Journal of Machine Learning Research 23 (2022) 1-36 Submitted 8/20; Revised 12/21; Published 1/22

Generalized Ambiguity Decomposition for Ranking Ensemble Learning

Hongzhi Liu[†]✉ LIUHZ@PKU.EDU.CN
Yingpeng Du[†] DYP1993@PKU.EDU.CN
School of Software and Microelectronics
Peking University
Beijing, P.R.China, 102600

Zhonghai Wu ✉ WUZH@PKU.EDU.CN
National Engineering Center of Software Engineering
Peking University
Beijing, P.R.China, 100871

Editor: Samy Bengio

Abstract

Error decomposition analysis is a key problem for ensemble learning, which indicates that proper combination of multiple models can achieve better performance than any individual one. Existing theoretical research of ensemble learning focuses on regression or classification tasks. There is limited theoretical research for ranking ensemble. In this paper, we first generalize the ambiguity decomposition theory from regression ensemble to ranking ensemble, which proves the effectiveness of ranking ensemble with consideration of list-wise ranking information. According to the generalized theory, we propose an explicit diversity measure for ranking ensemble, which can be used to enhance the diversity of ensemble and improve the performance of ensemble model. Furthermore, we adopt an adaptive learning scheme to learn query-dependent ensemble weights, which can fit into the generalized theory and help to further improve the performance of ensemble model. Extensive experiments on recommendation and information retrieval tasks demonstrate the effectiveness and theoretical advantages of the proposed method compared with several state-of-the-art methods.

Keywords: ensemble learning, ambiguity decomposition theory, ranking ensemble, diversity measure, adaptive learning

1. Introduction

Ensemble learning aims at combining multiple base models to obtain better prediction performance. It has been widely used for various tasks such as gene identification (Manavalan et al., 2019), computer vision (Yu et al., 2021), natural language processing (Fang et al., 2019), and etc.

Existing theoretical research of ensemble learning focuses on regression or classification tasks, such as the ambiguity decomposition theory (Krogh and Vedelsby, 1995; Jiang et al.,

. † These authors contributed equally to this work.

2017) and the bias-variance-covariance decomposition theory (Ueda and Nakano, 1996). Krogh and Vedelsby (1995) revealed the relationship between the error of the ensemble model and the error of base models for regression tasks. For each instance o, we have

$$(f^{ens}(o) - y)^2 = \sum_i w_i (f^i(o) - y)^2 - \sum_i w_i (f^i(o) - f^{ens}(o))^2$$

where y is the truth value of instance o, and the ensemble model f^{ens} is a convex combination of base models, i.e.,

$$f^{ens}(o) = \sum_i w_i f^i(o)$$

where w_i is the weight of the i-th base model f^i, $w_i \geq 0$ and $\sum_i w_i = 1$. $f^i(o)$ and $f^{ens}(o)$ denote the predictions on instance o by the i-th base model f^i and the ensemble model f^{ens}, respectively. Taking expectation on sample space $\mathcal{D}$ yields the classic ambiguity decomposition theory for regression tasks, which is written as follows,

$$E = \bar{E} - \bar{A}$$

where $E = E_{\mathcal{D}}[(f^{ens}(o) - y)^2]$ denotes the generalization error of the ensemble model f^{ens}, $\bar{E} = \sum_i w_i E_{\mathcal{D}}[(f^i(o) - y)^2]$ denotes the weighted average of generalization error of base models, and $\bar{A} = \sum_i w_i E_{\mathcal{D}}[(f^i(o) - f^{ens}(o))^2]$ is called "ambiguity" which has been considered to be relevant to the diversity of ensemble. For classification tasks, Jiang et al. (2017) proved an ambiguity decomposition theory with a variety of differentiable loss functions, such as logistic loss and exponential loss.

Existing theoretical research results for regression and classification ensemble are helpful in understanding the mechanism of ensemble learning and designing ensemble methods for regression and classification tasks (Brown et al., 2005a; Yin et al., 2014; Liu et al., 2019). However, it still lacks related theoretical research for ranking tasks and most of existing ranking ensemble methods are heuristic. Although we can transform a ranking task into a regression task (e.g., score regression) with consideration of only point-wise information or a classification task (e.g., positive-negative/order-pair classification) with consideration of point-wise and/or pair-wise information (Ailon and Mohri, 2010), it ignores some useful information and limits the prediction performance. Specifically, methods based on point-wise information ignore the order information of ranking lists (Rendle et al., 2009), and methods based on pair-wise information focus on the relative order of each pair of items while ignoring the ranking positions of items (Wang et al., 2016). Suppose the ground truth of a ranking list is $\pi^* = [o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5 \succ o_6]$ and we have two predicted ranking lists $\pi_1 = [o_2 \succ o_3 \succ \boldsymbol{o_1} \succ o_4 \succ o_5 \succ o_6]$ and $\pi_2 = [o_1 \succ o_2 \succ o_3 \succ \boldsymbol{o_6} \succ o_4 \succ o_5]$. For pair-wise losses or metrics (e.g., Area Under the ROC Curve, AUC), we have $l_{pair}(\pi_1, \pi^*) = l_{pair}(\pi_2, \pi^*)$. However, for ranking tasks like recommender systems (RSs) and information retrieval (IR), we hope that the loss $l_{list}(\pi_1, \pi^*) > l_{list}(\pi_2, \pi^*)$ because we pay more attention on the top positions of ranking lists.

In this paper, we derive some theories for ranking ensemble learning with consideration of list-wise information, which can be widely used in real-world applications like RSs and IR. Compared with point-wise loss functions (e.g., squared loss, logistic loss, exponential loss, etc.) used in regression or classification tasks, list-wise loss functions used in ranking tasks

for exploring rich ranking information in the training data are more complex. We generalize the ambiguity decomposition theory from regression tasks to ranking tasks, which proves the effectiveness of ranking ensemble with consideration of list-wise ranking information.

As the classic ambiguity decomposition theory has proved that diversity is another key factor besides accuracy for regression ensemble learning (Krogh and Vedelsby, 1995; Kuncheva and Whitaker, 2003; Brown et al., 2005b), many different kinds of diversity measures have been proposed, such as disagreement, double-fault measure and entropy measure (Brown et al., 2005a; Schwenker, 2013). However, these methods are designed for regression or classification tasks, which cannot meet the requirements of ranking ensemble learning. To solve this problem, we propose an explicit diversity measure for ranking ensemble learning based on our generalized ambiguity decomposition theory. It contributes to enhance the diversity of ensemble and improve the performance of ensemble model.

Besides the lack of theoretical support, most of existing ranking ensemble methods assume the ensemble weights of base models are query-independent. That is, each base model may have a different weight w_i but they are invariant with different queries. This assumption may degrade the performance of ensemble learning. Each base model may have different importance on different queries or users, e.g., a base model trained on a data set of female users may be unsuitable for male users. To deal with this problem, we adopt an attention mechanism and fit it into the proposed theory to learn query-dependent weights for ranking ensemble tasks.

In this paper, we contribute to the field of ranking ensemble both theoretically and practically. The main contributions of this paper are summarized as follows:

- We generalize the ambiguity decomposition theory from regression ensemble to ranking ensemble, which proves the effectiveness of ranking ensemble.
- We propose an explicit diversity measure for ranking ensemble learning based on the generalized ambiguity decomposition theory, which can be used to improve the performance of ensemble model.
- We adopt an adaptive learning scheme and fit it into the proposed theory to learn query-dependent weights for ranking ensemble, which can help to further improve the performance of ensemble model.
- We conduct extensive experiments on two kinds of ranking tasks, RSs and IR, whose results demonstrate the effectiveness and theoretical advantages of the proposed method.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 formulates the problem and introduce a list-wise likelihood function suitable for ranking ensemble learning. Section 4 presents the derived theories and the proposed method. Extensive experiments are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

In this section, we first review some related work about ensemble learning for regression and classification tasks, and then provide an overview of existing research about ranking ensemble.

2.1 Regression Ensemble and Classification Ensemble

Weighted ensemble is a kind of widely used ensemble method, which can be adopted with flexible base models such as homogenous or heterogenous ones (Liu et al., 2019). Theoretically, vote ensemble (Breiman, 1996) and selective ensemble (Rayana and Akoglu, 2016) are special cases of weighted ensemble. The key of weighted ensemble is to determine the weights of base models for ensemble. A common assumption of weighted ensemble is that base models with better performance should have larger ensemble weights. A simple and commonly used method is directly calculating the ensemble weights w_i according to base models' performance (Kuncheva, 2014), e.g.,

$$w_i = \log \frac{p_i}{1 - p_i}$$

where p_i denotes the normalized performance of the i-th base model on the training or validation data set.

Ambiguity decomposition is an important theory for weighted ensemble (Brown et al., 2005a). Krogh and Vedelsby (1995) and Jiang et al. (2017) revealed and generalized the ambiguity decomposition theory for regression ensemble and classification ensemble, respectively. They proved that diversity is another key factor for ensemble learning besides the performance of base models. Specifically, the generalization error of ensemble model can be decomposed into the weighted average generalization error of base models and an ambiguity term related to the diversity of base models. Based on these theoretical results, several ensemble learning methods have been proposed to learn the ensemble weights of base models with consideration of both the generalization error and the diversity. Yin et al. (2014) proposed to combine the square loss and a diversity term with sparseness regularization and took it as a convex quadratic programming problem. Liu et al. (2019) proposed to learn personalized ensemble weights based on the generalization error and a diversity measure for classification ensemble.

Existing theoretical research of ensemble learning focused on regression and classification tasks. There is limited theoretical research for ranking ensemble. In this paper, we derive a generalized ambiguity decomposition theory for ranking ensemble, based on which an explicit diversity measure for ranking ensemble is proposed as well.

2.2 Ranking Ensemble

Ranking ensemble, also known as rank aggregation or rank fusion, is an important technique for ranking tasks like RSs and IR. It combines the predictions generated by multiple base rankers into one ranking list. Existing ranking ensemble methods can be roughly divided into two categories: unsupervised ranking ensemble and supervised ranking ensemble (Lin, 2010; Volkovs and Zemel, 2014).

Unsupervised ranking ensemble tries to combine the predictions of base rankers in an unsupervised way, which attracts extensive attention due to its ease of use. Well-known unsupervised ranking ensemble methods include the combination rank fusion family (Fox and Shaw, 1994), Borda rank fusion (Aslam and Montague, 2001), manifold learning based rank aggregation (Liang et al., 2018b), spectral method based ensemble (Parisi et al., 2014), and learning to rank (LETOR) based methods (Bhowmik and Ghosh, 2017). Combination

rank fusion is a family of heuristic rank ensemble methods, which directly take some simple calculations like summation of base predictions as the ensemble predictions (Fox and Shaw, 1994). Borda rank fusion generates the ensemble ranking list according to the summation of each item's Borda counts in base ranking lists, which are determined by the positions of items in base ranking lists (Aslam and Montague, 2001). Although simple, combination rank fusion and Borda rank fusion have achieved competitive results on some ranking tasks (Cormack et al., 2009; Valcarce et al., 2017). Several methods tried to make use of extra information to enhance the performance of rank aggregation. For example, Liang et al. (2018b) assumed that similar items should have similar fusion scores, and adopted a manifold learning framework for rank aggregation. Bhowmik and Ghosh (2017) proposed to use item attributes to augment standard rank aggregation frameworks. However, these methods rely on unsupervised and heuristic schemes. Therefore, the ensemble results may be misled by some irrelevant information.

Supervised ranking ensemble tries to learn another model to combine base ranking models according to a set of labeled data, which could be the total or partial ordering of candidate items. Liu et al. (2007) proposed a weighted ranking ensemble framework to minimize the disagreements between ensemble ranking lists and the labeled data. Qin et al. (2010) proposed a ranking ensemble method to combine the list-wise Luce model and the Mallows model, which models the generation of a permutation (i.e., a ranking list) as a stagewise process. Vargas Muñoz et al. (2015) proposed a supervised genetic programming approach to search combinations of rank aggregation techniques for a given set of ranking lists. In addition, several studies used supervised ranking ensemble for link prediction (Pujari and Kanawati, 2012), influencer prediction in social networks (Subbian and Melville, 2011), and preference fusion (Volkovs and Zemel, 2013; Volkovs et al., 2012).

Most of existing ranking ensemble methods assume that the importance of base models for ensemble is query-independent. A few methods attempted to learn query-dependent importance of base models, such as Semi-supervised ensemble ranking (Hoi and Jin, 2008) and L2RA (Macdonald and Ounis, 2011). However, these methods rely on auxiliary information, e.g., similarities of queries' content (Hoi and Jin, 2008) or query-document's features (Macdonald and Ounis, 2011), which are not suitable for tasks without these information, such as recommender systems with only users' behavior information.

3. Preliminaries

In this section, we first formulate the problem of ranking ensemble learning, and then introduce a list-wise likelihood function suitable for this kind of ensemble learning.

3.1 Problem Definition

Let $\mathcal{F} = \{f^1, f^2, \cdots, f^T\}$ be a set of base ranking models, $\mathcal{O} = \{o_1, o_2, \cdots, o_K\}$ be the set of candidate items, and $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$ be the whole set of queries. $L_j^{(q)} = [f^1(o_j, q), \cdots, f^T(o_j, q)]$ denotes the predictions of base models $\mathcal{F}$ on item $o_j \in \mathcal{O}$ (e.g., scores or ranking positions) for query $q \in \mathcal{Q}$. Table 1 lists the symbols and notations used in this paper.

Table 1: Symbols and notations

Notation	Description
$\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$	The whole set of queries
$\mathcal{O} = \{o_1, o_2, \cdots, o_K\}$	The set of candidate items
$\mathcal{F} = \{f^1, f^2, \cdots, f^T\}$	A set of base models
$L_j^{(q)} = [f^1(o_j, q), \cdots, f^T(o_j, q)]$	Predictions on item o_j by base models $\mathcal{F}$ for query q
$\pi^{(q)} = [o_{(1)} \succ \cdots \succ o_{(K)}]$	Ground truth order of candidate items for query q
$x_j^{(q)} = f(o_{\pi_{(j)}}, q)$	Predicted score by model f on the j-th item $o_{(j)}$ of $\pi^{(q)}$
$\boldsymbol{x^{ens}} = \sum_{i=1}^{T} w_i \boldsymbol{x^i}$	Predicted scores by the ensemble model f^{ens} on all items
$y_j^{(q)} = x_j^{(q)} - x_{j+1}^{(q)}$	Margin between adjacent x_j and x_{j+1}
$\boldsymbol{\Delta x}_{m:K} = (x_m - x_{m+1}, \cdots, x_m - x_K)$	Difference vector of $(x_m, \cdots, x_K)$
$\boldsymbol{\Omega y}_{m:K} = (y_m, y_m + y_{m+1}, \cdots, \sum_{n=m}^{K-1} y_n)$	Composition vector of $(y_m, \cdots, y_{K-1})$
$l(\boldsymbol{x}, \pi^{(q)}, q)$	Loss function of P-L model with score vector $\boldsymbol{x}$
$l^*(\boldsymbol{y}, \pi^{(q)}, q)$	Loss function of P-L model with margin vector $\boldsymbol{y}$
$\mathcal{D} = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q \in \mathcal{Q}}$	The whole set of query specific samples
$\mathcal{D}_{\mathcal{T}} = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q \in \mathcal{Q}'}$	A training set of query specific samples, where $\mathcal{Q}' \subseteq \mathcal{Q}$
$E_{\mathcal{D}}(\cdot)$	The expectation on the sample space $\mathcal{D}$
$w_{i,u}$	Weight of the base model f^i w.r.t. query q_u for ensemble
$\alpha_{i,u} = w_{i,u} / (\sum_i w_{i,u})$	Contribution of the i-th base model f^i for ensemble

The goal of ranking ensemble is to construct or learn an ensemble function f^{ens} to fuse the prediction results of base ranking models. In this paper, we focus on the weighted ranking ensemble, i.e.,

$$f^{ens}(o_j, q) = \sum_{i=1}^{T} w_i f^i(o_j, q)$$

where $w_i \in \mathbb{R}$ denotes the weight of the i-th base model f^i. The goal of weighted ranking ensemble learning is to learn the weights $(w_1, w_2, \cdots, w_T)$ based on a training sample set $\mathcal{D}_{\mathcal{T}} = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q \in \mathcal{Q}'}$, where $\pi^{(q)}$ denotes the (partial) ground truth order of candidate items for query q and $\mathcal{Q}' \subseteq \mathcal{Q}$ denotes a subset of queries in the sample space.

3.2 List-wise Likelihood Function

To fully make use of the list-wise ranking information and pay more attention to the top positions of ranking lists, we adopt Plackett-Luce (P-L) model to define the likelihood function of ranking predictions. P-L model (Plackett, 1975) defines a parameterized probability distribution over the permutations of items given the prediction scores.

For a given query q with the ground truth ranking list π on an item set $\mathcal{O} = \{o_1, o_2, \cdots, o_K\}$, the likelihood function for the prediction scores generated by a model f can be defined as follows,

$$P(\pi | \mathcal{O}, q, f) = \prod_{m=1}^{K} \frac{\exp(f(o_{\pi_{(m)}}, q))}{\sum_{n=m}^{K} \exp(f(o_{\pi_{(n)}}, q))}$$

where $f(o_{\pi_{(m)}}, q)$ denotes the prediction score generated by model f for query q on the m-th item in the ranking list π. Maximizing the likelihood function $P(\pi | \mathcal{O}, q, f)$ enforces that

items at higher positions of the ground truth ranking list π should have larger prediction scores (i.e., $f(o_{\pi_{(i)}}, q) > f(o_{\pi_{(j)}}, q)$ if $i < j$) and that the relative margins between adjacent items $o_{\pi_{(m)}}$ and $o_{\pi_{(m+1)}}$ (i.e., $[\exp(f(o_{\pi_{(m)}}, q)) - \exp(f(o_{\pi_{(m+1)}}, q))]/\exp(f(o_{\pi_{(m)}}, q))$) are as large as possible.

To simplify the representations, we denote

$$x_m^{(q)} = f(o_{\pi_{(m)}}, q)$$

We will omit the superscript (q) without affecting the understanding given the context in the rest of this paper. Furthermore, we take the logarithm of the likelihood function and simplify it with a pseudo-sigmoid function as follows:

$$L(\pi|\mathcal{O}, q, f) = \ln P(\pi|\mathcal{O}, q, f) = \ln \prod_{m=1}^{K} \frac{\exp(x_m)}{\sum_{n=m}^{K} \exp(x_n)} = \sum_{m=1}^{K} \ln \frac{\exp(x_m)}{\sum_{n=m}^{K} \exp(x_n)}$$
$$= \sum_{m=1}^{K} \ln \frac{1}{\sum_{n=m}^{K} \exp(x_n - x_m)} = \sum_{m=1}^{K} \ln \frac{1}{1 + \sum_{n=m+1}^{K} \exp(-(x_m - x_n))} = \sum_{m=1}^{K} \ln \sigma_m(\Delta \boldsymbol{x}_{m:K})$$

where $\Delta \boldsymbol{x}_{m:K} = (x_m - x_{m+1}, \cdots, x_m - x_K) \in \mathbb{R}^{K-m}$ is named as **difference vector** and $\sigma_m(\cdot)$ is a multivariate function named as **pseudo-sigmoid function** that maps $\boldsymbol{z} \in \mathbb{R}^{K-m} \to \mathbb{R}$, i.e.,

$$\sigma_m(\boldsymbol{z}) = \frac{1}{1 + \sum_{n=1}^{K-m} \exp(-z_n)}$$

The pseudo-sigmoid function $\sigma_m(\boldsymbol{z})$ degrades into the standard sigmoid function when $m = K - 1$.

Therefore, we can denote the P-L likelihood function $L(\pi|\mathcal{O}, q, f)$ as a function of the difference vectors, i.e.,

$$L(\pi|\mathcal{O}, q, f) = g(\Delta \boldsymbol{x}) = \sum_{m=1}^{K} g_m(\Delta \boldsymbol{x}_{m:K}) \tag{1}$$

where $g_m(\Delta \boldsymbol{x}_{m:K}) = \ln \sigma_m(\Delta \boldsymbol{x}_{m:K})$, $\Delta \boldsymbol{x} = \Delta \boldsymbol{x}_{1:K} \bigoplus \Delta \boldsymbol{x}_{2:K} \bigoplus \cdots \bigoplus \Delta \boldsymbol{x}_{K-1:K}$ and $\bigoplus$ denotes the concatenation operation of vectors.

4. The Proposed Theory and Method

In this section, we first generalize the ambiguity decomposition theory from regression ensemble to ranking ensemble. Then, we propose an explicit diversity measure for ranking ensemble based on the generalized theory. Finally, we adopt an adaptive learning scheme to model and learn query-dependent ensemble weights to further improve the performance of ensemble model.

4.1 Generalized Ambiguity Decomposition Theory for Ranking Ensemble

The classic ambiguity decomposition theory proves the effectiveness of regression ensemble whose performance is measured by the squared loss, which is a point-wise loss function. In

this section, we generalize the ambiguity decomposition theory to ranking ensemble whose performance is measured by a list-wise loss function. This generalized theory provides evidence for the design of better ranking ensemble methods and diversity measures.

To reveal the relationship between the losses of base ranking models and the loss of ensemble model, we adopt P-L model to define the ranking loss function, which is defined as follows,

$$l(\boldsymbol{x}, \pi, q) = -L(\pi|O, q, f) = -g(\Delta\boldsymbol{x}) = \sum_{m=1}^{K} \ln(1 + \sum_{n=m+1}^{K} \exp(-(x_m - x_n))) \geq 0 \qquad (2)$$

where $L(\pi|O, q, f)$ is the P-L likelihood function defined in Equation (1), $x_m = f(o_{\pi_{(m)}}, q)$ denotes the prediction score generated by model f for query q on the m-th item in the ground truth ranking list π, and $\boldsymbol{x} = [x_m|m = 1, \cdots, K]$ denotes the score vector.

To simplify the proof of the generalized ambiguity decomposition theory, we first give two lemmas about the properties of the difference vectors $\boldsymbol{\Delta x}_{m:K}$ and the Hessian matrix of the **logarithm pseudo-sigmoid function**

$$g_m(\boldsymbol{z}) = \ln(1/(1 + \sum_{n=1}^{|\boldsymbol{z}|} \exp(-z_n)))$$

The proof of these lemmas are given in the Appendix.

Lemma 1 ***Property of Difference Vectors*** $\boldsymbol{\Delta x}_{m:K}$**.** *With a set of base ranking models* $\{f^1, \cdots, f^T\}$ *and a weighted ensemble model* $f^{ens}(o_{(j)}, q) = \sum_{i=1}^{T} w_i f^i(o_{(j)}, q)$*, we have* $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K} = \sum_{i=1}^{T} w_i \boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$*, where* $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}$ *and* $\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$ *denote the difference vectors of ensemble model* f^{ens} *and base model* f^i*, respectively.*

Lemma 2 ***Property of Hessian Matrix.*** *The Hessian matrix* $H_m(\boldsymbol{z})$ *of the logarithm pseudo-sigmoid function* $g_m(\boldsymbol{z}) = \ln(1/(1 + \sum_{n=1}^{|\boldsymbol{z}|} \exp(-z_n)))$ *is a negative definite matrix, where* $\boldsymbol{z} \in \mathbb{R}^r$*.*

Based on these two lemmas, we can prove a generalized ambiguity decomposition theory for ranking ensemble.

Theorem 1 ***Generalized Ambiguity Decomposition Theory for Ranking Ensemble.*** *Given a set of base ranking models* $\{f^1, \cdots, f^T\}$ *and a weighted ensemble model* $f^{ens} = \sum_{i=1}^{T} w_i f^i$ *with* $\sum_{i=1}^{T} w_i \neq 0$*, the expected P-L loss of the ensemble model* f^{ens} *on a sample space* $\mathcal{D} = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q \in \mathcal{Q}}$ *can be decomposed into two components as follows,*

$$\underbrace{E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{ens}}, \pi, q))}_{E} = \underbrace{\sum_{i=1}^{T} \alpha_i E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{i}}, \pi, q))}_{\bar{E}} - \underbrace{\sum_{i=1}^{T} \alpha_i E_{\mathcal{D}}(\sum_{m=1}^{K} A_{im})}_{\bar{A}}$$

where $\alpha_i = w_i/(\sum_{i=1}^{T} w_i)$ *and*

$$A_{im} = -\frac{1}{2}[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}]^{\intercal} H_m(\boldsymbol{\Delta \widetilde{x}}^{\boldsymbol{i}}_{m:K})[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}] \geq 0 \qquad (3)$$

$\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$ and $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}$ denote the difference vectors of base model f^i and the ensemble model f^{ens}, respectively. $H_m(\cdot)$ denotes the Hessian matrix of the logarithm pseudo-sigmoid function $g_m(\cdot)$ and $\boldsymbol{\Delta}\widetilde{\boldsymbol{x}}^{\boldsymbol{i}}_{m:K}$ is an interpolation point between $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}$ and $\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$. $E_{\mathcal{D}}(\cdot)$ denotes the expectation on the sample space $\mathcal{D} = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q\in\mathcal{Q}}$, where $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$ denotes the whole set of queries. $\bar{A} = \sum_{i=1}^{T} \alpha_i E_{\mathcal{D}}(\sum_{m=1}^{K} A_{im})$ is called the ambiguity of ensemble.

Proof For each base model f^i, we expand the logarithm pseudo-sigmoid function $g_m(\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K})$ around point $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}$ by Taylor expansion with Lagrange type reminder as follows,

$$\begin{aligned} g_m(\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}) =& g_m(\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}) + [\nabla g_m(\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K})]^{\intercal}[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}] + \\ & \frac{1}{2!}[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}]^{\intercal} H_m(\boldsymbol{\Delta}\widetilde{\boldsymbol{x}}^{\boldsymbol{i}}{}_{m:K})[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}] \end{aligned}$$

where vector $\nabla g_m(\boldsymbol{z})$ and Hessian matrix $H_m(\boldsymbol{z})$ denote the first-order and the second-order partial derivatives of the multivariate function $g_m(\boldsymbol{z}) = \ln(1/(1 + \sum_{n=1}^{|\boldsymbol{z}|} \exp(-z_n)))$, respectively. $\boldsymbol{\Delta}\widetilde{\boldsymbol{x}}^{\boldsymbol{i}}_{m:K}$ is an interpolation point between $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}$ and $\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$.

According to Equation (1), we can get $g(\Delta \boldsymbol{x}^{\boldsymbol{i}})$ by summarizing $\{g_m(\Delta \boldsymbol{x}^{\boldsymbol{i}}_{m:K}) | m = 1, \cdots, K\}$, i.e.,

$$g(\boldsymbol{\Delta x}^{\boldsymbol{i}}) = \sum_{m=1}^{K} g_m(\Delta \boldsymbol{x}^{\boldsymbol{i}}_{m:K}) = g(\boldsymbol{\Delta x}^{\boldsymbol{ens}}) + \sum_{m=1}^{K} [\nabla g_m(\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K})]^{\intercal}[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}] - \sum_{m=1}^{K} A_{im}$$

where

$$A_{im} = -\frac{1}{2}[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}]^{\intercal} H_m(\boldsymbol{\Delta}\widetilde{\boldsymbol{x}}^{\boldsymbol{i}}_{m:K})[\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}]$$

Based on Lemma 1, i.e., $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K} = \sum_{i=1}^{T} w_i \boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$, we can get the weighted summarization of $g(\boldsymbol{\Delta x}^{\boldsymbol{i}})$ as follows,

$$\begin{aligned} \sum_{i=1}^{T} w_i g(\boldsymbol{\Delta x}^{\boldsymbol{i}}) =& \sum_{i=1}^{T} w_i g(\boldsymbol{\Delta x}^{\boldsymbol{ens}}) + \sum_{i=1}^{T} w_i \sum_{m=1}^{K} [\nabla g_m(\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K})]^{\intercal}(\boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K}) - \sum_{i=1}^{T} w_i \sum_{m=1}^{K} A_{im} \\ =& \sum_{i=1}^{T} w_i g(\boldsymbol{\Delta x}^{\boldsymbol{ens}}) + \sum_{m=1}^{K} [\nabla g_m(\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K})]^{\intercal} \underbrace{(\sum_{i=1}^{T} w_i \boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K} - \boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K})}_{0} - \sum_{i=1}^{T} w_i \sum_{m=1}^{K} A_{im} \end{aligned}$$

i.e.,

$$\sum_{i=1}^{T} w_i \underbrace{g(\boldsymbol{\Delta x}^{\boldsymbol{i}})}_{=-l(\boldsymbol{x}^{\boldsymbol{i}},\pi,q)} = \sum_{i=1}^{T} w_i \underbrace{g(\boldsymbol{\Delta x}^{\boldsymbol{ens}})}_{=-l(\boldsymbol{x}^{\boldsymbol{ens}},\pi,q)} - \sum_{i=1}^{T} w_i \sum_{m=1}^{K} A_{im}$$

Due to $l(\boldsymbol{x}, \pi, q) = -g(\boldsymbol{\Delta x})$ and $\sum_{i=1}^{T} w_i \neq 0$, we have

$$\begin{aligned} l(\boldsymbol{x}^{\boldsymbol{ens}}, \pi, q) =& \sum_{i=1}^{T} \frac{w_i}{\sum_{i=1}^{T} w_i} l(\boldsymbol{x}^{\boldsymbol{i}}, \pi, q) - \sum_{i=1}^{T} \sum_{m=1}^{K} \frac{w_i}{\sum_{i=1}^{T} w_i} A_{im} \\ =& \sum_{i=1}^{T} \alpha_i l(\boldsymbol{x}^{\boldsymbol{i}}, \pi, q) - \sum_{i=1}^{T} \sum_{m=1}^{K} \alpha_i A_{im} \end{aligned}$$

where $\alpha_i = w_i/(\sum_{i=1}^{T} w_i)$. Taking expectation of $l(\boldsymbol{x}^{\boldsymbol{ens}}, \pi, q)$ on the sample space $\mathcal{D}$, Equation (1) is proved. Lemma 2 proves that $H_m(\cdot)$ is negative definite, thus $A_{im} \geq 0$ is established. ■

This theory proves that the prediction loss of ranking ensemble model can be decomposed into the weighted average loss of base models and an ambiguity term related to the diversity of ensemble, which shows an important relation among the ensemble model, base models, and the ambiguity for ranking ensemble. Furthermore, the ambiguity decomposition theories for regression and classification ensemble can be seen as special cases of our theory. For example, our theory can degenerate into the classic ambiguity decomposition theory for regression ensemble by setting π as the regression value and using the squared loss function. And our theory can also degenerate into an ambiguity decomposition theory for classification ensemble by setting π as the classification label and using a convex loss function as in Jiang et al. (2017).

Similar as the ambiguity decomposition theory for regression/classification tasks, the generalized ambiguity decomposition theory proves the effectiveness of ensemble learning for ranking tasks. Suppose we have $\boldsymbol{x}^{\boldsymbol{ens'}} = \sum_{i=1}^{T} w_i' \boldsymbol{x}^{\boldsymbol{i}}$ where $w_i' \geq 0$ and $\sum_{i=1}^{T} w_i' = 1$, we have $\alpha_i = w_i'/(\sum_{i=1}^{T} w_i') = w_i'$. According to Theorem 1, we have $A_{im} \geq 0$ and can further get the following inequality,

$$E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{ens'}}, \pi, q)) \leq \sum_{i=1}^{T} w_i' E_{\mathcal{D}}(l(\boldsymbol{x_i}, \pi, q)) \tag{4}$$

Therefore, the prediction loss of the ranking ensemble model is no greater than the average loss of base models with non-negative weights w_i'. In other words, the weighted ensemble of several base ranking models is better than randomly selecting the predictions of one base model according to the sampling weights w_i'. As base ranking models' results may contain "wrong" partial orders, we consider both "positive" and "negative" results with $w_i \in \mathbb{R}$ as in classification tasks (Jiang et al., 2017). Therefore, we can get a better ensemble model, because

$$\min_{\substack{w_1,\cdots,w_T \in \mathbb{R} \\ \sum_{i=1}^{T} w_i \neq 0}} E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{ens}}, \pi, q)) \leq \min_{\substack{w_1',\cdots,w_T' \geq 0 \\ \sum_{i=1}^{T} w_i' = 1}} E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{ens'}}, \pi, q)) \tag{5}$$

where $\boldsymbol{x}^{\boldsymbol{ens}} = \sum_{i=1}^{T} w_i \boldsymbol{x}^{\boldsymbol{i}}$. This inequality can be easily established because the set of real numbers $\mathbb{R}$ contains the set of non-negative real numbers and $w_i \in \mathbb{R}$ can take the value of $w_i' \geq 0$ if the less-than relation is not satisfied. According to Equation (4) and Equation (5), we can get the following inequality,

$$\min_{\substack{w_1,\cdots,w_T \in \mathbb{R} \\ \sum_{i=1}^{T} w_i \neq 0}} E_{\mathcal{D}}(l(\boldsymbol{x}^{\boldsymbol{ens}}, \pi, q)) \leq \min_{\substack{w_1',\cdots,w_T' \geq 0 \\ \sum_{i=1}^{T} w_i' = 1}} \sum_{i=1}^{T} w_i' E_{\mathcal{D}}(l(\boldsymbol{x_i}, \pi, q))$$

which proves the effectiveness of ensemble learning for ranking tasks.

The expected loss of weighted ranking ensemble model depends on an ambiguity term $\bar{A} = \sum_{i=1}^{T} \alpha_i E_{\mathcal{D}}(\sum_{m=1}^{K} A_{im})$, which is related to the diversity of ensemble. When there is no diversity, i.e., all base ranking models are the same, we get $\boldsymbol{\Delta x}_{m:K}^{\boldsymbol{i}} = \boldsymbol{\Delta x}_{m:K}^{\boldsymbol{ens}}$ and

$\bar{A} = 0$. A_{im} in Equation (3) can be seen as a measure of the disagreement between the i-th base ranking model f^i and the ensemble model f^{ens} for the prediction of the m-th position in a given ranking list. The larger of the ambiguity $\bar{A}$ means the greater of the ensemble diversity. However, it is difficult to calculate the ambiguity $\bar{A}$ directly.

4.2 Diversity Measure for Ranking Ensemble

Based on the generalized ambiguity decomposition theory proved in the previous section, we can derive an estimation of the ambiguity term $\bar{A}$ as an explicit diversity measure for ranking ensemble, which can be used to improve the performance of ensemble model.

Inspired by the margin theory (Gao and Zhou, 2013), we reformulate the loss function defined in Equation (2) with margins. Specifically, we denote the margins $\boldsymbol{y} = (y_1, \cdots, y_{K-1})$ as the distances between the prediction scores of adjacent items in the given ranking list π, i.e., $y_i = x_i - x_{i+1}$. Therefore, $x_m - x_n = y_m + \cdots + y_{n-1}$ and we can replace the loss function $l(\boldsymbol{x}, \pi, q)$ of difference vector $\boldsymbol{x}$ with the loss function $l^*(\boldsymbol{y}, \pi, q)$ of margin vector $\boldsymbol{y}$, i.e.,

$$l^*(\boldsymbol{y}, \pi, q) = \sum_{m=1}^{K} \underbrace{\ln(1 + \sum_{n=m+1}^{K} \exp(-(y_m + \cdots + y_{n-1})))}_{\triangleq -g_m(\boldsymbol{\Omega y}_{m:K})}$$

where $\boldsymbol{\Omega y}_{m:K} = (y_m, y_m + y_{m+1}, \cdots, y_m + \cdots + y_{K-1})$ denotes the composition vector of margins $(y_m, y_{m+1}, \cdots, y_{K-1})$. In addition, we denote $\boldsymbol{y^i}$ and $\boldsymbol{y^{ens}}$ as the special cases of margin vector $\boldsymbol{y}$ when $f = f^i$ and $f = f^{ens}$, respectively. $\boldsymbol{\Omega y}^{\boldsymbol{ens}}_{m:K}$ and $\boldsymbol{\Omega y}^{\boldsymbol{i}}_{m:K}$ follow the same idea with the definition of composition vector $\boldsymbol{\Omega y}_{m:K}$.

To simplify the proof of the proposed diversity measure for ranking ensemble, we first give a lemma about the property of Hessian matrix of the loss function $l^*(\boldsymbol{y}, \pi, q)$. The proof of this lemma is given in the Appendix.

Lemma 3 *The Hessian matrix $H^*(\boldsymbol{y})$ of function $l^*(\boldsymbol{y}, \pi, q)$ is a positive definite matrix, which is the sum of positive ($m = 1$) and semi-positive ($m > 1$) definite matrices $\{H^*_m(\boldsymbol{y}) | m = 1, \cdots, K\}$ that are the Hessian matrices of $\{l^*_m(\boldsymbol{y}, \pi, q) = -g_m(\boldsymbol{\Omega y}_{m:K}) | m = 1, \cdots, K\}$, i.e.,*

$$H^*(\boldsymbol{y}) = \sum_{m=1}^{K} H^*_m(\boldsymbol{y})$$

$$H^*_m(\boldsymbol{y}) = \begin{bmatrix} 0_{(m-1)\times(m-1)} & 0_{(m-1)\times(K-m)} \\ 0_{(K-m)\times(m-1)} & -\frac{\partial g_m(\Omega \boldsymbol{y}_{m:K})}{\partial \boldsymbol{y}_{m:K} \partial \boldsymbol{y}^\intercal_{m:K}} \end{bmatrix}$$

Based on Lemma 3 and Theorem 1, we can derive an estimation for the ambiguity term $\bar{A}$, which can be used as a diversity measure for ranking ensemble.

Theorem 2 ***Diversity Measure for Ranking Ensemble.*** *The ambiguity term $\bar{A}$ in Theorem 1 has an estimation $\overline{Div}$ as follows,*

$$\bar{A} \approx \overline{Div} = E_{\mathcal{D}}\left(\sum_i^T \alpha_i C'_i + (\frac{1}{\sum_i w_i} - 1) \cdot [\nabla l^*(0, \pi, q)]^\intercal \boldsymbol{y^{ens}} - \frac{1}{2} [\boldsymbol{y^{ens}}]^\intercal H^*(c_d \cdot \boldsymbol{y^{ens}}) [\boldsymbol{y^{ens}}] \right) \tag{6}$$

where $\boldsymbol{y^{ens}} = (y_1^{ens}, \cdots, y_{K-1}^{ens})$ *denotes the margin vector of the predictions of the weighted ensemble model* $f^{ens} = \sum_{i=1}^{T} w_i f^i$, $\alpha_i = w_i / (\sum_{i=1}^{T} w_i)$, $\nabla l^*(0, \pi, q)$ *denotes the partial derivative of loss function* $l^*(\boldsymbol{y}, \pi, q)$ *at point* $\mathbf{0}$, $C_i' = \frac{1}{2}[\boldsymbol{y^i}]^\intercal H^*(c_i \cdot \boldsymbol{y^i})[\boldsymbol{y^i}]$ *is a constant term,* $c_1, \cdots, c_T, c_d \in [0, 1]$ *are interpolation constants, and* $H^*(\cdot)$ *is the Hessian matrix of loss function* $l^*(\boldsymbol{y}, \pi, q)$*, which is a positive definite matrix and can be calculated as follows,*

$$H^*(\boldsymbol{y})[i,j] = \frac{\partial l^*(\boldsymbol{y}, \pi, q)}{\partial y_i \partial y_j} = \sum_{m=1}^{\min(i,j)} \frac{h(m, \max(i,j), K)[1 + h(m, m, \min(i,j))]}{[1 + h(m, m, K)]^2} \tag{7}$$

where

$$h(m, s, t) = \sum_{n=s+1}^{t} \exp(-\sum_{k=m}^{n-1} y_k)$$

Proof For each base model f^i and the weighted ensemble model f^{ens}, their loss functions can be expanded near zero according to the Taylor's Theorem as follows,

$$l^*(\boldsymbol{y^i}, \pi, q) = l^*(0, \pi, q) + [\nabla l^*(0, \pi, q)]^\intercal \boldsymbol{y^i} + \frac{1}{2!}[\boldsymbol{y^i}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{i}}) \boldsymbol{y^i}$$
$$l^*(\boldsymbol{y^{ens}}, \pi, q) = l^*(0, \pi, q) + [\nabla l^*(0, \pi, q)]^\intercal \boldsymbol{y^{ens}} + \frac{1}{2!}[\boldsymbol{y^{ens}}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{ens}}) \boldsymbol{y^{ens}}$$

where $H^*(\cdot)$ is a positive definite matrix according to Lemma 3.

Furthermore, we can get the ambiguity term $\bar{A}$ as follows according to Theorem 1:

$$\begin{aligned}
\bar{A} &= E_{\mathcal{D}}(\sum_i^T \alpha_i l(\boldsymbol{x^i}, \pi, q) - l(\boldsymbol{x^{ens}}, \pi, q)) = E_{\mathcal{D}}(\sum_i^T \alpha_i l^*(\boldsymbol{y^i}, \pi, q) - l^*(\boldsymbol{y^{ens}}, \pi, q)) \\
&= E_{\mathcal{D}}\Bigg(\underbrace{(\sum_i \alpha_i - 1)}_{=0} l^*(0, \pi, q) + [\nabla l^*(0, \pi, q)]^\intercal \underbrace{[\sum_i \alpha_i \boldsymbol{y^i} - \boldsymbol{y^{ens}}]}_{=\boldsymbol{y^{ens}}/(\sum_i w_i) - \boldsymbol{y^{ens}}} \\
&\qquad + \sum_i^T \frac{\alpha_i}{2}[\boldsymbol{y^i}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{i}})[\boldsymbol{y^i}] - \frac{1}{2}[\boldsymbol{y^{ens}}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{ens}})[\boldsymbol{y^{ens}}] \Bigg) \\
&= E_{\mathcal{D}}\Bigg((\frac{1}{\sum_i w_i} - 1) \cdot [\nabla l^*(0, \pi, q)]^\intercal \boldsymbol{y^{ens}} + \sum_i^T \alpha_i C_i - \frac{1}{2}[\boldsymbol{y^{ens}}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{ens}})[\boldsymbol{y^{ens}}] \Bigg)
\end{aligned}$$

where $C_i = \frac{1}{2}[\boldsymbol{y^i}]^\intercal H^*(\widetilde{\boldsymbol{y}}^{\boldsymbol{i}})[\boldsymbol{y^i}]$ denotes a constant which is independent with the weight vector $\boldsymbol{w}$. Due to the interpolation $\widetilde{\boldsymbol{y}}^{\boldsymbol{i}}$ is a point between 0 to $\boldsymbol{y^i}$, we can estimate it by $\widetilde{\boldsymbol{y}}^{\boldsymbol{i}} \approx c_i \cdot \boldsymbol{y^i}$, where $c_i \in [0, 1]$ is an interpolation constant. Similarly, we can get $\widetilde{\boldsymbol{y}}^{\boldsymbol{ens}} \approx c_d \cdot \boldsymbol{y^{ens}}$, where $c_d \in [0, 1]$ is an interpolation constant. Therefore, we can get an estimation of the ambiguity term $\bar{A}$ as follows,

$$\bar{A} \approx \overline{Div} \triangleq E_{\mathcal{D}}\Bigg(\sum_i^T \alpha_i C_i' + (\frac{1}{\sum_i w_i} - 1) \cdot [\nabla l^*(0, \pi, q)]^\intercal \boldsymbol{y^{ens}} - \frac{1}{2}[\boldsymbol{y^{ens}}]^\intercal H^*(c_d \cdot \boldsymbol{y^{ens}})[\boldsymbol{y^{ens}}] \Bigg)$$

where $C_i' = \frac{1}{2}[\boldsymbol{y^i}]^\intercal H^*(c_i \cdot \boldsymbol{y^i})[\boldsymbol{y^i}]$. The calculation formula (Equation 7) of Hessian matrix $H^*(\cdot)$ and partial derivative $\nabla l^*(0, \pi, q)$ are derived in Appendix.

■

This theorem provides an explicit diversity measure for ranking ensemble, which is related to the margin vector $\boldsymbol{y^{ens}}$ and the weight vector $\boldsymbol{w}$ according to Equation (6). Taking weight vector $\boldsymbol{w}$ as the parameter, the diversity term can be directly adopted into a diversity-based ensemble learning scheme to improve the performance of ensemble model.

4.3 Adaptive Weight Learning

Base models may have different predictive abilities on different queries. To explore the specificity of different queries for ranking ensemble, we propose to learn query-dependent weights $w_{i,u}$ of base models, i.e.,

$$f^{ens}(o_j, q_u) = \sum_i w_{i,u} f^i(o_j, q_u)$$

where $w_{i,u}$ represents the weight of the i-th base model f^i with respect to query q_u. It is difficult or even impossible to learn the query-dependent weights $w_{i,u}$ directly because of the large number and the uncertainty of queries in real applications. To deal with this problem, we adopt an attention mechanism to adaptively learn the query-dependent weights. The attention mechanism aims at learning the relevance between base models and queries according to their embedding representations in the same latent space, which describes the importance of base models with respect to queries. Specifically, we embed each base model f^i and query q_u into the same latent space, denoted as $\boldsymbol{p}_i \in \mathbb{R}^d$ and $\boldsymbol{h}_u(\boldsymbol{\theta}) \in \mathbb{R}^d$ respectively, where $\boldsymbol{\theta}$ denotes the parameters for query embedding.

The attentional weight $w_{i,u}$ can be calculated based on the similarity between the latent representations of base model f^i and query q_u, e.g.,

$$w_{i,u} =< \boldsymbol{p}_i, \boldsymbol{h}_u(\boldsymbol{\theta}) > \tag{8}$$

where $< \cdot, \cdot >$ denotes the inner product of two vectors. When the queries appear in both the training data set and the test data set, such as users in RS tasks, we can directly learn their embeddings $\boldsymbol{h_u}(\boldsymbol{\theta}) = \boldsymbol{h_u} \in \mathbb{R}^d$ as the LFM model (Koren et al., 2009) without knowing or using their content features. When the test queries are not in the training set, such as for IR tasks, we can make use of queries' content features such as topic words to construct queries' embeddings $\boldsymbol{h_u}(\boldsymbol{\theta}) \in \mathbb{R}^d$ as the AEM model (Liu et al., 2019). The base models' embedding $\boldsymbol{p_i}$ and the queries' embeddings $\boldsymbol{h_u}$ (or parameters $\boldsymbol{\theta}$) are parameters of our ensemble model.

To make sure that the query-dependent weights can fit into the proposed theories and methods, we derive a corollary as follows:

Corollary 4 *The query-dependent weights $w_{i,u}$ can fit into the generalized ambiguity decomposition theory (Theorem 1) and the explicit diversity measure (Theorem 2) by replacing w_i with $w_{i,u}$.*

Proof Due to the elements of sample space $\mathcal{D}$ are query q_u-specific, we can replace w_i with $w_{i,u}$ in the proof of Theorem 1 and Theorem 2. Therefore, this corollary is established. ■

We have two goals to learn the query-dependent weights. One goal is to minimize the prediction loss $\sum_{\mathcal{D}_T}\{l^*(\boldsymbol{y^{ens}}, \pi, q)\}$ on the training data set $\mathcal{D}_T$, i.e., maximizing the fitting degree of the ensemble model on the training data. However, simply minimizing the prediction loss on the training data set $\mathcal{D}_T \subset \mathcal{D}$ may cause the overfitting problem. To alleviate this problem, we add another goal to maximize the diversity of ensemble $\overline{Div}$ as defined in Theorem 2, which can help to reduce the generalization error on the whole sample space $\mathcal{D}$ and get a better ensemble model. Detailed analysis of how the diversity mechanism works is shown in Section 5.8. As a result, the whole objective function for ensemble model learning is defined as follows,

$$\min \text{aWELv-OPT} = \sum_{\mathcal{D}_T}(l^*(\boldsymbol{y^{ens}}, \pi, q) - \lambda\overline{Div}) \tag{9}$$

where $\mathcal{D}_T = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q\in\mathcal{Q}'}$ denotes a training sample set, and λ is a coefficient to trade off the two goals.

Algorithm 1 Parameter Learning Algorithm

Input: A training set $\mathcal{D}_T = \{L_1^{(q)}, \cdots, L_K^{(q)}, \pi^{(q)}\}_{q\in\mathcal{Q}'}$, partial-sampling number κ, embedding dimension d, initial learning rate τ, trade-off coefficient λ, interpolation constants $c_1, \cdots, c_T, c_d$, maximum iteration threshold ξ.

Output: Base models' embeddings $\{\boldsymbol{p_i}|f^i \in \mathcal{F}\}$, parameters $\boldsymbol{\theta}$ for query embedding $\{\boldsymbol{h_u}(\boldsymbol{\theta})|q_u \in \mathcal{Q}\}$.

Randomly initialize base models' embeddings $\{\boldsymbol{p_i}|f^i \in \mathcal{F}\}$ and parameters $\boldsymbol{\theta}$ for query embedding $\{\boldsymbol{h_u}(\boldsymbol{\theta})|q_u \in \mathcal{Q}\}$;
repeat
  Randomly select a sample $\{L_1^{(q_u)}, \cdots, L_K^{(q_u)}, \pi^{(q_u)}\} \in \mathcal{D}_T$;
  # Adding disturbance to avoid $\sum_{i=1}^T w_{i,u} = 0$:
  if $\sum_{i=1}^T < \boldsymbol{p_i}, \boldsymbol{h_u}(\boldsymbol{\theta}) >= 0$ **then**
    repeat
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\delta}$; # $\boldsymbol{\delta} \in \mathbb{R}^{|\boldsymbol{\theta}|}$ denotes small random disturbance;
    until $\sum_{i=1}^T < \boldsymbol{p_i}, \boldsymbol{h_u}(\boldsymbol{\theta}) >\neq 0$
  end if
  # Partial-sampling strategy:
  Sample κ integers $\Gamma = [\gamma_1, \cdots, \gamma_\kappa]$ s.t. $\pi^{(q_u)}(\gamma_1) \succ \cdots \succ \pi^{(q_u)}(\gamma_\kappa)$;
  Select the partial ranking lists: $\mathfrak{s} = \{L_1^{(q_u)}[\Gamma], \cdots, L_K^{(q_u)}[\Gamma], \pi^{(q_u)}[\Gamma]\}$;
  # Update parameters:
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau\frac{\partial \text{aWELv-OPT}}{\partial \boldsymbol{h}_u(\boldsymbol{\theta})} \cdot \frac{\partial \boldsymbol{h}_u(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$; # Update the parameters for query embedding;
  $\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i - \tau\frac{\partial \text{aWELv-OPT}}{\partial \boldsymbol{p}_i}$ for $f_i \in \mathcal{F}$; # Update the embedding of base models;
  Adjust the learning rate τ by AdaGrad;
until convergence or iteration number reaches the maximum iteration threshold ξ

For model training, i.e., learning base models' embeddings $\boldsymbol{p_i}$ and queries' embeddings $\boldsymbol{h_u}$, we optimize the objective function Equation (9) by stochastic gradient descent (SGD) and adaptively adjust the learning rate by AdaGrad (Duchi et al., 2011). In real-world scenarios, the ground truth ranking lists usually contain lots of items which leads to tremendous computational complexity for model training, and the full ranking list for each query may be not available (Liu et al., 2007). To deal with this problem, we adopt a partial-sampling strategy to sample the items in the ranking lists for model training as Xia et al. (2008). To avoid $\sum_{i=1}^{T} w_{i,u} = 0$ which is intractable for the proposed diversity measure, we add small random disturbance to the parameters when $\sum_{i=1}^{T} w_{i,u} = 0$. Specifically, the pseudocode of the algorithm for parameter learning is shown in Algorithm 1.

5. Experiments

In this section, we conduct several experiments to study the following research questions:

- **RQ1**: Whether the proposed ranking ensemble method outperforms state-of-the-art ensemble methods on different ranking tasks, such as RSs and IR?
- **RQ2**: Whether ranking ensemble with consideration of list-wise information outperforms those with consideration of only pair-wise information?
- **RQ3**: Whether the proposed ranking ensemble method benefits from diversity-based learning?
- **RQ4**: Whether the proposed ranking ensemble method benefits from the adaptive weight learning?
- **RQ5**: To what extent the proposed ranking ensemble method outperforms the base models?
- **RQ6**: Whether the proposed ranking ensemble method with several simple base models can outperform state-of-the-art ranking models?

5.1 Baselines and Evaluation Metrics

To evaluate the performance of the proposed ensemble method, we compare it with several competitive baseline methods, including Combination Fusion Family (Fox and Shaw, 1994), Borda Count Fusion (Aslam and Montague, 2001), aMANx (Liang et al., 2018b), BaggingReg (Louppe and Geurts, 2012), AdaBoostReg (Drucker, 1997), SER (Hoi and Jin, 2008), ConvexDS (Yin et al., 2014), and AEM (Liu et al., 2019).

- **Combination Fusion Family (Fox and Shaw, 1994)**: It is a family of unsupervised ranking ensemble methods, including CombSum, CombANZ and CombMNZ. Previous work (Cormack et al., 2009; Valcarce et al., 2017) showed that Combination Fusion Family achieved the best performance in their work for RS and IR tasks.
- **Borda Count (Aslam and Montague, 2001)**: It is an unsupervised ranking ensemble method, which directly calculates the ensemble score of each item according to its positions in the predicted ranking lists by base models.

- **aMANx (Liang et al., 2018b)**: It is a manifold-based unsupervised ranking ensemble method. In RSs, we calculate items' relevance matrix based on their cosine similarities as in Deshpande and Karypis (2004).

- **BaggingReg (Louppe and Geurts, 2012)**: It is a bagging based ensemble method for regression tasks.

- **AdaBoostReg (Drucker, 1997)**: It is a boosting based weighted ensemble method for regression tasks.

- **SER (Hoi and Jin, 2008)**: It is a supervised weighted ensemble method for ranking tasks, which tries to minimize the disagreements between the ground truth ranking lists and the ranking lists generated by the ensemble model.

- **ConvexDS (Yin et al., 2014)**: It is a weighted ensemble learning method for classification tasks, which learns the weights of base models with consideration of both sparsity and diversity. As ConvexDS is designed for classification tasks, we transform the ranking tasks into classification tasks as in Pan et al. (2008).

- **AEM (Liu et al., 2019)**: It is a weighted ensemble learning method for classification tasks, which learns personalized classifier weights based on instances' features. We also transform the ranking tasks into classification tasks as for ConvexDS.

Among these methods, ConvexDS and AEM are ensemble methods for classification tasks while BaggingReg and AdaBoostReg are ensemble methods for regression tasks. These methods have been adopted into ranking tasks for comprehensive comparisons. The other baselines are ranking ensemble methods with a supervised or unsupervised learning scheme.

We adopt two commonly used ranking evaluation metrics, Recall and NDCG (Normalized Discounted Cumulative Gain) to measure the performance of different methods. The former is used to measure methods' ability of distinguishing related items, while the latter focuses on the positions of related items in the ranking list. Specifically, we adopt the same formulations as in Shenbin et al. (2020), i.e., Recall is formulated as follows:

$$Recall@n = \frac{1}{|\mathcal{Q}|} \sum_{q_u \in \mathcal{Q}} \frac{\#Hit@n(q_u)}{\min(\#GT(q_u), n)}$$

where $\#Hit@n(q_u)$ denotes the number of ground truth items in the top-n ranking list for query q_u and $\#GT(q_u)$ denotes the number of all ground truth items for query q_u. NDCG is formulated as follows:

$$NDCG@n = \frac{1}{|\mathcal{Q}|} \sum_{q_u \in \mathcal{Q}} \frac{DCG@n(q_u)}{IDCG@n(q_u)}$$

where

$$DCG@n(q_u) = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$IDCG@n(q_u) = \sum_{i=1}^{\#GT(q_u)} \frac{1}{\log_2(i+1)}$$

where $rel_i \in \{0, 1\}$ denotes whether the item at the i-th position of the predicted ranking list is a ground truth item for query q_u, and $IDCG@n(q_u)$ denotes the ideal $DCG@n(q_u)$ which is used to normalize DCG into $[0, 1]$.

5.2 Application to Recommender Systems

The goal of this experiment is to evaluate the performance and effectiveness of the proposed ranking ensemble method for top-n recommendation tasks. We assume the sample space $\mathcal{D}$ includes partial orderings of items e.g., $o_{(2)} \succ o_{(1)} \succ o_{(0)}$, where $o_{(2)} \in Rating(5)$, $o_{(1)} \in Rating(1 \sim 4)$ and $o_{(0)} \in Rating(NULL)$. The goal of top-n recommendation is to recommend n items for each user that the user may prefer but have not seen, i.e., those items $o \in Rating(5)$ in the test set.

Each user u is regarded as a "query" that triggers the recommender systems, which can be embedded into a latent space i.e., $\boldsymbol{h}_u(\boldsymbol{\theta}) = \boldsymbol{h}_u \in \mathbb{R}^d$. For each base ranking model f^i, we can embed it into the same latent space as users, i.e., $\boldsymbol{p}_i \in \mathbb{R}^d$. The embedding representations of users and base ranking models are model parameters to be learned.

5.2.1 Experimental Design

We conduct experiments on four real world data sets, including Amazon[1]-Movies, Amazon-Kindle, Epinion[2], and ML-20m[3]. The Amazon-Movies and Amazon-Kindle data sets were collected from Amazon.com, which contain users' ratings on videos (e.g., movies, TV shows) and e-book respectively. The Epinion data set was collected from a rating website, which contains users' ratings on many different types of items (software, music, television shows, and etc.). The ML-20m data set was collected from the MovieLens website (movielens.umn.edu), which contains users' ratings on movies. Therefore, the task of RS on these data sets is to predict users' behaviors and recommend items to users that users may prefer and interact with. We only keep users who have engaged at least 25 items in these data sets except for 5 items in ML-20m data set as in Liang et al. (2018a). The characteristics of the four data sets are summarized in Table 2.

Table 2: Statistics of the experimental data sets for RS tasks

Data set	#Users	#Items	#Interactions
Amazon_Movies	11,396	7,074	473,878
Amazon_Kindle	9,371	2,932	119,509
Epinion	8,577	3,769	203,275
ML-20m	136,677	20,720	9,990,682

Each data set is randomly divided into two subsets: 80% for training and 20% for testing. The training set is used for training the base models and learning the parameters of ensemble models. The test set is used for evaluating the performance of ensemble models.

1. http://jmcauley.ucsd.edu/data/amazon/links.html
2. http://www.trustlet.org/wiki/Epinions_datasets
3. https://grouplens.org/datasets/movielens/

In addition, we split 20% data from the training set as the validation set, which is used for hyper-parameter selection.

We adopt seven existing methods as base recommendation methods, including one popularity based method (Tang and Wang, 2018), two memory-based methods and four model-based methods. The two memory-based methods are User-KNN and Item-KNN (Deshpande and Karypis, 2004). The four model-based methods consist of two point-wise collaborative filtering methods MF (Koren et al., 2009) and CDAE (Wu et al., 2016), a pair-wise learning to rank method BPR (Rendle et al., 2009), and a list-wise learning to rank method ListRank-MF (Shi et al., 2010). To enhance the diversity of base models, we generate five base models with different random initializations of model parameters for each model-based method. To unify the predictions generated by different base models, we transform each ranking list generated by a base model into a relative score list by $s(p) = 1/(p + \mu)$ as in Cormack et al. (2009), where p denotes the position in ranking list and μ denotes a constant which is set as 10.

For all baseline models, we tune the parameters to their best. For the proposed model, we set embedding dimension $d = 16$, initial learning rating $\tau = 0.01$, maximum iteration threshold $\xi = 100 \cdot |\mathcal{D}_T|$ for all data sets except $\xi = 300 \cdot |\mathcal{D}_T|$ for ML-20m, trade-off coefficient $\lambda = 0.1$ and interpolation constants[4] $c_1 = \cdots = c_T = c_d = 0.25$ and 1.0 for Amazon (Movie and Kindle) and ML-20m data sets, $\lambda = 1.0$ and $c_1 = \cdots = c_T = c_d = 0.5$ for Epinion data set. Experimental results are recorded as the average of five runs with different random initializations of model parameters.

5.2.2 Experimental Results

Table 3 shows the performance of different ensemble methods for RS tasks. To make the table more notable, we bold the best results and underline the best baseline results for each data set with one specific evaluation metric. Experimental results show that the proposed method aWELv performs significantly better than all baselines in all cases, which demonstrates the effectiveness of the proposed ensemble method (RQ1). In addition, we find that most of supervised ensemble methods, including ConvexDS, AEM, SER and our method, perform better than the unsupervised ensemble methods, including combination fusion family, Borda count and aMANx, in most cases. It confirms the necessity of making use of the supervised information for model ensemble. Methods with consideration of ensemble diversity (ConvexDS and AEM) perform the best among baselines, which confirms the importance of diversity for ensemble learning.

5.3 Application to Information Retrieval

The goal of this experiment is to evaluate the performance and effectiveness of the proposed ensemble method for IR tasks. We assume the sample space $\mathcal{D}$ includes partial orderings of documents $o_{(2)} \succ o_{(1)} \succ o_{(0)}$, where $o_{(2)} \in \mathit{Label}(2)$, $o_{(1)} \in \mathit{Label}(1)$ and $o_{(0)} \in \mathit{Label}(0)$. The goal of IR systems is to find the n most relevant documents $o \in \mathit{Label}(2)$ for each query.

4. To reduce the number of hyper-parameters for efficient tuning, we set $c_1 = \cdots = c_T = c_d$ inspired by (Jiang et al., 2017).

Table 3: Performance of different ensemble methods for RS tasks. * indicates statistically significant improvement by an independent-samples t-test ($p < 0.01$) compared with all baseline methods.

Method	Amazon_movie		Amazon_Kindle		Epinion		ML-20m	
	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5
CombSum	0.0354	0.0245	0.1088	0.0489	0.0480	0.0241	0.2716	0.2803
CombANZ	0.0351	0.0246	0.0830	0.0411	0.0454	0.0235	0.1910	0.1987
CombMNZ	0.0362	0.0243	0.1191	0.0520	0.0461	0.0231	0.2853	0.2953
Borda Count	0.0329	0.0255	0.1049	0.0587	0.0567	0.0315	0.2900	0.3041
BaggingReg	0.0294	0.0207	0.1400	0.0674	0.0474	0.0241	0.1825	0.1793
AdaBoostReg	0.0413	0.0279	0.1579	0.0810	0.0533	0.0275	0.3130	0.3209
aMANx	0.0382	0.0267	0.1767	0.0901	0.0586	0.0323	0.2713	0.2802
SER	0.0441	0.0317	0.1760	0.0897	0.0631	0.0349	0.3379	0.3474
ConvexDS	0.0537	0.0379	0.1855	0.0948	0.0674	0.0371	0.2845	0.2951
AEM	0.0589	0.0419	0.1799	0.0930	0.0710	0.0402	0.3621	0.3717
aWELv(ours)	**0.0629***	**0.0440***	**0.1912***	**0.0989***	**0.0794***	**0.0430***	**0.3695***	**0.3813***
Improve.	6.69%	5.00%	3.05%	4.26%	11.82%	7.13%	2.04%	2.56%

Due to the uncertainty of input queries to IR systems, we cannot directly learn the query-level embedding representations during the training phase, e.g., some test queries may never appear in the training set. Therefore, we appeal to the topics of queries. Let the topics of each query q_u denote as a set of words $\boldsymbol{V}_u = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{|\boldsymbol{V}_u|}\}$. The embedding representations of topic words $\boldsymbol{v}_i \in \mathbb{R}^{100}$ are 100-dimension vectors pre-trained by GloVe (Pennington et al., 2014). We calculate the embedding representation $\boldsymbol{h}_u$ of query q_u as a linear aggregation of its topic words' embedding representations as Su et al. (2021) and Li et al. (2020) with a neural network, i.e.,

$$\boldsymbol{h}_u(\boldsymbol{\theta}) = J \cdot \frac{1}{|\boldsymbol{V}_u|} \sum_{\boldsymbol{v}_i \in \boldsymbol{V}_u} \boldsymbol{v}_i + \boldsymbol{b}$$

where $J \in \mathbb{R}^{100 \times d}$ and $\boldsymbol{b} \in \mathbb{R}^d$ are model parameters $\boldsymbol{\theta}$ to be learned.

5.3.1 Experimental Design

We conduct experiments on two benchmark data sets, MQ2007-agg and MQ2008-agg, which are provided by LETOR4.0 (Qin and Liu, 2013). These data sets contain queries associated with ranking lists generated by different base models. The characteristics of the two data sets are summarized in Table 4 .

For each data set, we divide it into two equal subsets as in Shen et al. (2014), one for learning the parameters of ensemble models (including 10% for hyper-parameter selection) and the other for evaluating the performance of ensemble models. Experimental results show that the proposed method performs well when embedding dimension $d = 5$, initial learning rating $\tau = 0.01$, interpolation constants $c_1 = \cdots = c_T = c_d = 0.5$, trade-off coefficient $\lambda = 0.1$ and $\lambda = 0.01$, maximum iteration threshold $\xi = 200 \cdot |\mathcal{D}_T|$ and $\xi = 500 \cdot |\mathcal{D}_T|$ for

Table 4: Statistics of the experimental data sets for IR tasks

Data set	#Queries	#Documents	#Interactions
MQ2007	1,692	65,323	69,623
MQ2008	784	14,384	15,211

Table 5: Performance of different ensemble methods for IR tasks. * indicates statistically significant improvement by an independent-samples t-test ($p < 0.01$) compared with all baseline methods.

Method	MQ2007		MQ2008	
	Recall@5	NDCG@5	Recall@5	NDCG@5
CombSum	0.2925	0.2466	0.6880	0.5399
CombANZ	0.2972	0.2496	0.4417	0.2601
CombMNZ	0.3435	0.2806	0.6934	0.5367
Borda Count	0.2961	0.2508	0.6947	0.5430
BaggingReg	0.3171	0.2638	0.7172	0.5482
AdaBoostReg	0.3268	0.2756	0.6833	0.5303
SER	0.3220	0.2719	0.6876	0.5389
ConvexDS	0.3020	0.2555	0.6983	0.5454
AEM	0.3373	0.2822	0.7289	0.5763
aWELv(ours)	**0.3497***	**0.2919***	**0.7518***	**0.5939***
Improve.	1.82%	3.44%	3.14%	3.05%

MQ2007 and MQ2008, respectively. Experimental results are recorded as the average of five runs with different random initializations of model parameters.

5.3.2 Experimental Results

Table 5 shows the performance of different ensemble methods for IR tasks. The same as for RS tasks, the proposed method aWELv performs significantly better than all baseline methods in all cases, which confirms the research assumption RQ1. In addition, the supervised ensemble method AEM performs the best among baselines in most cases, which also confirms the necessity of making use of supervised information for model ensemble. Compared with AEM which makes use of only point-wise loss information, the proposed method aWELv performs significantly better, which confirms the necessity of making use of the list-wise information for ranking ensemble.

5.4 Ablation Experiment

The goal of this experiment is to evaluate the effectiveness of module design of the proposed method aWELv.

5.4.1 Experimental Design

There are several key modules in aWELv, including list-wise loss function, diversity-based learning and adaptive weight learning. To evaluate the effectiveness of these modules, we degrade or remove them from aWELv and compare the performance of these variants.

- **WEP**: **W**eighted **E**nsemble with **P**air-wise loss is a special case of the proposed method. The P-L model degrades into the pair-wise Bradley-Terry model when $K = 2$, and then the proposed method aggregates ranking lists using only pair-wise information. In addition, it ignores the diversity and the adaptive weight learning components.

- **WEL**: **W**eighted **E**nsemble with consideration of **L**ist-wise loss is a special case of the proposed method. It aggregates ranking lists with consideration of list-wise information. It ignores the diversity and the adaptive weight learning components.

- **WELv**: WEL with di**v**ersity-based learning is a special case of the proposed method. It takes into consideration of the diversity of ensemble while still ignoring the adaptive weight learning component, i.e., all the queries share an ensemble weight set $\{w_i|i = 1, \cdots, T\}$ and $f^{ens} = \sum_{i=1}^{T} w_i f^i$.

- **aWEL**: Adaptive WEL is a special case of the proposed method. It takes into consideration of the **a**daptive weight learning component while ignoring the diversity component, i.e., $\lambda = 0$.

- **aWELv**: It is the proposed method, which takes into consideration of both the di**v**ersity and the **a**daptive weight learning components. Instead of directly learning a shared weight set $\{w_i|i = 1, \cdots, T\}$ as in WELv, aWELv tries to learn query-specific weights $w_{i,u}$ with the embeddings of queries q_u and base models f^i according to Equation (8).

5.4.2 Experimental Results

Table 6 shows the performance of different special cases of the proposed method. From the experimental results, we can get the following conclusions. First, WEL with consideration of list-wise information performs better than WEP with consideration of only pair-wise information in most cases, which confirms that making use of list-wise information is important for ranking ensemble (RQ2). Second, WELv and aWELv with diversity-based learning perform better than WEL and aWEL in most cases, respectively. It not only confirms the generalized ambiguity decomposition theory, but also shows the effectiveness of the proposed diversity-based learning scheme (RQ3). Third, aWELv and aWEL with adaptive weight learning consistently outperform WELv and WEL, respectively. It not only confirms that each base model may have different predictive abilities on different queries, but also shows that the proposed ranking ensemble method benefits from adaptive weight learning (RQ4).

5.5 Effects of Different Parameter Settings

The goal of this experiment is to evaluate the effects of different parameter settings on the performance of our proposed ensemble method aWELv.

Table 6: Results of ablation experiments for RS and IR tasks. * indicates statistically significant improvement by an independent-samples t-test ($p < 0.01$) compared with the special cases of the proposed method.

Method	Amazon_movie		Amazon_Kindle		Epinion	
	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5
WEP	0.0418	0.0297	0.1826	0.0935	0.0613	0.0337
WEL	0.0555	0.0389	0.1869	0.0968	0.0725	0.0396
WELv	0.0574	0.0399	0.1883	0.0973	0.0731	0.0402
aWEL	0.0609	0.0427	0.1900	0.0982	0.0767	0.0420
aWELv	**0.0629***	**0.0440***	**0.1912***	**0.0989***	**0.0794***	**0.0430***

Method	ML-20m		MQ2007		MQ2008	
	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5
WEP	0.2830	0.2933	0.3081	0.2601	0.6997	0.5519
WEL	0.3650	0.3761	0.3097	0.2572	0.7022	0.5537
WELv	0.3679	0.3800	0.3162	0.2631	0.7007	0.5529
aWEL	0.3675	0.3784	0.3455	0.2826	0.7471	0.5924
aWELv	**0.3695***	**0.3813***	**0.3497***	**0.2919***	**0.7518**	**0.5939**

5.5.1 Experimental Design

We evaluate the effects of different settings of the key parameters on the performance of aWELv, including the trade-off coefficient λ and the interpolation constant $c_1 = \cdots = c_T = c_d = c$. The trade-off coefficient is chosen from $\lambda \in \{0, 0.001, 0.01, 0.1, 1, 5\}$. The interpolation constant is chosen from $c \in \{0, 0.25, 0.5, 0.75, 1\}$.

5.5.2 Experimental Results

Figure 1 shows the performance of aWELv measured by Recall with different settings of the trade-off coefficient λ and the interpolation constant c. The trade-off coefficient λ controls the relative importance of the diversity term $\overline{Div}$. The results show that it can improve the performance of aWELv with proper consideration of the proposed diversity measure. Too large or too small λ may degrade the performance of the proposed ranking ensemble method in most cases. For the interpolation constant c, the performance of aWELv is relatively stable with different settings of c, which shows the stability of the proposed interpolation method for ranking ensemble.

5.6 Comparison with Base Models

The goal of this experiment is to explore the gain of ranking ensemble with respect to base ranking models (RQ5).

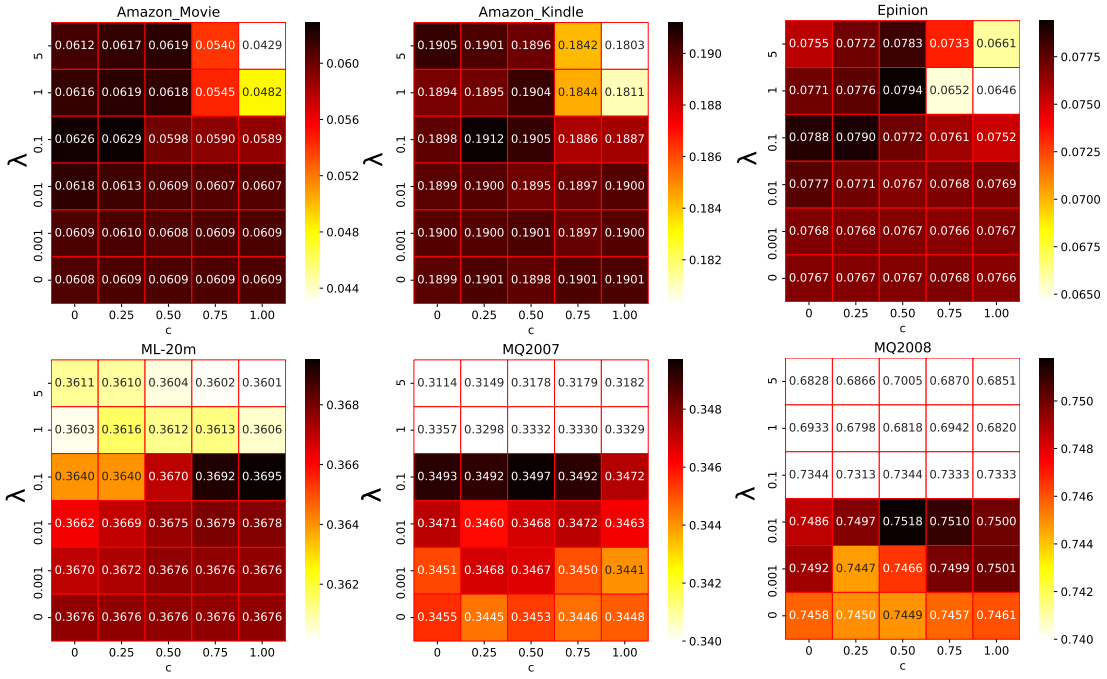


Figure 1: Effects of different settings of trade-off coefficient λ and interpolation constant c

5.6.1 Experimental Design

We compare the performance of our proposed ranking ensemble model with that of base models. For RS tasks, the base models include one popularity based method Heat, two memory-based methods User-KNN and Item-KNN, and four model-based methods MF, CDAE, BPR and ListRank-MF. For IR tasks, there are 21 and 25 base models in the MQ2007 and MQ2008 benchmark, respectively. Each base model can be seen as an individual predictor for RS or IR tasks.

5.6.2 Experimental Results

Table 7 and Table 8 show the performance of base models and our ensemble model for RS tasks and IR tasks, respectively. To make the tables more notable, we bold the best results and underline the best base model results for each data set with one specific evaluation metric. Experimental results show that the proposed ensemble method performs significantly better than all base models for different tasks on different data sets. This confirms that proper combination of multiple models can achieve better performance than any individual one, and also demonstrates the effectiveness of the proposed ensemble method.

5.7 Comparison with State-of-the-art Methods

The goal of this experiment is to evaluate whether the proposed ranking ensemble method with several simple base models can outperform state-of-the-art ranking models (RQ6).

Table 7: Performance of base models and our ensemble model for RS tasks. * indicates statistically significant improvement by an independent-samples t-test ($p < 0.01$) compared with base models.

Method	Amazon_movie		Amazon_Kindle		Epinion		ML-20m	
	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5	Recall@5	NDCG@5
Heat	0.0110	0.0082	0.0096	0.0056	0.0258	0.0153	0.1570	0.1625
UserKNN	0.0272	0.0189	0.1371	0.0717	0.0443	0.0243	0.3574	0.3704
ItemKNN	0.0421	0.0289	0.1421	0.0696	0.0581	0.0311	0.1435	0.1447
MF	0.0341	0.0246	0.1361	0.0670	0.0379	0.0206	0.2695	0.2695
CDAE	0.0109	0.0082	0.1273	0.0727	0.0354	0.0191	0.3383	0.3505
BPR	0.0375	0.0276	0.1323	0.0682	0.0504	0.0283	0.3138	0.3217
ListRank-MF	0.0425	0.0298	0.1420	0.0671	0.0509	0.0273	0.2035	0.2079
aWELv	**0.0629***	**0.0440***	**0.1912***	**0.0989***	**0.0794***	**0.0430***	**0.3695***	**0.3813***
Improve.	47.81%	47.42%	34.57%	36.02%	36.77%	38.52%	3.39%	2.93%

Table 8: Performance of base models and our ensemble model for IR tasks. * indicates statistically significant improvement by an one-sample t-test ($p < 0.01$) compared with base models.

Method	MQ2007		MQ2008		Method	MQ2007		MQ2008	
	Recall@5	NDCG@5	Recall@5	NDCG@5		Recall@5	NDCG@5	Recall@5	NDCG@5
BM1	0.1534	0.1178	0.4839	0.3295	BM14	0.1785	0.1393	0.5973	0.4380
BM2	0.1508	0.1158	0.5692	0.4042	BM15	0.2197	0.1722	0.6654	0.5320
BM3	0.1803	0.1451	0.5730	0.4100	BM16	0.1507	0.1195	0.5558	0.4181
BM4	0.2216	0.1806	0.5088	0.3682	BM17	0.1806	0.1399	0.5383	0.3658
BM5	0.2603	0.2079	0.4969	0.3509	BM18	0.2323	0.1751	0.6230	0.4704
BM6	0.2541	0.2107	0.6536	0.4864	BM19	0.2233	0.1786	0.6264	0.4899
BM7	0.2541	0.2107	0.4216	0.2902	BM20	0.2462	0.2037	0.3200	0.1870
BM8	0.2553	0.2102	0.6664	0.5074	BM21	0.2263	0.1849	0.5927	0.4421
BM9	0.2317	0.1868	0.6715	0.5175	BM22	NULL	NULL	0.6474	0.4948
BM10	0.1684	0.1376	0.6173	0.4433	BM23	NULL	NULL	0.4398	0.2853
BM11	0.1631	0.1248	0.6514	0.4940	BM24	NULL	NULL	0.3152	0.1805
BM12	0.2357	0.1882	0.4021	0.2496	BM25	NULL	NULL	0.3131	0.1762
BM13	0.1919	0.1510	0.3032	0.1787	aWELv	**0.3497***	**0.2919***	**0.7518***	**0.5939***
-	-	-	-	-	Improve.	34.37%	38.53%	11.95%	11.63%

5.7.1 Experimental Design

To compare our proposed model with state-of-the-art ranking methods, we test it on a well-studied benchmark for RS tasks as in (Liang et al., 2018a; Kim and Suh, 2019; Steck, 2019; Shenbin et al., 2020) with the ML-20m data set. The benchmark contains multiple state-of-the-art models for RS tasks, including Mult-VAE (Liang et al., 2018a), H+Vamp (Kim and Suh, 2019), ESAE (Steck, 2019), and RecVAE (Shenbin et al., 2020).

5.7.2 Experimental Results

Table 9 shows the performance of different methods. The proposed method aWELv, which aggregates several simple base models, can get comparable or even better performance than state-of-the-art deep models (RQ6). In addition, the experimental results show that the proposed method aWELv achieves better performance for top positions, e.g., measured by Recall@20 and NDCG, which is consistent with the characteristics of list-wise ranking ensemble that pays more attention to the top positions of ranking lists.

Table 9: Comparison with state-of-the-art methods for RS tasks

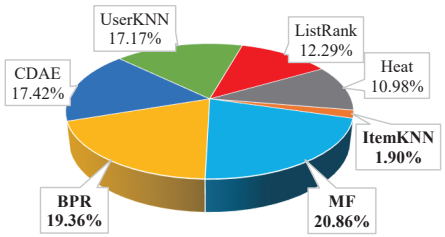
	Method	Recall@20	Recall@50	NDCG@50
base models	Heat	0.183	0.254	0.181
	UserKNN	0.403	0.519	0.397
	ItemKNN	0.164	0.223	0.156
	MF	0.366	0.501	0.347
	CDAE	0.377	0.481	0.372
	BPR	0.376	0.499	0.366
benchmark	Mult-VAE	0.395	0.537	0.426
	H+Vamp	0.413	0.551	0.445
	ESAE	0.391	0.521	0.420
	RecVAE	0.414	**0.553**	0.442
Ours	aWELv	**0.424**	0.543	**0.450**

5.8 Understanding Diversity Mechanism

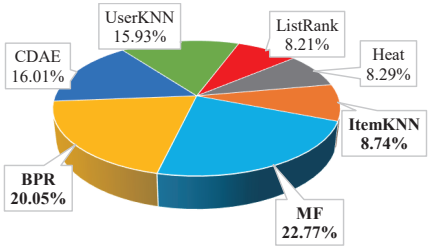
The goal of this experiment is to explore how the proposed diversity measure can help to improve the performance of the proposed ranking ensemble method.

5.8.1 Experimental Design

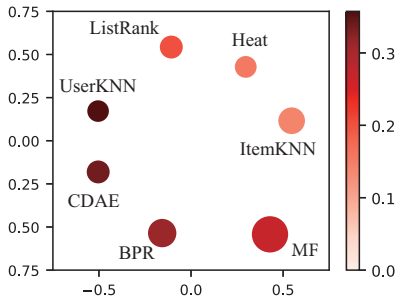
We explore how the proposed diversity measure works for ranking ensemble by a case study on the ML-20m data set. Specifically, we explicitly compare the learned ensemble weights by WELv with different settings of trade-off coefficient λ, and further analyze their relations with another ranking diversity measure, named Jaccard ranking distance. Given two ranking models A and B, their Jaccard ranking distance on a data set $\mathcal{D}_T$ for top-n



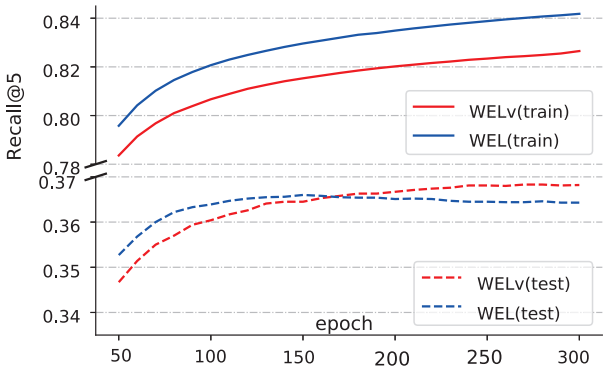
(a) Percentage contributions of base models $p_i = |w_i| / \sum_{i=1}^{T} |w_i|$ learned by WEL ($\lambda = 0.0$) .



(b) Percentage contributions of base models $p_i = |w_i| / \sum_{i=1}^{T} |w_i|$ learned by WELv ($\lambda = 0.1$) .



(c) Performance and distribution of base models. The color bar and the size of circle show the performance and the average distance to others, respectively.



(d) Performance of WEL (λ = 0.0) and WELv (λ= 0.1) on the training and test sets with increasing training iterations.

Figure 2: Effects of the proposed diversity measure for ranking ensemble learning.

recommendation is defined as follows,

$$d_J(A, B) = \frac{1}{|\mathcal{D}_T|} \sum_{q \in \mathcal{D}_T} \frac{1}{n} \sum_{l=1}^{n} (1 - \frac{|A(q,l) \cap B(q,l)|}{|A(q,l) \cup B(q,l)|}) \tag{10}$$

where $A(q,l)$ and $B(q,l)$ denote the item sets in the top-l ranking lists generated by ranking model A and ranking model B for query q, respectively. In addition, we explore and compare the performance of WEL (without considering the diversity term $\overline{Div}$) and WELv (with consideration of the diversity term $\overline{Div}$) on training and test sets with increasing training iterations.

5.8.2 Experimental Results

Figure 2(a) and Figure 2 (b) show the contribution of base models to the ensemble model learned by WEL and WELv, respectively. Figure 2(c) shows the distribution of base models generated by Multidimensional scaling (MDS) (O'Connell, 1999) according to their pairwise distances $d_J(A, B)$. Figure 2(d) shows the performance of WEL and WELv on training and test sets with increasing training iterations.

From Figure 2(a) and Figure 2(c), we can see that base models' contribution are roughly proportional to their performance when $\lambda = 0.0$, i.e., without considering the diversity term $\overline{Div}$. Compared with Figure 2(a), Figure 2(b) shows that the contributions of base ranking models MF, BPR and ItemKNN, which are far to others (i.e., different from other base models), increase with consideration of the diversity term $\overline{Div}$. On the other hand, the contributions of base ranking models CDAE, UserKNN, ListRank and HEAT, which are close to others (i.e., similar to other base models), decrease with consideration of the diversity term $\overline{Div}$. Finally, Figure 2(d) shows that the proposed method WELv can alleviate the overfitting problem with the help of the diversity term $\overline{Div}$.

6. Conclusion

This paper contributes to the field of ranking ensemble learning both theoretically and practically. First, we generalize the ambiguity decomposition theory from regression ensemble to ranking ensemble, which proves the effectiveness of ranking ensemble with consideration of list-wise ranking information. Second, to make use of the ambiguity term $\bar{A}$ of the generalized ambiguity decomposition theory to improve the performance of ensemble model, we derive an explicit diversity measure $\overline{Div}$ for ranking ensemble, which is an estimation of the ambiguity term $\bar{A}$. Ablation experiments (methods with/without diversity-based learning) in Section 5.4 and a case study in Section 5.8 show the effectiveness of the proposed diversity measure. Third, we adopt an adaptive weight learning scheme to learn query-dependent ensemble weights in a latent space. Extensive experiments on RS and IR tasks show the effectiveness of the proposed method compared with state-of-the-art methods.

In the future, we will evaluate the performance of the proposed method on other ranking tasks such as answer selection (Feng et al., 2015). In addition, we will further study the pruning of base models for ranking ensemble.

Acknowledgments

This work was partially sponsored by National Natural Science Fund of China (Grant No. 61232005) and Peking University Education Big Data Project (Grant No.2020YBC10).

Appendix A. The Proof of Lemmas and Theorems

In this appendix, we prove some lemmas and theorems used in the paper.

Lemma 1 Property of Difference Vectors. With a set of base ranking models $\{f^1, \cdots, f^T\}$ and a weighted ensemble model $f^{ens}(o_{(j)}, q) = \sum_{i=1}^{T} w_i f^i(o_{(j)}, q)$, we have $\boldsymbol{\Delta x}^{\boldsymbol{ens}}_{m:K} = \sum_{i=1}^{T} w_i \boldsymbol{\Delta x}^{\boldsymbol{i}}_{m:K}$.

Proof Due to $\boldsymbol{x^{ens}} = [f^{ens}(o_{\pi_{(m)}}, q)|m = 1, \cdots, K]$ and $\boldsymbol{x^i} = [f^i(o_{\pi_{(m)}}, q)|m = 1, \cdots, K]$, we have

$$\begin{aligned}\boldsymbol{\Delta x^{ens}_{m:K}} &= [\boldsymbol{x^{ens}_m} - \boldsymbol{x^{ens}_{m+1}}, \cdots, \boldsymbol{x^{ens}_m} - \boldsymbol{x^{ens}_K}] \\ &= [\sum_{i=1}^{T} w_i \boldsymbol{x^i_m} - \sum_{i=1}^{T} w_i \boldsymbol{x^i_{m+1}}, \cdots, \sum_{i=1}^{T} w_i \boldsymbol{x^i_m} - \sum_{i=1}^{T} w_i \boldsymbol{x^i_K}] \\ &= [\sum_{i=1}^{T} w_i (\boldsymbol{x^i_m} - \boldsymbol{x^i_{m+1}}), \cdots, \sum_{i=1}^{T} w_i (\boldsymbol{x^i_m} - \boldsymbol{x^i_K})] = \sum_{i=1}^{T} w_i \boldsymbol{\Delta x^i_{m:K}}\end{aligned}$$

■

Property 1 Judgement of Negative Definite Matrices. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is negative definite if and only if its k-th leading principle minor A^*_k satisfies $(-1)^k A^*_k > 0$ for all $k = 1, \cdots, n$. The k-th leading principle minor of matrix A is the determinant of its k-th leading principal submatrix which is defined as the $k \times k$ submatrix taken from the upper-left-hand corner of matrix A.

Proof Theories about the judgment of positive definite matrix in linear algebra show that symmetric matrix $B \in \mathbb{R}^{n\times n}$ is positive definite if and only if its k-th leading principle minor B^*_k satisfy $B^*_k > 0$ for all $k = 1, \cdots, n$.

Sufficiency. Suppose matrix $A \in \mathbb{R}^{n\times n}$ is negative definite. $B = -A \in \mathbb{R}^{n\times n}$ is positive definite due to $x^\intercal Bx = -x^\intercal Ax > 0$ for any x. Due to $A^*_k = (-B)^*_k = (-1)^k B^*_k$, we have $(-1)^k A^*_k = (-1)^{2k} B^*_k > 0$ for all $k = 1, \cdots, n$.

Necessity. Suppose $(-1)^k A^*_k > 0$ for all $k = 1, \cdots, n$ and $B = -A \in \mathbb{R}^{n\times n}$. Due to $B^*_k = (-A)^*_k = (-1)^k A^*_k > 0$, we have matrix B is positive definite. Therefore, A is negative definite due to $x^\intercal Ax = -x^\intercal Bx < 0$ for any x. ■

Lemma 2 Property of Hessian Matrix. Hessian matrix $H_m(\boldsymbol{z})$ of function $g_m(\boldsymbol{z}) = \ln(1/(1 + \sum_{n=1}^{r} \exp(-z_n)))$ is a negative definite Matrix, where $r = |z| = K - m$.

Proof The Hessian matrix of function $g_m(\boldsymbol{z})$ shows as follows:

$$H_m(\boldsymbol{z}) = \begin{bmatrix} \frac{\partial g_m(\boldsymbol{z})}{\partial^2 z_1} & \cdots & \frac{\partial g_m(\boldsymbol{z})}{\partial z_1 \partial z_r} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(\boldsymbol{z})}{\partial z_r \partial z_1} & \cdots & \frac{\partial g_m(\boldsymbol{z})}{\partial^2 z_r} \end{bmatrix}$$

where

$$\begin{aligned}\frac{\partial g_m(\boldsymbol{z})}{\partial^2 z_p} &= \sigma^2_m(\boldsymbol{z}) \exp(-2z_p) - \sigma_m(\boldsymbol{z}) \exp(-z_p) \\ \frac{\partial g_m(\boldsymbol{z})}{\partial z_p \partial z_q} &= \sigma^2_m(\boldsymbol{z}) \exp(-z_p - z_q) \\ \sigma_m(\boldsymbol{z}) &= 1/(1 + \sum_{n=1}^{r} \exp(-z_n))\end{aligned}$$

Let $s_p = \sigma_m(\boldsymbol{z})\exp(-z_p)$, we have:

$$H_m(\boldsymbol{z}) = \begin{bmatrix} s_1^2 - s_1 & \cdots & s_1 s_r \\ \vdots & \ddots & \vdots \\ s_r s_1 & \cdots & s_r^2 - s_r \end{bmatrix}$$

To prove $H_m(\boldsymbol{z})$ is negative definite, we calculate its k-th leading principle minor, which is denoted as $A_k^*(\boldsymbol{z})$:

$$A_k^*(\boldsymbol{z}) = \det(H_m(\boldsymbol{z})[:k,:k]) = \det(\begin{bmatrix} s_1^2 - s_1 & \cdots & s_1 s_k \\ \vdots & \ddots & \vdots \\ s_k s_1 & \cdots & s_k^2 - s_k \end{bmatrix})$$

Furthermore, we make some determinant transformations as follows:

$$\begin{aligned}
A_k^*(\boldsymbol{z}) &= \begin{vmatrix} s_1^2 - s_1 & \cdots & s_1 s_k \\ \vdots & \ddots & \vdots \\ s_k s_1 & \cdots & s_k^2 - s_k \end{vmatrix} \\
&\overset{(a)}{=} \begin{vmatrix} s_1 - 1 & \cdots & s_p & \cdots & s_k \\ \vdots & \ddots & \vdots & & \vdots \\ s_1 & \cdots & s_p - 1 & \cdots & s_k \\ \vdots & & \vdots & \ddots & \vdots \\ s_1 & \cdots & s_p & \cdots & s_k - 1 \end{vmatrix} \prod_{i=1}^{k} s_i \\
&\overset{(b)}{=} \begin{vmatrix} 1 & \cdots & s_p & \cdots & s_k \\ \vdots & \ddots & \vdots & & \vdots \\ 1 & \cdots & s_p - 1 & \cdots & s_k \\ \vdots & & \vdots & \ddots & \vdots \\ 1 & \cdots & s_p & \cdots & s_k - 1 \end{vmatrix} (\sum_{i=1}^{k} s_i - 1) \prod_{i=1}^{k} s_i \\
&\overset{(c)}{=} \begin{vmatrix} 1 & \cdots & s_p & \cdots & s_k \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -1 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & -1 \end{vmatrix} (\sum_{i=1}^{k} s_i - 1) \prod_{i=1}^{k} s_i \\
&= (-1)^{k-1} (\sum_{i=1}^{k} s_i - 1) \prod_{i=1}^{k} s_i
\end{aligned}$$

Step(a) extracts common factors s_i for each row. Step(b) adds each column to the first column and then extracts the first column. Step(c) subtracts each row from the first row.

Due to $s_i = \sigma_m(\boldsymbol{z})\exp(-z_i) > 0$ and $k \leq r$, we have

$$\begin{aligned}\sum_{i=1}^{k} s_i - 1 &= \sum_{i=1}^{k} \sigma_m(\boldsymbol{z})\exp(-z_i) - 1 \\ &= \frac{\sum_{i=1}^{k}\exp(-z_i)}{1+\sum_{i=1}^{r}\exp(-z_i)} - 1 \\ &< 0\end{aligned}$$

Furthermore, we have $(-1)^k A_k^*(z) > 0$ for $0 < k \leq r$. According to Property 1, $H_m(\boldsymbol{z})$ is a negative definite matrix. ■

Lemma 3 The Hessian matrix $H^*(\boldsymbol{y})$ of function $l^*(\boldsymbol{y}, \pi, q)$ is a positive definite matrix, which is the sum of positive ($m = 1$) and semi-positive ($m > 1$) definite matrices $H_m^*(\boldsymbol{y})$ that are the Hessian matrices of $l_m^*(\boldsymbol{y}, \pi, q) = -g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})$, i.e.,

$$H^*(\boldsymbol{y}) = \sum_{m=1}^{K} H_m^*(\boldsymbol{y})$$

$$H_m^*(\boldsymbol{y}) = \begin{bmatrix} 0_{(m-1)\times(m-1)} & 0_{(m-1)\times(K-m)} \\ 0_{(K-m)\times(m-1)} & -\frac{\partial g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})}{\partial \boldsymbol{y}_{m:K}\partial \boldsymbol{y}_{m:K}^\intercal} \end{bmatrix}$$

Proof Due to $x_m - x_n = y_m + \cdots + y_{n-1}$, we have $\boldsymbol{\Delta}\boldsymbol{x}_{m:K} = \boldsymbol{\Omega}\boldsymbol{y}_{m:K} = P_m \cdot \boldsymbol{y}_{m:K}$, where $\boldsymbol{y}_{m:K} = (y_m, \cdots, y_{K-1})$ and $P_m \in \mathbb{R}^{(K-m)\times(K-m)}$ is a lower triangular matrix with all non-zero elements equaling to 1. Therefore, the Hessian matrices $H_m^*(\boldsymbol{y})$ of $l_m^*(\boldsymbol{y}, \pi, q) = -g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})$ can be calculated as follows,

$$H_m^*(\boldsymbol{y}) = \begin{bmatrix} 0_{(m-1)\times(m-1)} & 0_{(m-1)\times(K-m)} \\ 0_{(K-m)\times(m-1)} & -\frac{\partial g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})}{\partial \boldsymbol{y}_{m:K}\partial \boldsymbol{y}_{m:K}^\intercal} \end{bmatrix}$$

where

$$\begin{aligned}\frac{\partial g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})}{\partial \boldsymbol{y}_{m:K}\partial \boldsymbol{y}_{m:K}^\intercal} &= \frac{\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}}{\partial \boldsymbol{y}_{m:K}} \cdot \frac{\partial g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})}{\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}^\intercal} \cdot \frac{\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}^\intercal}{\partial \boldsymbol{y}_{m:K}^\intercal} \\ &= P_m \cdot \frac{\partial g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})}{\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}\partial \boldsymbol{\Omega}\boldsymbol{y}_{m:K}^\intercal} \cdot P_m^\intercal\end{aligned}$$

Lemma 2 shows that the Hessian Matrix of function $g_m(\boldsymbol{\Delta}\boldsymbol{x}_{m:K}) = g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})$ is negative definite with $\boldsymbol{\Delta}\boldsymbol{x}_{m:K} = \boldsymbol{\Omega}\boldsymbol{y}_{m:K}$. Due to P_m is a full rank square matrix, $H_m^*(\boldsymbol{y})$ is a positive definite matrix when $m = 1$, otherwise it is a semi-positive definite matrix. This Lemma is clearly proved due to $l^*(\boldsymbol{y}, \pi, q) = \sum_{m=1}^{K} -g_m(\boldsymbol{\Omega}\boldsymbol{y}_{m:K})$. ■

The augment proof of Theorem 2. The partial derivative $\nabla l^*(\boldsymbol{y}, \pi, q)[i]$ and Hessian matrix $H^*(\cdot)$ of loss function $l^*(\boldsymbol{y}, \pi, q)$ can be calculated as follows,

$$\nabla l^*(\boldsymbol{y},\pi,q)[i] = \frac{\partial l^*(\boldsymbol{y},\pi,q)}{\partial y_i} = -\sum_{m=1}^{i}\frac{h(m,i,K)}{1+h(m,m,K)}$$

$$H^*(\boldsymbol{y})[i,j] = \frac{\partial l^*(\boldsymbol{y},\pi,q)}{\partial y_i \partial y_j} = \sum_{m=1}^{\min(i,j)}\frac{h(m,\max(i,j),K)[1+h(m,m,\min(i,j))]}{[1+h(m,m,K)]^2}$$

where

$$h(m,s,t) = \sum_{n=s+1}^{t}\exp(-(y_m+\cdots+y_{n-1}))$$

Proof For convenience, we define $h(m,s,t) = \sum_{n=s+1}^{t}\exp(-(y_m+\cdots+y_{n-1}))$, where $m \le s \le t$. Its partial derivatives show as follows:

$$\frac{\partial h(m,s,t)}{\partial y_i} = \begin{cases} 0 & i < m \text{ or } i > t \\ -h(m,s,t) & m \le i < s \\ -h(m,i,t) & s \le i \le t \end{cases}$$

Therefore, the loss function can be formulated as follows:

$$l^*(\boldsymbol{y},\pi,q) = \sum_{m=1}^{K}\ln(1+h(m,m,K))$$

The first-order partial derivatives of $l^*(\boldsymbol{y},\pi,q)$ are

$$\frac{\partial l^*(\boldsymbol{y},\pi,q)}{\partial y_i} = -\sum_{m=1}^{i}\frac{h(m,i,K)}{1+h(m,m,K)}$$

Furthermore, the second-order partial derivatives of $l^*(\boldsymbol{y},\pi,q)$ are

$$\begin{aligned}\frac{\partial l^*(\boldsymbol{y},\pi,q)}{\partial y_i \partial y_j} &= \sum_{m=1}^{\min(i,j)}\frac{h(m,\max(i,j),K)}{1+h(m,m,K)} - \sum_{m=1}^{\min(i,j)}\frac{h(m,i,K)h(m,j,K)}{[1+h(m,m,K)]^2} \\ &= \sum_{m=1}^{\min(i,j)}\frac{h(m,\max(i,j),K)[1+h(m,m,\min(i,j))+h(m,\min(i,j),K)]-h(m,i,K)h(m,j,K)}{[1+h(m,m,K)]^2} \\ &= \sum_{m=1}^{\min(i,j)}\frac{h(m,\max(i,j),K)[1+h(m,m,\min(i,j))]}{[1+h(m,m,K)]^2}\end{aligned}$$

■

References

Nir Ailon and Mehryar Mohri. Preference-based learning to rank. *Machine Learning*, 80 (2):189–211, 2010.

Javed A Aslam and Mark Montague. Models for metasearch. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 276–284, 2001.

Avradeep Bhowmik and Joydeep Ghosh. LETOR methods for unsupervised rank aggregation. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17)*, pages 1331–1340, 2017.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005a.

Gavin Brown, Jeremy L. Wyatt, and Peter Tiňo. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6:1621–1650, December 2005b.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 758–759, 2009.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

Harris Drucker. Improving regressors using boosting techniques. *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, pages 107–115, 08 1997.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61): 2121–2159, 2011.

Lanting Fang, Yong Luo, Kaiyu Feng, Kaiqi Zhao, and Aiqun Hu. Knowledge-enhanced ensemble learning for word embeddings. In *Proceedings of the 28th International Conference on World Wide Web (WWW'19)*, pages 427–437, 2019.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'15)*, pages 813–820, 2015.

Edward A Fox and Joseph A Shaw. Combination of multiple searches. In *Proceedings of the 2nd Text REtrieval Conference (TREC2)*, pages 243–252, 1994.

Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artificial Intelligence*, 203:1–18, 2013.

Steven CH Hoi and Rong Jin. Semi-supervised ensemble ranking. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI'08)*, pages 634–639, 2008.

Zhengshen Jiang, Hongzhi Liu, Bin Fu, and Zhonghai Wu. Generalized ambiguity decompositions for classification with applications in active learning and unsupervised ensemble pruning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, pages 2073–2079, 2017.

Daeryong Kim and Bongwon Suh. Enhancing vaes for collaborative filtering: Flexible priors & gating mechanisms. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys'19)*, pages 403–407, 2019.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems (NeurIPS'00)*, pages 231–238, 1995.

Ludmila I Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, 2014.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

B. Li, H. Zhou, J. He, M. Wang, and L. Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 9119–9130, 2020.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 27th International Conference on World Wide Web (WWW'18)*, pages 689–698, 2018a.

Shangsong Liang, Ilya Markov, Zhaochun Ren, and Maarten de Rijke. Manifold learning for rank aggregation. In *Proceedings of the 27th International Conference on World Wide Web (WWW'18)*, pages 1735–1744, 2018b.

Shili Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010.

Hongzhi Liu, Yingpeng Du, and Zhonghai Wu. AEM: Attentional ensemble model for personalized classifier weight learning. *Pattern Recognition*, 96:106976, 2019.

Yuting Liu, Tieyan Liu, Tao Qin, Zhiming Ma, and Hang Li. Supervised rank aggregation. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*, pages 481–490, 2007.

Gilles Louppe and Pierre Geurts. Ensembles on random patches. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD'12)*, pages 346–361, 2012.

Craig Macdonald and Iadh Ounis. Learning models for ranking aggregates. In *European Conference on Information Retrieval (ECIR'11)*, pages 517–529, 2011.

Balachandran Manavalan, Shaherin Basith, Tae Hwan Shin, Da Yeon Lee, Leyi Wei, and Gwang Lee. 4mcpred-el: An ensemble learning framework for identification of dna n4-methylcytosine sites in the mouse genome. *Cells*, 8(11), 2019.

Ann A O'Connell. Modern multidimensional scaling: Theory and applications. *Journal of the American Statistical Association*, 94(445):338–340, 1999.

Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM'17)*, pages 502–511, 2008.

Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 1532–1543, 2014.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

Manisha Pujari and Rushed Kanawati. Supervised rank aggregation approach for link prediction in complex networks. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*, pages 1189–1196, 2012.

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

Tao Qin, Xiubo Geng, and Tie-Yan Liu. A new probabilistic model for rank aggregation. In *Advances in Neural Information Processing Systems (NeurIPS'10)*, pages 1948–1956, 2010.

Shebuti Rayana and Leman Akoglu. Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data*, 10(4):1–33, 2016.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09)*, pages 452–461, 2009.

Friedhelm Schwenker. Ensemble methods: Foundations and algorithms. *IEEE Computational Intelligence Magazine*, 8(1):77–79, 2013.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM'14)*, pages 101–110, 2014.

Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM'20)*, pages 528–536, 2020.

Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM Conference on Recommender Systems (RecSys'10)*, pages 269–272, 2010.

Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *Proceedings of the 28th International Conference on World Wide Web (WWW'19)*, pages 3251–3257, 2019.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Karthik Subbian and Prem Melville. Supervised rank aggregation for predicting influencers in twitter. In *IEEE Third International Conference on Social Computing*, pages 661–665, 2011.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM'18)*, pages 565–573, 2018.

Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, pages 90–95, 1996.

Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. Combining top-n recommenders with metasearch algorithms. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*, pages 805–808, 2017.

Javier Alvaro Vargas Muñoz, Ricardo da Silva Torres, and Marcos André Gonçalves. A soft computing approach for learning to aggregate rankings. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*, pages 83–92, 2015.

Maksims N Volkovs and Richard S Zemel. CRF framework for supervised preference aggregation. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM'13)*, pages 89–98, 2013.

Maksims N Volkovs and Richard S Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15(1):1135–1176, 2014.

Maksims N Volkovs, Hugo Larochelle, and Richard S Zemel. Learning to rank by aggregating expert preferences. In *Proceedings of the 21st ACM International Conference on Information & Knowledge Management (CIKM'12)*, pages 843–851, 2012.

Shuaiqiang Wang, Shanshan Huang, Tie-Yan Liu, Jun Ma, Zhumin Chen, and Jari Veijalainen. Ranking-oriented collaborative filtering: A listwise approach. *ACM Transactions on Information Systems*, 35(2):1–28, 2016.

Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the nineth ACM International Conference on Web Search and Data Mining (WSDM'16)*, pages 153–162, 2016.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, pages 1192–1199, 2008.

Xu-Cheng Yin, Kaizhu Huang, Chun Yang, and Hong-Wei Hao. Convex ensemble learning with sparsity and diversity. *Information Fusion*, 20:49–59, 2014.

Yankun Yu, Huan Liu, Minghan Fu, Jun Chen, Xiyao Wang, and Keyan Wang. A two-branch neural network for non-homogeneous dehazing via ensemble learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, pages 193–202, 2021.