selections:** The set of items entered into the preference collection interface.
- **Ranks:** The number of times the participant clicked the "Rank!" button to advance from the Build view to the Explore view.
- **Refines:** The number of times the participant clicked the "Edit Preferences" button to return to the Build view from the Explore view.

To evaluate the system performance, we consider the size of the training dataset provided to the ranking engine using each interface. As detailed in Section 3 the training data consists of pairs of data objects generated from user preferences. We measure the training data size in two ways:

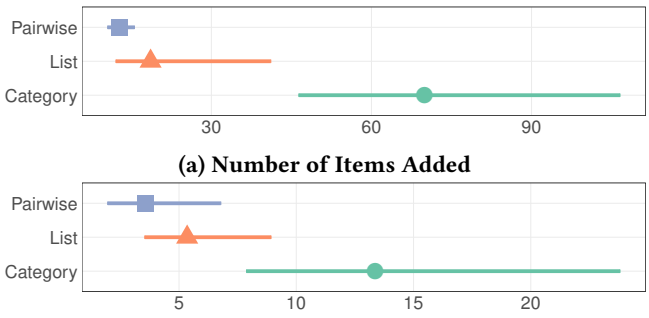
- **Pair growth rate:** The number of pairs p generated from m items entered by the user. [2]
- **Actual pairs:** An empirical count of the number of training data pairs generated in practice.

In addition, self-reported quantitative measures were collected using the post-test survey. Finally, using free-response questions, we also collect participant comments on their ranking strategy and experience using the College Ranker.

Pilots, Analyses, and Experiment Planning

To ensure our experiments included enough participants to reliably detect meaningful differences between the conditions, we conducted effect size and statistical power analyses. Specifically, we estimated the variance in our quantitative measures based on results from two pilot studies, and combined these with the observed means to approximate how many participants were needed. In response to concerns about the limitations of null hypothesis significance testing [12, 45], we model our analyses on HCI research that seeks to move beyond these limitations (e.g. Dragicevic [15]). Following Cumming's [12] recommendations regarding group comparison using confidence intervals (CIs), we compute 95% CIs using the bootstrap method, and use Cohen's d to measure effect sizes (the difference in means of the conditions divided by the pooled standard deviation).

[2]This is a distinct measure from the total number of interactions performed by the user since they may add, remove, and swap many items during the Build phase before ranking.



(a) Number of Items Added

(b) Number of Items Removed.

Figure 3: Comparing the number of user interactions across preference collection modes.

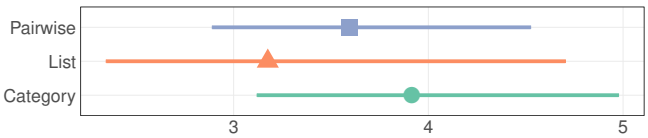
4 RESULTS

144 participants were recruited for the study. Out of the total, 49 participants were randomly assigned to the sub-list preference collection mode, 45 to categorical binning , and 50 to the pairwise mode. For each measure we compute quantitative results comparing each study condition. Error bars are 95% confidence intervals (CIs).

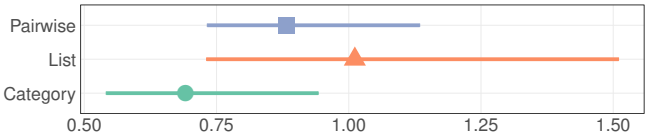
Elicitation Techniques and Observed User Behavior

Effect on Number of Interactions. We found that the average participant assigned to the categorical binning mode interacted with significantly more items from the dataset (M = 69.9 items added with 95% CI [46.4, 106.6]) than those participants assigned to the sub-list or pairwise modes (M = 18.6 items added with 95% CI [12.1, 41.1], and M = 12.7 items added with 95% CI [10.5, 15.6] respectively) as seen in Figure 3. Following Cumming [12], we can interpret the upper and lower limits of the confidence intervals as meaning that the average participant in the categorical binning group added at least 5 items and up to 95 additional items during the build phase compared to the other two conditions. The effect size as measured by Cohen's d is large between categorical binning and pairwise modes $d = 0.79$ [0.57, 1.06] and between categorical binning and sub-list modes $d = 0.66$ [0.27, 0.96]. While there is a small effect observed between sub-list and pairwise with $d = 0.22$ [−0.24, 0.5], the negative lower-bound suggests that differences in these two groups should be considered inconclusive.

We also count the number of items removed from the preference collection interface during the build phase. While there are fewer remove interactions on average, we observe a similar effect across modes. The average user assigned to categorical binning removed more items (M = 13.4 items removed with 95% CI [7.9, 23.8]) than in sub-list mode (M = 5.3 items removed with 95% CI [3.5, 8.9] or pairwise (M = 3.6 items removed with 95% CI [1.9, 6.7]). The effect sizes

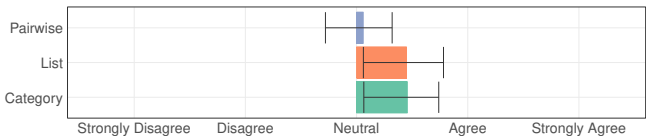


(a) Time spent building in minutes.

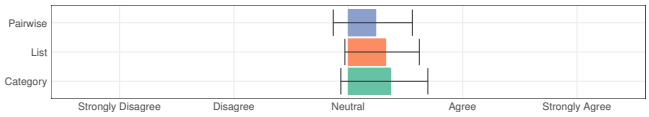


(b) Time spent exploring in minutes.

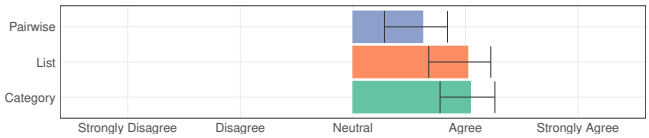
Figure 4: Comparing the time spent interacting with the preference collection interface during the build phase.



(a) The final ranking generated by the system reflected my personal preferences about colleges.



(b) I would use this system to make decisions about colleges.



(c) When I was entering colleges into the build tool, I could easily express my preferences.

Figure 5: Qualitative assessment. Users were asked to indicate their level of agreement with each statement.

are $d = 0.51$ $[0.23, 0.73]$ between categorical and pairwise, $d = 0.42$ $[0.11, 0.71]$ between categorical and sub-list, and $d = 0.28$ $[-0.37, 0.78]$ between sub-list and pairwise.

Effect on Time Spent Interacting. Despite the significant difference in the number of items added, we do not see a corresponding difference in the amount of time spent entering preferences in the rank building phase (re 4a). Sub-list time building (M = 3.2 minutes 95% CI [2.3, 4.7]), categorical binning time building (M = 3.9 minutes 95% CI [3.0, 5.2]), and pairwise time building (M = 3.6 minutes 95% CI [2.8, 4.6]) do not exhibit any significant effect from the preference collection mode used. Rank exploration time is not significantly impacted by preference elicitation technique either (Figure 4b). Results show similar sub-list time exploring (M = 1.0 minutes 95% CI [0.7, 1.4]), categorical binning time exploring(M = 0.7 minutes 95% CI [0.5, 0.9]), and pairwise time exploring (M = 0.9 minutes 95% CI [0.7, 1.1]).

Effect on User Satisfaction. We include examples of the qualitative statements presented to users in the Post-study survey (Figure 5). No significant differences between preference collection modes were observed. While participants didn't report a difference in their experience of the three modes, the difference in number of interactions tells a different story.

Elicitation Techniques and ML Implications

Effect on Pair Growth Rate. As discussed in Section 3, to learn a global ranking over the dataset the ranking engine is trained over data pairs. To evaluate the effect of alternative preference collection methods on system performance, we consider the number of pairs that can be extracted using each elicitation technique. We derive the *pair growth rate* for each mode (shown in Figure 6a) which captures the number of pairs n that is generated given m data items entered into the preference collection interface. Here we consider the number of items collected when the user clicks "Rank!" to generate the global ranking. Since objects are arranged differently in each preference collection mode, there is a different pair growth rate associated with each.

Sub-list Ranking Pair Growth Rate: The sub-List interface specifies an explicit order over the set of items. Its pair growth rate is thus $\binom{m}{2} = m(m-1)/2$, reflecting all possible ways of choosing ordered pairs from the list. This is a quadratic growth rate, meaning if m items have been added into the list, then the number of pairs implicitly specified is on the order of $O(m^2)$.

Categorical Binning Pair Growth Rate: The categorical binning mode is more difficult to quantify, since a different number of objects can be added to each of the 3 bins. In the worst case, $m-1$ items will be placed in one bin, and only one item placed in a second bin. In this case, only $m-1$ pairs would be formed ($O(m)$ linear growth rate). However, assuming an equal distribution of $m/3$ objects in each bin, many more pairs would be formed. In this best case, the growth rate is $\frac{m^2}{3}$ possible pairs. While this method is also quadratic in the best case, it has a slower growth rate than the sub-list mode. The max and min rates are both shown in Figure 6a, with the shaded region covering the possible range of pairs resulting from an uneven distribution of items across bins for the categorical mode.

Pairwise Comparison Pair Growth Rate: This mode is most directly aligned with the pairwise formulation of the underlying ranking algorithm. Here the user specifies each pair explicitly, meaning this is the most labor-intensive of the three modes. Since two items are required to form every pair, the growth rate is linear, namely, $m/2$, and even slower than the minimum rate of the categorical mode.

Figure 6b shows the actual number of pairs generated by users laid over the pair growth rates, this time in log scale for readability. We can see that while the number of pairs generated by the sub-list and pairwise modes are fixed dependent on m, for the categorical mode values can fall anywhere in the shaded region. On the whole we can see that users tend to distribute data evenly among the bins in practice, yielding pair numbers of close to the maximum categorical rate.

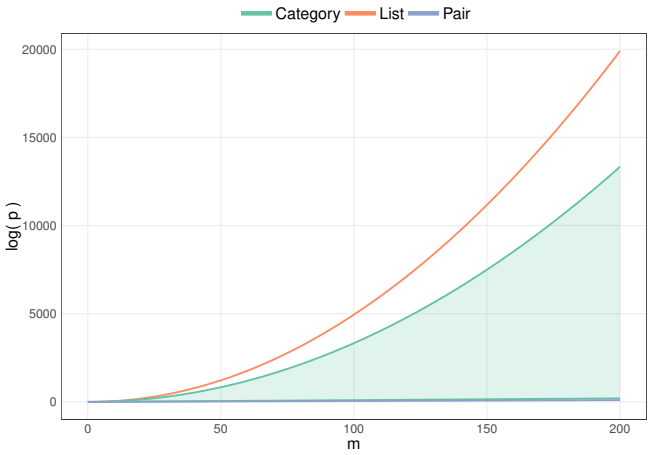
Effect on Training Data Size. We found that the average participant who was assigned to the categorical binning mode generated significantly more training data pairs (M = 2185 pairs 95% CI [1170,3805.5]) than those assigned to the pairwise mode (M=4.5 pairs 95% CI [3.9,5.2]), as indicated by Cohen's d with effect size d=0.63 [0.45,0.86]. The sub-list mode also resulted in fewer pairs on average (M = M=401.4 pairs 95% CI [55.1,1875.2]). These results are shown in Figure 7. The effect size as measured by Cohen's d between categorical binning and sub-list modes d=0.44 [0.01,0.69], and between sub-list and pairwise modes d=0.2 [0.16,0.26].

5 DISCUSSION AND FUTURE WORK

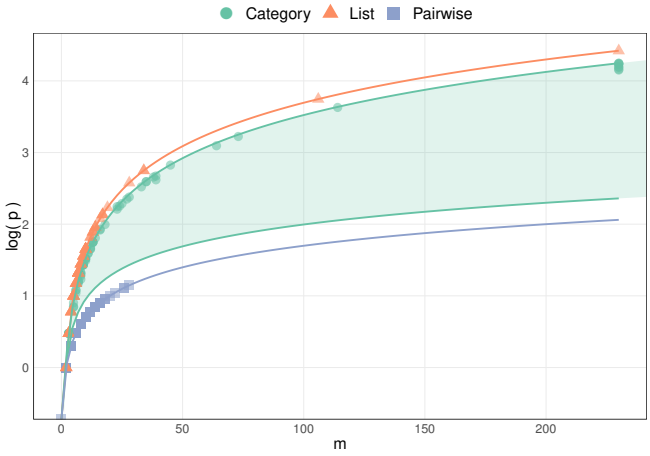
The results of our study suggest that the mode of preference collection employed can significantly influence the number of interactions performed by users (Figure 3) as well as the amount of training data provided to the ranking engine (Figure 7), without impacting the amount of time spent (Figure 4) nor the perceived ease of use of the ranking tool (Figure 5). One general implication of these results is that the categorical binning mode provides the best tradeoff between user effort and training dataset size. We next consider other possible explanations for these findings followed by implications for the design of interactive ranking systems.

Categorical Binning: High User Engagement and Expressiveness?

We targeted preference collection techniques which allow the user to make relative judgements, based on previous studies [9, 10]. Our results demonstrate appreciable differences in user behavior when specifying the relative value of items in different ways. The cognitive process of binning items in groups appears to impose less of a burden on the user, as opposed to the fine-grained distinctions required by the pair and sub-list modes. The categorical binning mode allows users to organize information using broad strokes, interacting with a large amount of data quickly, and providing the most training data to the ranking engine on average. It appears to facilitate sense-making over an entire decision space, as many participants in our study categorized the majority of colleges in the dataset. This result has the potential to provide utility for many tasks in addition to ranking, and



(a) Growth rate curves. Includes min and max rates for Categorical Binning, with the shaded region covering possible rates due to an unequal distribution of items in each bin.



(b) Actual number of pairs generated shown over growth rate curves, in log scale.

Figure 6: Pair growth rates comparing m the number of items in the preference collection interface against p the number of pairs extracted.

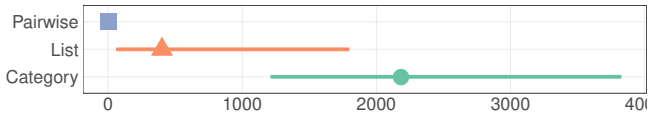


Figure 7: Number of pairs generated from user preferences.

aligns with the findings of other recent studies. Loepp *et al.* [35] observed that collecting user preferences between small subsets of items improved the user experience of interactive recommendation, compared to a more labor-intensive manual search for preferred items. Using categories was also shown by Hu *et al.* [26] to facilitate understanding of

recommendation results, for instance helping users identify qualities such as diversity.

Following from Horvitz's work on mixed-initiative systems [24], an interactive ranking system must be able to learn models that contribute meaningful insights to the ranking problem. In this setting, there is no absolute "ground truth" ranking as the target. Thus we cannot expect users to input objects labeled with an exact position in the final ranking, as might be provided in a traditional training dataset. This would require them to know not only how many items in total are being ranked, but also the precise position of one particular item in exact relation to all others. Previous work by Amatriain *et al.* [3] showed that such absolute evaluations elicited as numeric scores for individual items were an unreliable indicator of users' true preferences. A benefit of categorical binning is that it can capture *ambiguity* on the part of the user. For example, items placed in the top category may be perceived by the user to be preferred to other items, however, they are not forced to impose a strict order among them. The ease of understanding and expressing high level differences between items allowed by the categorical binning approach seems to facilitate the data exploration task well, allowing users to provide large amounts of training information to the system with ease.

Future work might more closely investigate the possible variations in user behavior and intent *within* binning modes, as it is possible that a person would want to organize both between categories, as in the college ranker tool, as well as within categories. This line of research may be shaped by significant existing research targeting cognition and user interface (UI) elements in human-computer interaction. For example, Aula *et al.*, [4] examined search behavior using cognitive proxies such as experience to further disambiguate observed differences in behavior with search interfaces. Connecting cognitive principles to preference elicitation tasks and UI elements may be a promising means for developing and evaluating interaction modes that better enable people to express their intent in human-computer collaborative contexts.

Supporting Diverse Ranking Strategies

Interaction modes using pairwise or sub-list preferences resulted in smaller input from users. While the sub-list mode has the fastest pair growth rate, in practice many fewer training pairs were generated using this mode. Lists have a high potential, but rarely do people use them to their full capacity. The pair preference format received the lowest rating from study participants. This aligns with the fact that the slow pair growth rate means that much more user effort is required to enter enough data to learn a useful ranking. Future research could draw on work in human-computer interaction targeting search elicitation strategies, such as work from Agapie *et al.* which explored UI components that led people to longer search queries [1]. Merging "nudging" threads of research with rank preference elicitation may yield additional evidence-driven design guidelines that better optimize the relationship between the user and the underlying ranking algorithms explored in systems today [19, 30, 31, 43]. It remains unclear what control schemes best align with individuals' ranking strategies. New studies could be designed that evaluate the consistency and diversity of ranking strategies, perhaps by varying not only the interface, but also the framing and underlying datasets under consideration.

Towards Compositions of User Elicitation Techniques

Although the study results indicate that users add significantly more data with categorical binning, it should not be taken to mean that categorical techniques are strictly superior to other elicitation modes. We posit that compositional approaches to user preference elicitation may be a path towards mitigating the drawbacks of each mode while maximizing the amount and quality of information the user provides to the system. For example, given that the sub-list mode has the fastest growth rate, and the fact that some (outlier) participants were observed to use the list technique to its full potential, future interfaces may possibly combine the benefits of both preference collection modes. Exploring this would require the creation and then evaluation of novel elicitation interaction techniques. Recent work from Wall *et al.* on the Podium system can be taken as a point in this design space [43], given their system allows users to directly manipulate a ranking result list. A recent approach of "blended recommendation" [34] which explores the use of manual interactions such as data attribute filtering and weight adjustment combined with automated recommendation could also inform design for interactive ranking.

Rankings are often used to capture an ill-defined or complex quality, which is difficult to measure directly, and in the end hinges on subjective evaluation by the user. As this design space grows, it will likely become necessary to further develop trustworthy and transparent experimental methodologies while also exploring how ranking algorithms can be modified to better include human-in-the-loop input.

6 CONCLUSION

Ranking is a powerful tool often employed by decision makers to understand complex data. Automated tools designed to facilitate the ranking process can provide insight based on users' domain knowledge and intuition about the objects being ranked. However, the impact of preference elicitation methods on both the user experience of interactive ranking systems as well as the quality of the learned rankings has not been well understood. In this work we evaluated three preference collection mechanisms designed to allow

users to directly specify relative judgments about items in a dataset in a crowdsourced study. Our results indicate that a categorical binning preference collection mode provides the best trade-off between user effort and training dataset size. The results of this study have practical implications for the design of interactive ranking systems, in how best to engage users and derive sufficient information from which to generate meaningful rankings.

ACKNOWLEDGMENTS

The authors thank the Computing Resources Association for Women for support through the CREU program. This work was also partially funded by NSF IIS-1815866, IIS-1560229, IIS-1815866 and US Department of Education GAANN Fellowship P200A150306.

REFERENCES

[1] Elena Agapie, Gene Golovchinsky, and Pernilla Qvarfordt. 2013. Leading people to longer queries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3019–3022.

[2] Charu C Aggarwal et al. 2016. *Recommender systems*. Springer.

[3] Xavier Amatriain, Josep M Pujol, Nava Tintarev, and Nuria Oliver. 2009. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 173–180.

[4] Anne Aula, Natalie Jhaveri, and Mika Käki. 2005. Information search and re-access strategies of experienced web users. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 583–592.

[5] Jeremy Boy, Francoise Detienne, and Jean-Daniel Fekete. 2015. Storytelling in information visualizations: Does it engage users to explore data?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1449–1458.

[6] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 89–96.

[7] André Calero Valdez, Martina Ziefle, and Katrien Verbert. 2016. HCI for recommender systems: the past, the present and the future. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 123–126.

[8] Giuseppe Carenini and John Loyd. 2004. ValueCharts: analyzing linear models expressing preferences and evaluations. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 150–157.

[9] Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. 2008. Here or there. In *European Conference on Information Retrieval*. Springer, 16–27.

[10] Li Chen and Pearl Pu. 2004. *Survey of preference elicitation methods*. Technical Report.

[11] R Jordan Crouser, Lyndsey Franklin, Alex Endert, and Kris Cook. 2017. Toward theoretical techniques for measuring the use of human effort in visual analytic systems. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 121–130.

[12] Geoff Cumming. 2013. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge Publishing.

[13] Ross DeVol, Joe Lee, and Minoli Ratnatunga. 2016. 2016 State Technology and Science Index: Sustaining America's Innovation Economy. (2016).

[14] Evanthia Dimara, Anastasia Bezerianos, and Pierre Dragicevic. 2017. Narratives in crowdsourced evaluation of visualizations: A double-edged sword?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[15] Pierre Dragicevic. 2016. Fair statistical communication in HCI. In *Modern Statistical Methods for HCI*. Springer, 291–330.

[16] Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 473–482.

[17] Mi Feng, Cheng Deng, Evan M Peck, and Lane Harrison. 2017. HindSight: Encouraging exploration through direct encoding of personal interaction history. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 351–360.

[18] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research* 4, Nov (2003), 933–969.

[19] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. 2013. Lineup: Visual analysis of multi-attribute rankings. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2277–2286.

[20] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P Bigham. 2018. A Data-Driven Analysis of Workers' Earnings on Amazon Mechanical Turk. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 449.

[21] Steve Haroz, Robert Kosara, and Steven L Franconeri. 2015. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1191–1200.

[22] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27.

[23] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 203–212.

[24] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 159–166.

[25] White House. 2013. College Scorecard. (2013). https://collegescorecard.ed.gov/data/

[26] Rong Hu and Pearl Pu. 2011. Helping Users Perceive Recommendation Diversity.. In *DiveRS@ RecSys*. 43–50.

[27] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 133–142.

[28] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

[29] Joseph A Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction* 22, 1-2 (2012), 101–123.

[30] Caitlin Kuhlman and Elke Rundensteiner. 2017. Towards an Interactive Learn-to-Rank System for Economic Competitiveness Understanding. In *KDD 2017 Interactive Data Exploration and Analytics Workshop*.

[31] Caitlin Kuhlman, MaryAnn VanValkenburg, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Elke Rundensteiner, and Lane Harrison. 2018. Preference-driven Interactive Ranking System for Personalized Decision Support. In *Proceedings of the International Conference on Information and Knowledge Management*. ACM.

[32] Ian Li, Anind Dey, and Jodi Forlizzi. 2010. A stage-based model of personal informatics systems. In *Proceedings of the SIGCHI Conference*

on Human Factors in Computing Systems. ACM, 557–566.

[33] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[34] Benedikt Loepp, Katja Herrmanny, and Jürgen Ziegler. 2015. Blended recommending: Integrating interactive information filtering and algorithmic recommender techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 975–984.

[35] Benedikt Loepp, Tim Hussein, and Jüergen Ziegler. 2014. Choice-based preference elicitation for collaborative filtering recommender systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3085–3094.

[36] Winter Mason and Duncan J Watts. 2009. Financial incentives and the performance of crowds. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*. ACM, 77–85.

[37] Tapio Pahikkala, Evgeni Tsivtsivadze, Antti Airola, Jorma Boberg, and Tapio Salakoski. 2007. Learning to rank with pairwise regularized least-squares. In *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, Vol. 80. Citeseer, 27–33.

[38] Stephan Pajer, Marc Streit, Thomas Torsney-Weir, Florian Spechtenhauser, Torsten Möller, and Harald Piringer. 2017. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 611–620.

[39] Filip Radlinski and Thorsten Joachims. 2007. Active exploration for learning rankings from clickthrough data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 570–579.

[40] Klaus Schwab, Xavier Sala-i Martin, et al. 2017. The global competitiveness report 2016-2017. World Economic Forum.

[41] Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. 2012. RankExplorer: Visualization of ranking changes in large time series data. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2669–2678.

[42] Martin Szummer and Emine Yilmaz. 2011. Semi-supervised learning to rank with preference regularization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 269–278.

[43] Emily Wall, Subhajit Das, Ravish Chawla, Bharath Kalidindi, Eli T Brown, and Alex Endert. 2018. Podium: Ranking data using mixed-initiative visual analytics. *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 288–297.

[44] Jagoda Walny, Samuel Huron, Charles Perin, Tiffany Wun, Richard Pusch, and Sheelagh Carpendale. 2018. Active Reading of Visualizations. *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 770–780.

[45] Ronald L Wasserstein and Nicole A Lazar. 2016. The ASA's statement on p-values: Context, process, and purpose. *The American Statistician* (2016).

[46] Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. 2013. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*. 109–117.

[47] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *International Conference on Information and Knowledge Management*.

[48] Li Zhou. 2015. Obama's New College Scorecard Flips the Focus of Rankings. *The Atlantic* (2015). https://www.theatlantic.com/education/archive/2015/09/obamas-new-college-scorecard-flips-the-focus-of-rankings/405379/