Suppressing Outliers in Pairwise Preference Ranking

Vitor R. Carvalho, Jonathan L. Elsas, William W. Cohen and Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University
{vitor,jelsas,wcohen,jgc}@cs.cmu.edu

ABSTRACT

Many of the recently proposed algorithms for learning feature-based ranking functions are based on the pairwise preference framework, in which instead of taking documents in isolation, document pairs are used as instances in the learning process [3, 5]. One disadvantage of this process is that a noisy relevance judgment on a single document can lead to a large number of mis-labeled document pairs. This can jeopardize robustness and deteriorate overall ranking performance. In this paper we study the effects of outlying pairs in rank learning with pairwise preferences and introduce a new meta-learning algorithm capable of suppressing these undesirable effects. This algorithm works as a second optimization step in which any linear baseline ranker can be used as input. Experiments on eight different ranking datasets show that this optimization step produces statistically significant performance gains over state-of-the-art methods.

Categories: H.3.3 Information Search and Retrieval: Retrieval models **General Terms:** Algorithms.

1. OUTLIERS IN PAIRWISE RANKING

Learning effective feature-based ranking functions is a fundamental task for search engines, and has recently become an active area of research [3, 7]. One popular approach to learning feature-based ranking functions is the pairwise ranking framework, where the goal is to learn a *preference function* over pairs of documents given a query.

There are many practical advantages in adopting the pairwise preference ranking framework. First, most classification methods can be easily adapted to this formulation of the ranking problem. Second, this framework can be generalized to any graded relevance levels (e.g. definitely relevant, somewhat relevant, non-relevant). Third, in many scenarios it is easier to obtain large amounts of pairwise preference data [5]. In addition, there is evidence that assessment of pairwise preferences is easier for assessors and yields higher inter-annotator agreement [1].

Using pairwise preferences, however, does pose some risks. In the presence of labeling errors or other "noise" in the document relevance information, creating a training set by pairing documents causes a quadratic increase in the number of noisy outlier observations, and this can have a strong negative impact on the quality of the learned ranking function. Specifically, mis-labeling of a single document's absolute judgement will lead to many "mis-labeled" document pair preferences. When using graded relevance levels, confusion or inconsistencies between different relevance levels may make mis-labeling a common problem.

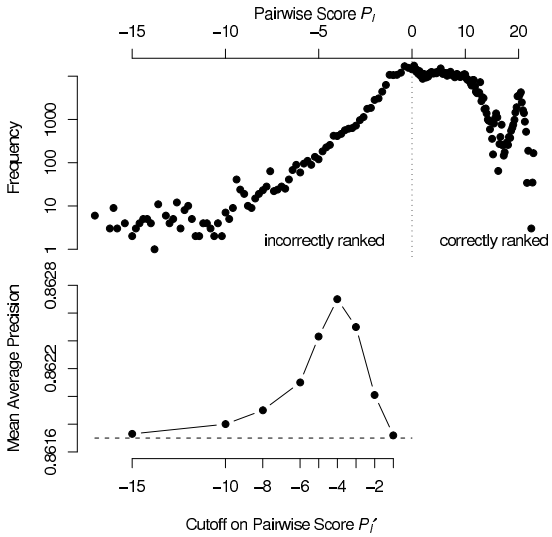

Figure 1: Example of outliers in pairwise ranking. (top) Histogram of pairwise scores. (bottom) Test MAP when excluding training instances whose scores were below cutoff.

Mis-labeling the absolute relevance level of a document is not the only source of outliers. Due to the nature of keyword search, we have an extremely impoverished view of the information need — typically only 2-3 terms per query. For this reason, the query-document features may not be expressive enough to truly distinguish relevant from non-relevant documents. This may result in many non-relevant documents "looking similar" to relevant in the query-document feature space. These non-relevant documents can also yield a quadratic increase in the number of pairwise outliers.

To illustrate the effect of outliers on rank learning, we trained a RankSVM model [5] on SEAL-1, a Set Expansion dataset [2]. Given the model learned, we calculated the pairwise decision scores P_l [2] for all training data instances and constructed a histogram, as shown in the top of Figure 1. Most pairwise instances had positive scores, showing that the learned ranking model correctly ordered most of the

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.
ACM 978-1-59593-991-3/08/10.

training instances. Some instances, however, had negative scores and the few having the most negative scores may be outliers.

We then retrained the same RankSVM model excluding from the training data a few instances whose scores were below a cutoff value, P'_l, and evaluated the learned model on the same test set. The bottom of Figure 1 shows test MAP results of this experiment. The dashed horizontal line shows performance when all instances are used for training. The leftmost point shows the performance when instances with score below -15 were removed from training. As the removal cuttoff increases up to -4, performance goes up, indicating that the removal of outliers improves the ranker's performance. For larger cutoffs, this effect is curtailed by the larger numbers of instances being discarded and performance drops.

Further evidence also suggests that pairwise ranking can be improved by removing or down-weighting outliers. In perceptron-based algorithms, outliers were identified as document pairs that consistently mis-ranked in several iterations through the training data, and removal of these outliers improved performance and stability of the learned ranking function [3, 4]. This technique, known as the α-bound [6], limits the influence of potential outlier observations on the final learned hypothesis, but it is still unclear how it generalizes to other learning algorithms.

Collection	Percep	Percep +Sigmoid	RankSVM	RankSVM +Sigmoid
OHSUMED	0.318	**0.451**††	0.447	**0.448**
TREC-03	0.067	**0.254**††	0.203	**0.244**
TREC-04	0.324	**0.385**†	0.385	**0.393**
SEAL-1	0.851	**0.866**††	0.862	**0.866**††
SEAL-2	0.869	**0.893**††	0.890	**0.894**††
SEAL-3	0.906	**0.924**††	0.916	**0.920**†
TOCCBCC	0.425	**0.479**††	0.472	**0.480**††
CCBCC	0.463	**0.524**††	0.516	**0.521**

Table 1: MAP test values for all datasets. Statistical significance tests over the previous column values are marked with † or †† for the Wilcoxon Matched-Pairs Signed-Ranks test with $p < 0.05$ or 0.01, respectively.

2. ROBUST PAIRWISE RANKING

In order to develop a new general mechanism to down-weight the influence of outliers in pairwise preference learning, we first observed that many competitive rank learners, such as RankSVM [5], utilize convex loss functions in order to optimize rank orderings. One of the disadvantages of convex loss function is its sensitivity to outliers. Outlier points have a strong contribution to the global loss, giving these outliers an important role in determining the final learned hypothesis.

To address this problem, we propose to approximate the number of misranks (the empirical 0/1 loss) using a non-convex sigmoidal function, instead of a convex one[1]. There are at least two advantages in using this particular loss function. First, this non-linear penalty suppresses the effect of outliers, i.e., not giving larger loss values to instances with very large negative pairwise scores. Second, this penalty can arbitrarily approximate the empirical 0/1, leading to potentially higher generalization accuracy.

The sigmoid loss function is not convex, thus the learning procedure is only guaranteed to reach a local maximum. To avoid learning poor locally optimal solutions, the sigmoid ranker is used as a second optimization step, refining the hypothesis produced by another ranker. Specifically, sigmoid-based optimization is seeded with the hypothesis learned from a base ranker, such as RankSVM, and then it converges to a local optimum close to the (presumably good) seed hypothesis. Among other methods, gradient descent can be used to learn parameters on this model [2].

We performed experiments on eight different ranking datasets: three datasets from LETOR (TREC-03, TREC-04, and Ohsumed), two from email recipient recommendation task (TOCCBCC and CCBCC) [2] and three other datasets from the set expansion task (SEAL-1, SEAL-2 and SEAL-3)[2]. Experiments were conducted with the sigmoid ranker using three baseline rankers: RankSVM and the averaged perceptron [3] trained using only 5 passes over the data. Description and details on the aforementioned datasets and algorithms can be found elsewhere [2].

Performance results for all ranking tasks are illustrated in Table 1. On all SEAL datasets there were statistically significant MAP improvements for the sigmoid ranker on the top of both base rankers. On both CCBCC and TOCCBCC tasks, the proposed ranker produced significantly better results than the averaged perceptron ranker. There are also visible performance gains for sigmoid ranker applied to RankSVM, although more modest.

In all LETOR datasets, the sigmoid optimization significantly improved results for the averaged perceptron ranker. For RankSVM, the sigmoid ranker produced improvements in all collections, with the largest gain for TREC-03. Although the proposed ranker improved performance on average, these improvements were not statistically significant. Because the LETOR collections have a relatively larger number of features and a smaller number of queries, we speculate that these ranking models are overfitting the training data.

Surprisingly, the perceptron+Sigmoid performance numbers were comparable, and sometimes slightly better, than those using stronger base rankers. This may be an indication that initially using a method that is sensitive to outliers can lead the learner astray, yielding a seed model that is too strongly influenced by those outliers. Please refer to the longer version of this paper [2] for a more detailed description of the algorithm and further experimental analysis.

[1]Details on the complete optimization procedure with the sigmoid-based loss function can be found elsewhere [2].

3. REFERENCES

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

[2] V. R. Carvalho, J. L. Elsas, W. W. Cohen, and J. G. Carbonell. A meta-learning approach for robust rank learning. In *SIGIR '08: Proceedings of the Learning to Rank Workshop*, 2008.

[3] J. Elsas, V. R. Carvalho, and J. G. Carbonell. Fast learning of document ranking functions with the committee perceptron. In *ACM WSDM*, 2008.

[4] J. Gao, H. Qi, X. Xia, and J.-Y. Nie. Linear discriminant model for information retrieval. In *SIGIR*, 2005.

[5] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[6] R. Khardon and G. Wachman. Noise tolerant variants of the perceptron algorithm. *Journal of Machine Learning Research*, 8:227–248, 2007.

[7] T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR '07: Proceedings of the Learning to Rank Workshop*, 2007.