Research Article

Stability Analysis of Learning Algorithms for Ontology Similarity Computation

Wei Gao[1] and Tianwei Xu[1,2]

[1] *School of Information and Technology, Yunnan Normal University, Kunming, Yunnan 650500, China*
[2] *Key Laboratory of Educational Informatization for Nationalities, Ministry of Education, Yunnan Normal University, Kunming 650500, China*

Correspondence should be addressed to Wei Gao; gaowei@ynnu.edu.cn

Ontology, as a useful tool, is widely applied in lots of areas such as social science, computer science, and medical science. Ontology concept similarity calculation is the key part of the algorithms in these applications. A recent approach is to make use of similarity between vertices on ontology graphs. It is, instead of pairwise computations, based on a function that maps the vertex set of an ontology graph to real numbers. In order to obtain this, the ranking learning problem plays an important and essential role, especially k-partite ranking algorithm, which is suitable for solving some ontology problems. A ranking function is usually used to map the vertices of an ontology graph to numbers and assign ranks of the vertices through their scores. Through studying a training sample, such a function can be learned. It contains a subset of vertices of the ontology graph. A good ranking function means small ranking mistakes and good stability. For ranking algorithms, which are in a well-stable state, we study generalization bounds via some concepts of algorithmic stability. We also find that kernel-based ranking algorithms stated as regularization schemes in reproducing kernel Hilbert spaces satisfy stability conditions and have great generalization abilities.

1. Introduction and Motivations

The study of ontology deals with questions concerning what entities exist and how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences. The developed tools have been widely applied in medicine, biology, and social science. In computer science, ontology is defined as a model for sharing formal concepts and has been applied in intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management. After a decade's development, ontology technology has matured as an effective model of hierarchical structure and semantics for concepts, supported by systematic and comprehensive engineering theory, representation, and construction tools.

Ontology similarity computation is an essential part in practical applications. In information retrieval, it has been used to compute semantic similarity and search for concepts. We take a graph-theory approach and represent an ontology by a weighted graph $G = (V, E, w)$. In this setting, $V = \{v_1, \ldots, v_n\}$ is the (finite) set of vertices corresponding to concepts or objects of the ontology, $E \subset V \times V$ is a set of edges, and $w : E \to \mathbb{R}_+$ is a weight function. For two vertices v_i and v_j representing two concepts, the weight $w(v_i, v_j)$ measures their similarity in the ontology.

Example 1. In some applications of ontology similarity computation, the weight function w takes values on $[0, 1]$. Then the case $w(v_i, v_j) = 1$ means that v_i and v_j represent the same concept while $w(v_i, v_j) = 0$ means that these two concepts have no similarity. In information retrieval, with a threshold parameter $0 < \epsilon < 1$, when one tries to find related information of the concept v_i, all concepts v_j satisfying $w(v_i, v_j) > \epsilon$ are returned, which means that v_j and v_i have a high similarity.

Traditional methods for ontology similarity computation are based on pairwise similarity calculation. Their computational complexity is high, and they required selection of many parameters, which are not so intuitive. In this paper,

we use a learning theory approach. The idea is to learn a scoring function $f : V \to \mathbb{R}$ and then to determine the similarity between vertices (concepts) v_i and v_j by their value difference $|f(v_i) - f(v_j)|$: the smaller the difference the higher the similarity. Formally, $\text{Sim}(v_i, v_{j_1}) \geq \text{Sim}(v_i, v_{j_2})$ if and only if $|f(v_i) - f(v_{j_1})| \leq |f(v_i) - f(v_{j_2})|$. Such an inspiring approach was introduced from the viewpoint of ranking in [1] where a ranking algorithm is used for learning from samples a scoring function f with small ranking error. The method was employed in the ontology setting in [2] which demonstrates accuracy and efficiency. Another possible way to learn such a function f is by a graph Laplacian and taking an eigenvector associated with its second smallest eigenvalue. See [3–6] for details. This method requires a positive definiteness condition for a similarity matrix which is hard to check in our setting. Also, when the size of the graph is large, the computational complexity is high.

In this paper, we explore the learning theory approach for ontology similarity computations in a setting when the ontology graph is a tree. It is a connected graph without cycle. Thus, there is a unique path between any two vertices. The tree structure gives restrictions on similarity of vertices (concepts). For example, we assign a top vertex v_{top} and let it be the root, then denote k the degree (the number of edges that link to a vertex) of the top vertex. Let $N_G(v_{\text{top}}) = \{v_1, v_2, \ldots, v_k\}$ be the neighbor set of v_{top}. If there is a path from one vertex to v_{top} through v_i, then it belongs to branch i. Thus, we have k branches in the tree and any two vertices belonging to different branches have no edge between them. The concepts in the same branch of the tree should have higher similarity, compared with concepts in different branches. This observation motivates us to apply the k-partite ranking algorithm [7] in which the k parts correspond to the k classes of vertices of k rates. The rate values of all classes are decided by experts. Intuitively, a vertex of a higher rate b is ranked higher than any vertex of rate a if $1 \leq a < b \leq k$. Thus, the k-partite ranking algorithm is reasonable to learn a similarity function for an ontology graph with a tree structure.

The main contribution of this paper is to state some ontology computations as a k-partite ranking problem and to conduct stability analysis of the algorithms with mild conditions, which leads to useful error bounds for ontology applications.

The organization of the rest part of this paper is as follows. The setting and main results are given in the next section. The generalization bounds for learning algorithms will be shown in Section 4. The stability and generalization bounds for the learning algorithms stated as regularization schemes in reproducing kernel Hilbert spaces will be discussed in Section 5.

2. Formal Setting and Main Results

Now, we state our learning algorithm for ontology similarity computation.

Let $V = \{v_1, \ldots, v_n\}$ be the finite set of vertices of an ontology graph. It is divided into k disjoint subsets $\{V_i\}_{i=1}^k$ corresponding to k rates. Let $\mathcal{D}$ be a probability measure on V.

The performance of a ranking function $f : V \to \mathbb{R}$ can be measured by the following concept.

Definition 2. A ranking loss function is a function $l : \mathbb{R}^V \times V \times V \to \mathbb{R}_+ \cup \{0\}$ that assigns, for $f : V \to \mathbb{R}$ and $v, v' \in V$, a nonnegative real number $l(f, v, v')$ interpreted as the loss of f in its relative ranking of v and v'. The expected k-partite ranking error on the ontology graph for a ranking function $f : V \to \mathbb{R}$ associated with the ranking loss function l is defined as

$$er_l(f) = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{v \sim \mathcal{D}_a, v' \sim \mathcal{D}_b} \left\{ l\left(f, v, v'\right) \right\}, \tag{1}$$

where $\mathcal{D}_a$ is the conditional distribution of $\mathcal{D}$ on V_a.

Example 3. One commonly used ranking loss function is the hinge ranking loss defined as

$$l_h\left(f, v, v'\right) = \left(1 - \left(f(v) - f\left(v'\right)\right)\right)_+, \tag{2}$$

where $x_+ = \max\{x, 0\}$. Another ranking loss function is the γ-ranking loss with a smoothing parameter $\gamma > 0$ defined as

$$l_\gamma\left(f, v, v'\right)$$
$$= \begin{cases} 1, & \text{if } \left(f(v) - f\left(v'\right)\right) < 0, \\ 1 - \dfrac{1}{\gamma}\left(f(v) - f\left(v'\right)\right), & \text{if } 0 \leq \left(f(v) - f\left(v'\right)\right) \\ & \qquad \leq \gamma\tau\left(v, v'\right), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Learning algorithms are implemented with a sample $\mathcal{T} = \{t_i\}_{i=1}^M$ of size M, called a preference graph, which is assumed here to be independently drawn according to $\mathcal{D}$. It can also be divided into k parts $\{\mathcal{T}_1, \ldots, \mathcal{T}_k\}$, where $\mathcal{T}_a = \{t_i : t_i \in V_a\}$ consists of those sampling points of rate a.

In [8], Agarwal and Niyogi have studied the algorithmic stability in a general setting, where the training examples take labels $y \in [0, M_1]$ for some $M_1 > 0$. A goal ranking function ranks future instances with larger labels higher than those with smaller labels. Here, our setting is more specific. The learner is given a preference graph $\mathcal{T}$ consisting of k disjoint parts corresponding to the k classes of vertices. Every part has a rate value. The target ranking function ranks future instances in higher-rate parts higher than those in lower-rate parts.

A large class of learning algorithms is generated by regularization schemes. They penalize an empirical error which is chosen here to be the *empirical k-partite ranking error* on the ontology graph defined for a $f : V \to \mathbb{R}$ associated with the sample $\mathcal{T}$ as

$$\widehat{er}_{\mathcal{T}, l}(f) = \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \sum_{i : t_i \in \mathcal{T}_a} \sum_{j : t_j \in \mathcal{T}_b} l\left(f, t_i, t_j\right). \tag{4}$$

In this paper, we study a learning algorithm generated by a regularization scheme in a reproducing kernel Hilbert space

(RKHS) $(\mathscr{H}_K, \|\cdot\|_K)$ associated with a Mercer kernel $K : V \times V \to \mathbb{R}$. Now, the regularization scheme is defined by

$$f_{\mathscr{T}} = \arg \min_{f \in \mathscr{H}_K} \left\{ \widehat{er}_{\mathscr{T},l}(f) + \lambda \|f\|_K^2 \right\}, \tag{5}$$

where $\lambda > 0$ is a regularization parameter. On the selection of the regularized parameter, readers are referred to [9, 10] for more details about the method of cross-validation.

One point we need to emphasize that we abuse terminology for the sake of better readability. If the ranking function f does not associate with RKHS (for instance, in Lemma 9, Theorems 10 and 12), then the second term in the right-hand side of (5) vanishes.

Our error analysis provides a learning rate of algorithm (5) when the ranking loss is σ-admissible.

Definition 4. Let l be a ranking loss, $\sigma > 0$, and $\mathscr{F}$ a class of real-valued functions on V. We say that l is σ-admissible with respect to $\mathscr{F}$ if for any $f_1, f_2 \in \mathscr{F}$ and $v, v' \in V$,

$$\begin{aligned} &\left| l\left(f_1, v, v'\right) - l\left(f_2, v, v'\right) \right| \\ &\leq \sigma \left(\left| f_1(v) - f_2(v) \right| + \left| f_1(v') - f_2(v') \right| \right). \end{aligned} \tag{6}$$

Let us state the estimate of learning rates which will be proved in Section 5.

Theorem 5. *Let $\mathscr{H}_K$ be a RKHS such that $K(v, v) \leq \kappa^2 < \infty$ for all $v \in V$. Let l be a ranking loss, σ-admissible with respect to $\mathscr{H}_K$, and bounded by some $B > 0$, such that $l(f, v, v')$ is convex with respect to f. Let f_l^* be a fixed function in $\mathscr{H}_K$ satisfying*

$$er_l(f_l^*) \leq \inf_{f \in \mathscr{H}_K} er_l(f) + \frac{1}{M'} \tag{7}$$

for some $M' > 0$. Then, for any $0 < \delta < 1$, with confidence at least $1 - \delta$, one has

$$\begin{aligned} er_l(f_{\mathscr{T}}) \leq{}& \inf_{f \in \mathscr{H}_K} er_l(f) + \lambda \|f_l^*\|_K^2 \\ &+ \frac{1}{M'} + \frac{16\sigma^2\kappa^2}{\lambda M} + \left(\frac{24\sigma^2\kappa^2}{\lambda} + B \right) \sqrt{\frac{\ln(2/\delta)}{2M}}. \end{aligned} \tag{8}$$

Form Theorem 5, we see that if $\lambda \to 0$ and $\lambda\sqrt{M} \to \infty$ (e.g., $\lambda = M^{-1/4}$), then $er_l(f_{\mathscr{T}})$ converges with confidence to $\inf_{f \in \mathscr{H}_K} er_l(f)$. The quantity $\inf_{f \in \mathscr{H}_K} er_l(f)$ is well understood in the literature related to learning theory (e.g., [11–14]).

3. Stability Analysis

An algorithm is stable if any change of a single point in a training set yields only a small change in the output. It is natural to consider that a good ranking algorithm is one with good stability; that is, a mild change of samples does not necessarily lead to too much change in the ranking function. Some analysis of the stability of ranking algorithms is given in [1, 8, 15].

Let $n_a = |\mathscr{T}_a|$ and $t_a^{i_a}$ be the i_ath element in $\mathscr{T}_a$ for $1 \leq a \leq k, 1 \leq i_a \leq n_a$. Let $\mathscr{T}^{(t_a^{i_a}, t^a)}$ be the sequence obtained by replacing $t_a^{i_a}$ in $\mathscr{T}$ by a new sampling point t^a of rate a. We define some notions of stability for k-partite ranking algorithms.

Definition 6 (uniform loss stability for a k-partite ranking algorithm on ontology graph). Let A be a k-partite ranking algorithm for ontology whose output on a preference graph $\mathscr{T} = (\mathscr{T}_1, \ldots, \mathscr{T}_k)$ is denoted by $f_{\mathscr{T}}$. Let l be a ranking loss function and $\alpha_a : \mathbb{N}^k \to \mathbb{R}$ for $1 \leq a \leq k$. We say that A has uniform loss stability $(\alpha_1, \ldots, \alpha_k)$ with respect to l if for all $1 \leq a \leq k, n_a \in \mathbb{N}, t_a^{i_a} \in \mathscr{T}_a$ and $i_a \in \{1, \ldots, n_a\}$, we have for all $t^a \in V_a, v, v' \in V$ but belong to different rate,

$$\left| l\left(f_{\mathscr{T}}, v, v'\right) - l\left(f_{\mathscr{T}^{(t_j^{i_j}, t_j)}}, v, v'\right) \right| \leq \alpha_j(n_1, \ldots, n_k), \tag{9}$$

$$j = 1, \ldots, k.$$

Definition 7 (uniform score stability for a k-partite ranking algorithm on ontology graph). Let A be a k-partite ranking algorithm for ontology whose output on a preference graph $\mathscr{T} = (\mathscr{T}_1, \ldots, \mathscr{T}_k)$ is denoted by $f_{\mathscr{T}}$. Let l be a ranking loss function and $\mu_a : \mathbb{N}^k \to \mathbb{R}$ for $1 \leq a \leq k$. We say that A has uniform score stability $(\mu_1, \ldots, \mu_k)$ with respect to l if for all $1 \leq a \leq k, n_a \in \mathbb{N}, t_a^{i_a} \in \mathscr{T}_a, i_a \in \{1, \ldots, n_a\}$, and $t^a \in V_a$, and for all $v \in V$,

$$\left| f_{\mathscr{T}}(v) - f_{\mathscr{T}^{(t_j^{i_j}, t_j)}}(v) \right| \leq \mu_j(n_1, \ldots, n_k), \quad j = 1, \ldots, k. \tag{10}$$

The main tool used here is McDiarmid's inequality, which bounds the deviation of any function of a sample on which a single change in the sample has limited effect.

Theorem 8 (see [16]). *Let $X_1, \ldots, X_N$ be independent random variables, each taking values in a set A. Let $\phi : A^N \to \mathbb{R}$ such that for each $i \in \{1, \ldots, N\}$, there exists a constant $c_i > 0$ such that*

$$\begin{aligned} \sup_{x_1, \ldots, x_N \in A, x_i' \in A} &\left| \phi(x_1, \ldots, x_N) \right. \\ &\left. - \phi(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_N) \right| \leq c_i. \end{aligned} \tag{11}$$

Then, for any $\varepsilon > 0$,

$$\mathbf{P}\left\{ \phi(X_1, \ldots, X_N) - \mathbf{E}\left\{ \phi(X_1, \cdots, X_N) \right\} \geq \varepsilon \right\} \leq e^{-2\varepsilon^2 / \sum_{i=1}^N c_i^2}. \tag{12}$$

In what follows, $\mathscr{T}^{(t_1^{i_1}, t^1), \ldots, (t_k^{i_k}, t^k)}$ denotes a training sample set obtained by replacing $t_1^{i_1}$ in $\mathscr{T}$ by $t^1, \ldots, t_k^{i_k}$ by t^k for $t_a^{i_a} \in \mathscr{T}_a$, and $i_a \in \{1, \ldots, n_a\}$. Also, $\alpha_a(n_1, \ldots, n_k)$ and $\mu_a(n_1, \ldots, n_k)$ are simply denoted by α_a and μ_a, respectively. We only consider the case of sample replacements with the same rate: for some ontology graphs, the graph structure is fixed; hence the members of vertices and edges in each branch are fixed.

4. Generalization Bounds for Stable k-Partite Ranking Algorithms on Ontology Graph

From this section, our analysis for stability of k-partite ranking algorithms is stated on an ontology graph and our organization follows [8]. In this section, generalization bounds for ranking algorithms that exhibit good stability properties will be derived. Our tricks are based on those of [17]. We start with the following technical lemma.

Lemma 9. *Let A be a symmetric k-partite ranking algorithm for ontology whose output on a preference graph $\mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_k)$ is denoted by $f_\mathcal{T}$, and let l be a ranking loss function. Then, for all $t_a^{i_a} \in \mathcal{T}_a$, $i_a \in \{1, \ldots, n_a\}$, and $t^a \in V_a$, $t^b \in V_b$, one has*

$$\mathbf{E}_\mathcal{T} \left[er_l (f_\mathcal{T}) - \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \right]$$

$$= \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathcal{T},t^a,t^b} \left\{ n_a n_b \left[l \left(f_\mathcal{T}, t^a, t^b \right) \right. \right.$$

$$\left. \left. - l \left(f_{\mathcal{T}^{(t_a^{i_a},t^a),(t_b^{i_b},t^b)}}, t^a, t^b \right) \right] \right\}. \tag{13}$$

Proof. We have

$$\mathbf{E}_\mathcal{T} \left[\widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \right]$$

$$= \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_\mathcal{T} \left[\sum_{i:t_i \in \mathcal{T}_a} \sum_{j:t_j \in \mathcal{T}_b} l \left(f_\mathcal{T}, t_i, t_j \right) \right]. \tag{14}$$

By symmetry, the term in the summation is the same for all i, j. Therefore, we get

$$\mathbf{E}_\mathcal{T} \left[\widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \right]$$

$$= \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathcal{T},t^a,t^b} \left[\sum_{i:t_i \in \mathcal{T}_a} \sum_{j:t_j \in \mathcal{T}_b} l \left(f_\mathcal{T}, t_i, t_j \right) \right]. \tag{15}$$

Interchanging the roles of t_i with t^a and t_j with t^b, we get

$$\mathbf{E}_\mathcal{T} \left[\widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \right]$$

$$= \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathcal{T},t^a,t^b} \left\{ n_a n_b \left[l \left(f_{\mathcal{T}^{(t_a^{i_a},t^a),(t_b^{i_b},t^b)}}, t^a, t^b \right) \right] \right\}. \tag{16}$$

Since

$$\mathbf{E}_\mathcal{T} \left[er_l (f_\mathcal{T}) \right] = \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathcal{T},t^a,t^b} \left\{ n_a n_b \left[l (f_\mathcal{T}) \right] \right\}, \tag{17}$$

the results follow. $\qquad \square$

We are now ready to give our main result of this section, which bounds the expected l-error of a ranking function learned by a k-partite ranking algorithm with good uniform loss stability in terms of its empirical l-error on the training sample. The proof follows [18].

Theorem 10. *Let A be a symmetric k-partite ranking algorithm for ontology whose output on a preference graph $\mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_k)$ is denoted by $f_\mathcal{T}$, and let l be ranking loss function such that $0 \le l(f, v, v') \le B$ for all $f : V \to \mathbb{R}$ and $v, v' \in \mathcal{T}$. Let $\alpha_a : \mathbb{N}^k \to \mathbb{R}$ for $1 \le a \le k$ such that A has uniform loss stability $(\alpha_1, \ldots, \alpha_k)$ with respect to l. Let $\alpha_{\max} = \max\{\alpha_1, \ldots, \alpha_k\}$. Then, for any $0 < \delta < 1$, with confidence at least $1 - \delta$, one has*

$$er_l (f_\mathcal{T}) \le \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) + 2\alpha_{\max} + \left(2\alpha_{\max} + \frac{B}{M} \right) \sqrt{\frac{M \ln (1/\delta)}{2}}. \tag{18}$$

Proof. Let $\phi : V^M \to \mathbb{R}$ be defined by

$$\phi (\mathcal{T}) = er_l (f_\mathcal{T}) - \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}). \tag{19}$$

We show that ϕ satisfies the condition for McDiarmid's inequality. To this end, let $t^a \in V_a$. For each $i_1 \in \{1, \ldots, n_1\}$, we have

$$\left| er_l (f_\mathcal{T}) - er_l \left(f_{\mathcal{T}^{(t_1^{i_1},t^1)}} \right) \right|$$

$$= \left| \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{v \sim \mathcal{D}_a, v' \sim \mathcal{D}_b} \left\{ l \left(f_\mathcal{T}, v, v' \right) - l \left(f_{\mathcal{T}^{(t_1^{i_1},t^1)}}, v, v' \right) \right\} \right| \le \alpha_1,$$

$$\left| \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) - \widehat{er}_{\mathcal{T},l} \left(f_{\mathcal{T}^{(t_1^{i_1},t^1)}} \right) \right|$$

$$\le \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \sum_{i:\{t_i \in \mathcal{T}_a, t_i \ne t_1^{i_1}\}} \sum_{j:t_j \in \mathcal{T}_b} \left| l \left(f_\mathcal{T}, t_i, t_j \right) \right.$$

$$\left. - l \left(f_{\mathcal{T}^{(t_1^{i_1},t^1)}}, t_i, t_j \right) \right|$$

$$+ \frac{1}{M^2} \sum_{b=2}^{k} \sum_{j:t_j \in \mathcal{T}_b} \left| l \left(f_\mathcal{T}, t_1^{i_1}, t_j, \right) - l \left(f_{\mathcal{T}^{(t_1^{i_1},t^1)}}, t_1^{i_1}, t_j \right) \right|$$

$$\le \alpha_1 + \frac{B \sum_{b=2}^{k} n_b}{M^2} \le \alpha_1 + \frac{B}{M}. \tag{20}$$

These give

$$\left| \phi (\mathcal{T}) - \phi \left(\mathcal{T}^{(t_1^{i_1},t^1)} \right) \right| \le 2\alpha_1 + \frac{B}{M}. \tag{21}$$

Similarly, it can be shown that for any $i_a \in \{1, \ldots, n_a\}, 1 \le a \le k$,

$$\left| \phi (\mathcal{T}) - \phi \left(\mathcal{T}^{(t_a^{i_a},t^a)} \right) \right| \le 2\alpha_a + \frac{B}{M}. \tag{22}$$

Thus, applying McDiarmid's inequality to ϕ, we get for any $\varepsilon > 0$,

$$\mathbf{P}_\mathcal{T} \left\{ \{ er_l (f_\mathcal{T}) - \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \} - \mathbf{E}_\mathcal{T} \{ er_l (f_\mathcal{T}) - \widehat{er}_{\mathcal{T},l} (f_\mathcal{T}) \} \ge \varepsilon \right\}$$

$$\le \exp \left\{ - \frac{2\varepsilon^2}{M(2\alpha_{\max} + B/M)^2} \right\}. \tag{23}$$

Now, by Lemma 9, we know that the expectation $\mathbf{E}_{\mathscr{T}}[er_l(f_{\mathscr{T}}) - \widehat{er}_{\mathscr{T},l}(f_{\mathscr{T}})]$ can be bounded as

$$
\frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathscr{T},t^a,t^b} \left\{ n_a n_b \left[l\left(f_{\mathscr{T}}, t^a, t^b\right) \right. \right.
$$
$$
\left. \left. - l\left(f_{\mathscr{T}^{(i_a^a,t^a),(i_b^b,t^b)}}, t^a, t^b\right)\right]\right\}
$$
$$
= \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathscr{T},t^a,t^b} \left\{ n_a n_b \left[l\left(f_{\mathscr{T}}, t^a, t^b\right) \right. \right.
$$
$$
\left. \left. - l\left(f_{\mathscr{T}^{(i_a^a,t^a)}}, t^a, t^b\right)\right]\right\}
$$
$$
+ \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathscr{T},t^a,t^b} \left\{ n_a n_b \left[l\left(f_{\mathscr{T}^{(i_a^a,t^a)}}, t^a, t^b\right) \right. \right.
$$
$$
\left. \left. - l\left(f_{\mathscr{T}^{(i_a^a,t^a),(i_b^b,t^b)}}, t^a, t^b\right)\right]\right\}
$$
$$
\leq \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathscr{T},t^a,t^b} \left\{ n_a n_b \alpha_a \right\}
$$
$$
+ \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbf{E}_{\mathscr{T},t^a,t^b} \left\{ n_a n_b \alpha_b \right\} \leq 2\alpha_{\max}.
$$

(24)

Thus, for any $\varepsilon > 0$,

$$
\mathbf{P}_{\mathscr{T}} \left\{ er_l\left(f_{\mathscr{T}}\right) - \widehat{er}_{\mathscr{T},l}\left(f_{\mathscr{T}}\right) - 2\alpha_{\max} \geq \varepsilon \right\}
$$
$$
\leq \exp \left\{ -\frac{2\varepsilon^2}{M\left(2\alpha_{\max} + B/M\right)^2} \right\}.
$$

(25)

The result follows by setting the right-hand side equal to δ and solving it for ε. □

For any $\gamma > 0$, and any k-partite ranking algorithm with good uniform loss stability with respect to l_γ, Theorem 10 can be applied to bound the expected ranking error of a learned ranking function in terms of its empirical l_γ-error on the training sample. The following lemma shows that, for every $\gamma > 0$, a ranking algorithm with good uniform score stability also has good uniform loss stability with respect to l_γ. Using the techniques of Lemma 2 in [17], and taking $\tau(v_i, v_j)$ in Example 3 as 1, the following lemma can be obtained immediately.

Lemma 11. *Let A be a k-partite ranking algorithm for ontology whose output on a preference graph $\mathscr{T} = (\mathscr{T}_1, \ldots, \mathscr{T}_k)$ is denoted by $f_{\mathscr{T}}$. Let $\mu_a : \mathbb{N}^k \to \mathbb{R}$ for $1 \leq a \leq k$ such that A has uniform score stability $(\mu_1, \ldots, \mu_k)$. Then, for every $\gamma > 0$, A has uniform loss stability $(\alpha_1^\gamma, \ldots, \alpha_k^\gamma)$ with respect to the γ ranking loss l_γ, where for all $n_1, \ldots, n_k \in \mathbb{N}$,*

$$
\alpha_j^\gamma\left(n_1, \ldots, n_k\right) = \frac{2\mu_j}{\gamma}, \quad j = 1, \ldots, k.
$$

(26)

Combining Theorem 10 and Lemma 11, we get the following result which bounds the expected ranking error of a learned ranking function in terms of its empirical l_γ-error for any ranking algorithm with good uniform score stability.

Theorem 12. *Let A be a k-partite ranking algorithm for ontology whose output on a preference graph $\mathscr{T} = (\mathscr{T}_1, \ldots, \mathscr{T}_k)$ is denoted by $f_{\mathscr{T}}$. Let $\mu_a : \mathbb{N}^k \to \mathbb{R}$ for $1 \leq a \leq k$ such that A has uniform score stability $(\mu_1, \ldots, \mu_k)$, and $\gamma > 0$. Denote $\mu_{\max} = \max\{\mu_1, \ldots, \mu_k\}$. If l is a ranking loss satisfying $0 \leq l(f, v, v') \leq B$ for all $f : V \to \mathbb{R}$ and $v, v' \in \mathscr{T}$, then for any $0 < \delta < 1$, with probability of at least $1 - \delta$,*

$$
er_l\left(f_{\mathscr{T}}\right) \leq \widehat{er}_{\mathscr{T},l}\left(f_{\mathscr{T}}\right) + \frac{4}{\gamma}\mu_{\max} + \left(\frac{4}{\gamma}\mu_{\max} + \frac{B}{M}\right)\sqrt{\frac{M\ln(1/\delta)}{2}}.
$$

(27)

Proof. One applies Theorem 10 to A with the ranking loss l_γ (using Lemma 11), which satisfies $0 \leq l_\gamma \leq B$. One finishes the proof thanks to the fact that $er \leq er_{l_\gamma}$ □

5. Stable Ranking Algorithms

In this section, we will demonstrate stability of some ranking algorithms in which a ranking function is selected by minimizing a regularized objective function. A general result for regularization-based k-partite ranking algorithms will be derived in Section 5.1. In Section 5.2, this result is used to illustrate stability of kernel-based k-partite ranking algorithms that perform regularization in a reproducing kernel Hilbert space. These stability results are also used to achieve consistency theorem for kernel-based k-partite ranking algorithms in Section 5.3.

5.1. General Regularizers. Let l be given a ranking loss function, $\mathscr{F}$ a class of real-valued functions on V, and $N : \mathscr{F} \to \mathbb{R}_+ \cup \{0\}$ a regularization functional. Consider the following regularized empirical l-error of a ranking function $f \in \mathscr{F}$ (with respect to a preference graph $\mathscr{T}$) with regularization parameter $\lambda > 0$,

$$
\widehat{er}_{\mathscr{T},l}^\lambda = \frac{1}{M^2} \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \sum_{i:t_i \in \mathscr{T}_a} \sum_{j:t_j \in \mathscr{T}_b} l\left(f, t_i, t_j\right) + \lambda N\left(f\right).
$$

(28)

We consider k-partite ranking algorithms that minimize such a regularized objective function; that is, ranking algorithms that, given a preference graph $\mathscr{T}$, output a ranking function $f_{\mathscr{T}} \in \mathscr{F}$ that satisfies

$$
f_{\mathscr{T}} = \arg\min_{f \in \mathscr{F}} \widehat{er}_{\mathscr{T},l}^\lambda = \arg\min_{f \in \mathscr{F}} \left\{ \widehat{er}_{\mathscr{T},l} + \lambda N\left(f\right) \right\},
$$

(29)

for some fixed choice of ranking loss l, function class $\mathscr{F}$, regularized N, and regularization parameter λ. We derive a general result below that will be useful for showing stability of such regularization-based algorithms.

Lemma 13. *Let l be a ranking loss such that $l(f, v, v')$ is convex in f. Let $\mathscr{F}$ be a convex set of real-valued functions on V, and let $\sigma > 0$ such that l is σ-admissible with respect to $\mathscr{F}$. Let $\lambda > 0$, and let $N : \mathscr{F} \to \mathbb{R}_+ \cup \{0\}$ be a functional defined on $\mathscr{F}$ such that for preference graph $\mathscr{T} = (\mathscr{T}_1, \ldots, \mathscr{T}_k)$, the regularized empirical l-error $\widehat{er}^{\lambda}_{\mathscr{T},l}$ has a minimum (not necessarily unique) in $\mathscr{F}$. Let A be a k-partite ranking algorithm for ontology defined by (29). Let $t^a \in V_a$, $t_a^{i_a} \in \mathscr{T}_a$, and $i_a \in \{1, \ldots, n_a\}$. For brevity, denote*

$$f \equiv f_{\mathscr{T}}, \qquad f_{1,i_1} \equiv f_{\mathscr{T}((t_1^{i_1}, t^1))}, \ldots, f_{k,i_k} \equiv f_{\mathscr{T}((t_k^{i_k}, t^k))}, \tag{30}$$

and let

$$\Delta f_{1,i_1} = (f_{1,i_1} - f), \ldots, \Delta f_{k,i_k} = (f_{k,i_k} - f). \tag{31}$$

Then for any $t \in [0,1]$ and $q = 1, \ldots, k$, one has

$$
\begin{aligned}
N(f) &- N\left(f + t\Delta f_{q,i_q}\right) \\
&+ N\left(f_{q,i_q}\right) - N\left(f_{q,i_q} - t\Delta f_{q,i_q}\right) \\
&\leq \frac{t\sigma}{\lambda M^2} \sum_{b \neq q} \sum_{j:t_j \in \mathscr{T}_b} \left\{ \left| \Delta f_{q,i_q}\left(t_q^{i_q}\right)\right| \right. \\
&\left. \qquad + 2\left|\Delta f_{q,i_q}\left(t_j\right)\right| + \left|\Delta f_{q,i_q}\left(t^q\right)\right| \right\}.
\end{aligned}
\tag{32}
$$

Proof. Recall that a convex function g satisfies

$$g\left(x + t(y - x)\right) - g(x) \leq t\left(g(y) - g(x)\right), \quad \forall t \in [0,1]. \tag{33}$$

Since $l(f, v, v')$ is convex in f, $\widehat{er}_{\mathscr{T},l}(f)$ is convex in f. Therefore, for any $t \in [0,1]$, we have

$$\widehat{er}_{\mathscr{T},l}\left(f + t\Delta f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}(f) \leq t\left(\widehat{er}_{\mathscr{T},l}\left(f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}(f)\right), \tag{34}$$

and also (interchanging the roles of f and f_{1,i_1}),

$$
\begin{aligned}
\widehat{er}_{\mathscr{T},l}&\left(f_{1,i_1} - t\Delta f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1}\right) \\
&\leq t\left(\widehat{er}_{\mathscr{T},l}(f) - \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1}\right)\right).
\end{aligned}
\tag{35}
$$

Adding the above two inequalities yields

$$
\begin{aligned}
\widehat{er}_{\mathscr{T},l}&\left(f + t\Delta f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}(f) \\
&+ \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1} - t\Delta f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1}\right) \leq 0.
\end{aligned}
\tag{36}
$$

Now, since $\mathscr{F}$ is convex, $(f + t\Delta f_{1,i_1}) \in \mathscr{F}$ and $(f_{1,i_1} - t\Delta f_{1,i_1}) \in \mathscr{F}$. Since f minimizes $\widehat{er}^{\lambda}_{\mathscr{T},l}(f)$ in $\mathscr{F}$ and f_{1,i_1} minimizes $\widehat{er}^{\lambda}_{\mathscr{T}(t_1^{i_1}, t^1),l}(f)$ in $\mathscr{F}$, we have

$$
\begin{aligned}
\widehat{er}^{\lambda}_{\mathscr{T},l}(f) - \widehat{er}^{\lambda}_{\mathscr{T},l}\left(f + t\Delta f_{1,i_1}\right) &\leq 0, \\
\widehat{er}^{\lambda}_{\mathscr{T}(t_1^{i_1}, t^1),l}\left(f_{1,i_1}\right) - \widehat{er}^{\lambda}_{\mathscr{T}(t_1^{i_1}, t^1),l}\left(f_{1,i_1} - t\Delta f_{1,i_1}\right) &\leq 0.
\end{aligned}
\tag{37}
$$

Adding these two inequalities and applying (36), we get

$$
\begin{aligned}
\lambda\Big(&N(f) - N\left(f + t\Delta f_{1,i_1}\right) + N\left(f_{1,i_1}\right) - N\left(f_{1,i_1} - t\Delta f_{1,i_1}\right)\Big) \\
&\leq \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1}\right) - \widehat{er}_{\mathscr{T}(t_1^{i_1}, t^1),l}\left(f_{1,i_1}\right) \\
&\quad + \widehat{er}_{\mathscr{T}(t_1^{i_1}, t^1),l}\left(f_{1,i_1} - t\Delta f_{1,i_1}\right) - \widehat{er}_{\mathscr{T},l}\left(f_{1,i_1} - t\Delta f_{1,i_1}\right) \\
&= \frac{1}{M^2} \sum_{b=2}^{k} \sum_{j:t_j \in \mathscr{T}_b} \Big\{ \left(l\left(f_{1,i_1}, t_1^{i_1}, t_j\right) - l\left(f_{1,i_1} - t\Delta f_{1,i_1}, t_1^{i_1}, t_j\right)\right) \\
&\qquad + \left(l\left(f_{1,i_1} - t\Delta f_{1,i_1}, t^1, t_j\right) - l\left(f_{1,i_1}, t^1, t_j\right)\right)\Big\} \\
&\leq \frac{t\sigma}{M^2} \sum_{b=2}^{k} \sum_{j:t_j \in \mathscr{T}_b} \left\{ \left|\Delta f_{1,i_1}\left(t_1^{i_1}\right)\right| + 2\left|\Delta f_{1,i_1}\left(t_j\right)\right| + \left|\Delta f_{1,i_1}\left(t^1\right)\right| \right\}.
\end{aligned}
\tag{38}
$$

Similarly, for $q = 2, \ldots, k$, we have

$$
\begin{aligned}
\lambda\Big(&N(f) - N\left(f + t\Delta f_{q,i_q}\right) + N\left(f_{q,i_q}\right) - N\left(f_{q,i_q} - t\Delta f_{q,i_q}\right)\Big) \\
&\leq \frac{t\sigma}{M^2} \sum_{b \neq q} \sum_{j:t_j \in \mathscr{T}_b} \left\{ \left|\Delta f_{q,i_q}\left(t_q^{i_q}\right)\right| \right. \\
&\qquad \left. + 2\left|\Delta f_{q,i_q}\left(t_j\right)\right| + \left|\Delta f_{q,i_q}\left(t^q\right)\right| \right\}.
\end{aligned}
\tag{39}
$$

The results follow. $\qquad\square$

As we will see below, the above result can be used to establish stability of some regularization-based ranking algorithms.

5.2. Regularization in Reproducing Kernel Hilbert Spaces. Let $\mathscr{F}$ be a reproducing kernel Hilbert space (RKHS) of real-valued functions on V associated with a Mercer kernel $K : V \times V \to \mathbb{R}$. Here, $K_v : V \to \mathbb{R}$ is defined as $K_v(v') = K(v, v')$, and the reproducing property of $\mathscr{F}$ gives that for all $f \in \mathscr{F}$ and all $v \in V$,

$$f(v) = \langle f, K_v \rangle_K, \tag{40}$$

where $\langle \cdot, \cdot \rangle_K$ denotes the RKHS inner product in $\mathscr{F}$. By the Schwartz inequality, it is easy to show that for all $f \in \mathscr{F}$ and all $v \in V$,

$$|f(v)| \leq \|f\|_K \|K_v\|_K = \|f\|_K \sqrt{K(v, v)}, \tag{41}$$

where $\|\cdot\|_K$ denotes the RKHS norm in $\mathscr{F}$. We consider ranking algorithms that perform regularization in the RKHS $\mathscr{F}$ using the squared norm in $\mathscr{F}$ as regularizers. Specifically, let $N : \mathscr{F} \to \mathbb{R}_+ \cup \{0\}$ be the regularizer defined by

$$N(f) = \|f\|_K^2. \tag{42}$$

It will be demonstrated below that if for some $0 \leq \kappa < \infty$, $K(v, v) \leq \kappa$ for any $v \in V$, then a ranking algorithm that minimizes an appropriate regularized error over $\mathscr{F}$, with regularizer N defined as above, has good uniform score stability.

Theorem 14. *Let $\mathcal{F}$ be an RKHS with kernel K such that for all $v \in V$, $K(v, v) \le \kappa^2 < \infty$. Let l be a ranking loss such that $l(f, v, v')$ is convex in f and l is σ-admissible with respect to $\mathcal{F}$. Let $\lambda > 0$, and let N be given by (42). Let A be the k-partite ranking algorithm for ontology that, given a preference graph $\mathcal{T}$, outputs a ranking function $f_{\mathcal{T}} \in \mathcal{F}$ defined by (29). Then, A has uniform score stability $(\mu_1, \ldots, \mu_k)$ with*

$$\mu_q(n_1, \ldots, n_k) = \frac{4\sigma\kappa^2}{\lambda M}, \quad q = 1, \ldots, k. \quad (43)$$

Proof. Let $v^a \in V_a$ and $i_a \in \{1, \ldots, n_a\}$.

Applying Lemma 13 with $t = 1/2$, we get (using the notation in the proof of Lemma 13) that

$$\|f\|_K^2 - \left\|f + \frac{1}{2}\Delta f_{1,i_1}\right\|_K^2$$

$$+ \|f_{1,i_1}\|_K^2 - \left\|f_{1,i_1} - \frac{1}{2}\Delta f_{1,i_1}\right\|_K^2$$

$$\le \frac{\sigma}{2\lambda M^2} \sum_{b=2}^{k} \sum_{j:t_j \in \mathcal{T}_b} \left\{ \left|\Delta f_{1,i_1}\left(t_1^{i_1}\right)\right| \right.$$

$$\left. + 2\left|\Delta f_{1,i_1}\left(t_j\right)\right| + \left|\Delta f_{1,i_1}\left(t^1\right)\right| \right\}. \quad (44)$$

Since $\mathcal{F}$ is a vector space, $\Delta f_{1,i_1} \in \mathcal{F}$, $(f + (1/2)\Delta f_{1,i_1}) \in \mathcal{F}$, and $(f_{1,i_1} - (1/2)\Delta f_{1,i_1}) \in \mathcal{F}$, so $\|f + (1/2)\Delta f_{1,i_1}\|_k^2$ and $\|f_{1,i_1} - (1/2)\Delta f_{1,i_1}\|_k^2$ are well defined. It is easy to check that

$$\|f\|_K^2 - \left\|f + \frac{1}{2}\Delta f_{1,i_1}\right\|_K^2 + \|f_{1,i_1}\|_K^2$$

$$- \left\|f_{1,i_1} - \frac{1}{2}\Delta f_{1,i_1}\right\|_K^2 = \frac{1}{2}\|\Delta f_{1,i_1}\|_K^2. \quad (45)$$

Combined with (44), this gives

$$\|\Delta f_{1,i_1}\|_K^2$$

$$\le \frac{\sigma}{\lambda M^2} \sum_{b=2}^{k} \sum_{j:t_j \in \mathcal{T}_b} \left\{ \left|\Delta f_{1,i_1}\left(t_1^{i_1}\right)\right| \right.$$

$$\left. + 2\left|\Delta f_{1,i_1}\left(t_j\right)\right| + \left|\Delta f_{1,i_1}\left(t^1\right)\right| \right\}. \quad (46)$$

Since (as noted above) $\Delta f_{1,i_1} \in \mathcal{F}$, this together with (41) gives

$$\|\Delta f_{1,i_1}\|_K^2$$

$$\le \frac{\sigma}{\lambda M^2} \|\Delta f_{1,i_1}\|_K \sum_{b=2}^{k} \sum_{j:t_j \in \mathcal{T}_b} \left\{ \sqrt{K\left(t_1^{i_1}, t_1^{i_1}\right)} \right.$$

$$\left. + 2\sqrt{K\left(t_j, t_j\right)} + \sqrt{K\left(t^1, t^1\right)} \right\}$$

$$\le \frac{4\sigma\kappa}{\lambda M} \|\Delta f_{1,i_1}\|_K. \quad (47)$$

It follows that

$$\|\Delta f_{1,i_1}\|_K \le \frac{4\sigma\kappa}{\lambda M}. \quad (48)$$

This together with (41) tells us that for any $v \in V$,

$$\left|f_{\mathcal{T}}(v) - f_{\mathcal{T}_1^{i_1}, \tau^1}(v)\right| \le \frac{4\sigma\kappa^2}{\lambda M}. \quad (49)$$

Similarly, we can also obtain

$$\left|f_{\mathcal{T}}(v) - f_{\mathcal{T}_q^{i_q}, \tau^q}(v)\right| \le \frac{4\sigma\kappa^2}{\lambda M}, \quad q = 2, \ldots, k. \quad (50)$$

The conclusion follows. $\square$

Theorems 12 and 14 give the following generalization bound for kernel-based ranking algorithms.

Corollary 15. *Under the conditions of Theorem 14, one has that for any $0 < \delta < 1$, with probability of at least $1 - \delta$ over the draw of $\mathcal{T}$, the expected ranking error of the ranking function $f_{\mathcal{T}}$ learned by the regularized algorithm associated with the l_1 ranking loss is bounded by*

$$er_{l_1}(f_{\mathcal{T}}) \le \widehat{er}_{\mathcal{T}, l_1}(f_{\mathcal{T}}) + \frac{16\sigma\kappa^2}{\lambda M}$$

$$+ \left(\frac{16\sigma\kappa^2}{\lambda M} + \frac{1}{M}\right)\sqrt{\frac{M\ln(1/\delta)}{2}}. \quad (51)$$

The result of Corollary 15 shows that a larger regularization parameter λ leads to better stability and, therefore, a tighter confidence interval in the resulting generalization bound.

Under the conditions of the above results, a kernel-based ranking algorithm minimizing a regularized empirical l-error also has good uniform loss stability with respect to l; this follows from the following simple lemma.

Lemma 16. *Let $\mathcal{F}$ be a class of real-valued functions on V, and let A be a k-partite ranking algorithm for ontology that, given a preference graph $\mathcal{T}$, outputs a ranking function $f_{\mathcal{T}} \in \mathcal{F}$. If A has uniform score stability $(\mu_1, \ldots, \mu_k)$ and l is a ranking loss that is σ-admissible with respect to $\mathcal{F}$, then A has uniform loss stability $(\alpha_1, \ldots, \alpha_k)$ with respect to l, where for all $m \in \mathbb{N}$,*

$$\alpha_j(n_1, \ldots, n_k) = 2\sigma\mu_j(n_1, \ldots, n_k), \quad j = 1, \ldots, k. \quad (52)$$

The proof of this result can follow the proof of Lemma 13 in [8]. Using Theorem 14 and Lemma 16, we can immediately get the following corollary.

Corollary 17. *Under the conditions of Theorem 14, A has uniform loss stability $(\alpha_1, \ldots, \alpha_k)$ with respect to l, where for all $n_a \in \mathbb{N}$,*

$$\alpha_j(n_1, \ldots, n_k) = \frac{8\sigma^2\kappa^2}{\lambda M}, \quad j = 1, \ldots, k. \quad (53)$$

5.3. Consistency. We can also use the above results to show consistency of kernel-based ranking algorithms. In particular, let $er_l^*(\mathscr{F})$ denote the optimal expected l-error in an RKHS $\mathscr{F}$ (for a given distribution):

$$er_l^*(\mathscr{F}) = \inf_{f \in \mathscr{F}} er_l(f). \tag{54}$$

Then, for a bounded loss function l, we can show that with an appropriate choice of the regularization parameter λ, the expected l-error $er_l(f_{\mathscr{T}})$ of the ranking function $f_{\mathscr{T}}$ learned by a kernel-based ranking algorithm that minimizes a regularized empirical l-error in $\mathscr{F}$ converges (in probability) to this optimal value. We first show the following simple lemma.

Lemma 18. *Let $f : V \to \mathbb{R}$ be a fixed ranking function, and let l be a bounded ranking loss function such that $0 \leq l(f, v, v') \leq B$ for all $f : V \to \mathbb{R}$ and $v, v' \in V$. Then, for any $0 < \delta < 1$, with probability of at least $1 - \delta$,*

$$\widehat{er}_{\mathscr{T},l}(f) \leq er_l(f) + \frac{8\sigma^2\kappa^2}{\lambda}\sqrt{\frac{\ln(1/\delta)}{2M}}. \tag{55}$$

Proof. Define ϕ as

$$\phi(\mathscr{T}) = \widehat{er}_{\mathscr{T},l}(f). \tag{56}$$

Then, $\mathbf{E}_{\mathscr{T}}[\phi(\mathscr{T})] = er_l(f)$. We show that ϕ satisfies the condition of McDiarmid's inequality. For each $i_a \in \{1, \ldots, n_a\}$ and $v^a \in V_a$, we have

$$\left| \phi(\mathscr{T}) - \phi\left(\mathscr{T}^{v_a^{i_a}, v^a}\right) \right|$$

$$= \left| \widehat{er}_{\mathscr{T},l}(f) - \widehat{er}_{\mathscr{T}^{(v_a^{i_a}, v^a)},l}(f) \right|$$

$$\leq \frac{1}{M^2} \sum_{b=a+1}^{k} \sum_{j:t_j \in \mathscr{T}_b} \left| l\left(f, v_a^{i_a}, t_j\right) - l\left(f, v^a, t_j\right) \right| \tag{57}$$

$$+ \frac{1}{M^2} \sum_{m=1}^{a-1} \sum_{i:t_i \in \mathscr{T}_m} \left| l\left(f, t_i, v_a^{i_a}\right) - l\left(f, t_i, v^a\right) \right|$$

$$\leq \frac{kB}{M}.$$

Therefore, applying McDiarmid's inequality, we know that for any $\varepsilon > 0$,

$$\mathbf{P}_{\mathscr{T}}\left(\widehat{er}_{\mathscr{T},l}(f) - er_l(f) \leq \varepsilon\right) \leq \exp\left\{-\frac{\varepsilon^2\lambda^2 M}{32\sigma^4\kappa^4}\right\}. \tag{58}$$

The result follows by setting the right-hand side equal to δ and solving it for ε. $\qquad \square$

We are now in a position to prove our main result (Theorem 5).

Proof of Theorem 5. We use Corollary 17 and apply Theorem 10 with $\delta/2$ to get that with probability of at least $1 - \delta/2$,

$$er_l(f_{\mathscr{T}}) \leq \widehat{er}_{\mathscr{T},l}(f_{\mathscr{T}}) + \frac{16\sigma^2\kappa^2}{\lambda M}$$

$$+ \left(\frac{16\sigma^2\kappa^2}{\lambda M} + \frac{B}{M}\right)\sqrt{\frac{M\ln(2/\delta)}{2}}. \tag{59}$$

Clearly,

$$\widehat{er}_{\mathscr{T},l}(f_{\mathscr{T}}) \leq \widehat{er}_{\mathscr{T},l}(f_{\mathscr{T}}) + \lambda\|f_{\mathscr{T}}\|_K^2 \leq \widehat{er}_{\mathscr{T},l}(f_l^*) + \lambda\|f_l^*\|_K^2. \tag{60}$$

Applying Lemma 18 to f_l^* with $\delta/2$, we, thus, get that with probability of at least $1 - \delta/2$:

$$\widehat{er}_{\mathscr{T},l}(f_{\mathscr{T}}) \leq er_l(f_l^*) + \frac{8\sigma^2\kappa^2}{\lambda}\sqrt{\frac{\ln(2/\delta)}{2M}} + \lambda\|f_l^*\|_K^2. \tag{61}$$

One finishes the proof by combining the inequalities in (59) and (61), each of which holds with probability at least $1 - \delta/2$, together with the condition in (7). $\qquad \square$

6. Conclusion

The main focus of this paper is on studying the stability and generalization properties of k-partite ranking algorithm used for ontology computation. This algorithm shows good intuition about the vertex in ontology graph mapping to a vertex in a line. The representation of vertices in ontology graph does not take real-valued labels, and the samples are given by preference graph (pairwise vertices in different ranking rates). This setting is suitable for ontology. We have derived generalization bounds for k-partite ranking algorithms in this setting using the notion of algorithmic stability. It is also shown that k-partite ranking algorithms revealing good stability properties have good generalization properties. Our results are applied to obtain generalization bounds for kernel-based k-partite ranking algorithms that perform regularization in a reproducing kernel Hilbert space.

Acknowledgments

References

[1] S. Agarwal, "Ranking on graph data," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 25–32, June 2006.

[2] Y. Y. Wang, W. Gao, Y. G. Zhang, and Y. Gao, "Ontology similarity computation using ranking learning method," in *Proceedings of the IEEE International Conference Computational Intelligence and Industrial Application*, pp. 20–23, 2010.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[4] W. Gao and D. X. Zhou, "Convergence of spectral clustering with a general similarity function," *Science China Mathematics*, vol. 42, no. 10, pp. 985–994, 2012.

[5] U. von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *The Annals of Statistics*, vol. 36, no. 2, pp. 555–586, 2008.

[6] S. Smale and D.-X. Zhou, "Geometry on probability spaces," *Constructive Approximation*, vol. 30, no. 3, pp. 311–323, 2009.

[7] S. Rajaram and S. Agarwal, "Generalization bounds for k-partite ranking," in *Proceedings of the NIPS Workshop on Learning to Rank*, pp. 28–33, British Columbia, Canada, 2005.

[8] S. Agarwal and P. Niyogi, "Generalization bounds for ranking algorithms via algorithmic stability," *Journal of Machine Learning Research*, vol. 10, pp. 441–474, 2009.

[9] A. Caponnetto and Y. Yao, "Cross-validation based adaptation for regularization operators in learning theory," *Analysis and Applications*, vol. 8, no. 2, pp. 161–183, 2010.

[10] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[11] T. Hu, "Online regression with varying Gaussians and nonidentical distributions," *Analysis and Applications*, vol. 9, no. 4, pp. 395–408, 2011.

[12] S. Smale and D.-X. Zhou, "Estimating the approximation error in learning theory," *Analysis and Applications*, vol. 1, no. 1, pp. 17–41, 2003.

[13] H.-Y. Wang, Q.-W. Xiao, and D.-X. Zhou, "An approximation theory approach to learning with ℓ^1 regularization," *Journal of Approximation Theory*, vol. 167, pp. 240–258, 2013.

[14] D. C. Yang and D. Y. Yang, "Real-variable characterizations of Hardy spaces associated with Bessel operators," *Analysis and Applications*, vol. 9, no. 3, pp. 345–368, 2011.

[15] S. Agarwal, "Learning to rank on graphs," *Machine Learning*, vol. 81, no. 3, pp. 333–357, 2010.

[16] C. McDiarmid, "On the method of bounded differences," in *Surveys in Combinatorics*, vol. 141, pp. 148–188, Cambridge University Press, 1989.

[17] S. Agarwal and P. Niyogi, "Stability and generalization of bipartite ranking algorithms," in *Proceedings of the 18th Annual Conference on Learning Theory*, vol. 3559 of *Lecture Notes in Computer Science*, pp. 32–47, 2005.

[18] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, no. 3, pp. 499–526, 2002.