Learning to Rank under Multiple Annotators

Ou Wu, Weiming Hu, Jun Gao
NLPR, Institute of Automation, Chinese Academy of Sciences,
{wuou,wmhu,jgao}@nlpr.ia.ac.cn

Abstract

Learning to rank has received great attention in recent years as it plays a crucial role in information retrieval. The existing concept of learning to rank assumes that each training sample is associated with an instance and a reliable label. However, in practice, this assumption does not necessarily hold true. This study focuses on the learning to rank when each training instance is labeled by multiple annotators that may be unreliable. In such a scenario, no accurate labels can be obtained. This study proposes two learning approaches. One is to simply estimate the ground truth first and then to learn a ranking model with it. The second approach is a maximum likelihood learning approach which estimates the ground truth and learns the ranking model iteratively. The two approaches have been tested on both synthetic and real-world data. The results reveal that the maximum likelihood approach outperforms the first approach significantly and is comparable of achieving results with the learning model considering reliable labels. Further more, both the approaches have been applied for ranking the Web visual clutter.

1 Introduction

Learning to rank is a relatively new research area which has emerged rapidly in the past decade. It plays a critical role in information retrieval. In a problem related to learning to rank, an instance is a set of objects and a label is a sorting applied over the instance. Learning to rank aims to construct a ranking model from training data. In the current scenario, each label is assumed to be objective and reliable. This assumption works well and is also used in other conventionally supervised settings such as classification. Recently, many studies have highlighted that for many real-world tasks, it may not be possible, or may be too expensive, to produce the accurate training labels. Instead, multiple (possibly subjective or noisy) labels can be provided by various experts or annotators. For example, the Amazon Mechanical Turk (AMT) allows the requesters to hire users from all over the world to perform data labeling. Any AMT user can opt for the labeling tasks of the user's own choices. Hence, this makes it easy and fast for an AMT requester to hire multiple labelers. As the AMT users have a little control, there is no guarantee of objective and accurate labels. Thus, learning under multiple annotators deserves a deep research.

A number of studies were carried out in the past to deal with the setting involving multiple annotators. One of the early works [Smyth *et al.*, 1995] that was proposed involved the estimation of the ground truth first and then use the estimated ground truth to learn a model. In 2010, a probabilistic framework was presented [Raykar *et al.*, 2010] to address the classification, regression and ordinal regression algorithms with multiple annotators. The probabilistic framework was based a simple yet reasonable assumption that a label by an annotator depends on both the true label and the reliability of the annotator. Their experimental results show that their framework is superior to the work by [Smyth *et al.*, 1995]. Donmez and Garnonell [Donmez and Garbonell, 2010] investigated the case when the reliability of annotators is time-varying and developed a sequential Bayesian estimation framework. There are some other related works that focus on somewhat different settings [Chen et al., 2010; Yan *et al.*, 2010].

The above studies paid little attention to learning to rank under multiple annotator setting. To complement the existing studies, this paper investigates the algorithms for learning to rank involving multiple annotators. Indeed, this study is also supported by a real world application as follows:

Visual clutter ranking for Webpages: Visual clutter (VisC) determines the accessibility of webpages and is a critical factor for accessible Web search engines (e.g., Google Accessible Search). Current VisC measuring algorithms are designed for images only. Although the algorithms have been claimed to be applicable for Web visual clutter ranking if a Webpage is transformed into an image, the computational load becomes very high because features are required to be extracted on transformed images. Therefore, the question that arises is whether some quick Webpage features (e.g., number of texts) obtained merely from source codes achieve the same or a comparable performance with the state-of-the-art algorithms for images? The answer is very important for accessible Web search.

To answer the question above, a learning to rank algorithm under multiple annotators is required. First, various state-of-the-art images' VisC measuring algorithms are taken as multiple annotators to rank training pages [1]. Then the algorithm of learning to rank under multiple annotators is leveraged to train the ranking model. If the performance of the ranking model is close to the state-of-the-art image VisC measuring algorithms, the answer to the posed problem is "Yes".

This study proposes two learning approaches. One is very direct: estimating the ground truth using rank aggregation techniques and learning with the estimated ground truth. The other is to use a maximization likelihood framework which is used in [Raykar *et al.*, 2010]. Nevertheless, unlike the studies in [Raykar *et al.*, 2010], here, each training instance is a set of objects and each label is an ordering applied over the instance. This gives rise to several new challenges and as a result, methods in [Raykar *et al.*, 2010] cannot be simply inherited. In our maximization likelihood framework, a new generalized ranking model based on both two existing probabilistic ranking models is introduced to describe the relationships among the true labels, annotators' labels, and annotators' expertise. A new EM procedure is introduced to iteratively estimate the ground truth, the expertise of each annotator, and the parameters of the ranking model to be learned.

Our main contributions can be summarized as follows:

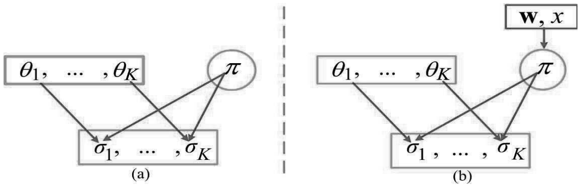
1. Unlike existing studies of learning under multi-annotators focus on classification, ordinal regression, and regression, this study focuses on learning to rank. As both learning to rank and labeling under multiple non-expert annotations will be more applied, our work will benefit related applications much.
2. Two learning approaches are proposed. The maximization likelihood approach, that jointly learns probabilistic true labels and ranking function, has been proved to be effective by experiments.
3. Finally, we have examined whether several quick features are enough to construct an effective VisC ranking function for Webpages or not.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the methodologies. Section 4 describes the learning algorithms for the maximum-likelihood approach. Section 5 reports experimental results. Finally, conclusions are given in Section 6.

2 Related work

2.1 Notation and Definitions

Let X be the input space whose instances are sets of objects, Y be the output space whose elements are the orderings of objects in the instance. An instance $x^{(i)}$ is represented by $(x_1^{(i)}, \cdots, x_{n^{(i)}}^{(i)})$, where $n^{(i)}$ denotes the number of objects in $x^{(i)}$; A label $y^{(i)} \in Y$ represented by $(y_1^{(i)}, \cdots, y_{n^{(i)}}^{(i)})$, where $y_j^{(i)}$ is the rank assigned to object $x_j^{(i)}$. For convenience, π and σ also denote orderings, where $\pi(i)$ $(\sigma(i))$ is the rank assigned to the i-th object and $\pi^{-1}(i)$ $(\sigma^{-1}(i))$ is the object index of the i-th rank. Let S_n be the set of all orderings over n objects, and $|S_n| = n!$. Let $d : S_n \times S_n \to R$ be the distance function between two orderings.



$\theta_1, \ \ldots \ , \theta_K$
π
$\sigma_1, \ \ldots \ , \sigma_K$
(a)
$\mathbf{w}, x$
$\theta_1, \ \ldots \ , \theta_K$
π
$\sigma_1 \ , \ \ldots \ , \ \sigma_K$
(b)

Figure 1: Two different generative processes.

2.2 Two probabilistic ranking models

This subsection reviews two probabilistic ranking models applied in our study. The first is the Mallows model [Mallows, 1957], which is also a typical distance-based ranking model. Given a truth ordering π and the expertise indicator parameter θ of an annotator, the Mallows model generates an ordering σ given by the annotator according to the formula:

$$P(\sigma|\pi, \theta) = \frac{1}{Z(\pi, \theta)} \exp(\theta \cdot d(\pi, \sigma)) \qquad (1)$$

where Z is a normalizing constant,

$$Z(\pi, \theta) = \sum\nolimits_{\sigma \in S_n} \exp(\theta \cdot d(\pi, \sigma)) \qquad (2)$$

The parameter θ is a non-positive quantity and the smaller the value of θ, the more expertise the annotator is said to be processing. When $\theta = 0$, the distribution is uniform meaning that the ordering by the annotator is independent of the truth and can have any values.

An extension of Mallows model was proposed as follows [Lebanon & Lafferty, 2002]:

$$P(\pi|\boldsymbol{\sigma}, \Theta) = \frac{1}{Z(\boldsymbol{\sigma}, \Theta)} P(\pi) \exp(\sum_{i=1}^{K} \theta_i \cdot d(\pi, \sigma_i)) \qquad (3)$$

where $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_K)$ belongs to S_n^K; $\Theta = (\theta_1, \cdots, \theta_K)$ belongs to R^K; $p(\pi)$ is a prior; and

$$Z(\boldsymbol{\sigma}, \Theta) = \sum\nolimits_{\pi \in S_n} P(\pi) \exp(\sum_{i=1}^{K} \theta_i \cdot d(\pi, \sigma_i)) \qquad (4)$$

In this extended model, each ordering σ_i is a quantity returned by an annotator for a particular set of objects. The free parameter θ_i represents the expertise degree of the i-th annotator. Equation (3) calculates the probability of that the truth ordering is π, given the orderings from the annotators and degrees of their expertise.

Klementiev *et al.* [Klementiev *et al.*, 2008] proved that if the distance is right-invariant, the following generative process can be derived based on Eq. (3):

$$P(\pi, \boldsymbol{\sigma}|\Theta) = P(\pi) \prod_{i=1}^{K} P(\sigma_i|\pi, \theta_i) \qquad (5)$$

This generative process can be described by Fig. 1(a). π is first drawn from prior $P(\pi)$. Then $\boldsymbol{\sigma}$ is generated by drawing $\sigma_1, \cdots, \sigma_K$ independently from K mallows models according to Eq. (1) with the same truth ordering π.

The second probabilistic ranking model is the Plackett-Luce (P-L) model [Plackett, 1975 and Luce, 1959]. P-L model is a distribution over orderings. It is parameterized

[1] It is impractical to recruit people as annotators to rank Webpages' visual clutter because people may find the job to be tedious when the number of training pages is large.

by a vector $v = (v_1, \cdots, v_M)$, where $v_i (> 0)$ is associated with index i:

$$P(\pi|v) = \prod_{i=1}^{M} \frac{v_{\pi^{-1}(i)}}{v_{\pi^{-1}(i)} + v_{\pi^{-1}(i+1)} + \cdots + v_{\pi^{-1}(M)}} \quad (6)$$

The meaning and value of v depend on application settings. P-L model has been applied in many machine learning problems [Cheng *et al.*, 2010].

The difference between Mallows model and P-L model is that the former refers to the relationships between orderings, while the latter refers to the probability about a single ordering.

2.3 Learning to rank

Previous studies carried out in learning to rank can be divided into three categories: position-based, pairwise, and listwise [Cao *et al.*, 2007]. A detailed survey of learning to rank can be found in [Liu, 2009]. This study in particular focuses on the listwise approach as it is the most suitable approach for information retrieval as compared to others.

The first step of listwise learning to rank is to define a loss function over a truth ordering and a predicted ordering. Following the definition of loss function, the parameters of the ranking function are learned according to the minimization of training loss. The ranking function is usually assumed linear on features:

$$f(x) = <\mathbf{w}, x> = \mathbf{w}^T x \quad (7)$$

Xia *et al.* [Xia *et al.*, 2008] proposed an effective listwise approach called ListMLE. Actually, ListMLE minimizes the sum of likelihood losses with respect to all the training samples. The likelihood loss function can be defined as follows:

$$l(f(x), y) = -\log P(y|f, x) \quad (8)$$

where $P(y|x; f)$ is calculated by the P-L model:

$$P(y|f, x) = \prod_{i=1}^{N} \frac{\exp(f(x_{y^{-1}(i)}))}{\sum_{k=i}^{M} \exp(f(x_{y^{-1}(k)}))} \quad (9)$$

ListMLE chooses linear Neural Network (parameterized by $\mathbf{w}$) as the ranking model, and utilizes Stochastic Gradient Descent as the algorithm for computing the (local) optimal parameter $\mathbf{w}$. The detailed steps can be found in [Xia *et al.*, 2008].

3 Methodologies

As presented earlier, each instance is labeled by G annotators. Hence, the training set is denoted as $D = \{(x^{(i)}, y_1^{(i)}, \cdots, y_G^{(i)})\}_{i=1}^{N}$, which contains N independently and identically distributed (i.i.d) samples. Our task is to achieve the value of the parameter $\mathbf{w}$ (in Eq. (7)) as well as the values of Θ which is the annotators' degrees of expertise.

3.1 Direct approach

This approach is very intuitive. The Mallows model is applied to fuse together the orderings by the annotators to estimate the ground truth and the degrees of expertise of the annotators [Klementiev *et al.*, 2008]. Then conventional learning to rank algorithm (e.g., ListMLE) is used to train a ranking model based on the estimated ground truth. This approach is called LTRMA-D for simplicity.

3.2 Maximum-likelihood approach

Let $\Omega = (\mathbf{w}, \Theta)$. The likelihood function of the parameters based on the observation D can be factored into:

$$\begin{aligned} P(D|\Omega) &= \prod_{i=1}^{N} P(x^{(i)}, y_1^{(i)}, \cdots, y_G^{(i)}|\Omega) \\ &= \prod_{i=1}^{N} P(y_1^{(i)}, \cdots, y_G^{(i)}|x^{(i)}, \Omega) P(x^{(i)}) \\ &\propto \prod_{i=1}^{N} P(y_1^{(i)}, \cdots, y_G^{(i)}|x^{(i)}, \Omega) \\ &= \prod_{i=1}^{N} \sum_{y^{(i)} \in S_n} [P(y_1^{(i)}, \cdots, y_G^{(i)}|y^{(i)}, \Theta) P(y^{(i)}|x^{(i)}, \mathbf{w})] \end{aligned} \quad (10)$$

The above equation integrates both the Mallows model $(P(y_1^{(i)}, \cdots, y_G^{(i)}|y^{(i)}, \Theta))$ and the P-L model $(P(y^{(i)}|x^{(i)}, \mathbf{w}))$. The maximum-likelihood estimator is attained by taking the logarithm of the likelihood and maximizing it, that is,

$$\bar{\Omega}_{ML} = \{\overline{\mathbf{w}}, \overline{\Theta}\} = \arg\max_{\Omega} \{\ln P(D|\Omega)\}. \quad (11)$$

This approach is called **LTRMA-ML** for simplicity. The following section introduces the learning algorithm in detail.

4 Learning for LTRMA-ML

As the cardinality of an ordering space is large, Eq. (11) is intractable optimized directly. To estimate Ω, the EM algorithm [Demspster *et al.*, 1977] is utilized. The truth labels $(y^{(i)})$ are taken as missing data. Then the true labels and Ω can be estimated iteratively. Once a certain number of criterions are met, the iteration is stopped. At first, a new log-likelihood is written as:

$$\ln P(D, \mathbf{y}|\Omega) \propto \ln \prod_{i=1}^{N} P(y_1^{(i)}, \cdots, y_G^{(i)}, y^{(i)}|x^{(i)}, \Omega) \quad (12)$$

4.1 Inference

To factorize Eq. (12), a new generalized ranking model is proposed:

$$\begin{aligned} &P(y_1^{(i)}, \cdots, y_G^{(i)}, y^{(i)}|x^{(i)}, \Omega) \\ &= P(y^{(i)}|x^{(i)}, \mathbf{w}) \prod_{j=1}^{G} P(y_j^{(i)}|y^{(i)}, \theta_j) \end{aligned} \quad (13)$$

This model can be described by Fig. 1(b), where $y^{(i)}$ is π, $y_j^{(i)}$ is σ_j, and $K = G$. This generative process differs from the process by Eq. (5) (shown in Fig. 1(a)) in the sense that in Fig. 1(a), the distribution of the ground-truth $y(i)$ (or π) is dependent on a prior probability $P(\pi)$, while in Fig. 1(b), $y^{(i)}$ is assumed to be dependent on the parameter $\mathbf{w}$ and $x^{(i)}$. This assumption is reasonable and inherited from [Raykar, 2010], which was proved to be working well through experiments.

Based on Fig. 1(b) and the properties of the Mallows model, the following equation is obtained:

$$\begin{aligned} &P(y^{(i)}|y_1^{(i)}, \cdots, y_G^{(i)}, x^{(i)}, \mathbf{w}, \Theta) \\ &= \frac{P(y_1^{(i)}, \cdots, y_G^{(i)}, y^{(i)}|x^{(i)}, \mathbf{w}, \Theta)}{P(y_1^{(i)}, \cdots, y_G^{(i)})} \\ &= P(y^{(i)}|x^{(i)}, \mathbf{w}) \frac{1}{Z(\Theta, \mathbf{w}, x^{(i)}, y_1^{(i)}, \cdots, y_G^{(i)})} \exp(\sum_{j=1}^{G} \theta_j d(y^{(i)}, y_j^{(i)})) \end{aligned} \quad (14)$$

where $p(y^{(i)}|x^{(i)}, \mathbf{w})$ is obtained from P-L model and calculated in a way similar to Eq. (9). Equation (14) will be used in the following algorithm.

Eq. (12) can be represented by

$$\begin{aligned}
&\ln \prod_{i=1}^{N} P(y_1^{(i)}, \cdots, y_G^{(i)}, y^{(i)}|x^{(i)}, \Omega) \\
&= \ln \prod_{i=1}^{N} \{P(y_1^{(i)}, \cdots, y_G^{(i)}|y^{(i)}, \theta_j) P(y^{(i)}|x^{(i)}, \mathbf{w})\} \qquad (15) \\
&= \ln \prod_{i=1}^{N} \{ \prod_{j=1}^{G} P(y_j^{(i)}|y^{(i)}, \theta_j) P(y^{(i)}|x^{(i)}, \mathbf{w})\}
\end{aligned}$$

When using EM, the first step is to define:

$$\begin{aligned}
&Q(\mathbf{w}, \Theta; \mathbf{w}', \Theta') \\
&= E(\ln \prod_{i=1}^{N} \{ \prod_{j=1}^{G} P(y_j^{(i)}|y^{(i)}, \theta_j) P(y^{(i)}|x^{(i)}, \mathbf{w})\}|D, \mathbf{w}', \Theta')
\end{aligned} \tag{16}$$

At first, we have the following Lemma:

Lemma 1:

$$Q(\mathbf{w}, \Theta; \mathbf{w}', \Theta') = \sum_{(y^{(i)}, \ldots, y^{(N)}) \in S_n^N} L(\mathbf{w}, \Theta) U(\mathbf{w}', \Theta') \tag{17}$$

where

$$\begin{aligned}
L(\mathbf{w}, \Theta) = &\sum_{i=1}^{N} \ln P(y^{(i)}|x^{(i)}, \mathbf{w}) - N \sum_{j=1}^{G} \ln Z(\theta_j) \\
&+ \sum_{i=1}^{N} \sum_{j=1}^{G} \theta_j d(y^{(i)}, y_j^{(i)})
\end{aligned} \tag{18}$$

$$U(\mathbf{w}', \Theta') = \prod_{i=1}^{N} P(y^{(i)}|y_1^{(i)}, \cdots, y_G^{(i)}, x^{(i)}, \mathbf{w}', \Theta') \tag{19}$$

To maximize Eq. (17), we have the following Lemmas.

Lemma 2: For any $\mathbf{w}$, the maximization of Q by Θ is attained by $\Theta = (\theta_1, \cdots, \theta_G)$ such that

$$E_{\theta_j}(d) = \sum_{(y^{(1)}, \ldots, y^{(N)}) \in S_n^N} (\frac{1}{N} \sum_{i=1}^{N} d(y^{(\imath)}, y_j^{(i)})) U(\mathbf{w}', \boldsymbol{\Theta}') \tag{20}$$

Proof. Omitted due to lack of space. The proof procedure is similar to that in [Klementiev *et al.*, 2008].

Lemma 3: For any Θ, the maximization of Q by $\mathbf{w}$ is equivalent to the minimization of the cross-entropy as follows:

$$\begin{aligned}
CE = - &\sum_{(y^{(i)}, \ldots, y^{(N)}) \in S_n^N} \{\ln[\prod_{i=1}^{N} \mathrm{P}(y^{(i)}|x^{(i)}, \mathbf{w})] \\
&\times \prod_{i=1}^{N} \mathrm{P}(y^{(i)}|y_1^{(i)}, \ldots, y_G^{(i)}, x^{(i)}, \mathbf{w}', \Theta')\}
\end{aligned} \tag{21}$$

Proof.

$$\begin{aligned}
Q(\mathbf{w}, \Theta; \mathbf{w}', \Theta') = &\sum_{(y^{(i)}, \ldots, y^{(N)}) \in S_n^N} \{\ln[\prod_{i=1}^{N} \mathrm{P}(y^{(i)}|x^{(i)}, \mathbf{w})] \\
\times \prod_{i=1}^{N} \mathrm{P}(y^{(i)}|y_1^{(i)}, \ldots, y_G^{(i)}, &x^{(i)}, \mathbf{w}', \Theta')\} + g(\mathbf{w}', \Theta', \Theta)
\end{aligned} \tag{22}$$

The following subsection describes the detailed steps of solving Eqs. (20) and (21) to infer $\mathbf{w}$ and Θ.

4.2 Algorithm

In each iteration of EM, Θ is updated by solving Eq. (20) while $\mathbf{w}$ is updated by minimizing Eq. (21). In Eq. (20), $E_{\theta_j}(d)$ is:

$$E_{\theta_j}(d) = \frac{ne^{\theta_j}}{1 - e^{\theta_j}} - \sum_{l=1}^{n} \frac{le^{l\theta_j}}{1 - e^{l\theta_j}} \tag{23}$$

where n is the number of objects in each instance. This function is monotonous. The right-hand side of Eq. (20) cannot be calculated in practice. We adopted the sampling method introduced in [Klementiev *et al.*, 2008] to obtain the approximate value of the right-hand side of Eq. (20). Thereafter, Θ can be obtained by a binary search approach. The steps are shown below in Algorithm 1.

Algorithm 1 Update Θ

```
Input: $D$, $\Theta^{(t)}$, $\mathbf{w}^{(t)}$, $k = 1, Ns$, $\pi^{(i)}\{1\}, i = 1, \cdots, N$.
Output: $\Theta^{(t+1)}$.
Steps:
1. For each $\pi^{(i)}\{k\}, i \in [1, N]$, choose two indices $p, q$ random,
   and exchange the $p$-th and $q$-th elements of $\pi^{(i)}\{k\}$ to form a
   new ordering $\sigma^{(i)}$.
2. Calculate $\alpha_i = \frac{P(\sigma^{(i)}|y_1^{(i)},\ldots,y_R^{(i)},x^{(i)},\mathbf{w}^{(t)},\Theta^{(t)})}{P(\pi^{(i)}\{k\}|y_1^{(i)},\ldots,y_R^{(i)},x^{(i)},\mathbf{w}^{(t)},\Theta^{(t)})}$. For each $i \in$
   $[1, N]$, if $\alpha_i > 1$, $\pi^{(i)}\{k+1\} = \sigma^{(i)}$; else $\pi^{(i)}\{k+1\} = \pi^{(i)}\{k\}$
   with probability $1 - \alpha_i$ and otherwise $\pi^{(i)}\{k+1\} = \sigma^{(i)}$. If
   $k < Ns$, $k = k + 1$ and goto 1.
3. Calculate $\beta_j = \frac{1}{N \cdot Ns} \sum_{i=1}^{N} \sum_{k=1}^{Ns} d(\pi^{(i)}\{k\}, y_j^{(i)}), j = 1, \cdots, G$.
4. Apply binary search to obtain $\theta_j^{(t+1)}$ according to $\frac{ne^{\theta_j}}{1-e^{\theta_j}} -$
   $\sum_{l=1}^{n} \frac{le^{l\theta_j}}{1-e^{l\theta_j}} = \beta_j, j = 1, \cdots, G$.
```

Minimizing Eq. (21) is similar to determining the parameter $\mathbf{w}$ based on the cross-entropy loss introduced in [Cao *et al.*, 2007]. However, the computational complexity of minimizing Eq. (21) is $O(n!N)$. Instead, we introduce a heuristic yet an efficient solution[2]. Note that Eq. (21) measures the distance between two conditional distributions. Both distributions have only one maximum value. If their maximum values are equal, their distance is likely to be quite small. Assume $p(y^{(i)}|y_1^{(i)}, \cdots, y_G^{(i)}, x^{(i)}, \mathbf{w}', \Theta')$ attains its maximum value at $\pi_*^{(i)}$. Then to minimize Eq. (21), we apply a transformation to maximize the following function:

$$\ln \prod_{i=1}^{N} P(\pi_*^{(i)}|x^{(i)}, \mathbf{w}) \tag{24}$$

So ListMLE[3] can be leveraged to achieve a new value of $\mathbf{w}$.

[2] The solution is the same as the strategy used in [Raykar *et al.*, 2010]: the estimated ground truth in each iteration is used to train a prediction model.

[3] In fact, any other linear or un-linear learning to rank algorithms which aim of minimizing the likelihood loss can be used in the proposed algorithm. In this study, ListMLE is chosen due to its competing performances reported in previous literature.

In our algorithm, $\pi_*^{(i)}$ is selected from orderings according to the same sampling steps as mentioned in Algorithm 1.

The steps of LTRMA-ML are as follows.

Algorithm 2 Steps of LTRMA-ML

```
Input: $D$, $Ns$, $\mathbf{w}^{(0)}$ and $\Theta^{(0)}$, $\tau1$, $\tau2$, $t = 0$, $MaxT$.
Output: $\mathbf{w}$, $\Theta$.
Steps:
1. Calculate $\Theta^{(t+1)}$ using Algorithm 1.
2. Repeat the sampling steps 1 and 2 in Algorithm 1 to obtain an
   ordering set for each $x^{(i)}$.
3. Select the maximum elements in the sampling sets for each $x^{(i)}$.
   These maximum elements are the estimated ground truth orderings
   for this particular iteration.
4. Update $\mathbf{w}$ using ListMLE with estimated ground truth.
5. If $t > MaxT$, or $||\Theta^{(t)} - \Theta^{(t+1)}|| < \tau1$ and $||\mathbf{w}^{(t)} - \mathbf{w}^{(t+1)}|| <$
   $\tau2$, return $\mathbf{w}^{(t+1)}$ and $\Theta^{(t+1)}$; else $t = t + 1$, goto 1.
```

5 Experiments

5.1 Experimental setup

As there have been no benchmark data sets for learning to rank under multiple annotators, two data sets (a synthetic data set and a real data set) that are popular in learning to rank literature are used. To simulate multiple annotators, several ordering labels are generated based on the ground truth for each instance. The two proposed algorithms are implemented on training and validation sets to search optimal parameters. Then the learned model is used to rank the test data. Finally, the mean average precision (MAP) [Liu, 2009] and normalized discounted cumulative gain (NDCG) [Liu, 2009] are calculated. The higher their values, the better the results. In total, three learning strategies are compared: LTRMA-D, LTRMA-ML, and ListMLE that are based on ground truth orderings. In each experiment, for Algorithm 1, Ns is set to 500. For Algorithm 2, $\tau1$ and $\tau2$ are set to $0.01 * G$ and $0.01 * m$, respectively, where G is the number of annotators and m is the feature dimension; each entry of $\mathbf{w}^{(0)}$ is set to $1/m$; $\Theta^{(0)}$ is randomly initialized; $MaxT$ is set to 200.

5.2 Results on synthetic data

The rule of creating the synthetic data is the similar to that in [Xia *et al.*, 2008]. First, a point is randomly sampled according to the uniform distribution on a square area $[0, 1] \times [0, 1]$. Then a score is assigned to the point using the following rule, $y = x1 + 10x2 + \epsilon$, where ε is a random variable normally distributed with zero mean and a standard deviation of 0.005. In total, 15 points associated with scores are generated in this way. The permutation on their scores forms the ranking of the points. The process is repeated to make 100 training instances, 100 validation instances, and 500 testing instances. When calculating MAP, top-5 items are consider as relevant. When calculating NDCG, the i-th $(i = 1, \cdots, 15)$ ranked item's relevance score is $16 - i$.

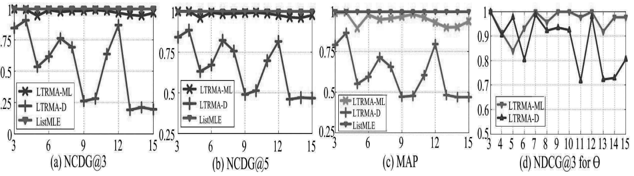


Figure 2: Performance comparison on Synthetic data.

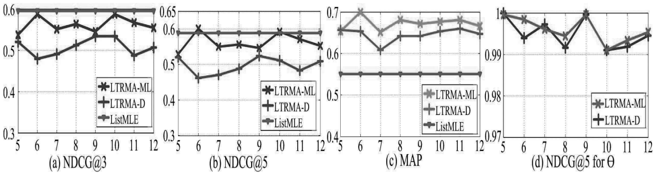


Figure 3: Performance comparison on OHSUMED data.

Assume there are G annotators. Their labels are simulated as follows: for the i-th annotator, two elements of a ground-truth label are exchanged, and the process repeated $2 \times i + 1$ times. In our experiment, G ranges from 3 to 15. The experiment is repeated ten times and average results are reported. Figures 2(a) and 2(b) shows the results of NDCG@3 and NDCG@5, respectively. Figure 2(c) shows the results of MAP when top-5 items are taken as relevant. It can be observed that the results of LTRMA-ML are very close to those of ListMLE when true ground-truth labels are employed.

Further more, LTRMA-D does not appear to be robust. For example, when the number of simulated annotators is 12, the performance of LTRMA-D is satisfactory. However, when the number is 9 or 10, LTRMA-D performs badly. According to Fig. 2(d), LTRMA-ML also outperforms LTRMA-D in terms of the estimated expertise of simulated annotators.

5.3 Results on OHSUMED data

The OHSUMED data collection is a benchmark set for learning to rank and is provide in LETOR [Liu *et al.*, 2007]. The data consists of query-document pairs upon which relevance judgments are made. The degree of relevance is divided into three categories: definitely relevant (score of 3), possibly relevant (score of 2), and not relevant (score of 1). The data split by LETOR is used to conduct five-fold cross validation. In MAP calculation, only definitely relevant objects are taken as relevant.

The ground-truth labels are constructed similar to [Xia *et al.*, 2008]: one perfect permutation is randomly selected for each query among all the possible perfect permutations based on the ground truth. The labels by annotators are simulated using a similar rule as applied on the Synthetic data. The number of annotators G ranges from 5 to 15.

Figures 3(a) and 3(b) shows the results of NDCG@3 and NDCG@5, respectively. Figure 3(c) shows the results of MAP. Observations similar to Figs. 2(a) and (b) can be made. The performances of LTRMA-ML are close to those of ListMLE and significantly better than LTRMA-D. According to Fig. 3(c), LTRMA-ML achieves the highest MAP values under different annotators. According to Fig. 3(d), LTRMA-ML slightly outperforms LTRMA-D in terms of the estimated expertise of the simulated annotators.

5.4 Web visual clutter ranking

In this experiment, 2000 homepages were collected, mainly from the websites of company and university as well as some personal sites. For each page, eight simple features are ex-

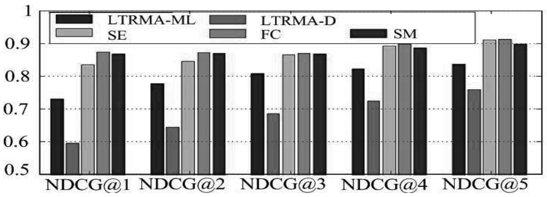


Figure 4: Results on Web visual clutter ranking

tracted: number of texts, number of linked texts, number of fonts, average font size, number of tables, number of background colors, number of images, and aspect ratio. All the collected pages were divided into 200 subsets and each subset consisted of ten pages. *Three state-of-the-art image visual clutter measuring algorithms were taken as three annotators.* They are Subband Entropy (SE), Feature Congestion (FC), and Segment Measuring (SM). Further details about them can be referred to from [Bravo and Farid, 2008] and [Rosenholtz *et al.*, 2007]. Consequently, each subset corresponds to three orderings as given by the three algorithms.

The 200 subsets are randomly divided into two heads: one for training and the other for testing. To evaluate the results on the test set, ten random subsets are selected and scored by seven volunteers aging from 20 to 30. The score ranges from 0 to 5 and the higher the score, the lower the visual clutter. The training and testing process is repeated ten times. The averaging values of NDCG are reported.

The performances of LTRMA-ML and LTRMA-D are shown in Fig. 4. The NDCG values of the three visual clutter measuring algorithms are also presented in Fig. 4. It was observed that LTRMA-ML outperforms LTRMA-D. LTRMA-ML is close to the visual clutter measuring algorithms on NDCG@3, NDCG@4, and NDCG@5.

5.5 Discussion

Based on the experimental results, LTRMA-ML is significantly better than LTRMA-D. The partial reason is that LTRMA-ML unifies both ground truth estimation and model learning in a probabilistic formulation, which better models their relationships. We also note that LTRMA-ML is more robust than LTRMA-D against the variation in the number of annotators. The partial reason is that in LTMA-D the model learning is very sensitive the ground truth estimation, while in LTRMA-ML, the model learning can bring feedback to the ground truth estimation in the next iteration. The results on the Web visual clutter data indicate that considering various simple and quick features can lead to a performance close to state-of-the-art algorithms that are based on image content analysis. Hence, if an application involving Web visual clutter ranking is time-sensitive, consideration of only some quick features is likely to be very helpful.

6 Conclusion

This paper has investigated learning to rank under multiple annotators providing labels that are not absolutely accurate. Two algorithms, namely, LTRMA-D and LTRMA-ML, are proposed. LTRMA-D is a direct approach that first estimates the ground truth and then uses conventional algorithms (e.g., ListMLE) to train a ranking model based on the estimated ground truth. LTRMA-ML integrates both ground truth estimation and ranking model learning by a maximum likelihood framework. Experiments suggest that LTRMA-ML outperforms LTRMA-D and is very close to ListMLE when true ground truth labels are employed. The experiment on Web visual clutter data indicates that based on LTRMA-ML, we can construct a ranking model whose performance is close to the state-of-the-art image visual clutter measuring algorithms, by considering only simple features.

Acknowledgments

This work is partly supported by NSFC (Grant No. 60825204, 60672040, 60723005).

References

[Bravi and Farid, 2008] Bravi, M. J., and Farid, H., A scale invariant measure of clutter, *Journal of Vision* (2008) 8(1):23, 1-9.

[Cao *et al*, 2007] Cao, Z., *et al.*, Learning to rank: from pairwise approach to listwise approach. *ICML*, 129-136, 2007.

[Cheng *et al.*, 2010] Cheng, W., *et al.*, Ranking Methods based on the Plackett-Luce Model, *ICML*, 215-222, 2010.

[Chen *et al.*, 2010] Chen, S., *et al.*, What if the Irresponsible Teachers Are Dominating? A Method of Training on Samples and Clustering on Teachers, *AAAI*, 419-424, 2010.

[Dekel and Shamir, 2009] Dekel, O., and Shamir, O., Vox populi: Collecting high-quality labels from a crowd. *COLT*, 2009.

[Dempster *et al.*, 1997] Dempster *et al.*, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1-38, 1977.

[Donmez and Garbonell, 2010] Donmez, P. and Garbonell, J., A Probabilistic Framework to Learn from Multiple Annotators with Time-Varying Accuracy, *SDM*, 862-871, 2010.

[Klementiev *et al.*, 2008] Klementiev, A., *et al.*, Unsupervised Rank Aggregation with Distance-Based Models, *ICML*, 2008.

[Lebanon & Lafferty, 2002] Lebanon, G., & Lafferty, J., Cranking: Combining rankings using conditional probability models on permutations. *ICML*, 2002.

[Liu *et al.*, 2007] Liu, T-Y., *et al.*, Letor: Benchmark dataset for research on learning to rank for information retrieval. *Proceedings of SIGIR Workshop on LTR4IR*, 2007.

[Liu, 2009] Liu, T-Y., Learning to Rank for Information Retrieval, *Foundation and Trends on Information Retrieval*, Now Publishers, 2009.

[Luce, 1959] Luce, R. D., Individual choice behavior: A theoretical analysis. Wiley, 1959.

[Mallows, 1957] Mallow, C. L., Non-null ranking models, *Biometrika*, 44, 114-130, 1957.

[Plackett, 1975] Plackett, R., The analysis of permutations. *Applied Stat.*, 24, 193-202, 1975.

[Raykar *et al.*, 2010] Raykar, V. C., *et al.*, Learning From Crowds, *JMLR*, 11 (2010) 1297-1322.

[Rosenholtz *et al.*, 2007] Rosenholtz, R., *et al.*, Measuring visual clutter. *Journal of Vision*, 7(2):17, 1-22, 2007.

[Sheng *et al.*, 2008] Sheng, V., *et al.*, Get another label? Improving data quality and data mining using multiple, noisy labelers. *ACM SIGKDD*, 614-622, 2008.

[Smyth *et al.*, 1995] Smyth, P., *et al.*, Inferring ground truth from subjective labelling of venus images. *NIPS*, 1085-1092. 1995.

[Xia *et al.*, 2008] Xia, F. *et al.*, Listwise Approach to Learning to Rank - Theory and Algorithm, *ICML*, 1192-1199, 2008.

[Yan *et al.*, 2010] Yan, Y., *et al.*, Modeling Annotator Expertise: Learning When Everybody Knows a Bit of Something, *AISTATS*, 932-939, 2010.