Query-Level Stability and Generalization in Learning to Rank

Yanyan Lan* LANYANYAN@AMSS.AC.CN
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, P. R. China.

Tie-Yan Liu TYLIU@MICROSOFT.COM
Microsoft Research Asia, Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing, 100190, P. R. China.

Tao Qin* QINSHITAO99@MAILS.THU.EDU.CN
Department of Electronic Engineering, Tsinghua University, Beijing, 100084, P. R. China.

Zhiming Ma MAZM@AMT.AC.CN
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, P. R. China.

Hang Li HANGLI@MICROSOFT.COM
Microsoft Research Asia, Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing, 100190, P. R. China.

Abstract

This paper is concerned with the generalization ability of learning to rank algorithms for information retrieval (IR). We point out that the key for addressing the learning problem is to look at it from the viewpoint of *query*. We define a number of new concepts, including query-level loss, query-level risk, and query-level stability. We then analyze the generalization ability of learning to rank algorithms by giving query-level generalization bounds to them using query-level stability as a tool. Such an analysis is very helpful for us to derive more advanced algorithms for IR. We apply the proposed theory to the existing algorithms of Ranking SVM and IRSVM. Experimental results on the two algorithms verify the correctness of the theoretical analysis.

1. Introduction

Recently, learning to rank has gained increasing attention in machine learning and information retrieval (IR). When applied to IR, learning to rank is a task as follows. Given a set of training queries, their associated documents, and the corresponding relevance judgments, a ranking model is created which best represents the relevance of documents with respect to queries. When a user submits a query to the IR system, the trained model assigns a score to each document associated with the query, sorts the documents based on their scores, and presents the top ranked documents to the user. Average ranking accuracy over a large number of queries is usually used to evaluate the effectiveness of a ranking model. Therefore, from the application's perspective, both training and evaluation should be conducted at query level.

Many learning to rank algorithms have been proposed in recent years. Examples include the pointwise ranking algorithms like MCRank (Li et al., 2007), the pairwise ranking algorithms like Ranking SVM (Herbrich et al., 1999) and RankBoost (Freund et al., 2003), and the listwise ranking algorithms like ListNet (Cao et al., 2007). Analysis on the algorithms in the light of statistical learning theory, however, was not sufficient, particularly that on the generalization ability of the proposed algorithms. The pointwise and pairwise approaches transform the ranking problem to classification or regression, and thus existing theory on classification and regression can be applied. However, it deviates from the direction of enhancing ranking accuracy at query level. Furthermore, the listwise approach lacks of analysis on generalization ability.

In this paper, we investigate the generalization ability of learning to rank algorithms, in particular from the viewpoint of query-level training and evaluation.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.
*The work was performed when the first and the third authors were interns at Microsoft Research Asia.

We propose a new probabilistic formulation of learning to rank for IR. The formulation can naturally represent the pointwise, pairwise and listwise approaches in a unified framework. Within the framework, we introduce the concepts of query-level loss, query-level risk, and particularly query-level stability. Query-level stability measures whether the output of a learning algorithm changes largely with small changes in the training queries. With query-level stability as a tool we can conduct analysis on query-level generalization bounds of learning algorithms. A query-level generalization bound indicates how well one can enhance the expected ranking accuracy (corresponding to the expected risk) by enhancing the average ranking accuracy in training (corresponding to the empirical risk).

We take the algorithms of Ranking SVM (Joachims, 2002; Herbrich et al., 1999) and IRSVM (Cao et al., 2006; Qin et al., 2007) as examples, and apply the proposed theory to them. Our theoretical result shows that the query-level generalization bound of Ranking SVM is not reasonably good, mainly because Ranking SVM is trained at document pair level, not query level. Furthermore, IRSVM does have a better generalization bound than Ranking SVM, due to its stronger query-level stability. We also conducted experiments and our experimental results agree with the theoretical findings.

The contributions of this paper are listed as follows. (1) A proposal on conducting analysis on learning to rank algorithms at query level is made. (2) A new probabilistic formulation of learning to rank is proposed. (3) A new methodology for analyzing generalization ability of learning to rank algorithms on the basis of query-level stability is proposed. (4) The proposed theory is applied to learning to rank algorithms of Ranking SVM and IRSVM. The correctness of the theory has been verified by experiments.

2. Previous Work

2.1. Ranking in IR

Ranking is a central issue for IR. Many methods for creating ranking models have been proposed, including heuristics and learning based methods, (Baeza-Yates & Ribeiro-Neto, 1999; Herbrich et al., 1999; Joachims, 2002; Freund et al., 2003; Burges et al., 2005; Cao et al., 2007). Typically a ranking model is defined as a function of features based on query-document pair, and is learned with training data containing a number of queries, associated documents, and corresponding relevance judgments. Measures for evaluating the performance of a ranking model, such as Precision, MAP (Baeza-Yates & Ribeiro-Neto, 1999), and NDCG (Järvelin & Kekäläinen, 2002) have been defined and used. All the measures are query-based; if the evaluation measure for a query q is $EV(q)$, then the averaged $EV(q)$ on a number of queries is used. From the application's perspective, both training and testing in learning to rank should be conducted at query level.

2.2. Learning to Rank

So far learning to rank has been addressed by the pointwise, pairwise, and listwise approaches. In the pointwise approach (Li et al., 2007), ranking is transformed to regression or classification, and the loss function in learning is defined as a function of a single document. In the pairwise approach (Herbrich et al., 1999; Joachims, 2002; Freund et al., 2003; Cao et al., 2006), ranking is transformed to pairwise classification, and the loss function is defined on a document pair. In the listwise approach (Cao et al., 2007; Qin et al., 2007), document lists are viewed as learning instances and the loss function is defined on that basis.

Although many learning methods have been proposed, theoretical investigations on them were not sufficient. Since training and testing should be conducted at query level, studies on query-level generalization ability of learning algorithms are really needed. Unfortunately, it was missing in the previous work.

2.3. Stability Theory

The notion of stability (Devroye & Wagner, 1979) has been proposed for analyzing the generalization bounds of learning algorithms.

Bousquet et al. (Bousquet & Elisseeff, 2002) propose the theory of uniform leave-one-out stability. Based on it, the generalization bounds of classification algorithms such as Support Vector Machines (SVM) can be derived. Agarwal et al. (Agarwal & Niyogi, 2005) apply the stability tool to bipartite ranking.

We can apply the existing stability theory to get document level and document pair level generalization bounds. However, they may be not suitable for the task of IR. In this paper, we propose query-level stability and reveal the relation between query-level stability and query-level generalization bound.

3. Probabilistic Formulation for Ranking

As explained in Section 2, ranking in IR is evaluated at query level. Therefore, to design and evaluate a learning to rank algorithm, we should also look at it from

the query perspective. To this end, we give a novel probabilistic formulation of ranking for IR, which contains queries and their *associates* (documents, document pairs, or document sets) in two layers. We then introduce the notions of query-level loss and query-level risk.

Assume that query q is a random sample from the query space $\mathcal{Q}$ according to a probability distribution $P_{\mathcal{Q}}$. For query q, an associate $\omega^{(q)}$ and its ground-truth $g(\omega^{(q)})$ are sampled from space $\Omega \times \mathcal{G}$ according to a joint probability distribution D_q, where Ω is the space of associates and $\mathcal{G}$ is the space of ground truth. Here the associate $\omega^{(q)}$ can be a single document, a pair of documents, or a set of documents, and correspondingly the ground truth $g(\omega^{(q)})$ can be a relevance score (or class label), an order on a pair of documents, or a permutation (list) of documents. Let $l(f; \omega^{(q)}, g(\omega^{(q)}))$ denote a loss (referred to as *associate-level loss*) defined on $(\omega^{(q)}, g(\omega^{(q)}))$ and a ranking function f.

Expected query-level loss is defined as:

$$L(f;q) = \int_{\Omega \times \mathcal{G}} l(f; \omega^{(q)}, g(\omega^{(q)}))\, D_q(d\omega^{(q)}, dg(\omega^{(q)})).$$

Empirical query-level loss is defined as:

$$\hat{L}(f;q) = \frac{1}{n_q} \sum_{j=1}^{n_q} l(f; \omega_j^{(q)}, g(\omega_j^{(q)})),$$

where $(\omega_j^{(q)}, g(\omega_j^{(q)})), j = 1 \cdots, n_q$ stands for n_q associates of q, which are sampled i.i.d. according to D_q. The empirical query-level loss can be an estimate of the expected query-level loss. It can be proven that the estimation is consistent.

The goal of learning to rank is to select the ranking function f which can minimize the *expected query-level risk* defined as:

$$R_l(f) = E_{\mathcal{Q}} L(f;q) = \int_{\mathcal{Q}} L(f;q)\, P_{\mathcal{Q}}(dq). \tag{1}$$

In practice, $P_{\mathcal{Q}}$ is unknown. What we have are the training samples $(q_1, S_1), \cdots, (q_r, S_r)$, where $S_i = \{(\omega_1^{(i)}, g(\omega_1^{(i)})), \cdots, (\omega_{n_i}^{(i)}, g(\omega_{n_i}^{(i)}))\}, i = 1, \cdots, r$, and n_i is the number of associates for query q_i. Here $q_1, \cdots, q_r$ can be viewed as data sampled *i.i.d.* according to $P_{\mathcal{Q}}$, and $(\omega_j^{(i)}, g(\omega_j^{(i)}))$ as data sampled *i.i.d.* according to $D_{q_i}, j = 1, \cdots, n_i, i = 1, \cdots, r$.

Empirical query-level risk is defined as:

$$\widehat{R_l}(f) = \frac{1}{r} \sum_{i=1}^{r} \hat{L}(f; q_i). \tag{2}$$

The empirical query-level risk is an estimate of the expected query-level risk. It can be proven that the estimation is consistent.

This probabilistic formulation can cover most of existing learning to rank algorithms. If we let the associate to be a single document, a document pair, or a document set, we can respectively define pointwise, pairwise, or listwise losses, and develop pointwise, pairwise, or listwise approaches to learning to rank.

(a) *Pointwise Case*

Let $\mathcal{D}$ denote the document space. We use a feature mapping function $\phi : \mathcal{Q} \times \mathcal{D} \to \mathcal{X}(= R^d)$ to create a d-dimensional feature vector for each query-document pair. For each query q, suppose that the feature vector of a document is $x^{(q)}$ and its relevance score (or class label) is $y^{(q)}$, then $(x^{(q)}, y^{(q)})$ can be viewed as a random sample from $\mathcal{X} \times R$ according to a probability distribution D_q. If $l(f; x^{(q)}, y^{(q)})$ is a pointwise loss (square loss for example), then the expected query-level loss becomes:

$$L(f;q) = \int_{\mathcal{X} \times R} l\left(f; x^{(q)}, y^{(q)}\right)\, D_q\left(dx^{(q)}, dy^{(q)}\right).$$

Given training samples $(q_1, S_1), \cdots, (q_r, S_r)$, where $S_i = \{(x_1^{(i)}, y_1^{(i)}), \cdots, (x_{n_i}^{(i)}, y_{n_i}^{(i)})\}, i = 1, \cdots, r$, the empirical query-level loss of query $q_i, (i = 1, \cdots, r)$ turns out to be:

$$\hat{L}(f;q_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} l(f; x_j^{(i)}, y_j^{(i)}).$$

(b) *Pairwise Case*

For each query q, $z^{(q)} = (x_1^{(q)}, x_2^{(q)})$ stands for a document pair associated with it. Moreover, $y^{(q)} = 1$ if $x_1^{(q)}$ is ranked above $x_2^{(q)}$, $y^{(q)} = -1$ otherwise. Let $\mathcal{Y} = \{1, -1\}$. $(x_1^{(q)}, x_2^{(q)}, y^{(q)})$ can be viewed as a random sample from $\mathcal{X}^2 \times \mathcal{Y}$ according to a probability distribution D_q. If $l(f; z^{(q)}, y^{(q)})$ is a pairwise loss (hinge loss for example, (Herbrich et al., 1999)), then the expected query-level loss becomes:

$$L(q) = \int_{\mathcal{X}^2 \times \mathcal{Y}} l\left(f; z^{(q)}, y^{(q)}\right)\, D_q\left(dz^{(q)}, dy^{(q)}\right).$$

Given training samples $(q_1, S_1), \cdots, (q_r, S_r)$, where $S_i = \{(z_1^{(i)}, y_1^{(i)}), \cdots, (z_{n_i}^{(i)}, y_{n_i}^{(i)})\}, i = 1, \cdots, r$, the empirical query-level loss of query $q_i, (i = 1, \cdots, r)$ turns out to be:

$$\hat{L}(f;q_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} l(f; z_j^{(i)}, y_j^{(i)}).$$

(c) *Listwise Case*

For each query q, let $s^{(q)}$ denote a set of m documents associated with it, $\pi(s^{(q)}) \in \Pi$ denote a permutation of documents in $s^{(q)}$ according to their relevance degrees to the query, where Π is the space of all permutations

on m documents. $(s^{(q)}, \pi(s^{(q)}))$ can be viewed as a random sample from $\mathcal{X}^m \times \Pi$ according to a probability distribution D_q. If $l(f; s^{(q)}, \pi(s^{(q)}))$ is a listwise loss (cross entropy loss for example, (Cao et al., 2007)), then the expected query-level loss becomes:

$$L(q) = \int_{\mathcal{X}^m \times \Pi} l\left(f; s^{(q)}, \pi\left(s^{(q)}\right)\right) D_q\left(ds^{(q)}, d\pi\left(s^{(q)}\right)\right).$$

Given training samples $(q_1, S_1), \cdots, (q_r, S_r)$, where $S_i = \{(s_1^{(i)}, \pi(s_1^{(i)})), \cdots, (s_{n_i}^{(i)}, \pi(s_{n_i}^{(i)}))\}, i = 1, \cdots, r$, the empirical query-level loss of query $q_i, (i = 1, \cdots, r)$ turns out to be:

$$\hat{L}(f, q_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} l(f; s_j^{(i)}, \pi(s_j^{(i)})).$$

4. Stability Theory For Query-level Generalization Bound Analysis

Based on the probabilistic formulation, we propose a novel concept named query-level stability. We further discuss how to use query-level stability to analyze the generalization ability of a learning to rank algorithm.

First, we give a definition to uniform leave-one-query-out associate-level loss stability. The stability of a learning algorithm represents the degree of change in the loss of prediction when randomly removing a query and its associates from the training data.

Definition 1. *Let $\mathcal{A}$ be a learning to rank algorithm, $\{(q_i, S_i), i = 1, \cdots, r\}$ be the training set, l be the associate-level loss function, and τ be a function mapping an integer to a real number. We say that $\mathcal{A}$ has uniform leave-one-query-out associate-level loss stability with coefficient τ with respect to l, if $\forall q_j \in \mathcal{Q}, S_j \in (\Omega \times \mathcal{G})^{n_j}, j = 1, \cdots, r, q \in \mathcal{Q}, (\omega^{(q)}, g(\omega^{(q)})) \in \Omega \times \mathcal{G}$, the following inequality holds:*

$$\left| l(f_{\{(q_i,S_i)\}_{i=1}^r}, \omega^{(q)}, g(\omega^{(q)})) - l(f_{\{(q_i,S_i)\}_{i=1,i\neq j}^r}, \omega^{(q)}, g(\omega^{(q)})) \right| \leq \tau(r).$$

Here $\{(q_i, S_i)\}_{i=1, i\neq j}^r$ stands for the samples $(q_1, S_1), \cdots, (q_{j-1}, S_{j-1}), (q_{j+1}, S_{j+1}), \cdots, (q_r, S_r)$, where (q_j, S_j) is deleted. $f_{\{(q_i,S_i)\}_{i=1}^r}$ stands for the ranking function learned from $\{(q_i, S_i)\}_{i=1}^r$. We will use the notations hereafter.

With the definition, we can obtain the following lemma. It states that, if an algorithm has uniform leave-one-query-out associate-level loss stability, it will be stable in terms of expected query-level loss and empirical query-level loss. For ease of explanation, we simply call the uniform leave-one-query-out associate-level loss stability *query-level stability*.

Lemma 1. *Let $\mathcal{A}$ be a learning to rank algorithm, $\{(q_i, S_i), i = 1, \cdots, r\}$ be the training set, and l be the associate-level loss function. If $\mathcal{A}$ has leave-one-query-out associate-level loss stability with coefficient τ with respect to l, then the following inequalities hold:*

$$\left| L(f_{\{(q_i,S_i)\}_{i=1}^r}, q) - L(f_{\{(q_i,S_i)\}_{i=1,i\neq j}^r}, q) \right| \leq \tau(r),$$
$$\left| \hat{L}(f_{\{(q_i,S_i)\}_{i=1}^r}, q) - \hat{L}(f_{\{(q_i,S_i)\}_{i=1,i\neq j}^r}, q) \right| \leq \tau(r).$$

Based on the concept of query-level stability, we can derive a query-level generalization bound, as shown in Theorem 1. The theorem states that if an algorithm has query-level stability, then with high probability over the samples, the expected query-level risk can be bounded by the empirical risk and a term which depends on the query number and parameters of the algorithm. Furthermore, the theorem quantifies the expected loss on new queries, which is exactly what we mean by query-level generalization.

Theorem 1. *Let $\mathcal{A}$ be a learning to rank algorithm, $(q_1, S_1), \cdots, (q_r, S_r)$ be r training samples, and let l be the associate-level loss function. If (1) $\forall (q_1, S_1), \cdots, (q_r, S_r), q \in \mathcal{Q}, (\omega^{(q)}, g\left(\omega^{(q)}\right) \in \Omega \times \mathcal{G}$, $\left| l\left(f_{(q_i,S_i)_{i=1}^r}, \omega^{(q)}, g\left(\omega^{(q)}\right)\right)\right| \leq B$, (2) $\mathcal{A}$ has query-level stability with coefficient τ, then $\forall \delta \in (0, 1)$ with probability at least $1 - \delta$ over the samples of $\{(q_i, S_i)\}_{i=1}^r$ in the product space $\prod_{i=1}^r \{\mathcal{Q} \times (\Omega \times \mathcal{G})^\infty\}$, the following inequality holds:*

$$R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) \leq \widehat{R_l}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) + 2\tau(r) + (4r\tau(r) + B)\sqrt{\frac{\ln\frac{1}{\delta}}{2r}}.$$

Proof. For clarity of the proof, we first give the following definitions:

$$\rho(\{(q_i, S_i)\}_{i=1}^r) \triangleq R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) - \widehat{R_l}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right),$$
$$\int_{\Omega_1} \triangleq \int_{\mathcal{Q}} \int_{(\Omega\times\mathcal{G})^{n_1}} \cdots \int_{\mathcal{Q}} \int_{(\Omega\times\mathcal{G})^{n_r}}, \int_{\Omega_2} \triangleq \int_{\mathcal{Q}} \int_{\Omega\times\mathcal{G}},$$

$$P_1(d\omega) \triangleq D_{q^r}^{n_r}(dS_r) P_{\mathcal{Q}}(dq_r) \cdots D_{q^1}^{n_1}(dS_1) P_{\mathcal{Q}}(dq_1),$$
$$P_2(d\omega') \triangleq D_q(d\omega^{(q)}, dg(w^{(q)})) P_{\mathcal{Q}}(dq).$$

We then prove the theorem in two steps.

1) *Get the bound of*

$$\left| \rho(\{(q_i, S_i)\}_{i=1}^r) - \int_{\Omega_1} \rho(\{(q_i, S_i)\}_{i=1}^r) P_1(d\omega) \right|.$$

For this purpose, we get the upper bound of the following term first:

$$\left| \rho(\{(q_i, S_i)\}_{i=1}^r) - \rho(\{(q_i, S_i)\}_{i=1}^{r,j,q_j'}) \right|$$

where $\{(q_i,S_i)\}_{i=1}^{r,j,q_j'}$ means that query (q_j,S_j) is changed for another query (q_j',S_j'), where S_j' refers to $(w_1^{(j')},g(w_1^{(j')})),\cdots,(w_{n_j'},g(w_{n_j'}^{(j')}))$.

To utilize the query-level stability, we divide ρ into two terms: $\rho=\rho_1-\rho_2$, and discuss either of them separately, as follows.

$$\begin{aligned}&\rho_1(\{(q_i,S_i)\}_{i=1}^r)\triangleq R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right)\\&=\int_{\Omega_2}l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega^{(q)},g(\omega^{(q)}))P_2(d\omega^{'}).\\&\rho_2(\{(q_i,S_i)\}_{i=1}^r)\triangleq \widehat{R_l}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right)\\&=\frac{1}{r}\sum_{i=1}^r\frac{1}{n_i}\sum_{j=1}^{n_i}l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega_j^{(i)},g(\omega_j^{(i)})).\end{aligned}$$

Based on query-level stability, we can obtain that $\forall q_j\in\mathcal{Q},S_j\in(\Omega\times\mathcal{G})^{n_j},j=1,\cdots,r,q,q_j'\in\mathcal{Q},S_j'\in\{\mathcal{Q}\times(\Pi\times\mathcal{G})^{n_j'}\},(\omega^{(q)},g(\omega^{(q)}))\in\Omega\times\mathcal{G}$, the following inequality holds:

$$\begin{aligned}&\Big|l(f_{\{(q_i,S_i)\}_{i=1}^r},\omega^{(q)},g(\omega^{(q)}))\\&-l(f_{\{(q_i,S_i)\}_{i=1}^{r,j,q_j'}},\omega^{(q)},g(\omega^{(q)}))\Big|\le 2\tau(r).\end{aligned}\tag{3}$$

With (3), as ρ_1 is an integral function, the following inequality holds:

$$|\rho_1(\{(q_i,S_i)\}_{i=1}^r)-\rho_1(\{(q_i,S_i)\}_{i=1}^{r,j,q_j'})|\le 2\tau(r).\tag{4}$$

As for ρ_2, we have

$$\begin{aligned}&|\rho_2(\{(q_i,S_i)\}_{i=1}^r)-\rho_2(\{(q_i,S_i)\}_{i=1}^{r,j,q_j'})|\\&\le\frac{1}{r}\sum_{i=1,i\neq j}^r\frac{1}{n_i}\sum_{j=1}^{n_i}|l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega_j^{(i)},g(\omega_j^{(i)}))\\&-l(f_{\{(q_i,S_i)\}_{i=1}^{r,j,q_j'}};\omega_j^{(i)},g(\omega_j^{(i)}))|\\&+\frac{1}{r}|\frac{1}{n_j}\sum_{s=1}^{n_j}l(f_{\{(q_i,S_i)\}_{i=1}^{n_j}};\omega_s^{(j)},g(\omega_s^{(j)}))\\&-\frac{1}{n_j'}\sum_{s=1}^{n_j'}l(f_{\{(q_i,S_i)\}_{i=1}^{r,j,q_j'}};\omega_s^{(j')},g(\omega_s^{(j')}))|\\&\le 2\tau(r)+\frac{B}{r}.\end{aligned}\tag{5}$$

By jointly considering (4) and (5), we obtain:

$$|\rho(\{(q_i,S_i)\}_{i=1}^r)-\rho(\{(q_i,S_i)\}_{i=1}^{r,j,q_j'})|\le 4\tau(r)+\frac{B}{r}.$$

Based on McDiarmid's inequality(McDiarmid, 1989), with probability at least $1-\delta$ over the samples of $\{(q_i,S_i)\}_{i=1}^r$ in the product space $\prod_{i=1}^r\{\mathcal{Q}\times(\Omega\times\mathcal{G})^\infty\}$, we have

$$\begin{aligned}\rho(\{(q_i,S_i)\}_{i=1}^r)\quad&\le\quad\int_{\Omega_1}\rho(\{(q_i,S_i)\}_{i=1}^r)\,P_1(d\omega).\\&+\quad(4r\tau(r)+B)\sqrt{\frac{\ln\frac{1}{\delta}}{2r}}.\end{aligned}\tag{6}$$

2) *Get the bound of* $\left|\int_{\Omega_1}\rho(\{(q_i,S_i)\}_{i=1}^r)\,P_1(d\omega)\right|$

$$\begin{aligned}&\int_{\Omega_1}\rho[\{(q_i,S_i)\}_{i=1}^r]P_1(d\omega)\\&=\int_{\Omega_1}\int_{\Omega_2}[l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega^{(q)},g(\omega^{(q)}))]\,P_2(d\omega^{'})\,P_1(d\omega)\\&-\int_{\Omega_1}l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega_j^{(i)},g(\omega_j^{(i)}))\,P_1(d\omega)\\&=\int_{\Omega}\int_{\Omega_2}[l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega^{(q)},g(\omega^{(q)}))\\&-l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega_j^{(i)},g(\omega_j^{(i)}))]\,P_2(d\omega^{'})\,P_1(d\omega).\\&=\int_{\Omega_1}\int_{\Omega_2}[l(f_{\{(q_i,S_i)\}_{i=1}^r};\omega^{(q)},g(\omega^{(q)}))\\&-l(f_{\{(q_i,S_i)\}_{i=1}^{r,i,q}};\omega_j^{(q)},g(\omega_j^{(q)}))]\,P_2(d\omega^{'})\,P_1(d\omega).\end{aligned}$$

The reason that the last equality holds is as follows. Because the integral is conducted over all of the samples, and the samples are *i.i.d.*, we can change the ith query in the training set for $(q,\omega^{(q)},g(\omega^{(q)}))$. Then by further using (3), we have:

$$\left|\int_{\Omega_1}\rho[\{(q_i,S_i)\}_{i=1}^r]P_1(d\omega)\right|\le 2\tau(r).\tag{7}$$

Merging Eq. (6) and (7) yields the inequality in the theorem. □

5. Case Study

Without loss of generality, we take existing algorithms of Ranking SVM (Joachims, 2002; Herbrich et al., 1999) and IRSVM (Cao et al., 2006; Qin et al., 2007) as examples to show how to analyze the query-level generalization bound of an algorithm, using the tool of query-level stability. Both of the two algorithms belong to the pariwise case of our probabilistic formulation. It should be noted that the framework is neither limited to these two algorithms nor to the pair-wise case, we leave the discussions on other algorithms or other approaches to our future work.

5.1. Generalization Bound of Ranking SVM

Ranking SVM is widely used in ranking for IR, which views document pair as associate of the query and minimizes:

$$\min_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n l_h(f;z_i,y_i)+\lambda\|f\|_K^2,\tag{8}$$

where $l_h(f;z_i,y_i)$ is the hinge loss, and K is a kernel function in the Reproducing Kernel Hilbert Space (RKHS).

Using the conventional stability theory (Bousquet & Elisseeff, 2002), we can get the following lemma which shows the query-level stability of Ranking SVM.

Lemma 2. *If* $\forall x \in \mathcal{X}, K(x,x) \leq \kappa^2 < \infty$, *then Ranking SVM has query-level stability with coefficient* $\tau(r) = \frac{4\kappa^2}{\lambda r} \times \max_{\forall n_i, S_i} \frac{n_i}{\frac{1}{r}\sum_{i=1}^{r} n_i}$.

As for this lemma, we have the following discussions. (1) When r approaches infinity, suppose the mean and variance of the distribution of n_q are μ and σ^2 respectively. Then by the Law of Large Numbers and Chebyshev's inequality, $\forall 0 < \delta < 1, \forall \epsilon > 0, \exists R(\epsilon)$, if $r > R(\epsilon)$, with probability at least $1-\delta$, the following inequality holds:

$$max_{\forall n_i, S_i} \frac{n_i}{\frac{1}{r}\sum_{i=1}^{r} n_i} \leq \frac{1 + \frac{\sigma}{\mu\sqrt{\frac{\delta}{r}}}}{1 - \frac{\varepsilon}{\mu}}.$$

Therefore, $\tau(r) \leq \frac{4\kappa^2}{\lambda r} \frac{1+\frac{\sigma}{\mu\sqrt{\frac{\delta}{r}}}}{1-\frac{\varepsilon}{\mu}}$. That is, $\tau(r)$ will approach zero, with a convergence rate of $O(\frac{1}{\sqrt{r}})$, when r goes to infinity.

(2) When r is finite (which is the case in practice), we have no reasonable statistical estimation of the term $\max_{\forall n_i, S_i} \frac{n_i}{\frac{1}{r}\sum_{i=1}^{r} n_i}$. As a result, we can only get a loose bound for $\tau(r)$ as $\frac{4\kappa^2}{\lambda}$. That is, when r increases but is still finite, $\tau(r)$ does not necessarily decrease.

Based on the above lemma, we can further derive the generalization bound of Ranking SVM. In particular, as the function $f_{\{(q_i,S_i)\}_{i=1}^r}$ is learned from the training samples $(q_1, S_1), \cdots, (q_r, S_r)$, there is a constant C, such that, $\forall (q_1,S_1), \cdots, (q_r,S_r)$, $\left\| f_{\{(q_i,S_i)\}_{i=1}^r} \right\|_K \leq C$. Then, $\forall (q_1,S_1), \cdots, (q_r,S_r), z \in \mathcal{Z}, y \in \mathcal{Y}$, $l_h\left(f_{\{(q_i,S_i)\}_{i=1}^r}, z, y\right) \leq 1 + 2C\kappa$. By further considering Theorem 1, we obtain the following theorems.

Theorem 2. *If* $\forall x \in \mathcal{X}, K(x,x) \leq \kappa^2 < \infty$, *then for Ranking SVM,* $\forall \delta \in (0,1), \forall \epsilon > 0, \exists R(\epsilon)$, *if* $r > R(\epsilon)$, *then with probability at least* $1-2\delta$ *over the samples of* $\{(q_i,S_i)\}_{i=1}^r$ *in the product space* $\prod_{i=1}^r \{\mathcal{Q} \times (\mathcal{X} \times \mathcal{X} \times \mathcal{Y})^\infty\}$, *we have:*

$$R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) \leq \widehat{R_l}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) + \frac{8\kappa^2}{\lambda r}\frac{1+\frac{\sigma}{\mu\sqrt{\frac{\delta}{r}}}}{1-\frac{\varepsilon}{\mu}} + \left(\frac{16\kappa^2 \frac{1+\frac{\sigma}{\mu\sqrt{\frac{\delta}{r}}}}{1-\frac{\varepsilon}{\mu}} + \lambda(1+2C\kappa)}{\lambda}\right)\sqrt{\frac{\ln\frac{1}{\delta}}{2r}}.$$

Theorem 3. *If* $\forall x \in \mathcal{X}, K(x,x) \leq \kappa^2 < \infty$ *and we have no constraint on* r, *then for Ranking SVM,* $\forall \delta \in (0,1)$, *with probability at least* $1-\delta$ *over the samples of* $\{(q_i,S_i)\}_{i=1}^r$ *in the product space* $\prod_{i=1}^r \{\mathcal{Q} \times (\mathcal{X} \times \mathcal{X} \times \mathcal{Y})^\infty\}$, *we only have:*

$$R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) \leq \widehat{R_l}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) + \frac{8\kappa^2}{\lambda} + \left(\frac{16r\kappa^2 + \lambda(1+2C\kappa)}{\lambda}\right)\sqrt{\frac{\ln\frac{1}{\delta}}{2r}}.$$

Theorem 2 states that when the number of training queries tends to be infinity, with high probability the empirical query-level risk of Ranking SVM will converge to its expected query-level risk. However, when the number of training queries is finite, the expected query-level risk and empirical query-level risk are not necessarily close to each other, and the bound in Theorem 3 quantifies the difference, which is an increasing function of the number of training queries.

5.2. Generalization Bound of IRSVM

In IR application, the numbers of document pairs associated with different queries vary largely (See LETOR or other public dataset). In consideration of this, IRSVM, studied in (Cao et al., 2006) and (Qin et al., 2007), is an adaptive version of Ranking SVM to the IR applications, which minimizes:

$$\min_{f \in \mathcal{F}} \frac{1}{r}\sum_{i=1}^{r}\frac{1}{n_i}\sum_{j=1}^{n_i} l_h(f; z_j^{(i)}, y_j^{(i)}) + \| f \|_K^2 . \qquad (9)$$

We can prove the query-level stability of IRSVM as shown in Lemma 3. Due to space limitations, we omit the proof.

Lemma 3. *If* $\forall x \in \mathcal{X}, K(x,x) \leq \kappa^2 < \infty$, *then IRSVM has query-level stability* $\tau(r) = \frac{4\kappa^2}{\lambda r}$.

With a similar analysis to that for Ranking SVM, we obtain the following theorem.

Theorem 4. *If* $\forall x \in \mathcal{X}, K(x,x) \leq \kappa^2 < \infty$, *then for IRSVM,* $\forall \delta \in (0,1)$, *with probability at least* $1-\delta$ *over the samples of* $\{(q_i,S_i)\}_{i=1}^r$ *in the product space* $\prod_{i=1}^r \{\mathcal{Q} \times (\mathcal{X} \times \mathcal{X} \times \mathcal{Y})^\infty\}$, *we have:*

$$R_l\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) \leq \widehat{R_{l_h}}\left(f_{\{(q_i,S_i)\}_{i=1}^r}\right) + \frac{8\kappa^2}{\lambda r} + \frac{16\kappa^2 + \lambda(1+2C\kappa)}{\lambda}\sqrt{\frac{\ln\frac{1}{\delta}}{2r}}.$$

The theorem states that when the number of training queries tends to be infinity, with high probability the empirical query-level risk of IRSVM will converge to its expected query-level risk. When the number of queries is finite, the bound in the theorem quantifies the difference between the two risks, which is a decreasing function of the number of training queries.

Remark 1. *By comparing Theorem 2 and Theorem 4, we can find that the convergence rates of the empirical query-level risk to the expected query-level risk for Ranking SVM and IRSVM are the same, i.e.* $O(\frac{1}{\sqrt{r}})$. *However, by comparing Theorem 3 to Theorem 4, we can see that for the case of finite* r, *the bound of IRSVM is much tighter than that of Ranking SVM.*

6. Experiments and Discussion

We conducted experiments on Ranking SVM and IRSVM to verify our theoretical results.

6.1. Query-level Stability

First, we conducted an experiment to compare the stabilities of Ranking SVM and IRSVM. We randomly sampled 1,200 queries from a search engine's data repository, each query associated with hundreds of documents and their relevance labels. There are five labels: "perfect", "excellent", "good", "fair", and "bad". We split the queries into three sets: a training set with 200 queries, a validation set with 500 queries, and a test set with 500 queries (we denote the test set as $\mathcal{T}$). The validation set was used to select the regularization parameter λ for Ranking SVM and IRSVM.

We first trained two ranking models with Ranking SVM and IRSVM, denoted as f_0 and $f_0^{'}$ respectively. Then we randomly deleted one query from the training set, and trained two new models with Ranking SVM and IRSVM, denoted as f_1 and $f_1^{'}$ respectively. We repeated this process 30 times, and created the models $f_1, f_2, \cdots, f_{30}$ an $f_1^{'}, f_2^{'}, \cdots, f_{30}^{'}$. Then on the test set, we compared the associate-level loss for f_0 with that for f_i, and obtained the difference Δ_i for Ranking SVM. Similarly, we computed $\Delta_i^{'}$ for IRSVM.

$$\Delta_i = \max_{q \in \mathcal{T}} \max_{z \in S_q} |l_h(f_0, z^{(q)}, y^{(q)}) - l_h(f_i, z^{(q)}, y^{(q)})|,$$

$$\Delta_i^{'} = \max_{q \in \mathcal{T}} \max_{z \in S_q} |l_h(f_0^{'}, z^{(q)}, y^{(q)}) - l_h(f_i^{'}, z^{(q)}, y^{(q)})|.$$

According to Definition 1, Δ_i can bound from below the query-level stability $\tau(r)(r = 200)$ of Ranking SVM. Similarly, $\Delta_i^{'}$ can bound from below the query-level stability $\tau(r)(r = 200)$ of IRSVM. In this regard, we can compare stabilities of Ranking SVM and IRSVM by comparing Δ_i and $\Delta_i^{'}$.

We list all the 30 values of Δ_i and $\Delta_i^{'}$ in Table 1. From it, we can see that Δ_i is always much larger than $\Delta_i^{'}$. The mean (or maximum) value of Δ_i over the 30 trials is 1.23 (or 4.53). It is about more than ten times higher than the mean (or maximum) value of $\Delta_i^{'}$, which is only 0.12 (or 0.27). Furthermore, the variance of Δ_i (i.e. 0.72) is also larger than that of $\Delta_i^{'}$ (i.e. 0.003). These results indicate that the query-level stability of RankSVM is not so good as that of IRSVM. (Note that Lemmas 2 and 3 hold for any r, the number of training queries. We simply set $r = 200$.)

6.2. Query-level Generalization Bounds

Next, we compared the performances of Ranking SVM and IRSVM, to verify the theoretical results on their query-level generalization bounds.

From Theorems 3 and 4 we can see that the bound for Ranking SVM is much looser than that for IRSVM, especially when the number of training queries r is large but finite. We interpret the result as follow.

The actual empirical risk and expected risk with respect to Ranking SVM are as follows.

$$\widehat{R_{l_h}}(f) = \frac{1}{n} \sum_{i=1}^{n} l_h(f; z^{(i)}, y^{(i)})), n = \sum_{i=1}^{r} n_i.$$

$$R_{l_h}(f) = \int_{\mathcal{X}^2 \times \mathcal{Y}} l_h(f; z, y) P(dz, dy).$$

In the definitions, only document pair but no query appears, and thus we call them the *pair-level risks*. For comparison, we also list the *query-level risks* for the learning to rank problem (See also Section 3) where hinge loss is used as associate-level loss.

$$\widehat{R_{l_h}}(f) = \frac{1}{r} \sum_{i=1}^{r} \frac{1}{n_i} \sum_{j=1}^{n_i} l_h(f; z^{(i)}, y^{(i)}).$$

$$R_{l_h}(f) = \int_{\mathcal{Q}} \int_{\mathcal{X}^2 \times \mathcal{Y}} l_h(f; z^{(q)}, y^{(q)})\, D_q(dz^{(q)}, dy^{(q)})\, P_{\mathcal{Q}}(dq).$$

By comparing the above formulas, we can clearly see that what is optimized in Ranking SVM (i.e. the pair-level risk) is not equal to what should be optimized (i.e. the query-level risks), unless every training query has the same number of document pairs, which is not true in practice. In contrast, it is easy to verify that what is optimized in IRSVM is exactly the query-level risk. Therefore, no surprisingly IRSVM has a better query-level generalization bound.

In summary, the theoretical results indicate that the performance of Ranking SVM on the test set in terms of a query-level measure should not be so good as that of IRSVM. We have verified this through experiments. We tested the ranking performances of Ranking SVM (RankSVM for short) and IRSVM on the test set, in terms of Precision and NDCG. The results are shown in Figure 1. Furthermore, MAP [1] for Ranking SVM is 0.39 and MAP for IRSVM is 0.41. From the results, we can see that IRSVM achieves better ranking performance than RankSVM, in terms of all the query-level measures. This is also consistent with the results reported in (Cao et al., 2006) and (Qin et al., 2007).

7. Conclusions

In this paper, we have studied the generalization ability of learning to rank algorithms for IR. A probabilistic formulation for ranking has been proposed, which covers ranking algorithms belonging to the pointwise,

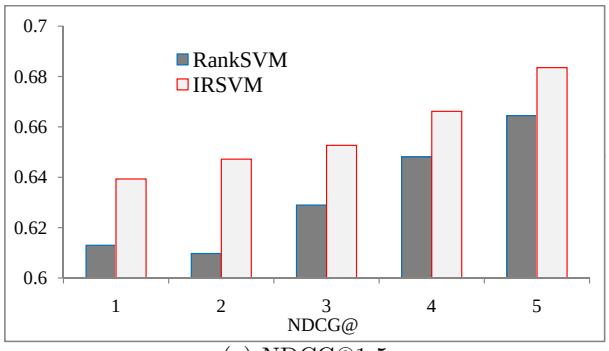
[1] To compute MAP, we treated "perfect", "excellent" and "good" as relevant, and "fair" and "bad" as irrelevant.

Table 1. Comparison of Query-level Stability

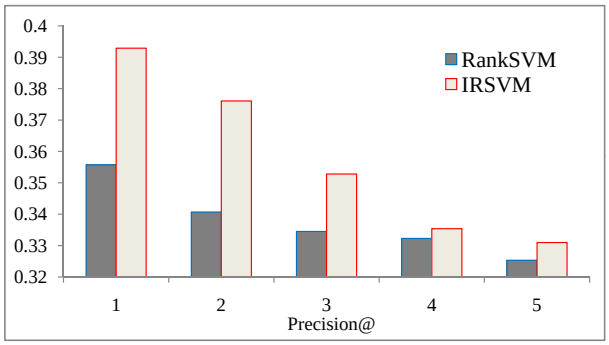
i	1	2	3	4	5	6
Δ_i	3.59	1.14	0.88	0.81	1.84	1.15
Δ_i'	0.07	0.07	0.06	0.06	0.05	0.24
	7	8	9	10	11	12
	0.89	1.30	0.90	1.42	1.38	1.39
	0.18	0.06	0.09	0.08	0.11	0.15
	13	14	15	16	17	18
	0.56	1.43	1.42	1.01	1.13	1.34
	0.11	0.13	0.14	0.11	0.06	0.11
	19	20	21	22	23	24
	1.04	0.86	0.43	0.51	0.64	0.92
	0.08	0.05	0.09	0.20	**0.27**	0.14
	25	26	27	28	29	30
	0.50	0.88	**4.53**	0.99	1.13	0.62
	0.18	0.08	0.12	0.09	0.21	0.14

pairwise and listwise approaches. The tool of query-level stability has been developed, which has been further used to analyze the generalization bound of a ranking algorithm. We have applied the tool to two existing ranking algorithms (Ranking SVM and IRSVM) and obtained theoretical results. We have also verified the correctness of the results by experiments.

As far as we know, this is the first work on query-level generalization bound of learning to rank algorithms. There are still many issues to investigate. (1) We have taken SVM based ranking algorithms as examples. We will try to obtain similar results for other algorithms, such as RankBoost. (2) We have focused on the pairwise approach. The proposed formulation for ranking and the tool of query-level stability can also be used to analyze other approaches. (3) It is worth checking whether new learning to rank algorithms can be derived under the guide of the theoretical study.



(a) NDCG@1-5



(b) Precision@1-5

Figure 1. Accuracies of Ranking SVM and IRSVM

References

Agarwal, S., & Niyogi, P. (2005). Stability and generalization of bipartite ranking algorithms. *Proc. of COLT'05* (pp. 32–47).

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval.* Addison Wesley.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research, 2*, 499–526.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. *ICML '05* (pp. 89–96).

Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y., & Hon, H.-W. (2006). Adapting ranking svm to document retrieval. *SIGIR '06* (pp. 186–193).

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. *ICML '07* (pp. 129–136).

Devroye, L., & Wagner, T. (1979). Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory, 25*, 601–604.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res., 4*, 933–969.

Herbrich, R., Obermayer, K., & Graepel, T. (1999). Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers.* (pp. 115–132).

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst., 20*, 422–446.

Joachims, T. (2002). Optimizing search engines using clickthrough data. *KDD '02* (pp. 133–142).

Li, P., Burges, C., & Wu, Q. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. *NIPS2007.*

McDiarmid, C. (1989). *On the method of bounded differences.* Cambridge University Press.

Qin, T., Zhang, X.-D., Tsai, M.-F., Wang, D.-S., Liu, T.-Y., & Li, H. (2007). Query-level loss functions for information retrieval. *Information Processing & Management.*