What Makes Data Robust: A Data Analysis in Learning to Rank

Shuzi Niu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Xiubo Geng
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P. R. China
{niushuzi,gengxiubo}@software.ict.ac.cn, {lanyanyan,guojiafeng,cxq}@ict.ac.cn

ABSTRACT

When applying learning to rank algorithms in real search applications, noise in human labeled training data becomes an inevitable problem which will affect the performance of the algorithms. Previous work mainly focused on studying how noise affects ranking algorithms and how to design robust ranking algorithms. In our work, we investigate what inherent characteristics make training data robust to label noise. The motivation of our work comes from an interesting observation that a same ranking algorithm may show very different sensitivities to label noise over different data sets. We thus investigate the underlying reason for this observation based on two typical kinds of learning to rank algorithms (i.e. pairwise and listwise methods) and three different public data sets (i.e. OHSUMED, TD2003 and MSLR-WEB10K). We find that when label noise increases in training data, it is the *document pair noise ratio* (i.e. *pNoise*) rather than *document noise ratio* (i.e. *dNoise*) that can well explain the performance degradation of a ranking algorithm.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Learning to Rank; Label Noise; Robust Data

1. INTRODUCTION

Learning to rank has gained much attention in recent years, especially in information retrieval [11]. When applying learning to rank algorithms in real Web search applications, a collection of training data is usually constructed, where human judges assign relevance labels to a document with respect to a query under some pre-defined relevance judgment guideline. In real scenario, the ambiguity of query intent, the lack of domain knowledge and the vague definition of relevance levels all make it difficult for human judges to give reliable relevance labels to some documents. Therefore, noise in human labeled training data becomes an inevitable issue that will affect the performance of learning to rank algorithms.

An interesting observation is that the performance degradation of ranking algorithms may vary largely over different data sets with the increase of label noise. On some data sets, the performances of ranking algorithms decrease quickly, while on other data sets they may hardly be affected with the increase of noise. This motivated us to investigate the underlying reason why different training data sets show different sensitivities to label noise. Previous work either only observed how noise in training data affects ranking algorithms [21, 2], or focused on how to design robust ranking algorithms to reduce the effect of label noise [17, 6]. So far as we know, this is the first work talking about data robustness to label noise in learning to rank.

To investigate the underlying reasons for our observations, we conducted data analysis in learning to rank based on three public data sets, i.e. the OHSUMED and TD2003 data sets in LETOR3.0 [13], and the MSLR-WEB10K data set. There are multiple types of judging errors [9] and they leads to different noise distributions [19, 10], but a recent study shows that the learning performance of ranking algorithms is dependant on label noise quantities and has little relationship with label noise distributions [10]. Therefore we randomly injected label errors (i.e. label noise) into training data (with fixed test data) to simulate human judgment errors, and investigated the performance variation of the same ranking algorithm over different data sets. In this study, we mainly focus on the widely used pairwise and listwise learning to rank algorithms.

We find that it is the *document pair noise ratio* (i.e. *pNoise*) rather than *document noise ratio* (i.e. *dNoise*) that can well explain the performance degradation of a ranking algorithm along with the increase of label noise. Here dNoise denotes the proportion of noisy documents (i.e. documents with error labels) to all the documents, while pNoise denotes the proportion of noisy document pairs (i.e. document pairs with wrong preference order) to all the document pairs. We show that the performance degradation of ranking algorithms over different data sets is quite consistent with respect to pNoise. It indicates that pNoise captures the intrinsic factor that determines the ranking performance. We also find the increase

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609542.

of pNoise w.r.t. dNoise varies largely over different data sets. This explains the original observation that the performance degradation of ranking algorithms varies largely over different data sets with the increase of label noise.

2. BACKGROUND

Before we conduct data analysis in learning to rank, we first introduce related work and our experimental settings.

2.1 Related work

How noise affects ranking algorithms has been widely studied in previous work. Voorhees [18] and Bailey et al. [2] show the relative order of ranking algorithms in terms of performance is actually quite stable despite of remarkable noise among human judgements. Recently various learning to rank algorithms have been proposed [7, 4, 22, 3]. Correspondingly, noise in training data has also attracted much attention in this area. Xu et al. [21] explored the effect of training data quality on learning to rank algorithms.

There are also various approaches proposed to learn robust ranking models [6, 20]. Another solution to deal with noisy training set in learning to rank is to employ pre-processing mechanisms, such as "pairwise preference consistency" [5] and repeated labeling techniques [16].

Similar to our work, there do exist a branch of studies on how to construct effective training data in learning to rank. They show us what characteristics of training data are needed to train a effective ranker [1, 8]. Especially, [12] makes an extensive analysis of the relation between the ranking performance and sample size. Different from existing work, our work focuses on investigating the inherent characteristics of training data related to noise sensitivity.

2.2 Experimental Setting

Here we present data sets and ranking algorithms used in our experiments.

Data Sets. In our experiments, we use three public data sets in learning to rank, i.e. OHSUMED, TD2003 and MSLR-WEB10K, for training and evaluation. Among the three data sets, OHSUMED and TD2003 in LETOR3.0 [13] are constructed based on two widely used data collections in information retrieval respectively, the OHSUMED collection and ".gov" collection for TREC 2003 topic distillation task respectively. MSLR-WEB10K is another data set released by Microsoft Research, where relevance judgments are obtained from a retired labeling set of a commercial Web search engine (i.e. Bing). We choose such three data sets for our experiments, since they have quite different properties described in Table 1. All the three data sets can be further divided into training set and test set with their "standard partition". In our experiments, all the evaluations are conducted using the 5-fold cross validation.

Ranking Algorithms. In our work, we mainly focus on the widely applied pairwise and listwise learning to rank algorithms. We employ two typical pairwise learning to rank algorithms, namely *RankSVM* [7] and *RankBoost* [4], and two typical listwise learning to rank algorithms, namely *ListNet* [3] and *AdaRank* [22]. These algorithms are chosen not only because they belong to different categories of learning to ranking approaches (i.e. pairwise and listwise), but also because they adopt different kinds of ranking functions (i.e. linear and non-linear). Specifically, *RankSVM* and *ListNet* use a linear function for scoring, which can be represented as $f(x) = w \cdot x$. Meanwhile, *RankBoost* and *AdaRank* are both ensemble methods that combine many weak ranking functions. Their non-linear ranking function can be expressed as $f(x) = \sum_{t=1}^{n} \alpha_t h_t(x)$, where $h_t(x)$ is the chosen weak ranking function at the t-th iteration.

3. DOCUMENT LABEL NOISE

In this section, we first simulate the label noise in training data with the same noise injection method as [14, 21]. Then we show our basic observations on data sensitivities to label noise.

3.1 Noise Injection Method

We take the original data sets as our ground truth (i.e. noise free data) in our experiments. To simulate label noise in real data sets, we randomly inject label errors into training data, while the test data is fixed. We randomly change some of the relevance labels to other labels uniformly. Since the noise is introduced at the document level, here we define the document noise ratio as the proportion of noisy documents (i.e. documents with error labels) to all the documents, referred to as dNoise. Given a dNoise γ, each query-document pair keeps its relevance label with probability $1 - \gamma$, and changes its relevance label with probability γ. Note that with the simple noise injection method, we assume that human judgment errors are independent and equally possible in different grades. In practice, the possibility of judgment errors in different grades are not equal [15], which can be considered in our future work.

3.2 Basic Observations

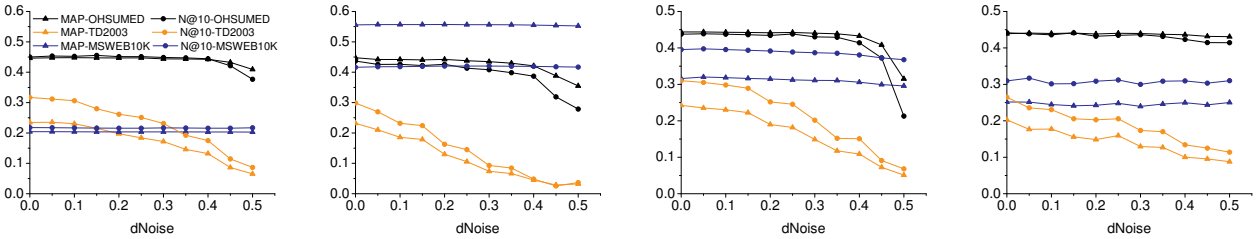
Here we show our basic observations which motivated our work. We consider the ranking performance variation in terms of MAP and NDCG@10 with dNoise from 0 to 0.5 with a step of 0.05. For each fold with a given dNoise, we will apply our noise injection method to a training set for 10 times, and obtain the average performance over the corresponding noise free test set. Each point in the performance curve is derived from the average over 5 folds in Figure 1.

The performance degradation of ranking algorithms may vary largely over different data sets with the increase of dNoise. Take *RankSVM* for instance in Figure 1(a), its performance on TD2003 in terms of MAP (i.e. orange curve with up triangles) decreases quickly as the dNoise increases. Its performance on OHSUMED (i.e. black curve with up triangles) keeps stable for a long range of dNoise, and then drops. Its performance on MSLR-WEB10K (i.e. blue curve with up triangles) is hardly affected even though the dNoise reaches 0.5. The results are consistent in terms of NDCG@10 corresponding to three curves with circles in Figure 1(a). Besides, we can also observe very similar performance degradation behavior with the other three algorithms in Figure 1(b), (c) and (d). In fact, similar results can also been found in previous work [21], but such observations are not the main concern in their work.

The above observations are actually contrary to the following two intuitions. 1)*Degradation Intuition*: For a machine learning algorithm, its performance usually would degrade along with the deterioration of the training data quality (i.e. increase of noise in the training data), no matter quickly or slowly. 2)*Consistency Intuition*: For a same machine learning algorithm, the performance degradation behavior against label noise usually would be similar across

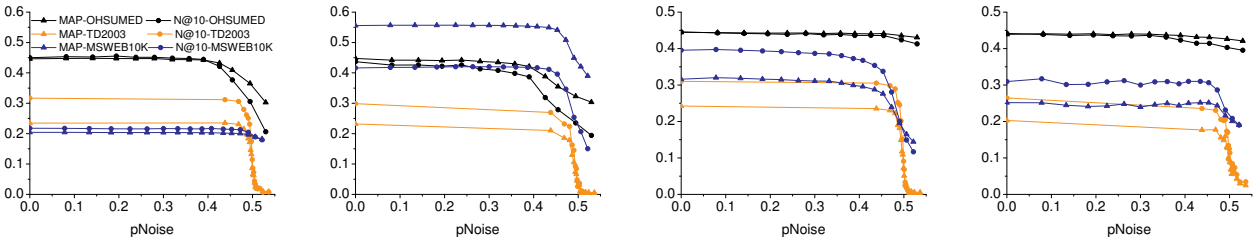
Table 1: Detailed Statistics of Three Data Sets

Data Sets	#queries	#docs	Ave. #docs/query	#features	relevance judgments	label proportions(%)
OHSUMED	106	16,140	152	45	0,1,2	70:16:14
TD2003	50	49,058	981	64	0,1	99.2:0.8
MSLR-WEB10K	10,000	1,200,192	120	136	0,1,2,3,4	51.7:32.5:13.3:1.7:0.8



(a) RankSVM (b) RankBoost (c) ListNet (d) AdaRank

Figure 1: Performance Evaluation against dNoise with four ranking algorithms in terms of MAP and NDCG@10



(a) RankSVM (b) RankBoost (c) ListNet (d) AdaRank

Figure 2: Performance Evaluation against pNoise with four ranking algorithms in terms of MAP and NDCG@10

.

the data sets. A possible reason for the above results is that the label noise (i.e. dNoise) cannot properly characterize the deterioration of the training data quality in learning to rank. This brings us the following question: what is the true noise that affects the performances of ranking algorithms?

4. DOCUMENT PAIR NOISE

To answer the above question, we need to briefly re-visit the learning to rank algorithms. As we know, the pairwise ranking algorithms transform the ranking problem to a binary classification problem, by constructing preference pairs of documents from human labeled data. The `FIFA` World Cup may help understand that a ranking list, a basic unit in listwise learning, is generated by pairwise contests. Thus, for both pairwise and listwise learning, the quality of pairs turns out to be the key to the ranking algorithms. In other words, errors (i.e. noise) in document pairs arise from the original document label noise might be the true reason for the performance degradation of ranking algorithms.

4.1 Definition

To verify our idea, we propose to evaluate the performance of ranking algorithms against the document pair noise. Similar to the definition of dNoise, here we define the document pair noise ratio as the proportion of noisy document pairs (i.e. document pairs with wrong preference order) to all the document pairs, referred to as pNoise. We would like to check whether the performance degradation of ranking algorithms against pNoise is consistent across the data sets.

For this purpose, we first investigate how document pair noise arises from document label noise and take a detailed look at pNoise. The document pairs $\langle d_i, d_j \rangle$ (i.e. document d_i is more relevant than d_j to the given query) generated from a noisy training data set can be divided into three

Table 2: Proportions of Predicted Positive Document Pairs with a tie

	RankSVM	RankBoost	ListNet	AdaRank
OHSUMED	0.49	0.45	0.48	0.44
TD2003	0.50	0.49	0.50	0.47
MSLR-WEB10K	0.50	0.49	0.49	0.44

categories according to the original relationship of the two documents in the noise free data set. (1)Correct-Order Pairs. Document d_i is indeed more relevant than d_j in the original noise free training data. It is clear that the correct-order pairs do not introduce any noise since they keep the same order as in the noise free case, even though the relevance labels of the two documents might have been altered. (2)Inverse-Order Pairs. Document d_j is more relevant than d_i in the original noise free training data instead. Obviously, the inverse-order pairs are noisy pairs since they are opposite to the true preference order between two documents. (3)New-Come Pairs. Document d_i and d_j are a tie in the original noise free training data instead, so we estimate its true preference order by the ranking model learned from the noise free training data. The probability being a noisy pair is estimated by the proportion that the prediction score of d_j is higher than d_i through experiments[1]. The experimental results in Table 2 show that there is approximately half a chance of being a noisy pair.

According to the above analysis, the pNoise of a given data set with known noisy labels can be estimated as follows

$$\text{pNoise} = \frac{N_{inverse} + 0.5 * N_{new}}{N_{all}},$$

[1]We randomly sample a collection of document pairs (e.g. 10,000 pairs) with a tie from the noise free training data, and compute the proportion of the predicted negative pairs. We denote a document pair, d_i and d_j, as positive if the predicted relevance score of d_i is higher than d_j, otherwise negative. Experiments are conducted over all the three data sets with both pairwise and listwise ranking algorithms and results are averaged over 5-fold training sets.

where $N_{inverse}$, N_{new} and N_{all} denote the number of inverse-order pairs, new-come pairs and all the document pairs, respectively.

With this definition, the performance variations of ranking algorithms against pNoise are depicted in Figure 2, where each sub figure corresponds to performances over three data sets using four ranking algorithms respectively. Surprisingly consistent behavior for the performance degradation is observed across different data sets. The performances (e.g. MAP and NDCG@10) of ranking algorithms keep quite stable as pNoise is low. When pNoise exceeds a certain point[2] (around 0.5 as indicated in our experiment), performances drop quickly. The results w.r.t. pNoise are in accordance with the degradation and consistency intuitions mentioned before, which are violated in the case of dNoise.

Therefore, the results indicate that pNoise captures the intrinsic factor that determines the performance of a ranking algorithm, and thus can well explain the consistency of performance degradation of various ranking algorithms.

4.2 pNoise VS. dNoise

Now we know that dNoise is a natural measure of label noise in training data as label noise is often introduced at document level in practice, while pNoise is an intrinsic measure of noise that can reflect the true noise for ranking algorithms. Here we explore the variation of pNoise against dNoise across different data sets in Figure 3.

The increase of pNoise with respect to dNoise varies largely over different data sets, which actually well explains our basic observations. Given a small dNoise (e.g. 0.1) on TD2003, pNoise reaches a very high value (>0.4) in Figure 3, which is the turning point of the performance curve in Figure 2. This explains why the performances of ranking algorithms on TD2003 drop quickly along with the increase of dNoise. On the contrary, on OHSUMED and MSLR-WEB10K, the variations of pNoise with respect to dNoise are more gentle, and correspondingly the performances of ranking algorithms are quite stable. Even when dNoise reaches 0.5 on MSLR-WEB10K, the corresponding pNoise is still below a threshold (about 0.4), which means the pNoise has not reached the turning point of the performance curve on MSLR-WEB10K according to blue curves in Figure 2. This explains why the ranking performance is hardly affected by dNoise on MSLR-WEB10K in our basic observations.

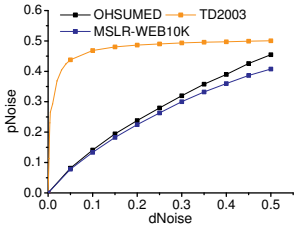


Figure 3: pNoise with respect to dNoise over Three Data Sets

5. CONCLUSION

In this paper, we conducted data analysis to first address the data robustness problem in learning to rank algorithms. In our study, we find that document pair noise captures the true noise of ranking algorithms, and can well explain the performance degradation of ranking algorithms. As a measure of labeling accuracy, it can be used for annotator filtering in a crowdsourced relevance labeling task in our next step. Additionally the current noise injection method is somehow simple and we may further improve the injection method for better analysis. In fact, the noise ratio could be different among queries due to various difficulty levels.

[2]There is a threshold of pNoise, after which the performance will be affected by the label noise much heavily. This guess needs further theoretical guarantee, which are not included in our work.

6. ACKNOWLEDGMENTS

This research work was funded by the 973 Program of China under Grants No. 2012CB316303, No. 2013CB329602, the 863 Program of China under Grants No. 2014AA015204, No. 2012AA011003, the National Natural Science of China under Grant No. 61232010, No. 61203298 and the National Key Technology R&D Program of China under Grants No. 2012BAH39B02.

7. REFERENCES

[1] J. A. Aslam, E. Kanoulas, and et. al. Document selection methodologies for efficient and effective learning-to-rank. SIGIR '09, pages 468–475, 2009.

[2] P. Bailey, N. Craswell, and et. al. Relevance assessment: are judges exchangeable and does it matter. SIGIR '08, pages 667–674, 2008.

[3] Z. Cao, T. Qin, and et. al. Learning to rank: from pairwise approach to listwise approach. ICML '07, pages 129–136.

[4] Y. Freund, R. Iyer, and et. al. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.

[5] X. Geng, T. Qin, and et. al. Selecting optimal training data for learning to rank. *IPM*, 47:730– 741, 2011.

[6] V. Jain and M. Varma. Learning to re-rank: query-dependent image re-ranking using click data. WWW '11, pages 277–286.

[7] T. Joachims. Optimizing search engines using clickthrough data. KDD '02, pages 133–142, 2002.

[8] E. Kanoulas, S. Savev, and et. al. A large-scale study of the effect of training set characteristics over learning-to-rank algorithms. SIGIR '11, pages 1243–1244, 2011.

[9] G. Kazai, N. Craswell, and et. al. An analysis of systematic judging errors in information retrieval. CIKM '12, pages 105–114.

[10] A. Kumar and M. Lease. Learning to rank from a noisy crowd. SIGIR '11, pages 1221–1222, 2011.

[11] T.-Y. Liu. Introduction. In *Learning to rank for information retrieval*, chapter 1, pages 3–30. 2011.

[12] C. Macdonald, R. Santos, and I. Ounis. The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5):584–628, 2013.

[13] T. Qin, T.-Y. Liu, and et. al. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval Journal*, 13:346–374, 2010.

[14] U. Rebbapragada and C. E. Brodley. Class noise mitigation through instance weighting. ECML '07, pages 708–715, 2007.

[15] F. Scholer, A. Turpin, and M. Sanderson. Quantifying test collection quality based on the consistency of relevance judgements. SIGIR '11, pages 1063–1072, 2011.

[16] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. KDD '08, pages 614–622, 2008.

[17] W. C. J. V. Carvalho, J. Elsas. A meta-learning approach for robust rank learning. In *Proceedings of SIGIR 2008 LR4IR*.

[18] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. SIGIR '98, pages 315–323.

[19] J. Vuurens, A. P. De Vries, and C. Eickho. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proceedings of SIGIR 2011 Workshop on CIR*.

[20] J. L. S.-F. C. Wei Liu, Yugang Jiang. Noise resistant graph ranking for improved web image search. In *CVPR*, 2011.

[21] J. Xu, C. Chen, and et. al. Improving quality of training data for learning to rank using click-through data. WSDM '10, pages 171–180, 2010.

[22] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. SIGIR '07, pages 391–398, 2007.