

Which noise affects algorithm robustness for learning to rank

Shuzi Niu¹ · Yanyan Lan¹ · Jiafeng Guo¹ ·
Shengxian Wan¹ · Xueqi Cheng¹

Received: 19 May 2014 / Accepted: 9 April 2015 / Published online: 28 April 2015
© Springer Science+Business Media New York 2015

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 and how to utilize them to guide labeling. 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 three typical kinds of learning to rank algorithms (i.e. pointwise, pairwise and listwise methods) and three public data sets (i.e. OHSUMED, TD2003 and MSLR-WEB10K) with different properties. We find that when label noise increases in training data, it is the *document pair noise ratio* (referred to as *pNoise*) rather than *document noise ratio* (referred to as *dNoise*) that can well explain the performance degradation of a ranking algorithm. We further identify two inherent characteristics of the training data, namely *relevance levels* and *label balance*, that have great impact on the variation of *pNoise* with respect to label noise (i.e. *dNoise*). According to these above results, we further discuss

✉ Yanyan Lan
lanyanyan@ict.ac.cn

Shuzi Niu
niushuzi@software.ict.ac.cn

Jiafeng Guo
guojiafeng@ict.ac.cn

Shengxian Wan
shengxianwan@software.ict.ac.cn

Xueqi Cheng
cxq@ict.ac.cn

¹ Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan Nan Lu, Beijing, Haidian District, People's Republic of China

some guidelines on the labeling strategy to construct robust training data for learning to rank algorithms in practice.

Keywords Learning to rank · Label noise · Robust data

1 Introduction

Learning to rank has gained much attention in recent years, especially in information retrieval (Liu 2011). 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. We try to understand the inherent characteristics of data sets related to noise sensitivity, so that we can obtain some guidelines to construct robust training data for learning to rank in real scenario. Previous work either only observed how noise in training data affects ranking algorithms (Bailey et al. 2008; Xu et al. 2010), or focused on how to design robust ranking algorithms to reduce the effect of label noise (Jain and Varma 2011; Carvalho et al. 2008). 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 with different properties, i.e. the OHSUMED and TD2003 data sets in LETOR3.0 (Qin et al. 2010) benchmark collection, and the MSLR-WEB10K data set. There are multiple types of judging errors (Kazai et al. 2012) and they lead to different noise distributions (Kumar and Lease 2011; Vuurens et al. 2011), but a recent study shows that the learning performance of ranking algorithms is dependent on label noise quantities and has little relationship with label noise distributions (Kumar and Lease 2011). For simplicity, 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 three kinds of learning to rank algorithms through experiments: (1) pointwise algorithms such as *PRank* (Crammer and Singer 2001) and *GBDT* (Friedman 2000); (2) pairwise algorithms such as *RankSVM* (Joachims 2002) and *RankBoost* (Freund et al. 2003); (3) listwise algorithms such as *ListNet* (Cao et al. 2007) and *AdaRank* (Xu and Li 2007).

We first find that it is the *document pair noise ratio* (referred to as *pNoise*) rather than *document noise ratio* (referred to as *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 p_{Noise} denotes the proportion of noisy document pairs (i.e. document pairs with wrong preference order) to all the document pairs. Note that d_{Noise} is a natural measure of label noise in training data since label noise is usually introduced at document level in practice, while p_{Noise} is an intrinsic measure of label noise according to the mechanism of evaluation measures used in information retrieval. We show that the performance degradation of ranking algorithms over different data sets is quite consistent with respect to p_{Noise} . It indicates that p_{Noise} captures the intrinsic factor that determines the performance of a ranking algorithm. We also find that the increase of p_{Noise} with respect to d_{Noise} 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.

We then study what affects the variation of p_{Noise} with respect to label noise (i.e. d_{Noise}). We identify two inherent characteristics of training data that have critical impact on the variation of p_{Noise} with respect to d_{Noise} , namely *relevance levels* and *label balance*. Relevance levels refer to the graded relevance judgments one adopts for labeling (e.g. 2-graded relevance judgments including “relevant” and “irrelevant”), while label balance refers to the balance among labeled data of different relevance levels, which can be reflected by the proportions of labeled data over different relevance levels. We show that for a fixed size of training data, with more relevance levels and better label balance, we can make p_{Noise} less affected by d_{Noise} , i.e. the training data more robust to label noise. These above results actually provide us some valuable guidelines on the labeling strategy to construct robust training data for learning to rank algorithms in practice.

The rest of the paper is organized as follows. Section 2 discusses some background and related work. Section 3 describes our experimental settings. Section 4 shows our basic observations and introduces document pair noise for better explanation. Section 5 further explores the two characteristics of training data that have critical impact on its robustness to label noise, and discusses some principles on constructing robust data set. The conclusion is made in Sect. 6.

2 Background and related work

Noise in data has been widely studied in machine learning literature. For example, Zhu and Wu (2003) presented a systematic evaluation on the effect of noise in machine learning. They investigated the relationship between attribute noise and classification accuracy, the impact of noise from different attributes, and possible noise handling solutions. Nettleton et al. (2010) systematically compared how different degrees of noise affect four supervised learners that belong to different paradigms. They showed that noise in the training data set is found to give the most difficulty to the learners. To handle noise in learning, various approaches have been integrated into existing machine learning algorithms (Abelln and Masegosa 2010; Crammer et al. 2009; Rebbapragada and Brodley 2007) to enhance their learning abilities in noisy environments. Besides, some other work (Jeatrakul et al. 2010; Verbaeten and Assche 2003) concentrated on employing some preprocessing mechanisms to handle noise before a learner is formed.

How noise affects ranking algorithms has been widely studied in previous work. Voorhees (1998) and Bailey et al. (2008) investigated how noise affects the evaluation of ranking algorithms. Their results showed that although considerable noise exists among

human judgements, the relative order of ranking algorithms in terms of performance is actually quite stable. In recent years, learning to rank has emerged as an active and growing research area both in information retrieval and machine learning. Various learning to rank algorithms have been proposed, such as *RankSVM* (Joachims 2002), *RankBoost* (Freund et al. 2003), *RankNet* (Burges et al. 2005) and *ListNet* (Cao et al. 2007). Correspondingly, noise in training data has also attracted much attention in this area. Xu et al. (2010) explored the effect of training data quality on learning to rank algorithms. They showed that judgment errors in training do affect the performance of the trained models.

There are also various approaches proposed to construct robust learning to rank algorithms. For example, Carvalho et al. (2008) proposed a meta-learning algorithm, which took any linear baseline ranker as input and used a non-convex optimization procedure to output a more robust ranking model. Jain and Varma (2011) presented a learning to re-rank method for image search, which cope with the label noise problem by leveraging user click data. In Liu et al. (2011), introduced a graph-theoretical framework amenable to noise resistant ranking to process query samples with noisy labels.

Another solution to deal with noisy data in learning to rank is to employ some pre-processing mechanisms. For example, Geng et al. (2011) proposed the concept “pairwise preference consistency” (PPC) to describe the quality of a training data collection, and selected a subset for training by optimizing PPC measure. Xu et al. (2010) attempted to detect and correct label errors using click-through data to improve the quality of training data for learning to rank. Besides, other methods like repeated labeling techniques (Sheng et al. 2008) and overlapping labeling scheme (Yang et al. 2010) have also been proposed to improve data quality in learning to rank.

Similar to our work, there do exist a branch of studies on how to construct effective training data in learning to rank. A comprehensive research (Aslam et al. 2009) on selecting documents for learning-to-rank data sets has been conducted, and the effect of different document selection methods on the efficiency and effectiveness of learning-to-rank algorithms has also been investigated. Scholer et al. (2011) found that the fraction of inconsistent judgments should be viewed as a lower bound for the measurement of assessor errors and these assessor errors are not made by chance but there are factors contributing to assessment mistakes, such as time between judgments and a judgment inertia in the labeling process. A detailed study on modeling how assessors might make errors has been presented (Carterette and Soboroff 2010) in a low-cost large scale test collection scenario. According to these assessor error models, they further investigated the effect of assessor errors on the evaluation estimation and propose two possible means to adjust for errors. Especially, Macdonald et al. (2013) made an extensive analysis of the relation between the ranking performance and sample size. A surprising finding by Kumar and Lease (2011) is the consistency of learning curves across different noise distributions, which indicates that there is no need to obtain all the learning curves under all kinds of noise distributions exhaustively. Kanoulas et al. (2011) suggested that the distribution of labels across different relevance grades in the training set has an effect on the performance of trained ranking functions.

Our work also has some relation with existing works on how to construct robust training data. In previous work, there are typically two labeling strategy, absolute judgment and relative judgment. The absolute judgments will typically easy to introduce noise to the training data since there are more factors to consider and the choice of label is not clear (Burgin 1992). The relative judgments such as pairwise judgments can be viewed as a good alternative, however, One concern of using this strategy is the complexity of judgment since the number of document pairs is polynomial in the number of document. Recently,

many research work attempt to reduce the complexity of relative labeling strategy to obtain a robust training data, successful examples include top-k labeling strategy (Niu et al. 2012). Active labeling strategies can be applied to further reduce the complexities of labeling.

Different from existing work, our work focuses on investigating the inherent characteristics of training data related to noise sensitivity with the document number per query given. By studying this problem, we aim to obtain some guidelines to construct robust training data for learning to rank.

3 Experimental settings

Before we conduct data analysis in learning to rank, we first introduce our experimental settings. In this section, we will cover four aspects, including the data sets, the noise injection method, ranking algorithms, and evaluation metrics.

3.1 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 come from the benchmark collection LETOR3.0 (Qin et al. 2010). They are constructed based on two widely used data collections in information retrieval, the OHSUMED collection and the “.gov” collection in topic distillation tasks of TREC 2003, respectively. MSLR-WEB10K is another data set released by Microsoft Research, where the relevance judgments are obtained from a retired labeling set of a commercial Web search engine (i.e. Bing). The detailed statistics of three data sets are shown in Table 1. We choose the three data sets typically for our experiments, since they have quite different properties as follows.

Data size Both OHSUMED and TD2003 are smaller data sets as compared with MSLR-WEB10K. There are in total 106 and 50 queries in OHSUMED and TD2003, respectively, while 10,000 queries in MSLR-WEB10K. Besides, there are <50,000 documents in both OHSUMED and TD2003, while more than 1.2 million documents in MSLR-WEB10K. However, if we compare the average document number per query, there are over 900

Table 1 Detailed statistics of three data sets

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

labeled documents per query in TD2003, which is much larger than that in OHSUMED and MSLR-WEB10K.

Feature space The feature spaces of OHSUMED and TD2003 are also smaller than MSLR-WEB10K. OHSUMED and TD2003 contain 45 and 64 features, respectively, while MSLR-WEB10K contains 136 features.

Relevance judgments The three data sets adopt different relevance judgment methods. TD2003 takes two-graded relevance judgments, i.e. 0 (irrelevant) and 1 (relevant). In OHSUMED, three-graded relevance judgments are used, i.e. 0 (irrelevant), 1 (partially relevant) and 2 (definitely relevant), while in MSLR-WEB10K the relevance judgments take 5 values from 0(irrelevant) to 4(perfectly relevant).

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.

3.2 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, with test data fixed. We follow a similar approach as proposed in previous work (Abellán and Masegosa 2009; Abelln and Masegosa 2010; Rebbapragada and Brodley 2007; Verbaeten and Assche 2003; Xu et al. 2010) for noise injection, and introduce noise 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 . For each training-test set pair, we will randomly inject label noise into the training data under a given dNoise for 10 times to obtain the average results.

Assume there are C relevance grades from 0 to $C - 1$ in the training data, and we employ the following two noise injection methods respectively to change its relevance labels.

- Uniform noise profile: randomly change some of the relevance labels to other labels uniformly, i.e. $P(r_i \rightarrow r_j) = \frac{1}{C-1}, \forall r_j \neq r_i, r_i, r_j \in \{0, 1, \dots, C-1\}$.
- Non-uniform noise profile: In practice, the possibility of judgment errors in different grades are not equal (Scholer et al. 2011), since the assessors are easier to label a document as the relevance level nearer to its ground-truth label. For example, a document with ground-truth label r_i are prone to be labeled as r_j with probability $P(r_i \rightarrow r_j) \propto |r_i - r_j|^{-1}$, so we define the error probability based on label differences, i.e. $P(r_i \rightarrow r_j) = \frac{|r_i - r_j|^{-1}}{\sum_{k=0, k \neq r_i}^{C-1} |r_i - k|^{-1}}$.

3.3 Ranking algorithms

In our work, we mainly focus on the widely applied pointwise, pairwise and listwise learning to rank algorithms. We employ two typical pointwise learning to rank algorithms, namely PRank (Crammer and Singer 2001) and GBDT (Friedman 2000), two typical pairwise learning to rank algorithms, namely *RankSVM* (Joachims 2002) and *RankBoost* (Freund et al. 2003), and two typical listwise learning to rank algorithms, namely *ListNet* (Cao et al. 2007) and *AdaRank* (Xu and Li 2007), in our experiments.

These algorithms are chosen not only because they belong to different categories of learning to ranking approaches (i.e. pointwise, pairwise and listwise), but also because they adopt different kinds of ranking functions (i.e. linear and non-linear). Specifically, *PRank*, *RankSVM* and *ListNet* use a linear function for scoring, which can be represented as $f(x) = w \cdot x$. While *GBDT*, *RankBoost* and *AdaRank* are all ensemble methods that combine many weak ranking functions. Meanwhile, 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.

3.4 Evaluation metrics

We use two metrics, i.e. Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto 1999) and Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2000), to evaluate ranking performance, which are both popular measures used in the literature of information retrieval.

MAP for a set of queries is the mean of the Average Precision (AP) for each query, defined as follows

$$MAP = \frac{\sum_{i=1}^Q AP_i}{Q},$$

where Q denotes the number of queries, and AP_i is the AP score of the i -th query, which can be calculated by

$$AP_i = \frac{1}{n_i} \sum_{k=1}^{N_i} \frac{\sum_{j \leq k} I(r(j) = 1)}{k} I(r(k) = 1),$$

where n_i is the number of relevant documents for the i -th query, N_i is the total number of documents for the i -th query, $r(j)$ is the label of the j -th document in the list (e.g. $r(j) = 1$ if the j -th document is relevant and $r(j) = 0$ otherwise), and $I(\cdot)$ is the indicator function.

The calculation of $NDCG@n$ is defined as follows

$$NDCG@n = Z_n \sum_{j=1}^n \frac{2^{r(j)} - 1}{\log(j + 1)},$$

where j is the position in the document list, $r(j)$ is the label of the j -th document in the list, and Z_n is the normalization factor so that $NDCG@n$ of the perfect list equals one.

4 Data sensitivities to label noise and the underlying reasons

In this section, we first show our basic observations on data sensitivities to label noise. We then introduce document pair noise which captures the true noise of ranking algorithms and helps us understand how label noise affects the performance of ranking algorithms. Finally, we explain our basic observations through the variation of document pair noise against document label noise.

4.1 Basic observations

Here we first show our basic observations which motivates our work. Using our noise injection methods, we run a number of experiments over the three data sets with dNoise

changing from 0 to 0.5 with a step of 0.05, and evaluate the performance of the six trained ranking algorithms based on the corresponding noise free test set. The results are depicted in Fig. 1 with uniform noise profile and Fig. 2 with non-uniform noise profile.

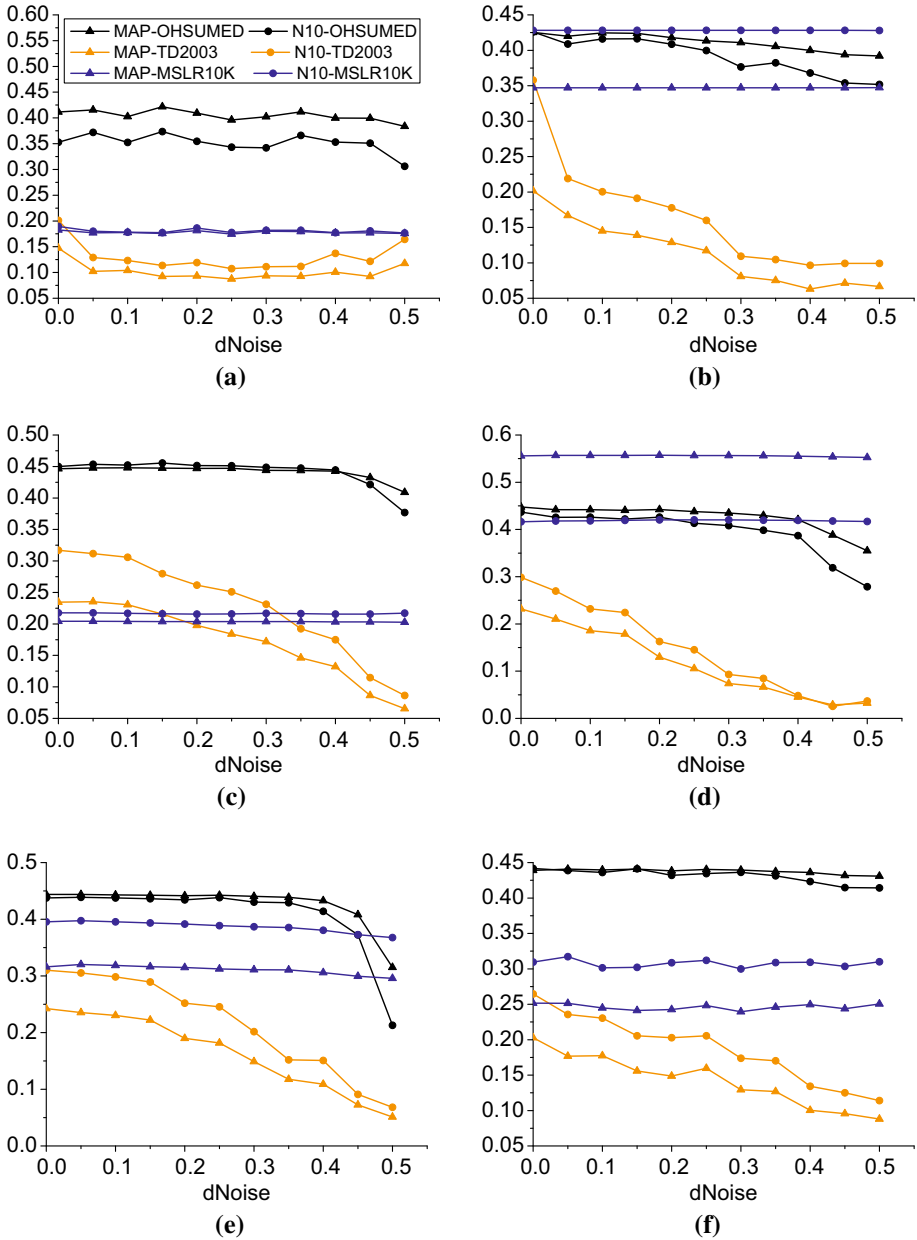


Fig. 1 Performance evaluation against dNoise (Uniform noise profile) with all three kinds of ranking algorithms in terms of MAP and NDCG@10. **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

From Fig. 1 we can clearly see that, the performance degradation of ranking algorithms may vary largely over different data sets with the increase of dNoise. Taking *RankSVM* for instance, from Fig. 1c we can see that, on TD2003 its performance in terms of MAP (i.e. orange curve with up triangles) decreases quickly as dNoise increases. On OHSUMED its performance (i.e. black curve with up triangles) keeps stable for a long range of dNoise, and

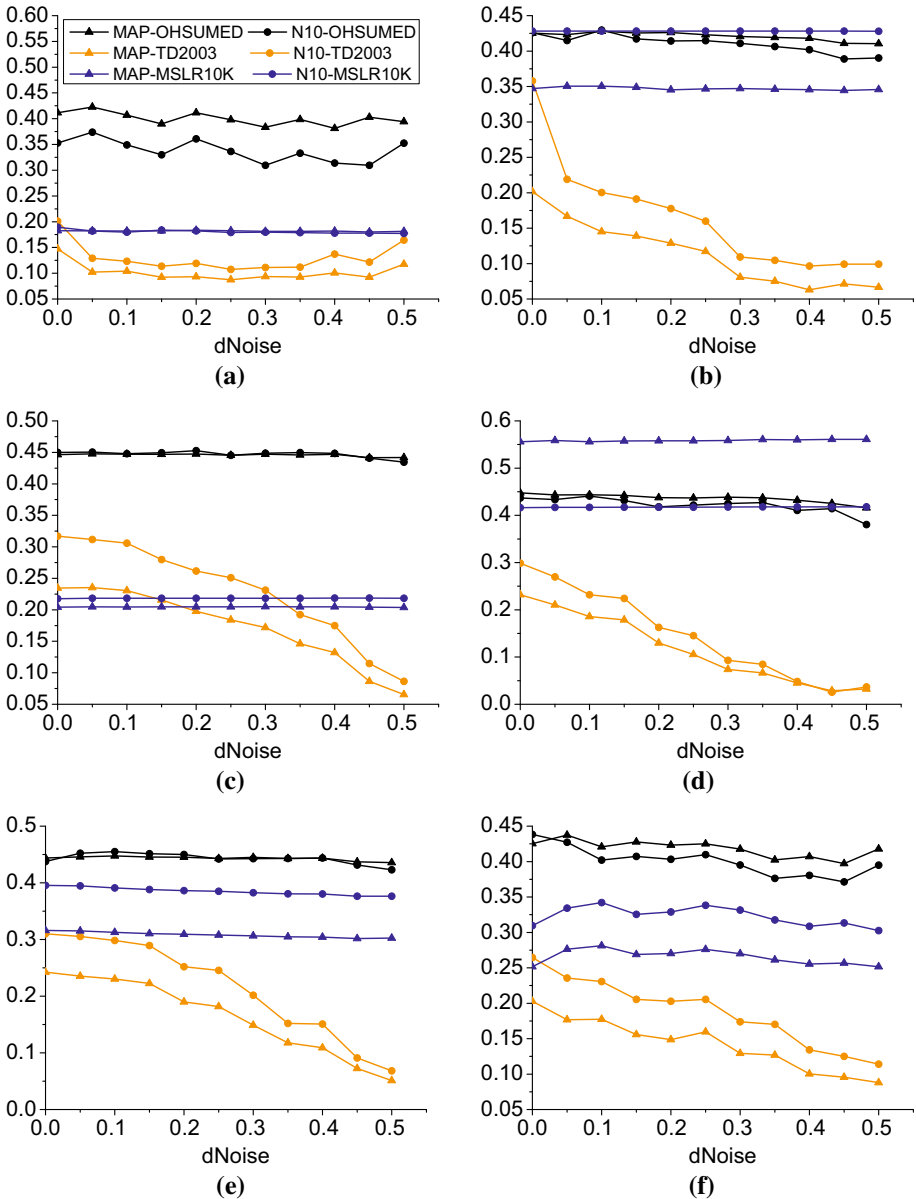


Fig. 2 Performance evaluation against dNoise (Non-uniform noise profile) with all three kinds of ranking algorithms in terms of MAP and NDCG@10. **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

then drops. On MSLR-WEB10K its performance (i.e. blue curve with up triangles) is hardly affected even though dNoise reaches 0.5. The evaluation results are consistent in terms of NDCG@10 as shown in three curves with circles of Fig. 1a. Besides, we can also observe very similar performance degradation behavior with *RankBoost* in Fig. 1b, *ListNet* in Fig. 1c, *AdaRank* in Fig. 1d. In fact, similar results can also be found in Fig. 2 and previous work (Xu et al. 2010), but such observations are not the main concern in their work.

Someone may feel that the results for pointwise algorithms such as PRank and GBDT are quite puzzling, since as pointwise algorithms, they should be sensitive to dNoise. Actually, the results are explainable. Though they are pointwise algorithms, the evaluation measures are mainly based on pairwise orders. That is the main reason why they are not sensitive to dNoise any more. To make it much clearer, we plot the curve of pointwise loss used in PRank as examples to show the trend, as shown in Fig. 3. The differences between this figure and previous ones only lie in the y axis. In the previous ones, the y axis stands for evaluation measures such as MAP and NDCG, while it becomes (normalized) testing loss adopted in PRank. From the results, we can clearly see that the pointwise loss is indeed sensitive to dNoise.

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 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?

4.2 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. Specifically, given a query q and two documents d_i and d_j , a document pair $\langle d_i, d_j \rangle$ is generated if d_i is more relevant than d_j with respect to q . As we can see, document pairs $\langle d_i, d_j \rangle$ comprise the true training data for pairwise ranking algorithms. Thus, the quality of the pairs should be key to the learning performances of pairwise ranking algorithms. This also holds for listwise ranking algorithms although it is not obvious to see. The FIFA

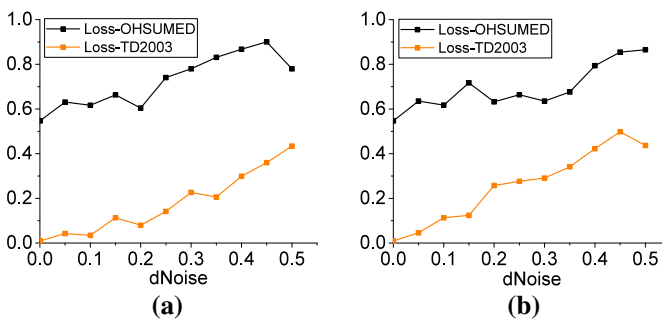


Fig. 3 Testing loss of PRank with respect to dNoise over two data sets. **a** Prank with uniform noise profile; **b** Prank with non-uniform noise profile

World Cup may help understand that a ranking list, a basic unit in listwise learning, is generated by pairwise contests. More specifically, the rank of a certain team is determined by the number of teams being beaten (Taylor et al. 2008). 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.

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 generated from a noisy training data set can be divided into three categories according to the original relationship of the two documents in the noise free data set.

Correct-order pairs: For a document pair $\langle d_i, d_j \rangle$ with preference order from noisy data, 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.

Inverse-order pairs: For a document pair $\langle d_i, d_j \rangle$ with preference order from noisy data, 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.

New-come pairs: For a document pair $\langle d_i, d_j \rangle$ with preference order from noisy data, document d_i and d_j are a tie in the original noise free training data instead. It is not clear whether the new-come pairs are noisy pairs or not, since the original labels of the two documents in the noise free data are the same. We thus conducted some experiments to investigate this problem.

The basic procedure is like this. We first train a ranking algorithm based on the noise free training data. We then randomly sample a collection of document pairs (e.g. 10,000 pairs) with the same original relevance labels from both the training and test data, and predict the preference orders between documents in such pairs. For a document pair $\langle d_i, d_j \rangle$, we denote it as a positive pair if the predicted relevance score of d_i is higher than d_j , otherwise negative. The experiments are conducted over all the three data sets with all the three kinds of ranking algorithms. The 5-fold cross validation results are shown in Table 2.

From Table 2, we can see that the proportion of positive or negative pairs is around 0.5 over all the sets. The results show that there is no strong preference order between documents with the same original relevance label, which is quite reasonable and intuitive.

Table 2 Proportions of predicted positive document pairs with same relevance levels

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

In this way, if two documents with the same relevance label are turned into a new-come pair due to label noise, there would be 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

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

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, we now apply the same noise injection method, run a number of experiments by varying dNoise as before, and evaluate the performance of algorithms against the pNoise. The results are depicted in Figs. 4 (uniform noise profile) and 5 (non-uniform noise profile), where each sub-figure corresponds to performances over three data sets (OHSUMED, TD2003, MSLR-WEB10K) using *PRank*, *GBDT*, *RankSVM*, *RankBoost*, *ListNet*, and *AdaRank*, respectively.

From the results in both figures, we can observe surprisingly consistent behavior for the performance degradation of ranking algorithms across different data sets. The performances of ranking algorithms keep quite stable as pNoise is low. When pNoise exceeds a certain point¹ (around 0.5 as indicated in our experiment), the performances drop quickly. Now we can see that, the results w.r.t pNoise are in accordance with the intuitions of degradation and consistency mentioned above, 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. Actually, it is not surprising to find that pNoise is the intrinsic noise which affect the ranking performances for all algorithms including pointwise, pairwise and listwise algorithms. The main reasons lie in that the measures used for evaluation in IR (MAP, NDCG) are mainly based on pairwise orders. Therefore, pNoise can capture the intrinsic factor, and well explain the consistency of performance degradation.

4.3 pNoise versus 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. Since document pair noise usually arises from document label noise, here we explore the variation of pNoise against dNoise across different data sets. We study the relationship between pNoise and dNoise theoretically and practically.

Reminder that pNoise is roughly defined as the proportion of the inverse pair and new coming pair to all the pairs, with mathematical descriptions as follows.

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

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

¹ 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.

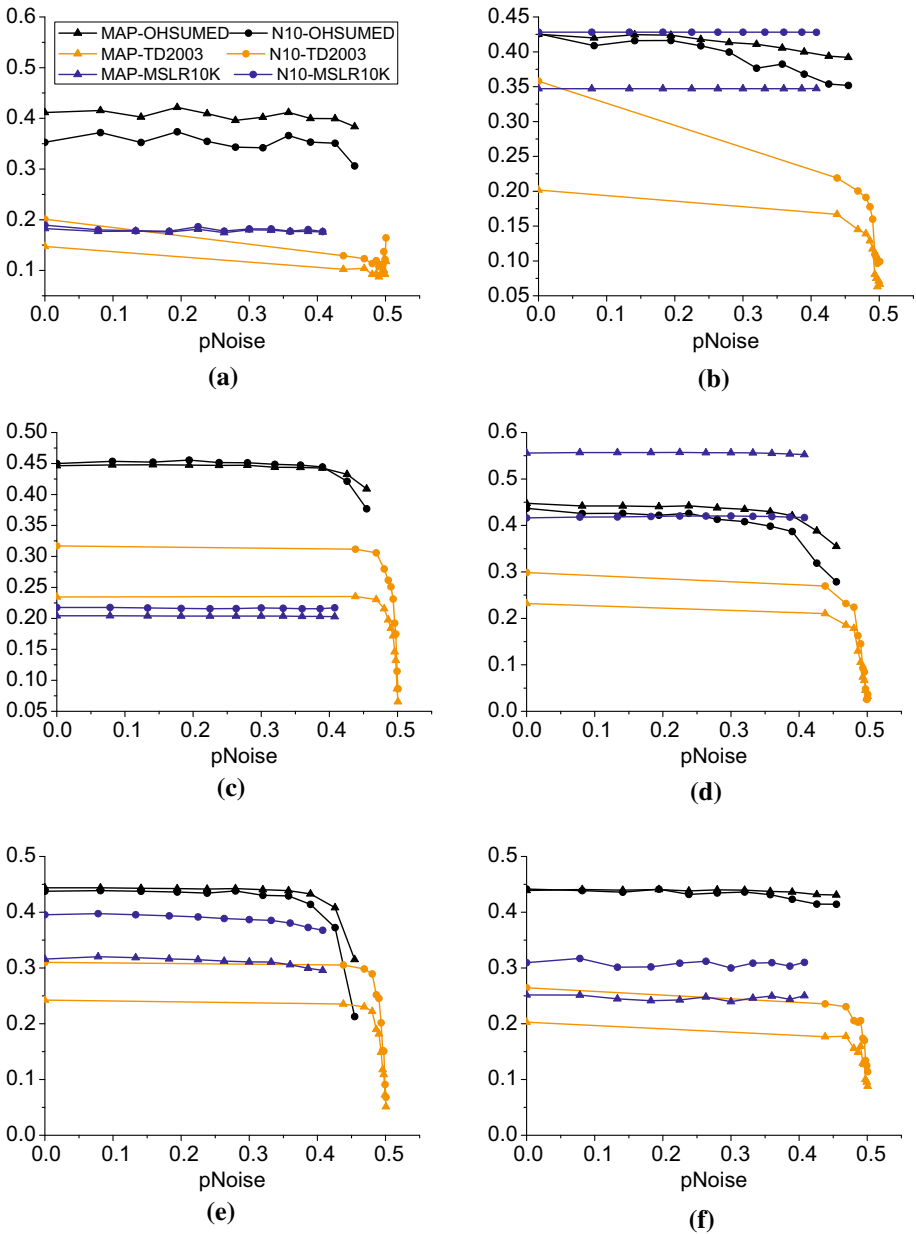


Fig. 4 Performance evaluation in terms of MAP and NDCG@10 against pNoise (uniform noise profile) with different algorithms on OHSUMED, TD2003, and MSLR-WEB10K. **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

Given the labeling strategy, the detailed computation of pNoise can be further derived from the original equation. Considering C -grades relevance levels, there are n_l documents labeled with $l \in \{0, 1, \dots, C - 1\}$ among n documents per query, and the

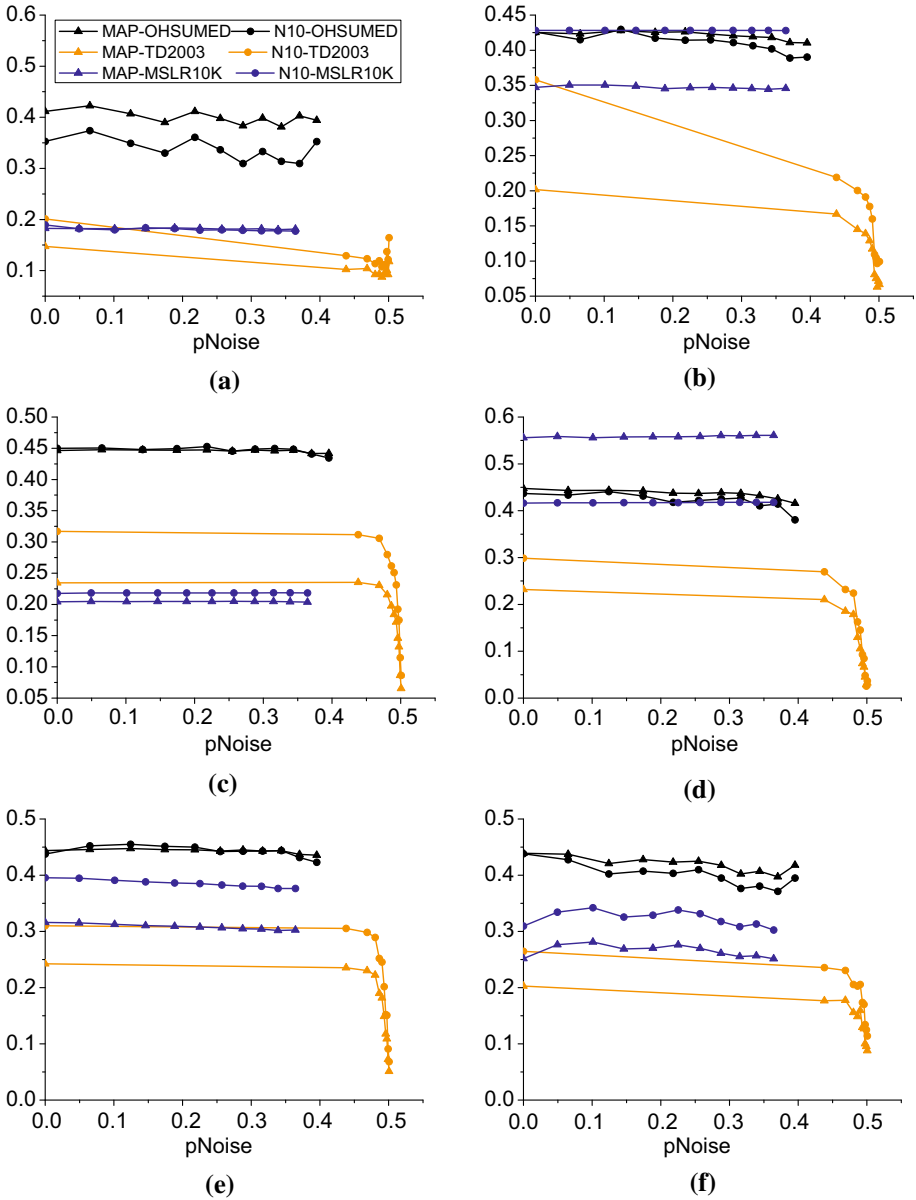


Fig. 5 Performance evaluation in terms of MAP and NDCG@10 against pNoise (non-uniform noise profile) with different algorithms on OHSUMED, TD2003, and MSLR-WEB10K. **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

proportion of documents labeled with l is $r_l = \frac{m_l}{n}$. With γ dNoise injected for both uniform and non-uniform noise profile, the pNoise can be calculated precisely as

$$pNoise(\{r_l\}_{l=0}^{C-1}, \gamma) = \frac{\sum_{l=0}^1 \sum_{j=l}^1 D(l, j) r_l r_j}{\sum_{l=0}^1 \sum_{j=l}^1 A(l, j) r_l r_j}, \tag{1}$$

where A and D are upper triangle coefficient matrices, with each $D(l, j), l \leq j$ stands for the the probability that original pair generated by label l and j become inverse or new coming pair (times 1/2 in this case) after noise injection, and each $A(l, j), l \leq j$ stands for the probability that original pair of documents labeled l and j become a real pair (with different labels) after noise injection.

Taking $C = 2$ with uniform noise as an example, $D(0, 0)$ stands for the probability that original pair generated by label 0 and 0 become inverse or new coming pair (times 1/2 in this case). Since pairs with label 0 and 0 cannot become inverse since they are not real pairs originally, therefore we only calculate the probability that they become new coming pair. With dNoise γ , the probability that they become new pair is $\gamma(1 - \gamma)$, thus we get $D(0, 0) = \frac{1}{2}\gamma(1 - \gamma)$. Easily we can get that $D(1, 1)$ is also equal to $\frac{1}{2}\gamma(1 - \gamma)$. While for $D(0, 1)$, since the original pairs generated by label 0 and 1 are real pairs, the probability that original pair become new pair is 0, therefore $D(0, 1)$ equal to the probability that original pair generated by label 0 and 1 become inverse. When dNoise is γ , we can get that $D(0, 1) = \gamma^2$. Similarly, we can get the value of each $A(l, j)$.

Here we take $C = 2, 3, 5$ for example to show the two matrices of A and D .

– $C = 2$.

$$D = \begin{pmatrix} \frac{1}{2}(\gamma - \gamma^2) & \gamma^2 \\ & \frac{1}{2}(\gamma - \gamma^2) \end{pmatrix}, A = \begin{pmatrix} \gamma - \gamma^2 & 2\gamma^2 - 2\gamma + 1 \\ & \gamma - \gamma^2 \end{pmatrix} \tag{2}$$

– $C = 3$.

(1) Uniform noise profile.

$$D = \begin{pmatrix} \frac{1}{2}(\gamma - \frac{3}{4}\gamma^2) & \frac{\gamma}{2} & \frac{3\gamma^2}{4} \\ & \frac{1}{2}(\gamma - \frac{3}{4}\gamma^2) & \frac{\gamma^2}{2} \\ & & \frac{1}{2}(\gamma - \frac{3}{4}\gamma^2) \end{pmatrix}, \tag{3}$$

$$A = \begin{pmatrix} \gamma - \frac{3}{4}\gamma^2 & \gamma - \frac{3}{4}\gamma^2 - 1 & \gamma - \frac{3}{4}\gamma^2 - 1 \\ & \gamma - \frac{3}{4}\gamma^2 & \gamma - \frac{3}{4}\gamma^2 - 1 \\ & & \gamma - \frac{3}{4}\gamma^2 \end{pmatrix}$$

(2) Non-uniform noise profile.

$$D = \begin{pmatrix} \frac{1}{2}(\gamma - \frac{7}{9}\gamma^2) & \frac{\gamma^2}{6} + \frac{\gamma}{3} & \frac{5\gamma^2}{9} \\ & \frac{1}{2}(\gamma - \frac{3}{4}\gamma^2) & \frac{\gamma^2}{2} \\ & & \frac{1}{2}(\gamma - \frac{7}{9}\gamma^2) \end{pmatrix}, \quad (4)$$

$$A = \begin{pmatrix} \gamma - \frac{7}{9}\gamma^2 & \gamma^2 - \frac{7}{6}\gamma + 1 & \frac{2}{9}\gamma^2 - \frac{2}{3}\gamma + 1 \\ & \gamma - \frac{3}{4}\gamma^2 & \gamma^2 - \frac{7}{6}\gamma + 1 \\ & & \gamma - \frac{7}{9}\gamma^2 \end{pmatrix}$$

– C = 5.

(1) Uniform noise profile.

$$D = \begin{pmatrix} \frac{\gamma}{2} - \frac{5\gamma^2}{16} & \frac{3\gamma}{4} - \frac{5\gamma^2}{16} & \frac{\gamma}{2} & \frac{5\gamma^2}{16} + \frac{\gamma}{4} & \frac{5\gamma^2}{8} \\ & \frac{\gamma}{2} - \frac{5\gamma^2}{16} & \frac{7\gamma^2}{16} - \frac{\gamma}{4} & \frac{\gamma}{2} & \frac{5\gamma^2}{16} + \frac{\gamma}{4} \\ & & \frac{\gamma}{2} - \frac{5\gamma^2}{16} & \frac{\gamma}{4} - \frac{5\gamma^2}{16} & \frac{\gamma}{2} \\ & & & \frac{\gamma}{2} - \frac{5\gamma^2}{16} & \frac{3\gamma}{4} - \frac{5\gamma^2}{16} \\ & & & & \frac{\gamma}{2} - \frac{5\gamma^2}{16} \end{pmatrix},$$

$$A = \begin{pmatrix} \gamma - \frac{5\gamma^2}{8} & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 \\ & \gamma - \frac{5\gamma^2}{8} & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 \\ & & \gamma - \frac{5\gamma^2}{8} & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 \\ & & & \gamma - \frac{5\gamma^2}{8} & \frac{5\gamma^2}{16} - \frac{\gamma}{2} + 1 \\ & & & & \gamma - \frac{5\gamma^2}{8} \end{pmatrix} \quad (5)$$

(2) Non-uniform noise profile.

$$D = \begin{pmatrix} \frac{\gamma}{2} - \frac{83\gamma^2}{250} & \frac{13\gamma}{25} - \frac{4\gamma^2}{85} & \frac{7\gamma}{25} + \frac{\gamma^2}{10} & \frac{3\gamma}{25} + \frac{16\gamma^2}{85} & \frac{41\gamma^2}{125} \\ & \frac{\gamma}{2} - \frac{11\gamma^2}{34} & \frac{47\gamma}{102} - \frac{5\gamma^2}{51} & \frac{4\gamma}{17} + \frac{\gamma^2}{17} & \frac{3\gamma}{25} + \frac{16\gamma^2}{85} \\ & & \frac{\gamma}{2} - \frac{23\gamma^2}{72} & \frac{47\gamma}{102} - \frac{5\gamma^2}{51} & \frac{7\gamma}{25} + \frac{\gamma^2}{10} \\ & & & \frac{\gamma}{2} - \frac{11\gamma^2}{34} & \frac{13\gamma}{25} - \frac{4\gamma^2}{85} \\ & & & & \frac{\gamma}{2} - \frac{83\gamma^2}{250} \end{pmatrix},$$

$$A = \begin{pmatrix} \gamma - \frac{83\gamma^2}{125} & -\frac{71\gamma^2}{425} - \frac{354\gamma}{425} + 1 & \frac{13\gamma^2}{75} - \frac{61\gamma}{150} + 1 & \frac{28\gamma^2}{425} - \frac{118\gamma}{425} + 1 & \frac{18\gamma^2}{625} - \frac{6\gamma}{25} + 1 \\ & \gamma - \frac{11\gamma^2}{17} & \frac{28\gamma^2}{51} - \frac{35\gamma}{51} + 1 & \frac{42\gamma^2}{289} - \frac{6\gamma}{17} + 1 & \frac{28\gamma^2}{425} - \frac{118\gamma}{425} + 1 \\ & & \gamma - \frac{23\gamma^2}{36} & \frac{28\gamma^2}{51} - \frac{35\gamma}{51} + 1 & \frac{13\gamma^2}{75} - \frac{61\gamma}{150} + 1 \\ & & & \gamma - \frac{11\gamma^2}{17} & \frac{299\gamma^2}{425} - \frac{354\gamma}{425} + 1 \\ & & & & \gamma - \frac{83\gamma^2}{125} \end{pmatrix} \tag{6}$$

A curve without markers in Fig. 6 is theoretical pNoise variation with dNoise on one data set using one noise profile, while the curve of the same color with markers is the corresponding experimental results on this data set using this noise profile. Both theoretical and experimental curves are consistent on three data sets using both uniform and non-uniform noise profile, so it is possible to analyze either of them.

Figure 6 shows the average pNoise with respect to dNoise over the three data sets under our previous fivefold cross validation experimental settings. From the results we can see that the increase of pNoise with respect to dNoise varies largely over different data sets. This actually well explains our basic observations. As we can see, given a small dNoise (e.g. 0.1) on TD2003, the pNoise will reach a very high value (> 0.4). According to red curves in Fig. 4a–f, such a pNoise has already reached the turning point of the performance curve on TD2003. 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. Comparing the two data sets, the increase of pNoise along with dNoise is faster on OHSUMED. This is also consistent with our basic observations where the performance degradation of the same algorithm on OHSUMED is quicker than that on MSLR-WEB10K. In fact, even when dNoise reaches 0.5 on MSLR-WEB10K, the corresponding pNoise is still below a threshold (In this case, the turning point is 0.4 or so), which means the pNoise has not reached the turning point of the performance curve on MSLR-WEB10K according to blue curves in Fig. 4a–f. This explains why the performance of a ranking algorithm is hardly affected with respect to dNoise on MSLR-WEB10K in our basic observations. Similar results can be found in Fig. 5a–f.

5 How to make data robust

From the above section, we know that pNoise captures the true noise of learning to rank algorithms. Higher pNoise leads to lower ranking performance. Furthermore, the performance degradation of the same ranking algorithm varies largely across data sets due to the fact that the increase of pNoise with respect to dNoise varies largely over different data sets. Without loss of generality, faster increase of pNoise with respect to dNoise corresponds to quicker performance degradation of the ranking algorithm. Now the remaining question is what factors determine the variation of pNoise with respect to dNoise from the data aspect. By answering this question, we are able to identify the inherent characteristics that are related to the data robustness to label noise in learning to rank.

Considering the differences among the three data sets and how pNoise arises from dNoise, we identify that two inherent characteristics of the training data may have impact on the variation of pNoise with respect to dNoise, namely relevance levels and label balance. In this section, we aim to verify whether these two data characteristics affect the variation of pNoise against dNoise, and if yes, how?

Note that in this section, we conducted our data analysis experiments by fixing the size of training data. There are two major reasons: (1) The labeling process is expensive and time consuming. Therefore, it is reasonable to guide the labeling process under some targeted labeling effort (i.e. fixed labeling size). (2) If one can afford more labeled data, it is well believed that one may obtain better performance and robustness. Therefore, here we aim to identify inherent characteristics related to data robustness beyond the data size.

5.1 Relevance levels

Relevance levels refer to the graded relevance judgments one adopts for labeling. For example, TD2003 takes 2-graded relevance judgments while MSLR-WEB10K takes 5-graded relevance judgments. We first investigate the influence of relevance levels on pNoise theoretically for each data set with different relevance levels, then experimental results are shown to verify similar influence on data robustness.

5.1.1 Relevance levels versus pNoise

According to the detailed computation equations of pNoise from Eqs. (1, 2, 3, 4, 5, 6), it is easy to obtain a complete picture of any data set with any grade relevance judgments, i.e. 2, 3, 5. Here we take TD2003, OHSUMED and MSLRWEB10K for example.

(1) TD2003

- C = 2 (true): $r_0 = 0.992, r_1 = 0.008$;
- C = 3 (simulation): $r_0 = 0.792, r_1 = 0.2, r_2 = 0.008$;
- C = 5 (simulation): $r_0 = 0.792, r_1 = 0.1, r_2 = 0.1, r_3 = 0.004, r_4 = 0.004$.

(2) OHSUMED

- C = 2 (simulation): $r_0 = 0.7, r_1 = 0.3$;
- C = 3 (true): $r_0 = 0.7, r_1 = 0.16, r_2 = 0.14$;
- C = 5 (simulation): $r_0 = 0.5, r_1 = 0.2, r_2 = 0.016, r_3 = 0.07, r_4 = 0.07$.

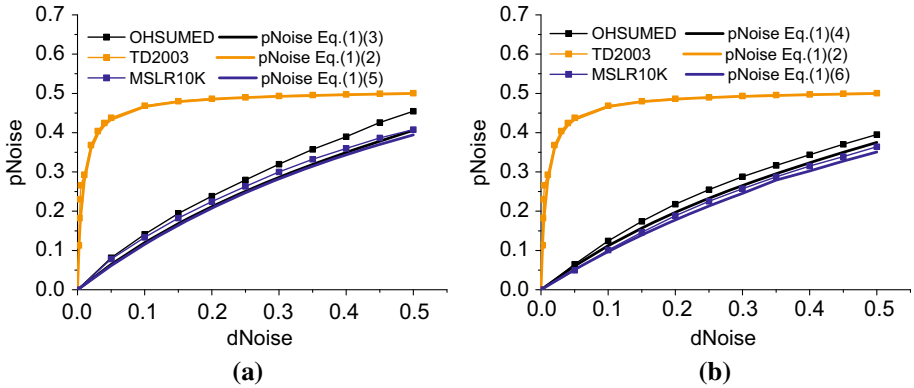


Fig. 6 Average pNoise with respect to dNoise over the three data sets. **a** Uniform noise profile; **b** Non-uniform noise profile

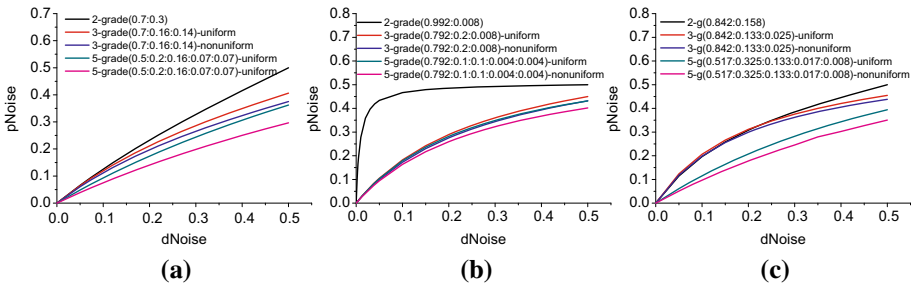


Fig. 7 Theoretical pNoise varies with dNoise over the data set with different relevance levels. **a** OHSUMED; **b** TD2003; **c** MSLRWEB10K

(3) MSLRWEB10K

- C = 2 (simulation): $r_0 = 0.842, r_1 = 0.158$;
- C = 3 (simulation): $r_0 = 0.842, r_1 = 0.133, r_2 = 0.025$;
- C = 5 (true): $r_0 = 0.517, r_1 = 0.325, r_2 = 0.133, r_3 = 0.017, r_4 = 0.008$.

Therefore the comparison results on one data set with different relevance levels is shown in Fig. 7.

In Fig. 7 it is obvious that pNoise decrease more quickly with dNoise when there are more relevance levels. This suggests we need more relevance levels in the labeling process to make pNoise lower directly.

5.1.2 Relevance levels versus robustness

Here we conduct experiments to investigate whether relevance levels has an effect on the variation of pNoise against dNoise. For this purpose, we need to obtain identical training data with fixed size but different relevance levels. The basic idea is that we turn the training data with multi-graded relevance judgments into that with lower graded ones, while the test data is fixed.

Specifically, both OHSUMED and MSLR-WEB10K data sets are leveraged in our experiments because they adopt more than 2-graded relevance judgments (i.e. OHSUMED takes 3-graded judgments while MSLR-WEB10K takes 5-graded). We then turn both multi-graded relevance judgments on the training data into 2-graded ones in the following way. On OHSUMED, we keep the relevance judgment 0 the same, while merge the relevance judgments 1 and 2 into a single grade 1. We denote the corresponding OHSUMED data set with 2-graded relevance judgments as OHSUMED2G. Similarly, on MSLR-WEB10K, we merge the relevance judgments 0 and 1 into a single grade 0, and merge the relevance judgments 2, 3, and 4 into a single grade 1. We denote the corresponding MSLR-WEB10K data set with 2-graded relevance judgments as MSLR-WEB10K2G. Note there may be other ways to turn the 5-graded relevance judgments into lower graded ones (e.g. 3-graded ones). Here we turn it into 2-graded ones to make the difference on relevance levels as large as possible.

We then apply the noise injection method on both the original data sets and the new lower graded data sets, and compare the variation of pNoise against dNoise. The results are depicted in Fig. 8, where Fig. 8a shows the comparison between OHSUMED and OHSUMED2G, and Fig. 8b shows the comparison between MSLR-WEB10K and MSLR-WEB10K2G. From both the results we can see, when there are less relevance levels, the increase of pNoise with respect to dNoise will be faster. For example, given the same dNoise 0.3, the corresponding pNoise is about 0.3 on MSLR-WEB10K, but reaches about 0.4 on MSLR-WEB10K2G. From previous analysis we know that, faster increase of pNoise with respect to dNoise corresponds to quicker performance degradation of the learning to rank algorithm. Therefore, we may expect that these lower graded data sets would be more sensitive to label noise. We verified our guess by evaluating the performance of trained ranking algorithms with respect to dNoise based on the above two groups of data sets. Similarly, a number of experiments were conducted with dNoise changing from 0 to 0.5 with a step of 0.05, and the performances of trained ranking algorithms were evaluated based on the corresponding noise free test set. The 5-fold cross validation results are depicted in Figs. 9 and 10, where Fig. 9 shows the results of the four ranking algorithms on OHSUMED and OHSUMED2G evaluated by MAP and NDCG@10,

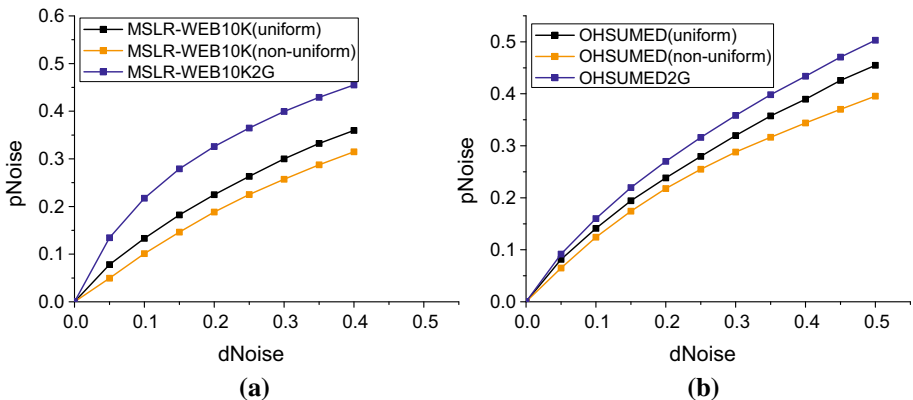


Fig. 8 Average pNoise with respect to dNoise over the data sets with different relevance levels, **a** OHSUMED (3-graded) and OHSUMED2G (2-graded), and **b** MSLR-WEB10K (5-graded) and MSLR-WEB10K2G (2-graded)

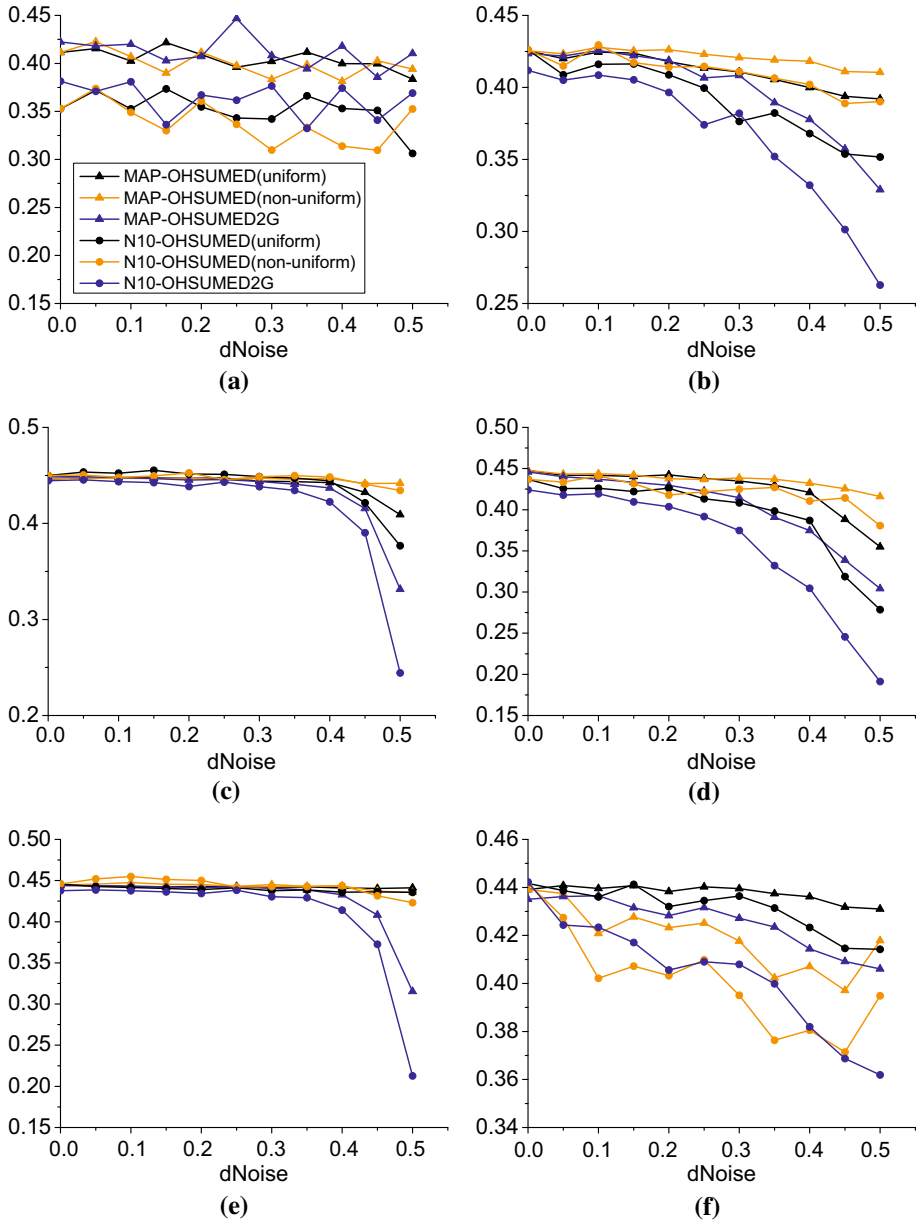


Fig. 9 Performance evaluation against dNoise with different relevance levels on OHSUMED (3-graded) and OHSUMED2G (2-graded). **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

respectively, and Fig. 10 shows the corresponding results on MSLR-WEB10K and MSLR-WEB10K2G. Not surprisingly, under both evaluation measures, the performance of a same ranking algorithm drops more quickly on the lower graded data set than on the corresponding higher graded one. Take RankBoost as an example, given the same dNoise 0.4,

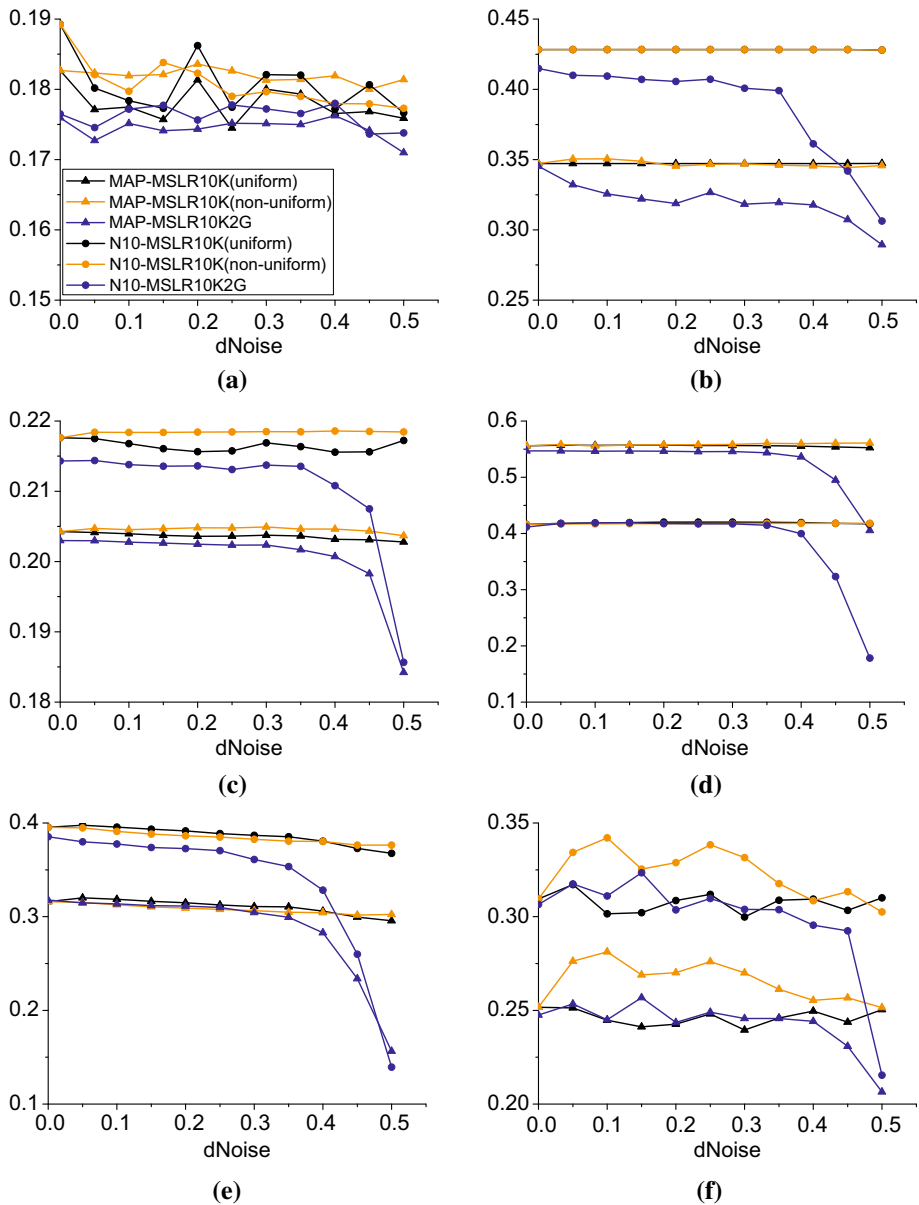


Fig. 10 Performance evaluation against dNoise (both uniform and non-uniform) with different relevance levels on MSLR-WEB10K (5-graded) and MSLR-WEB10K2G (2-graded). **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

the relative drop of MAP compared with noise free case is about 6% for the model trained on OHSUMED, while the relative drop of MAP reaches about 16% for the model trained on OHSUMED2G, as shown in Fig. 9d. Similarly, the MAP of RankBoost is hardly affected for the model trained on MSLR-WEB10K, but drops quickly when dNoise reaches 0.4 for the model trained on MSLR-WEB10K2G, as shown in Fig. 10d.

From the above results, we identify that relevance levels have crucial impact on the variation of pNoise with respect to dNoise. For a fixed size of training data, with more relevance levels, pNoise can be less affected by dNoise. Correspondingly we can make the training data more robust to label noise.

5.2 Label balance

The above section identifies the relationship between relevance levels and data robustness, now we turn to the label balance factor. Label balance refers to the balance among labeled data of different relevance levels, which can be reflected by the proportions of labeled data over different relevance levels. For example, the average proportions of labeled data over different relevance levels are about 99.2:0.8 on TD2003 (2-graded), and about 70:16:14 on OHSUMED (3-graded), as shown in Table 1. We first investigate the influence of label balance on pNoise theoretically for each data set with different label balances, then experimental results are shown to verify similar influence on data robustness.

5.2.1 Label balance versus pNoise

According to the detailed computation equations of pNoise from Eqs. (1, 2, 3, 4, 5, 6), it is easy to obtain the complete picture of label balance's influence on pNoise. Here we take TD2003, OHSUMED and MSLRWEB10K for example.

(1) TD2003(2-grade)

- R1 (true): $r_0 = 0.992, r_1 = 0.008$;
- R2 (simulation): $r_0 = 0.7, r_1 = 0.3$; (more balanced than R1)
- Rbest (simulation): $r_0 = 0.5, r_1 = 0.5$.

(2) OHSUMED (3-grade)

- R1 (simulation): $r_0 = 0.9, r_1 = 0.05, r_2 = 0.05$;
- R2 (true): $r_0 = 0.7, r_1 = 0.16, r_2 = 0.14$; (more balanced than R1)
- Rbest(simulation): $r_0 = 1/3, r_1 = 1/3, r_2 = 1/3$.

(3) MSLRWEB10K(5-grade)

- R1 (simulation): $r_0 = 0.89, r_1 = 0.05, r_2 = 0.05, r_3 = 0.005, r_4 = 0.005$;
- R2 (true): $r_0 = 0.517, r_1 = 0.325, r_2 = 0.133, r_3 = 0.017, r_4 = 0.008$; (more balanced than R1)
- Rbest (simulation): $r_0 = 0.2, r_1 = 0.2, r_2 = 0.2, r_3 = 0.2, r_4 = 0.2$.

Therefore the comparison results on one data set with different relevance levels is shown in Fig. 11.

In Fig. 11 it is obvious that pNoise decreases when the proportion becomes more balanced from R1, R2 to Rbest given dNoise. This suggests we need to balance the document proportion of each relevance level better in the labeling process to make pNoise lower directly.

5.2.2 Label balance versus robustness

Here we conduct experiments to investigate the relationship between label balance and data robustness. Similarly, we start from considering how label balance affects the

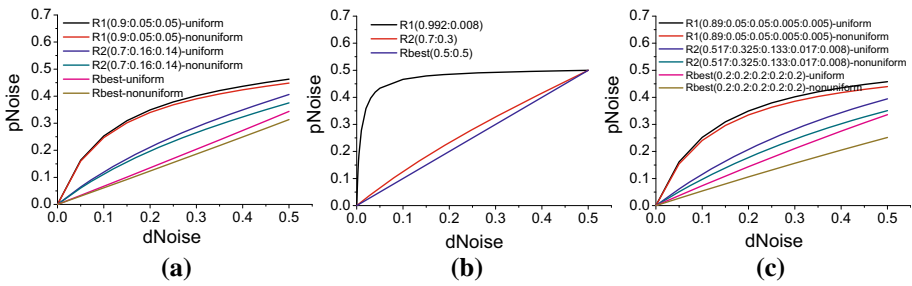


Fig. 11 Theoretical pNoise varies with dNoise over the data set with different label balances. **a** OHSUMED; **b** TD2003; **c** MSLRWEB10K

variation of pNoise against dNoise. For this purpose, we need to obtain training data with a fixed size but different label balance. The basic idea is that we can sample training data with fixed size under different proportions over relevance levels from the original data set, while the test data is fixed.

Specifically, we leveraged two data sets both with 2-graded relevance judgments in our experiments. One is TD2003 and the other is MSLR-WEB10K2G which is a 2-graded version of MSLR-WEB10K generated in the above section. We choose the two data sets due to the following considerations. (1) We aim to sample different proportions of labeled data over relevance levels with a fixed total size, and the difference should be sufficient for better comparison. The small data set (e.g. OHSUMED) and the original multi-graded data set (e.g. MSLR-WEB10K) lose the flexibility for such sampling. (2) We would like to demonstrate the effect of label balance. Therefore, with 2-graded relevance judgments, one can capture what corresponds to label balance more easily.

On TD2003, we sampled two training data sets, denoted as TD2003P1 and TD2003P7. Both data sets have exactly the same number of labeled documents (i.e. 16 labeled documents) for each query, but different proportions over the 2 relevance levels. For TD2003P1, the ratio between relevant and irrelevant labeled data for each query is 1:1, while for TD2003P7, the ratio is 1:7 for each query. Note that these ratios are chosen according to the maximum number of relevant documents available per query. In order to avoid possible bias in random sampling, such sampling process are repeated for 5 times under each ratio. In other words, both TD2003P1 and TD2003P7 have 5 random sampled sets.

On MSLR-WEB10K2G, we sampled two sets, denoted as MSLR-WEB10K2GP5 and MSLR-WEB10K2GP50, each of which has exactly the same number of labeled documents (i.e. 102 labeled documents) for each query, but different proportions. In MSLR-WEB10K2GP5, the ratio between relevant and irrelevant labeled data for each query is 1:5, while in MSLR-WEB10K2GP50, the ratio is 1:50 for each query. We also obtain 5 samples for each data set.

We apply the noise injection method on all the sampled data sets, and compare the variation of pNoise with respect to dNoise. The results are depicted in Fig. 12, where Fig. 12a shows the comparison between TD2003P1 and TD2003P7, and Fig. 12b shows the comparison between MSLR-WEB10K2GP5 and MSLR-WEB10K2GP50. From both results we can see, when the proportions of labeled data over different relevance levels are less balanced, the increase of pNoise with respect to dNoise will be faster. For example, given the same dNoise (0.1), the corresponding pNoise is about 0.16 on TD2003P1, but

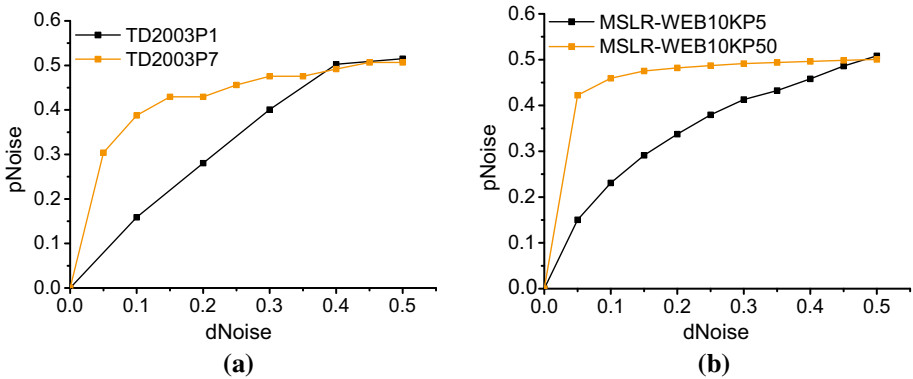


Fig. 12 Average pNoise with respect to dNoise over the data sets with different label balance, **a** TD2003P1 (1:1) and TD2003P7 (1:7), and **b** MSLR-WEB10K2GP5 (1:5) and MSLR-WEB10K2GP50 (1:50). **a** TD2003; **b** MSLR-WEB10K

reaches about 0.39 on TD2003P7. As aforementioned, faster increase of pNoise with respect to dNoise corresponds to quicker performance degradation of the ranking algorithm. Therefore, we may also expect that these less balanced data sets would be more sensitive to label noise.

Similar to what we have done in the previous section, we also evaluate the performance of the trained ranking algorithms with respect to dNoise based on the above four data sets to verify our guess. The evaluation results are depicted in Figs. 13 and 14, where Fig. 13 shows the results of the four ranking algorithms on TD2003P1 and TD2003P7 evaluated by MAP and NDCG@10 respectively, and Fig. 14 shows the corresponding results on MSLR-WEB10K2GP5 and MSLR-WEB10K2GP50. From the results we can see that, under both evaluation measures, the performance of the same ranking algorithm drops more quickly on the less balanced data set as compared with the corresponding more balanced one. Take RankSVM as an example, given the same dNoise 0.2, the relative drop of NDCG@10 as compared with noise free case is about 22 % for the model trained on TD2003P1, while the relative drop of MAP reaches about 55 % for the model trained on TD2003P7, as shown in Fig. 13a. Similarly, for the RankSVM trained on MSLR-WEB10K2GP5, its performance under NDCG@10 keeps quite stable until dNoise exceeds 0.4 as shown in Fig. 14a. However, for the same model trained on MSLR-WEB10K2GP50, its performance under NDCG@10 drops almost linearly with the increase of dNoise.

From the above results, we identify that label balance also has crucial impact on the variation of pNoise with respect to dNoise. For a fixed size of training data, with better label balance, pNoise can be less affected by dNoise. Correspondingly, we can make the training data more robust to label noise.

5.3 Guidelines for robust data construction

The above analysis tells us that for a fixed size of training data, with more relevance levels and better label balance, we can make the training data more robust to label noise. These results actually provide us some valuable guidelines on the labeling strategy in practice.

Firstly, the results on relevance levels provide us some guidelines on the number of grades adopted in labeling. In the dominating multi-graded relevance judgments scenario,

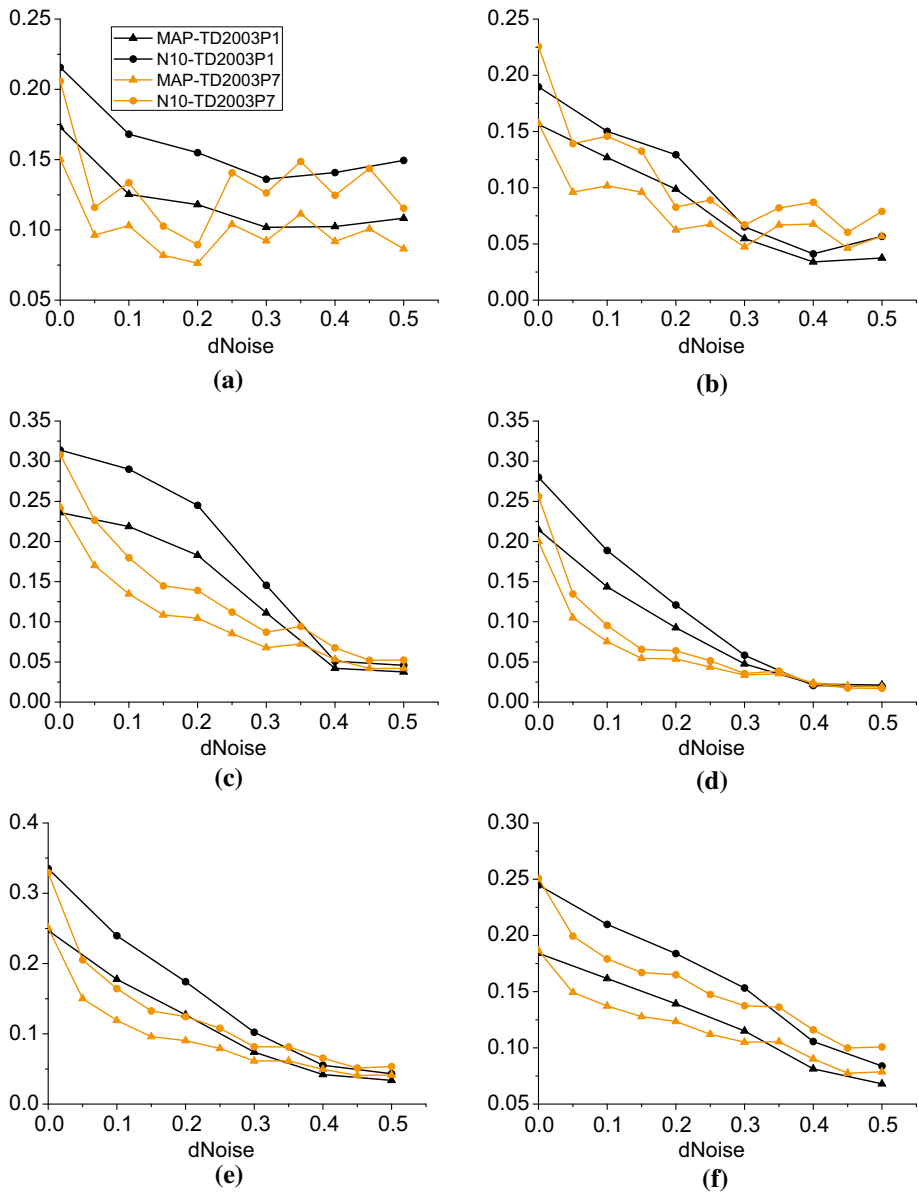


Fig. 13 Performance evaluation against dNoise(both uniform and non-uniform noise profile) with different Label Balance on TD2003P1 (1:1) and TD2003P7 (1:7). **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

2-graded, 3-graded, or 5-graded judgments are usually adopted. It is unclear how many relevance grades are better. Now from the robustness perspective, we show that we can make training data more robust to label noise if we take more relevance levels. However, more relevance levels may lead to more effort on defining a clear relevance judgment specification and more effort on judging for human assessors. Therefore, we cannot

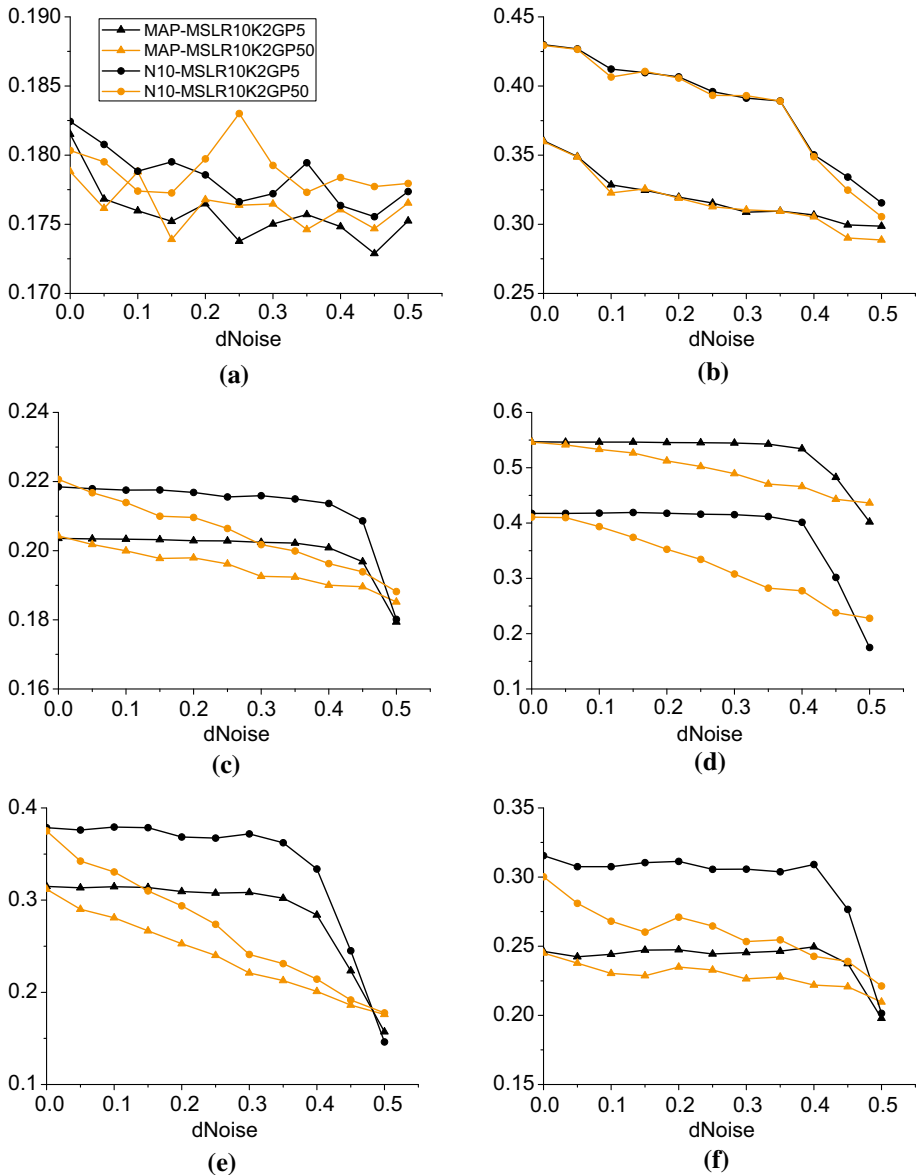


Fig. 14 Performance evaluation against dNoise with different label balance on MSLR-WEB10K2GP5 (1:5) and MSLR-WEB10K2GP50 (1:50). **a** PRank; **b** GBDT; **c** RankSVM; **d** RankBoost; **e** ListNet; **f** AdaRank

leverage arbitrary levels in multi-graded relevance judgments. There is clearly a tradeoff between robustness and label effort (e.g. maybe 5-graded is a good tradeoff), which is worth further study.

Secondly, the results on label balance provide us some guidelines on gradation distribution in labeling. In traditional labeling processes, one usually cares about how many documents to label per query, but seldom addresses the label balance problem. Given the

fact that there are always much more irrelevant documents than relevant ones for a query, if we randomly sample a collection of documents for a query to label, we may very probably obtain an unbalanced data set. One possible way is to keep a certain number of relative relevant documents labeled.

For a multi-graded relevance judgment approach, both principles are not easy for implementation in real application as indicated above. Fortunately, we find that the recently proposed Top- k Labeling strategy (Niu et al. 2012) is a good alternative. Top- k labeling strategy adopts the pairwise preference judgment to generate the top k ordering items from n documents in a manner of HeapSort. The obtained ground-truth from this top- k labeling strategy is a mixture of the total order of the top k items, and the relative preferences between the set of top k items and the set of the rest $n - k$ items, referred as top- k ground-truth. Therefore, we can see that Top- k Labeling strategy naturally satisfies the above two principles.

- (1) Top- k Labeling results convey finer gradations than multi-graded relevance judgments as they can distinguish those documents within one grade (top k documents) in multi-graded relevance judgments (k relevance levels). It is intrinsically a pairwise judgment method. As noted by Carterette et al. (2008), “By collecting preferences directly, some of the noise associated with difficulty in distinguishing between different levels of relevance may be reduced”.
- (2) Top- k Labeling results naturally satisfy the label balance requirement. The top- k ground-truth can be viewed as k relevant documents and the another $n - k$ irrelevant documents, and there are k relevance levels. Therefore it keeps the label balance in k relevance levels since each level has only one document. Someone may argue that the number of irrelevant documents are much more than relevant documents, we admit that it is true, however, this will not influence much to the results since the top k items and the order between them are the key to the problem.

Additionally, the basic judgment in Top- k Labeling is in a pairwise way, which is much easier than multi-graded relevance judgment through user studies in Carterette et al. (2008), Niu et al. (2012) in terms of time cost per judgment. This suggests that pairwise preference judgment methods are prone to be less noisy than multi-graded relevance judgment methods.

To this end, we can see that pairwise preference judgment is more suitable to construct robust training data set for learning to rank, according to our analysis. Here, we use a case study to verify the above guidelines in real labeling process.

5.3.1 Case study

We randomly selected 5 queries out of all 50 queries from the Topic Distillation task of TREC2003 as our query set. For each query, we then randomly sampled 50 documents from its associated documents for judgment. In TREC topics, most of the queries have clear intent in the form of query description. The descriptions for the chosen 5 queries are “local, state organizations and programs”, “polygraphs, polygraph exams”, “shipwrecks”, “cyber crime, internet fraud”, and “literature for children”, respectively.

Here we take the existing binary relevance judgments (“irrelevant”, “relevant”) for all the query-document pairs above as noise-free ground-truth. Three labeling strategies are adopted in our experiments, three-graded relevance judgment method $M1$ (“irrelevant”, “partially relevant”, “relevant”), five-graded relevance judgement method $M2$ (“bad”,

Table 3 Comparison of pNoise among three labeling strategies

Labeling method	Query1	Query2	Query3	Query4	Query5	Average
Three-grade judgement(M1)	0.390	0.451	0.426	0.354	0.278	0.380
Five-grade judgement(M2)	0.376	0.421	0.398	0.322	0.275	0.359
Top-k labeling(M3)	0.253	0.445	0.397	0.303	0.215	0.342

Bold values indicate the best one among the three judgement methods on each query

“fair”, “good”, “excellent”, “perfect”) and top-k labeling method *M3* with *k* set to 10. We asked three annotators to label the queries under these strategies.

To measure the degree of robustness, pNoise are computed between the existing binary noise-free ground truth and the three kinds of noisy labeling results. From Table 3, we can see that: (1) the pNoise generated by *M2* are lower than that by *M1* on all 5 queries, which verifies our first guideline that more relevant levels is better for robust dataset construction; (2) the pNoise generated by *M3* is lower than those by both *M1* and *M2* for all queries except the the second query, which verifies our guidelines that pairwise preference judgement is more suitable to construct robust training data set as compared to multi-graded relevance judgment approach. As for the result on the second query, we explain that the main reason may be due to the limitation of fixed *k* for all the queries in the top-k labeling strategy, considering there is only one relevant document for the the second query in the binary ground truth.

6 Conclusions and future work

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. We further identify two inherent characteristics of the training data, namely relevance levels and label balance, that have crucial impact on the variation of pNoise with respect to dNoise. We show that for a fixed size of training data, with more relevance levels and better label balance, we can make pNoise less affected by dNoise, i.e. the training data more robust to label noise. With these results, we discuss some guidelines on the labeling strategy for constructing robust training data in real scenario. We believe these guidelines would be helpful especially for people who want to apply learning to rank algorithms in real search systems.

As for the future work, the current noise injection methods are somehow simple and we may further improve the injection method for better analysis. In fact, the noise ratio could be different from query to query due to the different difficulty levels of queries. Besides, from the guidelines discussed above, one interesting direction would be to design novel robust data construction methods. For example, we may improve the existing pairwise preference labeling technique (as it is more finer graded and more balance) by effectively selecting document pairs for labeling, so that we can achieve lower complexity and higher robustness.

Acknowledgments This research work was funded by the 973 Program of China under Grants No. 2014CB340401, No. 2012CB316303, the 863 Program of China under Grants No. 2014AA015204, the National Natural Science of China under Grants No. 61472401, No. 61203298, and the National Key Technology R&D Program of China under Grants NO. 2012BAH46B04.

References

- Abellán, J., & Masegosa, A.R. (2009). An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. In: Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU '09, (pp. 446–456). Berlin: Springer.
- Abellán, J., Masegosa, A.R. (2010). Bagging decision trees on data sets with classification noise. In: FoIKS '10 (pp. 248–265).
- Aslam, J.A., Kanoulas, E., Pavlu, V., Savev, S., & Yilmaz, E. (2009). Document selection methodologies for efficient and effective learning-to-rank. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09, (pp. 468–475). New York: ACM.
- Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York: Addison-Wesley.
- Bailey, P., Craswell, N., Soboroff, I., Thomas, P., de Vries, A.P., & Yilmaz, E. (2008). Relevance assessment: Are judges exchangeable and does it matter. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '08 (pp. 667–674). New York: ACM.
- Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., et al. (2005). Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning, ICML '05 (pp. 89–96). New York: ACM.
- Burgin, R. (1992). Variations in relevance judgments and the evaluation of retrieval performance. *Inf. Process. Manage.* 28, pp. 619–627. doi:10.1016/0306-4573(92)90031-T. <http://dl.acm.org/citation.cfm?id=144934.144946>.
- Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th international conference on Machine learning, ICML '07 (pp. 129–136). New York: ACM. doi:10.1145/1273496.1273513.
- Carterette, B., Bennett, P.N., Chickering, D.M., & Dumais, S.T. (2008). Here or there: Preference judgments for relevance. In: Proceedings of the IR research, 30th European conference on Advances in information retrieval, ECIR'08 (pp. 16–27). Berlin: Springer.
- Carterette, B., Soboroff, I. (2010). The effect of assessor error on ir system evaluation. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10 (pp. 539–546). New York: ACM. doi:10.1145/1835449.1835540.
- Carvalho, V., Elsas, J., Cohen, W. (2008). A meta-learning approach for robust rank learning. In: Proceedings of SIGIR 2008 LR4IR—Workshop on learning to rank for information retrieval.
- Crammer, K., Singer, Y. (2001). Pranking with ranking. In: *Advances in neural information processing systems*, 14 (pp. 641–647). Cambridge: MIT Press.
- Crammer, K., Kulesza, A., & Dredze, M. (2009). Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems*, 22, 414–422.
- Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4, 933–969.
- Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 1189–1232.
- Geng, X., Qin, T., Liu, T. Y., Cheng, X. Q., & Li, H. (2011). Selecting optimal training data for learning to rank. *Information Processing and Management*, 47, 730–741.
- Jain, V., & Varma, M. (2011). Learning to re-rank: Query-dependent image re-ranking using click data. In: Proceedings of the 20th international conference on World wide web, WWW '11 (pp. 277–286). New York: ACM.
- Järvelin, K., & Kekäläinen, J. (2000). Ir evaluation methods for retrieving highly relevant documents. In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00 (pp. 41–48). New York: ACM.
- Jeatrakul, P., Wong, K. W., & Fung, C. C. (2010). Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14, 297–302.
- Joachims, T. (2002). Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '02 (pp. 133–142). New York: ACM.
- Kanoulas, E., Savev, S., Metrikov, P., Pavlu, V., & Aslam, J. (2011). A large-scale study of the effect of training set characteristics over learning-to-rank algorithms. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information, SIGIR '11 (pp. 1243–1244). New York: ACM.

- Kazai, G., Craswell, N., Yilmaz, E., Tahaghoghi, S. (2012). An analysis of systematic judging errors in information retrieval. In: Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12 (pp. 105–114). New York: ACM. doi:[10.1145/2396761.2396779](https://doi.org/10.1145/2396761.2396779).
- Kumar, A., & Lease, M. (2011). Learning to rank from a noisy crowd. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information, SIGIR '11 (pp. 1221–1222). New York: ACM.
- Liu, T.Y. (2011). Introduction. *Learning to rank for information retrieval, 978-3-642-14267-3, chap. 1* (pp. 3–30). Berlin: Springer. Microsoft Research Asia.
- Liu, W., Jiang, Y., Luo, J., Chang, S.F. (2011). Noise resistant graph ranking for improved web image search. In: Proceeding of Computer Vision and Pattern Recognition.
- Macdonald, C., Santos, R., & Ounis, I. (2013). The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5), 584–628. doi:[10.1007/s10791-012-9209-9](https://doi.org/10.1007/s10791-012-9209-9).
- Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33, 275–306.
- Niu, S., Guo, J., Lan, Y., Cheng, X. (2012). Top-k learning to rank: Labeling, ranking and evaluation. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '12 (pp. 751–760). New York: ACM. doi:[10.1145/2348283.2348384](https://doi.org/10.1145/2348283.2348384).
- Qin, T., Liu, T. Y., Xu, J., & Li, H. (2010). Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13, 346–374.
- Rebbapragada, U., & Brodley, C.E. (2007). Class noise mitigation through instance weighting. In: Proceedings of the 18th European conference on Machine Learning, ECML '07 (pp. 708–715). Berlin: Springer.
- Scholer, F., Turpin, A., & Sanderson, M. (2011). Quantifying test collection quality based on the consistency of relevance judgements. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information, SIGIR '11 (pp. 1063–1072). New York: ACM.
- Sheng, V.S., Provost, F., & Ipeirotis, P.G. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08 (pp. 614–622). New York: ACM.
- Taylor, M., Guiver, J., Robertson, S., Minka, T. (2008). Sofrank: Optimizing non-smooth rank metrics. In: Proceedings of the international conference on Web search and web data mining, WSDM '08 (pp. 77–86). New York: ACM. doi:[10.1145/1341531.1341544](https://doi.org/10.1145/1341531.1341544).
- Verbaeten, S., & Van Assche, A. (2003). Ensemble methods for noise elimination in classification problems. In: Proceedings of the 4th international conference on Multiple classifier systems, MCS'03 (pp. 317–325). Berlin: Springer.
- Voorhees, E.M. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '98 (pp. 315–323). New York: ACM.
- Vuurens, J., De Vries, A.P., Eickho, C. (2011). How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In: Proceedings of SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval.
- Xu, J., Chen, C., Xu, G., Li, H., & Abib, E.R.T. (2010). Improving quality of training data for learning to rank using click-through data. In: Proceedings of the third ACM international conference on Web search and data mining, WSDM '10 (pp. 171–180). New York: ACM.
- Xu, J., Li, H. (2007). Adarank: A boosting algorithm for information retrieval. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07 (pp. 391–398). ACM: New York. doi:[10.1145/1277741.1277809](https://doi.org/10.1145/1277741.1277809).
- Yang, H., Mityagin, A., Svore, K.M., & Markov, S. (2010). Collecting high quality overlapping labels at low cost. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '10 (pp. 459–466). New York: ACM.
- Zhu, X., & Wu, X. (2003). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22, 177–210.