A Document Rating System for Preference Judgements

Maryam Bashir, Jesse Anderton, Jie Wu, Peter B. Golbus, Virgil Pavlu, Javed A. Aslam
College of Computer and Information Science, Northeastern University
Boston, Massachusetts, USA
{maryam,jesse,evawujie,pgolbus,vip,jaa@ccs.neu.edu}

ABSTRACT

High quality relevance judgments are essential for the evaluation of information retrieval systems. Traditional methods of collecting relevance judgments are based on collecting binary or graded nominal judgments, but such judgments are limited by factors such as inter-assessor disagreement and the arbitrariness of grades. Previous research has shown that it is easier for assessors to make pairwise preference judgments. However, unless the preferences collected are largely transitive, it is not clear how to combine them in order to obtain document relevance scores. Another difficulty is that the number of pairs that need to be assessed is quadratic in the number of documents. In this work, we consider the problem of inferring document relevance scores from pairwise preference judgments by analogy to tournaments using the Elo rating system. We show how to combine a linear number of pairwise preference judgments from multiple assessors to compute relevance scores for every document.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory

Keywords

Evaluation, Preference Judgment

1. INTRODUCTION

Traditional methods of collecting relevance judgments make binary assumption about relevance i.e. a document is assumed to be either relevant or non-relevant to the information need of a user. This assumption turns relevance judgment into a classification problem. In the modern world, search engines can easily retrieve thousands of documents at least somewhat relevant to the user's information need. Therefore it becomes necessary to assign a ranking to these documents based on their degree of relevance. This somewhat more continuous notion of relevance cannot be expressed through binary relevance judgments; researchers have developed two ways to express non-binary relevance judgments: either consider relevance as a relative notion such that one document is more or less relevant than another document, or consider relevance as a quantitative notion and create multiple grades of relevance. The first notion of relevance can be expressed as *pairwise preference* judgments; the second notion can be expressed as *nominal graded* relevance judgments, which appear far more prevalently in the literature.

Graded relevance has two significant shortcomings. First, the total number of grades must be defined in advance, and it is not clear how this choice effects the relative measurement of system performance. Second, graded judgments require assessors to choose between arbitrarily defined grades, a choice on which different assessors can easily disagree. The alternative, pairwise preference judgments, allows the assessor to make a binary decision, freeing him or her from the difficulty of deciding between multiple relevance grades. Another advantage of using preferences is that many popular learning-to-rank algorithms, e.g. RankBoost and RankNet, are naturally trained on preferences; thus a better training set can be obtained from direct preference judgments, as opposed to pairwise preferences inferred from nominal judgments.

Pairwise preference judgments have not been explored extensively in the literature. There have been several attempts to use preference judgments by inferring them from absolute judgments [4] and from click data [8]. Nie et al. [9] used preferences for relevance assessments and showed that labelling effort can be reduced by focussing on top ranked documents. Chen et al. [2] also used preferences but focused more on estimating worker quality. To the best of our knowledge, the only work where assessors were asked for direct pairwise preferences as well as absolute relevance judgments for the comparison of the two assessment approaches is by Carterette et al. [1]. The authors showed that rate of inter-assessor agreement is higher on preference judgments, and that assessors take longer to make absolute judgments than preference judgments.

If a simple routine is to be used to infer document relevance from pairwise preferences, it is essential that the preferences be *transitive*, so that we may sort documents by

We gratefully acknowledge support provided by NSF IIS-1256172.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

preference and decide which and how many pairs to judge. Carterette et al., by collecting all $O(n^2)$ preference judgments found that the preferences they collected are transitive 99% of the time. However, the study used experts assessors. The critical property of transitivity might not hold when judgments are collected through the much noisier process of crowdsourcing.

In order to obtain document grades (or scores) from a smaller number of preference judgments, we draw an analogy to the *tournament problem*. In a typical tournament, pairs of players or teams compete in matches of one or more games. The desired outcome is a final ranking (or scoring) of each competitor. A common solution is to use the *Elo rating system* [3], in which players are assigned ratings which are updated iteratively each time the player competes in a match. Using the Elo rating system to combine preference judgments into document grades has the following benefits:

1. The judgments do not need to be transitive. We cannot simply sort the documents by preference since humans assessors can be intransitive in their assessments; especially when we are combining preference judgments from noisy assessments (e.g. through crowdsourcing). The Elo rating system produces a ranking of documents even if the preferences are not transitive.

2. We do not need a quadratic number of pairwise assessments for inferring the relevance of documents. The Elo rating system can be applied to any number of assessments. Indeed, it can infer highly reliable relevance scores using only a linear number of pairwise assessments.

3. For any pair of documents, the document scores produced using the Elo rating system can be used to compute the likelihood of one document is more relevant than the other. In this way we can predict all $O(n^2)$ preferences while only collecting $O(n)$ judgments.

2. THE ELO RATING SYSTEM

The Elo rating system is a method for calculating the relative rating of players in two player games [3]. The system assigns each player a rating score, with a higher number indicating a better player. Each player's rating is updated after he or she has played a certain number of matches, increasing or decreasing in value depending on whether the player won or lost each match, and on the ratings of both players competing in each match—beating a highly rated player increases one's rating more than beating a player with a low rating, while losing to a player with a low rating decreases one's score more than losing to a player with a high rating. These scores are used in two ways: 1) players are ranked by their scores, and 2) the scores are used to compute the likelihood that one player will beat another. If the matches are selected intelligently, the stable ratings can be achieved after only $O(n)$ matches played.

Given the two player's ratings before the match, denoted R_A and R_B, an expected match outcome is calculated for each player: E_A and E_B. The actual output of the match from the perspective of each player (since a win for player A is assumed to be a loss for player B) is denoted as S_A and S_B. The ratings are updated after each match, based on how the expected aligns with the actual outcome.

The Elo rating system can be applied directly to our problem by treating the documents as players, their scores as the ratings to be learned, and document-pairwise preference assessments as matches. All documents begin the "tournament" rated equally. After each document "plays" a match, we update its rating according to equation 2. Each match corresponds to a fixed number of assessors expressing a preference between the pair of documents. The actual outcome of the match for each document, S, is the number of assessors that preferred that document plus half the number of assessors who considered the documents to be "tied." After all the matches are played, we can rank the documents by their final score. This list can be thresholded to produce absolute relevance judgments. We can also use the scores directly to compute transitive preference judgments.

2.1 Math Details of the Elo Rating System

If, before a match, document A has a rating of R_A and document B has a rating of R_B, then the expected outcome of the match according to the Elo rating system is:

$$E_A = \frac{1}{1+10^{\frac{R_B-R_A}{F}}}; \qquad E_B = \frac{1}{1+10^{\frac{R_A-R_B}{F}}} \tag{1}$$

where F is a rating disparity parameter used to control how quickly ratings can change.

If E_A is greater than E_B, then we expect document A to win the match. Once the match is played and we can observe S_A and S_B, the documents' Elo rating is updated as follows:

$$R'_A = R_A + K(S_A - E_A); \quad R'_B = R_B + K(S_B - E_B) \tag{2}$$

where K is a game importance parameter that can be varied so as to give some matches more weight than others.

2.1.1 Elo Rating with Variance

The Elo rating system assumes that the uncertainty about a player's skill rating does not change over time. Therefore, all skill rating updates are computed with the same variance, and any change in the uncertainty about the player's skills over time is not modeled. Glickman proposed to solve this problem by incorporating the variance over time in the player's skill rating [5]. Other researchers have used Glickman's system for the purpose of ranking documents based on clickthrough data [10]. Glickman presented the idea of modeling the belief about a player's skills as a Gaussian distribution whose mean corresponds to the player's rating. As a player plays more matches, the uncertainty about his her her skills is decreased, and this is reflected by a decrease in the variance of the player's associated Gaussian distribution. Rather than using equation 2, the mean rating (R_A) and variance (σ^2) of each document is updated using equation 3 and equation 4 as follows:

$$R'_A = R_A + Kg(\sigma_B^2)(S_A - E_A) \tag{3}$$

$$g(\sigma^2) = \frac{1}{\sqrt{1+\frac{3q^2\sigma^2}{\pi^2}}} \tag{4}$$

where,

$$E_A = \frac{1}{1+10^{-g(\sigma_B^2)\frac{R_B-R_A}{F}}} \tag{5}$$

$$K = \frac{q}{\frac{1}{\sigma_A^2}+\frac{1}{\delta^2}}; \quad \sigma^2 = \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\delta^2}}; \quad q = \frac{\log 10}{F} \tag{6}$$

$$\delta^2 = \frac{1}{q^2 \sum_{j=1}^{m} n_j g(\sigma_j^2)^2 E_A(1 - E_A)} \quad (7)$$

Throughout this work, we set $F = 200$. Each document is initialized with a mean of 100 and a variance of 10.

2.2 Selection of Preference Pairs

For our preliminary experiments, we select $O(n)$ matches stochastically. Each document in the list will be compared against five other documents. We wish to sample pairs in such a way that we create a bias towards relevant documents. In this way, relevant documents will play more matches than non-relevant documents, giving them more opportunities to improve their ratings and move up the list. First, we calculate an initial relevance score for each document using BM25. This produces an initial ranking of the documents for each topic. We collected complete pairwise preferences between the top six documents. For each document below the top six, we select five documents from the set of documents with higher BM25 scores, uniformly at random. We collected four worker assessments for each preference pair which we selected for judgment. We sort all documents based on their Elo ratings after all $O(n)$ matches have been played.

3. EXPERIMENTS

We will compare our methodology for collecting relevance grades from pairwise preferences to the results of the TREC 2012 Crowdsourcing track[1]. The goal of the track was to evaluate approaches to crowdsourcing high quality relevance judgments for text documents and images. Track participants were asked to provide new binary relevance grades, as well as probabilities of relevance, for 18,260 documents that had previously been judged with respect to ten topics selected randomly from the TREC 8 ad-hoc collection.

3.1 Crowdsourcing

We crowdsourced our preference judgments using Amazon Mechanical Turk (AMT)[2]. Each crowd worker was shown the interface presented in Figure 1. Workers were shown the title, description, and narrative fields of a TREC topic, and presented with two documents. Worker's were asked which document "does a better job of answering the questions at the top of the page." They were allowed to select either document, as well as the options "They're Equally Good" and "They're Equally Bad." Internally, these latter two options were treated equivalently as ties. Each task, known on AMT as a HIT, consisted of 20 preference pairs for the same topic, and had a time limit of 30 minutes. Workers were paid $0.15 for each approved HIT. The order in which the document pairs were displayed, as well as which document appeared on which side of the interface, was randomized.

3.1.1 *Quality Control*

The workers we employed have no particular training in assessing document relevance. Therefore, we need a means of verifying the quality of their work. We used trap questions, a document pair for which the "correct" preference is already known, in our study to ensure that workers are giving us reasonable results, and not just clicking randomly. We asked five graduate students studying information retrieval to create our trap questions by pairing documents which they deemed highly relevant with documents they deemed highly non-relevant. We then inserted five of these trap questions, selected at random, into each HIT. As a result, each assignment consisted of five trap questions and fifteen "real" questions. Worker's submission were not accepted unless at least two of the five trap questions were answered correctly. Although, answering two of the five trap questions is not strict criteria but it makes sure that the worker's performance is not worse than random answers.

As another means of ensuring the quality of the collected judgments, we also employed Expectation Maximization (EM). In this context EM, is a means of estimating the "true" pairwise preferences from crowd workers as latent variables in a model of worker quality. For every pair of documents about which we collected judgments from workers, EM provides a probability that one document beats the other. EM has been shown to work well for aggregating labels from multiple crowd workers on AMT [7], and in particular with regarding to collecting relevance judgments [6].

[1]http://sites.google.com/site/treccrowd
[2]http://www.mturk.com

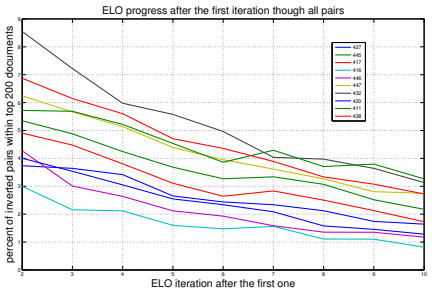

Figure 2: Relationship of Number of Elo rating iterations to percent of pairs inverted, separately for each query.

3.2 Iterations of Elo Rating

In Elo rating system, the score of each document depends on the score of its opponent document in a match. The order in which matches are played has an impact on scores of documents. For example, if a document wins a match against a relevant document, and the relevant document has not played any match yet, then the score of the document would not increase significantly. If the relevant document has already played few matches and has raised its score, then wining a match against it would increase the score of a document to a large extent. Because of this, if we run only one iteration of Elo rating algorithm (through all pairs) then some document scores may not be reliable; we instead run several iterations of Elo rating algorithm so that scores of documents converge. Figure 2 shows the relationship of number of Elo rating iterations to percentage of pairs inverted, after the initial run through all pairs. Note that as we run more iterations, the percentage of pairs whose order is changed decreases.

3.3 Baseline

In order to measure the quality of our Elo-based system, we also implemented a naive system as a baseline. In our naive system, each document is given a score based on the percentage of its matches that it won and the number of matches it competed in. The score of a document A is calculated as:

$$score(A) = \lambda \frac{wins_A}{matches_A} + (1 - \lambda) \frac{matches_A}{matches} \quad (8)$$

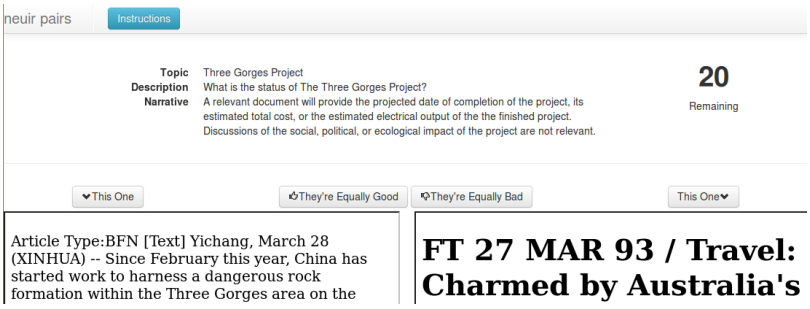

Figure 1: Preference pair selection interface

Topic ID	# Documents in Collection	# Relevant Documents	AUC				
			Median Score of TREC Participant Runs	Baseline	Elo Without Variance	Elo	Elo+EM
411	2056	27	0.86	0.809	0.811	0.857	0.862
416	1235	42	0.85	0.919	0.940	0.944	0.939
417	2992	75	0.75	0.848	0.897	0.887	0.914
420	1136	33	0.71	0.808	0.834	0.823	0.853
427	1528	50	0.73	0.864	0.871	0.882	0.907
432	2503	28	0.71	0.544	0.536	0.637	0.558
438	1798	173	0.78	0.725	0.731	0.708	0.774
445	1404	62	0.83	0.750	0.748	0.790	0.843
446	2020	162	0.82	0.700	0.716	0.720	0.865
447	1588	16	0.76	0.935	0.995	0.859	1.000
All	18260	668	Not Reported	0.790	0.808	0.811	0.851

Table 1: Evaluation Results using AUC for Preference based Relevance Judgements. Elo+EM is statistically significantly better than Baseline, Elo is not significantly better than baseline.

where $wins_A$ is number of matches won by document A, $matches_A$ is total number of matches played by a document A, and $matches$ is total number of matches played. Since we did not have enough data to properly tune λ, λ is set to 0.5.

3.4 Results

Table 1 shows the Area Under the ROC Curve (AUC), one of the primary measures used in the TREC 2012 Crowdsourcing, of our Elo and Baseline systems, with and without EM, and the median scores of the 33 systems that participated in the Crowdsourcing track. For most topics, our Elo-based system outperforms both the Baseline naive system and the median TREC participant. When we also use EM, our results improve. The results using Elo+EM are significantly[3] better than the simple baseline.

4. CONCLUSION AND FUTURE WORK

Preference judgments are easier for assessors to produce and are more useful for training learning-to-rank algorithms. However, their use has been limited due to the polynomial increase in the number of judgments that need to be collected. In this work, we have shown how the Elo rating system can be used to combine a linear number of preferences to obtain either an ordered list of documents or document relevance scores. The results of our experiments are encouraging and demonstrate the potential of our Elo-based system for inferring the relevance of documents from a linear number of pairwise preference judgments.

In future work, we plan to use active learning to intelligently select which pairs of documents to judge in an online manner. The pairwise preference judgments collected in each phase of active learning will dictate which pairs are selected to be judged in the next phase.

[3]Statistical significance is determined using a two-tailed T-Test and is measured at a significance level of 0.05.

5. REFERENCES

[1] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there. In *ECIR*, 2008.

[2] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of WSDM*. ACM, 2013.

[3] A. Elo and S. Sloan. *The Rating of Chess Players, Past and Present*. Arco Publishing, 1978.

[4] H. P. Frei and P. Schauble. Determining the effectiveness of retrieval algorithms. *Inf. Process. Manage.*, 27(2-3), 1991.

[5] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. In *Applied Statistics*, pages 48–377, 1999.

[6] M. Hosseini, I. J. Cox, N. Milić-Frayling, G. Kazai, and V. Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *ECIR*. Springer-Verlag, 2012.

[7] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *SIGKDD Workshop on Human Computation*. ACM, 2010.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*. ACM, 2002.

[9] S. Niu, J. Guo, Y. Lan, and X. Cheng. Top-k learning to rank: labeling, ranking and evaluation. In *Proceedings of SIGIR*. ACM, 2012.

[10] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of SIGKDD*. ACM, 2007.