Query-dependent ranking and its asymptotic properties

Dai, Ben; Wang, Junhui

Published in:
Electronic Journal of Statistics

Published: 01/01/2019

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1214/19-EJS1531

Publication details:
Dai, B., & Wang, J. (2019). Query-dependent ranking and its asymptotic properties. *Electronic Journal of Statistics*, *13*(1), 465-488. https://doi.org/10.1214/19-EJS1531

Electronic Journal of Statistics
Vol. 13 (2006) 465–488
ISSN: 1935-7524
https://doi.org/10.1214/19-EJS1531

Query-dependent ranking and its asymptotic properties

Ben Dai and Junhui Wang

School of Data Science,
City University of Hong Kong,
Kowloon Tong, Hong Kong
e-mail: bendai2-c@my.cityu.edu.hk; j.h.wang@cityu.edu.hk

Abstract: Ranking, also known as learning to rank in machine learning community, is to rank a number of items based on their relevance to a specific query. In literature, most ranking methods use a uniform ranking function to evaluate the relevance, which completely ignores the heterogeneity among queries. To admit different ranking functions for various queries, a general U-process formulation for query-dependent ranking is developed. It allows to incorporate neighborhood structure among queries via various forms of smoothing weights to improve the ranking performance. One of its salient features is its capability of producing reasonable rankings for novel queries that are absent in the training set, which is commonly encountered in practice but often neglected in the literature. The proposed method is implemented via an inexact alternating direction method of multipliers (ADMM) for each query parallelly. Its asymptotic risk bound is established, showing that it achieves desirable ranking accuracy at a fast rate for any query including the novel ones. Furthermore, simulated examples and a real application to the Yahoo! challenge dataset also support the advantage of the query-dependent ranking method against existing competitors.

MSC 2010 subject classifications: Primary 62F07, 62G20; secondary 62H30, 62P30.
Keywords and phrases: Empirical process, ranking, U-process, local smoothing, margin loss, SVM.

Received September 2018.

Contents

1. Introduction

Ranking has a long-standing history in statistical literature [2]. It arises in medical informatics, psychology, information retrieval, and many other fields, where a number of items are to be ranked. Specifically, in information retrieval, ranking finds applications in document retrieval [22], computational biology [1], and collaborative filtering [23]. Taking document retrieval as an illustrating example, its goal is to rank a large collection of text documents given a text-based query and retrieve the top-ranked documents. In 2010, Yahoo! initiated a learning to rank challenge on web search, and more than 1,000 teams had participated [9].

Broadly speaking, the existing ranking methods in literature can be categorized into three types: pointwise methods, pairwise methods, and listwise methods. The pointwise methods convert ranking to an ordinal regression problem, and then apply the existing ordinal regression methods to estimate the relevance score. For example, OC SVM [29] embeds ranking in an ordinal SVM formulation, which essentially decomposes ordinal regression into a series of binary classification problems. However, as pointed out in [22], the pointwise methods are often sub-optimal, since ranking concerns about the relative order of the documents while ordinal regression focuses on the absolute relevance score of each document-query pair. The pairwise methods try to correctly rank all the document pairs for the same query, and formulate the ranking problem as minimization of U-processes, such as RankSVM [17]. The listwise methods consider all the documents associated with the same query simultaneously. They attempt to find the most preferred ranking of the documents to maximize certain ranking performance measures, such as SVM MAP [42].

In most existing ranking methods, a uniform ranking function is often assumed for all queries. Yet its clear drawback is that it completely ignores the heterogeneity among different queries and thus the resultant ranking performance is suboptimal. In practice, the semantics, frequencies and ranking patterns can vary significantly from one query to another, and thus lead to sever heterogeneity among queries. For example, for two entirely different queries "Donald Trump" and "LeBron James", the number of political terms shall contribute differently to the ranking performance for these two queries. Hence, using a uniform ranking function would compromise different types of queries and deteriorate the ranking performance. In recent years, query-dependent ranking has been proposed to circumvent this problem. For example, [14] proposes to use the RankSVM to train a query-specific ranking function based on the nearest neighbors of the target query, and [6] employs the query categorization to develop the position-sensitive query-dependent loss functions to estimate the ranking function. Even though superior numerical performance has been widely reported, very little is known about the asymptotic properties of the query-dependent ranking methods, such as how the neighborhood structure of queries can affect the ranking accuracy.

The main contribution of this paper is two-fold. First, a general query-dependent ranking formulation is developed based on the idea of kernel smoothing, which not only contains the existing methods [14] as special cases, but also allows for more appropriate modelling about the kernel weights. The formulation admits query-specific ranking functions for different queries, incorporates neighborhood structure among queries to improve the ranking performance, and produces satisfactory rankings for novel queries that are not present in the training set. Second, asymptotic analysis is conducted to establish the uniform convergence rate of the developed ranking formulation. The uniform convergence is established for almost all queries, which provides theoretical guarantee for the developed ranking formulation in producing appropriate ranking for novel queries that are absent from the training set. More importantly, the established rate of convergence is fast as it goes to zero as the number of queries diverges. In sharp contrast, the existing results [11, 28], mainly based on the theory of U-processes [27], are governed by the small number of documents retrieved for each query, leading to a much slower rate of convergence.

The rest of the paper is organized as follows. Section 2 briefly introduces the formulation of ranking and some popular ranking metrics. Section 3 presents the proposed query-dependent ranking method and its implementation via the inexact ADMM algorithm. Section 4 establishes asymptotic risk bounds for the proposed method. Section 5 examines the numerical performance of the proposed method in simulated examples and a real application to the Yahoo! challenge dataset. A brief summary is given in Section 6, and the Appendix is devoted to the technical proofs.

2. Ranking metrics

In ranking problem, a training observation is a triples of (query, document, relevance), where each query is represented by a p-dimensional feature vector $Q \in \mathcal{R}^p$, its associated document is represented by a b-dimensional feature vector $D \in \mathcal{R}^b$, and $Y \in \mathcal{R}$ denotes the relevance score describing how related D is to Q. Here p and b are the number of features for the query and document, respectively. Popular features used in Q includes number of terms, frequency of each term, click-through rate of the query, and D includes document statistics, textual similarity between the query and the document, the BM25 and PageRank scores [9]. Typically, multiple documents are retrieved for the same Q, and the primary goal of ranking is to construct a ranking function f that can produce an appropriate ranking of all the retrieved documents for each query. Yet as discussed in the introduction, most of the existing ranking methods focus on a uniform ranking function regardless of the heterogeneity among queries, which in turn leads to suboptimal ranking performance. As a natural remedy, query-specific ranking function f_{q_0} shall be used for each query q_0 to overcome the limitation of a uniform ranking function.

To assess the ranking accuracy of f_{q_0}, a number of metrics in literature can be used, including Kendall's tau [19], the expected reciprocal rank (ERR; [12]), the discounted cumulative gain (DCG; [18]) and the normalized discounted cumulative gain (NDCG; [18]). Kendall's tau is a well-studied ranking metric, which allows for convex surrogate losses [17] with established asymptotic properties [11]. Both NDCG and ERR suggest to reward correctly top-ranked documents more than those correctly ranked in the bottom. This is sensible in assessing ranking accuracy in practice, but also casts great challenges both computationally and theoretically. Particularly, it is proved in [37] that NDCG converges to 1 almost surely for any ranking function when sample size goes to infinity, suggesting that NDCG may not be able to distinguish the ranking functions at limiting case. It has also been shown in [8] that no convex surrogate loss can be calibrated for ERR, suggesting that nonconvex optimization is inevitable in order to optimize ERR. Therefore, even though Kendall's tau may not distinguish the top-ranked and bottom-ranked documents, we still focus on it in the subsequent analysis, noting that the formulation can be naturally extended to other reasonable ranking metrics.

Denote the ranking function for query q_0 as $f_{q_0}(D, D') : \mathcal{R}^b \times \mathcal{R}^b \to \mathcal{R}$, which provides the relative rank of documents D and D'. If $f_{q_0}(D, D') > 0$, then f_{q_0} ranks D higher than D' for q_0, and vice versa. Kendall's tau measures the probability of correctly evaluating of the relative order of any two retrieved documents,

$$\tau(f_{q_0}) = P\big((Y - Y')f_{q_0}(D, D') \geq 0 \big| Q = q_0\big), \tag{2.1}$$

where D and D' are two independent documents retrieved for the same query q_0, and Y and Y' are their respective relevance scores. This definition slightly differs from the existing Kendall's tau in literature in that it takes conditional expectation over the documents retrieved for q_0. It still enjoys the desirable properties of the existing Kendall's tau, and admits many ranking metrics as its special cases, including DCG with a specific discount and gain function [21].

More importantly, Kendall's tau naturally leads to the pairwise ranking function $f_{q_0}(D, D')$ which focuses on the relative order of any pair of documents D and D'. Note that in (2.1), $(Y - Y')f_{q_0}(D, D') \geq 0$ indicates f_{q_0} correctly ranks D and D'. Therefore, the mis-ranking error $\mathrm{MRE}(f_{q_0})$ can be defined as

$$\mathrm{MRE}(f_{q_0}) = 1 - \tau(f_{q_0}) = P\big((Y - Y')f_{q_0}(D, D') \leq 0 \big| Q = q_0\big). \tag{2.2}$$

Lemma 1 gives the ideal ranking function $f^*_{q_0}(\cdot)$ that yields the smallest $\mathrm{MRE}(f_{q_0})$, or equivalently the largest $\tau(f_{q_0})$.

Lemma 1. *The global minimizer of $MRE(f_{q_0})$ must satisfy that*

$$\mathrm{sign}\left(f^*_{q_0}(d, d')\right) = \mathrm{sign}\left(P(Y \geq Y' | Q = q_0, D = d, D' = d') - 1/2\right), \tag{2.3}$$

for any two documents d and d' retrieved for the same query q_0.

It is clear that $\mathrm{MRE}(f_{q_0})$ is a sensible ranking metric as $f^*_{q_0}$ ranks each pair of documents according to the most probable order. A similar result for the uniform ranking function can be found in [11]. Essentially, MRE can be regarded as an analogy to the misclassification error in classification, and the ideal $f^*_{q_0}$ in (2.3) is analogous to the Bayes decision function [3]. One notable fact from Lemma 1 is that the ideal query-specific ranking function $f^*_{q_0}$ is solely determined by the documents retrieved for q_0. Naturally, $f^*_{q_0}$ can be estimated by an individual ranking method, which mimics the uniform ranking method but only use the documents retrieved for q_0. The individual ranking method accounts for the heterogeneity among queries through different ranking functions. However, it has its own limitations. First, the number of documents retrieved for a specific query is often small. As provided in the Yahoo! challenge dataset, the average number of documents per query is about 24, and thus the estimation accuracy of f_{q_0} can be unsatisfactory. Second and more severely, it is difficult to apply the individual ranking method to produce rankings for novel queries that are not present in the training set. Therefore, a flexible query-dependent ranking method that allows pooling and sifting information across queries is in demand.

3. Query-dependent ranking

This section develops a general query-dependent ranking formulation, which not only admits query-specific ranking function f_{q_0} to account for the heterogeneity among different queries, but also incorporates the neighborhood structure among similar queries to improve the ranking performance.

3.1. A general ranking formulation

Consider a training set of $(q_i, d_{ij}, y_{ij}); i = 1, \dots, n; j = 1, \dots, N_i$, where $q_i \in \mathcal{R}^p$ is the feature of the i-th query, $d_{ij} \in \mathcal{R}^b$ is the feature of the j-th document retrieved for q_i, and $y_{ij} \in \mathcal{R}$ is the relevance score between d_{ij} and q_i, n is the number of queries, and N_i is the number of documents retrieved for q_i. Here $\mathbf{x}_{ij}$ is often created based on the query-document pair (q_i, d_{ij}), such as BM25, PageRank, query statistics and document statistics. Furthermore, let $\mathbf{x}_{ij} = (q_i, d_{ij})$ for simplicity. Apparently, $(\mathbf{x}_{ij}, y_{ij})$'s are not independent as multiple documents are retrieved for the same query. A query-specific ranking function $f_{q_0}(d, d') = f_{q_0}(\mathbf{x}, \mathbf{x}')$ is to be constructed to determine the relative order of any pair of documents d and d' so that high ranking accuracy can be achieved.

For any given q_0, the proposed query-dependent ranking method is formulated as

$$\min_{f_{q_0} \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_i}{N_i(N_i-1)} \sum_{j \neq l} L\big(\operatorname{sign}(y_{ij} - y_{il}) f_{q_0}(\mathbf{x}_{ij}, \mathbf{x}_{il})\big) + \lambda J(f_{q_0}), \quad (3.1)$$

where $\mathcal{H}_K$ is a reproducing kernel Hilbert space (RKHS; [34]) specified by a kernel function $K(\cdot, \cdot)$, L is a surrogate loss function, $\pi_i = \pi(q_i, q_0)$ is a weight function, $J(f_{q_0}) = \|f_{q_0}\|^2_{\mathcal{H}_K}$ is a regularization term, and λ is a tuning parameter. The surrogate loss L can be set as the hinge loss $L(u) = (1-u)_+$ [35], the ψ-loss $L(u) = \min(1, (1-u)_+)$ [30, 39], or the logistic loss $\log(1 + \exp(-u))$ [44]. For illustration, we focus on the hinge loss $L(u) = (1-u)_+$ in the sequel, but the proposed formulation can be adapted to other loss functions.

The key advantage of the ranking formulation in (3.1) is that it allows pooling information from other queries for estimating f_{q_0}. More specifically, the first loss part in (3.1) can be regarded as an empirical version of

$$l^{\pi}(f_{q_0}) = \mathbb{E}\big(\pi(Q,q_0)L(\operatorname{sign}(Y-Y')f_{q_0}(D,D'))\big), \tag{3.2}$$

where the expectation is taken with respect to (Q,D,Y,D',Y'), and (D,D') are a pair of documents retrieved for any query Q. It is clear that all the document-query pairs can contribute to the estimation of f_{q_0}, where their contributions are controlled by $\pi(Q,q_0)$. This general formulation not only increases the effective sample size to improve estimation accuracy, but also provides a reasonable estimation formulation for novel q_0's which are not present in the training set.

The proposed formulation in (3.1) is general in that $\pi(q,q_0)$ can take various forms, leading to different ranking methods.

If $\pi(q,q_0) \equiv 1$, then the developed ranking formulation in (3.1) degenerates to the uniform ranking method (e.g. RankSVM; [17]), where $f_{q_0} = f$ for all q_0's and can be obtained by solving

$$\min_{f\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{N_i(N_i-1)}\sum_{j\neq l} L\big(\operatorname{sign}(y_{ij}-y_{il})f(\mathbf{x}_{ij},\mathbf{x}_{il})\big) + \lambda J(f). \tag{3.3}$$

If $\pi(q,q_0) = I(q=q_0)$, then the developed ranking formulation reduces to the individual ranking method, where each f_{q_i} is obtained by solving

$$\min_{f_{q_i}\in\mathcal{H}_K} \frac{1}{N_i(N_i-1)}\sum_{j\neq l} L\big(\operatorname{sign}(y_{ij}-y_{il})f_{q_i}(\mathbf{x}_{ij},\mathbf{x}_{il})\big) + \lambda J(f_{q_i}). \tag{3.4}$$

Note that individual method can only estimate f_{q_i} for the observed q_i's in the training set.

A more appropriate choice is $\pi(q,q_0) = h^{-p}\mathcal{W}\big(h^{-1}(q-q_0)\big)$, where $\mathcal{W}(\cdot)$ is the multivariate kernel function. Typically, $\mathcal{W}(\cdot)$ is set as $\mathcal{W}\big(h^{-1}(q-q_0)\big) = W(h^{-1}\|q-q_0\|)$, so that the contribution of documents retrieved for q decays as q deviates from q_0. Here h is the bandwidth for a kernel, and $W(\cdot)$ can be any popular univariate smoothing kernel [13], such as the rectangular kernel, the Epanechnikov kernel, the tri-cube kernel, or the Gaussian kernel. Note that if the rectangular kernel is used, the developed ranking formulation becomes the method in [14] using k-nearest neighbors. Moreover, $\mathcal{W}(\cdot)$ can be defined as a product kernel to admit ordinal and categorical features. More specifically, the Kendall and Mallows kernels can be used to compute a similarity between two ordinal features [20, 24], and the overlap kernels can be used for the categorical features [4]. We can also use various numerical embeddings, including ordinal embedding [31], Word2Vec [25] and graph embeddings [15], to pre-process the ordinal, categorical and textual query features into continuous features to be included in the kernel function.

3.2. Scalable computation

One salient aspect of the ranking formulation in (3.1) is that it estimates f_{q_0} for each q_0 separately, and thus the estimation can be parallelized to handle a large number of queries. In addition, with the hinge loss $L(u) = (1-u)_+$, solving (3.1) for each f_{q_0} is a large-scale constrained convex optimization task, and can be tackled by an efficient inexact alternating direction method of multipliers algorithm (ADMM; [7]).

Denote a documents pair by $\tilde{\mathbf{x}} = \binom{\mathbf{x}}{\mathbf{x}'}$, and a kernel matrix by $\mathbf{K}$ with $\mathbf{K}_{tt'} = K(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t'})$, where $(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t'})$ indexes all possible pairs of documents. By the representer theorem [34] for the RKHS, the solution to (3.1) must be of form $f_{q_0}(\tilde{\mathbf{x}}) = \sum_t \alpha_{t,q_0} K(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}) + \alpha_{0,q_0}$, and the penalty becomes $J(f_{q_0}) = \frac{1}{2}\boldsymbol{\alpha}_{q_0}^T \mathbf{K}\, \boldsymbol{\alpha}_{q_0}$. After introducing slack variables $\tilde{\gamma}_{ijl}$, (3.1) can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}_{q_0}, \tilde{\gamma}_{ijl}} && \frac{1}{n}\sum_{i=1}^{n} \tilde{\pi}_i \sum_{j \neq l} (\tilde{\gamma}_{ijl})_+ + \frac{\lambda}{2}\boldsymbol{\alpha}_{q_0}^T \mathbf{K} \boldsymbol{\alpha}_{q_0} && (3.5) \\ &\text{subject to} && 1 - y_{ijl}\Big(\sum_t \alpha_{t,q_0} K(\tilde{\mathbf{x}}_{ijl}, \tilde{\mathbf{x}}_t) + \alpha_{0,q_0}\Big) = \tilde{\gamma}_{ijl},\ i = 1, \cdots, n; j \neq l, \end{aligned}$$

where $\tilde{\pi}_i = \frac{1}{N_i(N_i-1)}\pi_i$, $y_{ijl} = \operatorname{sign}(y_{ij} - y_{il})$, and $\tilde{\mathbf{x}}_{ijl} = \binom{\mathbf{x}_{ij}}{\mathbf{x}_{il}}$. Clearly, the optimization task becomes a weighted SVM [40], which can be solved by the ADMM algorithm as follows.

First, let $\xi_{ijl} = (\tilde{\gamma}_{ijl})_+$ and $\gamma_{ijl} = \xi_{ijl} - \tilde{\gamma}_{ijl} \geq 0$, the augmented Lagrangian of (3.5) is

$$\begin{aligned} L_\rho(\boldsymbol{\alpha}_{q_0}, \boldsymbol{\xi}, \boldsymbol{\gamma}, \mathbf{u}) = {} & \frac{1}{n}\sum_{i=1}^{n} \tilde{\pi}_i \sum_{j \neq l} \xi_{ijl} + \frac{\lambda}{2}\boldsymbol{\alpha}_{q_0}^T \mathbf{K} \boldsymbol{\alpha}_{q_0} \\ & + \frac{\rho}{2}\sum_{i=1}^{n}\sum_{j\neq l}\Big(y_{ijl}\big(\sum_t \alpha_{t,q_0} K(\tilde{\mathbf{x}}_{ijl}, \tilde{\mathbf{x}}_t) + \alpha_{0,q_0}\big) - 1 + \xi_{ijl} - \gamma_{ijl} + u_{ijl}\Big)^2, \end{aligned}$$

subject to the positivity constraints $\xi_{ijl} \geq 0$ and $\gamma_{ijl} \geq 0$, for $i = 1, \cdots, n$ and $j \neq l$. Here ρ is a Lagrangian multiplier. The ADMM algorithm updates $\boldsymbol{\alpha}_{q_0}$, $\boldsymbol{\xi}$, $\boldsymbol{\gamma}$ and $\mathbf{u}$ separately with others fixed, which requires substantial effort in inverting large matrices. To alleviate the computing burden, the inexact ADMM [36] is developed to bypass the matrix inversion and approximate the update in the subproblems.

Specifically, the inexact ADMM iteratively updates $\tilde{\boldsymbol{\alpha}}_{q_0}$, $\boldsymbol{\xi}$, $\boldsymbol{\gamma}$ and $\mathbf{u}$ as follows,

$$\begin{aligned} \tilde{\boldsymbol{\alpha}}_{q_0}^{(k+1)} &= \frac{\zeta - \lambda\tilde{\mathbf{K}}}{\zeta}\tilde{\boldsymbol{\alpha}}_{q_0}^{(k)} - \frac{\rho}{\zeta}\boldsymbol{v}^{(k)}, \\ \xi_{ijl}^{(k+1)} &= \max\Big(1 - y_{ijl}\big(1, \mathbf{K}_{\tilde{\mathbf{x}}_{ijl}}^T\big)\tilde{\boldsymbol{\alpha}}_{q_0}^{(k+1)} - u_{ijl}^{(k)} - \frac{1}{n\rho}\tilde{\pi}_i,\ 0\Big), \\ \gamma_{ijl}^{(k+1)} &= \max\Big(y_{ijl}\big(1, \mathbf{K}_{\tilde{\mathbf{x}}_{ijl}}^T\big)\tilde{\boldsymbol{\alpha}}_{q_0}^{(k+1)} - 1 + u_{ijl}^{(k)},\ 0\Big), \end{aligned}$$

$$u_{ijl}^{(k+1)} \quad = \quad u_{ijl}^{(k+1)} + y_{ijl}\big(1, \mathbf{K}_{\tilde{\mathbf{x}}_{ijl}}^T\big)\tilde{\boldsymbol{\alpha}}_{q_0}^{(k+1)} - 1 + \xi_{ijl}^{(k+1)} - \gamma_{ijl}^{(k+1)}.$$

Here $\tilde{\boldsymbol{\alpha}}_{q_0} = \binom{\alpha_{0,q_0}}{\boldsymbol{\alpha}_{q_0}}$, $\tilde{\mathbf{K}}$ is obtained by inserting 0 as the first row and column in $\mathbf{K}$, $\boldsymbol{v}^{(k)} = \sum_{i=1}^n \sum_{j\neq l} \big(1, \mathbf{K}_{\mathbf{x}_{ijl}}^T\big)^T y_{ijl}(y_{ijl}\big(1, \mathbf{K}_{\mathbf{x}_{ijl}}^T\big)\boldsymbol{\alpha}_{q_0}^{(k)} - 1 + \xi_{ijl}^{(k)} - \gamma_{ijl}^{(k)} + u_{ijl}^{(k)})$, $\mathbf{K}_{\tilde{\mathbf{x}}_{ijl}}$ is a column in $\mathbf{K}$ corresponding to $\tilde{\mathbf{x}}_{ijl}$, $\zeta \geq \varphi_{max}(\lambda\tilde{\mathbf{K}} + \rho\,\mathbf{M}^T\mathbf{M})$ with $\mathbf{M} = \big(y_{ijl}(1, \mathbf{K}_{\mathbf{x}_{ijl}}^T)\big)_{i=1,\cdots,n;j\neq l}$, and $\varphi_{max}(\cdot)$ denotes the largest eigenvalue of a matrix.

The inexact ADMM algorithm converges to a global solution of (3.1), and often produces reasonable approximations after a small number of iterations [36]. Each iteration only involves some matrix multiplication, its computational complexity is of order $O\big((\sum_{i=1}^n N_i(N_i-1))^2\big)$. Furthermore, $\tilde{\pi}_i$ is usually truncated to include only a small number of queries close to the target query, which will further reduce the computational cost.

Once $\hat{f}_{q_0}$ is obtained, it can be used to determine the relative order of any pair (d, d'), but it still requires further adjustment to produce the ranking of all documents due to the possible inconsistent order of the document pairs. Specifically, the score for each document can be estimated by solving

$$\min_{\boldsymbol{s}} \sum_{j,l} (s_j - s_l - \hat{f}_{q_0}(\mathbf{x}_j, \mathbf{x}_l))^2, \tag{3.6}$$

where s_j and s_l are the relevance scores correspond to d_j and d_l respectively, and the documents can be ranked according to the magnitude of s. Another more direct way is to replace f_{q_0} by the difference between two scoring functions, $f_{q_0}(\mathbf{x}_{ij}, \mathbf{x}_{il}) = s_{q_0}(\mathbf{x}_{ij}) - s_{q_0}(\mathbf{x}_{il})$. Then the proposed formulation can be rewritten as

$$\min_{s_{q_0}\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^n \frac{2\pi_i}{N_i(N_i-1)} \sum_{j>l} L\big(\operatorname{sign}(y_{ij}-y_{il})(s_{q_0}(\mathbf{x}_{ij}) - s_{q_0}(\mathbf{x}_{il}))\big) + \lambda J(s_{q_0}). \tag{3.7}$$

This formulation can largely alleviate the computation burden, especially when the number of documents is large. However, a consistent scoring function does not always exist, unless certain structure is imposed as in the bipartite ranking problem [32, 11].

4. Asymptotic ranking theory

In this section, we establish some theoretical results to quantify the asymptotic behavior of the query-dependent ranking formulation in (3.1). Its ranking accuracy, measured by MRE, converges to that of the ideal ranking function at a fast rate, which is governed by the number of queries and various tuning parameters.

For simplicity, let $\mathcal{H}_K$ be a RKHS with the Gaussian kernel $K(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \exp\{-\frac{\|\tilde{\mathbf{x}}-\tilde{\mathbf{x}}'\|^2}{\sigma^2}\}$, $L(u) = (1-u)_+$, and let $N_1 = N_2 = \cdots N_n = N$; that is, the numbers of documents are the same for all queries. Furthermore, assume the ideal ranking function $f^*_{q_0} \in \mathcal{H}_K$, and $\{(Q_i, Z_{i1}, \cdots, Z_{iN})\}_{i=1}^n$ are sampled independently, where $Z_{ij} = (D_{ij}, Y_{ij})$. Note that all these conditions are assumed only for the clarity of the theoretical results, which can be naturally extended to more general cases. The following technical assumptions are made.

Assumption A. Denote $\phi_{z|q}(z|q)$ as the conditional density of Z given $Q = q$, and $q = (q^{(1)}, \cdots, q^{(p)})^T$. For each $q^{(j)}$, $\int \frac{\partial \phi_{z|q}(z|q)}{\partial q^{(j)}} dz$ and $\int \frac{\partial^2 \phi_{z|q}(z|q)}{\partial (q^{(j)})^2} dz$ are bounded. The density $\phi_Q(q)$ is also bounded and has bounded first and second derivatives with respect to q.

Assumption B. Assume that $\pi(q, q_0) = h^{-p} \mathcal{W}\big(h^{-1}(q - q_0)\big)$, where $\mathcal{W}(\cdot)$ is a Lipschitz continuous multivariate kernel function satisfying $\int_{\mathcal{R}^p} \mathcal{W}(u)du = 1$, $\int_{\mathcal{R}^p} u\mathcal{W}(u)du = 0$, and $\int_{\mathcal{R}^p} u^T u \mathcal{W}(u)du$ is bounded.

Assumptions A and B are standard assumptions in the literature of kernel smoothing, and similar assumptions can be found in [38, 10, 16]. Specifically, Assumption A is a smoothness condition quantifying the behavior of (Z, Q), and assures the smoothness of the conditional distribution of Z given Q, so that information across neighboring queries can be pooled together through the weight function $\pi(q, q_0)$. It is a mild assumption and is satisfied if Z has bounded support and $\phi_{z|q}(z|q)$ is twice-continuously differentiable with respect to q. Assumption B is satisfied by many popular kernels including the Gaussian kernel, the Epanechnikov kernel and the tri-cube kernel.

Lemma 2. *Suppose Assumptions A and B are met, and let* $f^\pi_{q_0} = \operatorname{argmin}_{f_{q_0}} l^\pi(f_{q_0})$. *Then for any* $\eta > 0$*, we have*

$$\lim_{h \to 0} \sup_{q_0 \in \mathcal{Q}_\eta} \big(MRE(f^\pi_{q_0}) - MRE(f^*_{q_0}) \big) = 0,$$

where $\mathcal{Q}_\eta = \{q : \phi_Q(q) \geq \eta\}$ *contains all the queries with marginal density bounded away from* η*.*

Lemma 2 shows that the query-dependent formulation in (3.2) is appropriate in that its global minimizer converges to the ideal ranking function $f^*_{q_0}$ when h approaches 0. Lemma 2 also provides important foundation for establishing the asymptotic consistency of the sample-based ranking function $\hat{f}_{q_0}$, estimated from minimizing (3.1).

To proceed, we rewrite h and λ as h_n and λ_n to emphasize their dependence on n. Theorem 1 provides an upper bound for the mis-ranking error of $\hat{f}_{q_0}$, implying that it converges to the ideal performance at a fast rate.

Theorem 1. *Suppose Assumptions A and B are met. There exists a constant* $C > 0$*, such that for any* $\eta > 0$*, with probability at least* $1 - \exp(-\delta_n^2)$*, it holds*

$$\sup_{q_0 \in \mathcal{Q}_\eta} \big(MRE(\hat{f}_{q_0}) - MRE(f^*_{q_0}) \big) \leq C\eta^{-1} (n^{-1/2} h_n^{-(p+1)} \lambda_n^{-1/2} \delta_n + \lambda_n^{-1/2} h_n^2 + \lambda_n).$$

Theorem 1 establishes an upper bound for $\mathrm{MRE}(\hat{f}_{q_0})$. As a direct implication of Theorem 1, we have

$$\sup_{q_0 \in \mathcal{Q}_\eta} \big(\mathrm{MRE}(\hat{f}_{q_0}) - \mathrm{MRE}(f^*_{q_0}) \big) = O_p(\lambda_n^{-1/2} h_n^2 + \lambda_n) + O_p(n^{-1/2} h_n^{-(p+1)} \lambda_n^{-1/2}).$$

Setting $h_n = n^{-1/(6+2p)}$, $\lambda_n = n^{-2/(9+3p)}$, the above convergence rate simplifies to

$$\sup_{q_0 \in \mathcal{Q}_\eta} \left(\mathrm{MRE}(\hat{f}_{q_0}) - \mathrm{MRE}(f^*_{q_0}) \right) = O_p(n^{-2/(9+3p)}).$$

It is clear that the proposed query-dependent ranking formulation can pool information across neighboring queries, and ensures the ranking accuracy for all queries in $\mathcal{Q}_\eta$ converges at a rate in n. Note that this rate of convergence still holds for diverging p, which corresponds to the case that the number of features may increase with the number of queries. As a sharp contrast, most individual ranking methods can only achieve a convergence rate in N [11], which is undesirable as N is often small in practice. For example, in the Yahoo! challenge dataset, the average number of documents per query is about 24, which is far less than the number of queries in the dataset. More importantly, the convergence rate is established uniformly for all queries in $\mathcal{Q}_\eta$, which provides theoretical guarantee on its ranking performance for the novel queries.

5. Numerical experiments

This section examines the performance of the query-dependent ranking methods, where $\pi(q, q_0) = h^{-p} W(h^{-1} \|q - q_0\|_2)$ with a Gaussian kernel, or set adaptively to the K-nearest neighbors as in [14]. For simplicity, we denote them as *q-SVM* or *kNN-SVM*, respectively. Their performance is compared against the uniform rankSVM [17] and the individual rankSVM, denoted as *rank-SVM* and *indv-SVM* respectively. The ranking performance is measured by three widely adopted metrics, including MRE, ERR and NDCG [9]. For a specific query q_0, we compute the predicted relevance score for each document by $\hat{y}_i = \hat{f}_{q_0}(\mathbf{x}_i)$, and denote the decreasing rank of $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_N)$ as $(\hat{\theta}_1, \cdots, \hat{\theta}_N)$. Then, MRE is estimated as

$$\widehat{\mathrm{MRE}} = \frac{2}{N(N-1)} \sum_{i<j} I(y_{\hat{\theta}_i} < y_{\hat{\theta}_j}),$$

which is an empirical average of MRE based on the testing set. Normalized Discounted Cumulative Gain (NDCG; [18]) uses graded relevance scale of documents to measure the ranking quality,

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\text{Ideal DCG}} \quad \text{and} \quad \mathrm{DCG} = \sum_{i=1}^{\min(10,N)} \frac{2^{y_{\hat{\theta}_i}} - 1}{\log_2(1+i)},$$

where the ideal DCG is the maximum possible DCG, obtained by sorting documents according to their relevance scores. Expected Reciprocal Rank (ERR; [12]) is the expectation of the reciprocal of the position where a user stops his search under cascade user model,

$$\mathrm{ERR} = \sum_{i=1}^{N} \frac{1}{i} R(y_{\hat{\theta}_i}) \prod_{j=1}^{i-1} (1 - R(y_{\hat{\theta}_j})),$$

where $R(y) = \frac{2^y - 1}{2^{\max(\mathbf{y})}}$, and $\max(\mathbf{y})$ is the maximal relevance score. To be consistent, we report $\widehat{\mathrm{MRE}}$, 1 - NDCG, and 1 - ERR in the numerical experiments. Note that all the metrics are defined for each individual query, and the overall ranking performance is measured as the average over all queries.

5.1. Simulated examples

The simulated examples $\{q_i, \mathbf{x}_{ij}, y_{ij}\}_{i=1,\cdots,n; j=1,\cdots,N}$ are generated as follows. First, $\mathbf{x}_{ij}$ is generated independently from $N(\mathbf{0}_b, 0.1\boldsymbol{I}_b)$, where $\mathbf{0}_b$ is a vector of b zeros and $\boldsymbol{I}_b$ is a b-dimensional identity matrix. Two examples are examined with linear and nonlinear scoring functions.

Example 1: $n = 40, N = 50, b = 40$ with linear scoring function $s_i^*(\mathbf{x}_{ij}) = \mathbf{x}_{ij}^T \boldsymbol{\beta}_i^*$.

Example 2: Similar as [41] with $n = 40, N = 30, b = 4$, where $s_i^*(\mathbf{x}_{ij}) = \beta_{1,i}^* s_1(x_{1,ij}) + \beta_{2,i}^* s_2(x_{2,ij}) + \beta_{3,i}^* s_3(x_{3,ij}) + \beta_{4,i}^* s_4(x_{4,ij})$ with $s_1(u) = u$, $s_2(u) = (2u-1)^2$, $s_3(u) = \frac{sin(\pi u)}{2 - sin(\pi u)}$ and $s_4(u) = 0.1\sin(\pi u) + 0.2\cos(\pi u) + 0.3\sin^2(\pi u) + 0.4\cos^3(\pi u) + 0.5\sin^3(\pi u)$.

Next, let $(y_{ij})_{j=1}^N$ be the rank corresponding to the values of $(s^*(x_{ij}))_{j=1}^N$ in an ascending order, and $q_i = \boldsymbol{\beta}_i^* + \epsilon N(\mathbf{0}_b, \boldsymbol{I}_b)$ with ϵ denoting the level of noise for constructing the query features. For each example, we fix $\epsilon_i = 0.1$ and consider four scenarios, composing of different neighborhood structures among q_i's.

Scenario I: $\boldsymbol{\beta}_1^* = \cdots = \boldsymbol{\beta}_{40} = \mathbf{1}_b + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$ with $\mathbf{1}_b$ being a vector of b ones.

Scenario II: $\boldsymbol{\beta}_1^* = \cdots = \boldsymbol{\beta}_{20}^* = (\mathbf{1}_{b/2}^T, \mathbf{0}_{b/2}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$, and $\boldsymbol{\beta}_{21}^* = \cdots = \boldsymbol{\beta}_{40}^* = (\mathbf{0}_{b/2}^T, \mathbf{1}_{b/2}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$.

Scenario III: $\boldsymbol{\beta}_1^* = \cdots = \boldsymbol{\beta}_{10}^* = (\mathbf{1}_{b/4}^T, \mathbf{0}_{3b/4}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$, $\boldsymbol{\beta}_{11}^* = \cdots = \boldsymbol{\beta}_{20}^* = (\mathbf{0}_{b/4}^T, \mathbf{1}_{b/4}^T, \mathbf{0}_{b/2}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$, $\boldsymbol{\beta}_{21}^* = \cdots = \boldsymbol{\beta}_{30}^* = (\mathbf{0}_{b/2}^T, \mathbf{1}_{b/4}^T, \mathbf{0}_{b/4}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$, $\boldsymbol{\beta}_{31}^* = \cdots = \boldsymbol{\beta}_{40}^* = (\mathbf{0}_{3b/4}^T, \mathbf{1}_{b/4}^T)^T + N(\mathbf{1}_b, 0.1\boldsymbol{I}_b)$.

Scenario IV: $\boldsymbol{\beta}_i^* \sim N(\mathbf{0}_b, \boldsymbol{I}_b)$, $i = 1, \cdots, 40$.

Furthermore, a proportion, denoted as γ, of queries and all of their retrieved documents are reserved in the testing set to mimic the novel queries. For the remaining queries, documents are assigned to the training, validation and testing sets with ratio 20% : 20% : 60%.

In Example 1, the linear kernel is used for all methods, and in Example 2 the Gaussian kernel is used. Scale parameter σ^2 in the Gaussian kernel is fixed as the median of pairwise Euclidean distances within the training set. A grid search on the validation set is conducted to find the optimal λ, where the grid for all methods is set as $\{10^{(\nu-31)/10}; \nu = 1, \cdots, 61\}$. For *kNN-SVM*, $k = 15$ is set, and for *q-SVM*, h is set as the median of pairwise Euclidean distance between all queries and the target query, and we truncate π_i by taking top 15 queries closest to the target query. Note that *kNN-SVM* treats the top 15 queries equally in

Table 1
The averaged $\widehat{\text{MRE}}$, 1-ERR and 1-NDCG and their standard errors (in parenthesis) of various methods in Example 1.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
Scenario I					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.04(0.001)	0.30(0.001)	0.08(0.001)	0.08(0.001)
	1 - ERR	0.05(0.003)	0.50(0.008)	0.11(0.004)	0.10(0.004)
	1 - NDCG	0.03(0.002)	0.45(0.006)	0.08(0.001)	0.08(0.001)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.04(0.001)	0.35(0.001)	0.09(0.001)	0.08(0.001)
	1 - ERR	0.05(0.003)	0.54(0.007)	0.11(0.004)	0.11(0.003)
	1 - NDCG	0.04(0.002)	0.51(0.006)	0.09(0.003)	0.08(0.002)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.05(0.001)	0.38(0.001)	0.09(0.001)	0.09(0.001)
	1 - ERR	0.05(0.002)	0.59(0.007)	0.12(0.005)	0.11(0.004)
	1 - NDCG	0.04(0.002)	0.56(0.006)	0.09(0.003)	0.09(0.003)
Scenario II					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.27(0.001)	0.30(0.002)	0.09(0.002)	0.09(0.002)
	1 - ERR	0.45(0.007)	0.52(0.005)	0.12(0.005)	0.12(0.005)
	1 - NDCG	0.40(0.005)	0.47(0.004)	0.09(0.001)	0.09(0.003)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.27(0.001)	0.35(0.001)	0.09(0.002)	0.09(0.001)
	1 - ERR	0.46(0.007)	0.55(0.006)	0.12(0.005)	0.12(0.005)
	1 - NDCG	0.41(0.005)	0.51(0.005)	0.09(0.002)	0.09(0.001)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.29(0.002)	0.38(0.001)	0.14(0.004)	0.11(0.002)
	1 - ERR	0.45(0.006)	0.59(0.006)	0.19(0.006)	0.14(0.005)
	1 - NDCG	0.42(0.005)	0.55(0.006)	0.16(0.004)	0.12(0.004)
Scenario III					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.30(0.002)	0.30(0.001)	0.23(0.004)	0.16(0.003)
	1 - ERR	0.52(0.008)	0.52(0.007)	0.37(0.008)	0.26(0.007)
	1 - NDCG	0.47(0.007)	0.47(0.006)	0.32(0.006)	0.21(0.003)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.30(0.002)	0.34(0.002)	0.28(0.004)	0.21(0.003)
	1 - ERR	0.51(0.006)	0.54(0.008)	0.44(0.006)	0.32(0.006)
	1 - NDCG	0.48(0.005)	0.50(0.007)	0.40(0.006)	0.28(0.005)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.31(0.001)	0.38(0.001)	0.33(0.004)	0.28(0.005)
	1 - ERR	0.52(0.007)	0.58(0.006)	0.50(0.007)	0.38(0.006)
	1 - NDCG	0.49(0.006)	0.55(0.006)	0.46(0.004)	0.28(0.005)
Scenario IV					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.48(0.002)	0.30(0.002)	0.42(0.003)	0.36(0.003)
	1 - ERR	0.78(0.006)	0.52(0.007)	0.60(0.007)	0.62(0.007)
	1 - NDCG	0.78(0.006)	0.47(0.006)	0.68(0.006)	0.59(0.007)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.48(0.002)	0.34(0.001)	0.42(0.003)	0.38(0.003)
	1 - ERR	0.79(0.005)	0.55(0.006)	0.72(0.006)	0.65(0.005)
	1 - NDCG	0.78(0.005)	0.50(0.005)	0.60(0.006)	0.61(0.003)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.48(0.003)	0.38(0.001)	0.44(0.003)	0.40(0.003)
	1 - ERR	0.78(0.006)	0.58(0.007)	0.74(0.006)	0.65(0.005)
	1 - NDCG	0.78(0.005)	0.55(0.006)	0.72(0.005)	0.62(0.005)

training, but *q-SVM* gives different weights to these 15 queries depending on their closeness to the target query. For *indv-SVM*, as it is not applicable to generate rankings for novel queries, a random ranking is given by *indv-SVM* to the documents associated with the novel queries.

All scenarios are replicated 50 times, and the averaged performance measures and their corresponding standard errors are summarized in Tables 1 and 2.

It is evident that *q-SVM* and *kNN-SVM* substantially outperform *rank-SVM* and *indv-SVM* in Scenario II, III and IV, except that *rank-SVM* yields better performance in Scenario I where a uniform scoring function is used to generate data for all queries. The amount of improvement is substantial, with the largest improvement of 62.1% and 71.1% over *rank-SVM* and *indv-SVM* in $\widehat{\text{MRE}}$, respectively. It is also interesting to note that both *q-SVM* and *kNN-SVM* are very robust against missing ratio γ compared with *indv-SVM*, whose performance can be severely deteriorated by the missing queries. Furthermore, *q-SVM* and *kNN-SVM* share similar performance in Scenarios I and II. How-

Table 2

The averaged $\widehat{\text{MRE}}$, 1-ERR and 1-NDCG and their standard errors (in parenthesis) of various methods in Example 2.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
Scenario I					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.08(0.001)	0.22(0.003)	0.11(0.003)	0.11(0.004)
	1 - ERR	0.09(0.006)	0.31(0.009)	0.12(0.005)	0.12(0.008)
	1 - NDCG	0.07(0.004)	0.25(0.007)	0.10(0.004)	0.10(0.006)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.08(0.001)	0.28(0.002)	0.11(0.004)	0.11(0.003)
	1 - ERR	0.09(0.006)	0.35(0.008)	0.12(0.007)	0.13(0.008)
	1 - NDCG	0.07(0.004)	0.30(0.007)	0.10(0.005)	0.10(0.005)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.08(0.001)	0.32(0.002)	0.12(0.004)	0.11(0.004)
	1 - ERR	0.09(0.005)	0.41(0.008)	0.14(0.006)	0.13(0.005)
	1 - NDCG	0.07(0.003)	0.36(0.007)	0.11(0.004)	0.10(0.004)
Scenario II					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.27(0.003)	0.22(0.006)	0.14(0.010)	0.15(0.011)
	1 - ERR	0.41(0.010)	0.32(0.012)	0.19(0.014)	0.20(0.017)
	1 - NDCG	0.34(0.009)	0.26(0.010)	0.15(0.012)	0.16(0.014)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.28(0.004)	0.28(0.003)	0.13(0.015)	0.15(0.018)
	1 - ERR	0.41(0.012)	0.36(0.012)	0.17(0.013)	0.21(0.018)
	1 - NDCG	0.35(0.011)	0.32(0.010)	0.14(0.012)	0.17(0.016)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.29(0.005)	0.33(0.005)	0.15(0.015)	0.16(0.015)
	1 - ERR	0.42(0.012)	0.42(0.014)	0.21(0.016)	0.22(0.017)
	1 - NDCG	0.37(0.011)	0.37(0.012)	0.17(0.015)	0.18(0.016)
Scenario III					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.34(0.008)	0.22(0.007)	0.19(0.007)	0.17(0.006)
	1 - ERR	0.51(0.017)	0.33(0.014)	0.27(0.011)	0.26(0.009)
	1 - NDCG	0.46(0.016)	0.27(0.012)	0.22(0.009)	0.21(0.008)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.34(0.009)	0.28(0.006)	0.21(0.007)	0.19(0.006)
	1 - ERR	0.53(0.018)	0.39(0.013)	0.28(0.008)	0.27(0.007)
	1 - NDCG	0.47(0.018)	0.33(0.011)	0.23(0.006)	0.22(0.006)
$\gamma = 0.2$	MRE	0.35(0.010)	0.32(0.004)	0.26(0.006)	0.23(0.008)
	1 - ERR	0.52(0.017)	0.41(0.012)	0.35(0.013)	0.31(0.010)
	1 - NDCG	0.47(0.017)	0.37(0.010)	0.30(0.010)	0.26(0.008)
Scenario IV					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.46(0.005)	0.22(0.003)	0.19(0.004)	0.17(0.004)
	1 - ERR	0.32(0.014)	0.30(0.008)	0.26(0.008)	0.24(0.008)
	1 - NDCG	0.75(0.014)	0.25(0.006)	0.21(0.006)	0.19(0.006)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.47(0.005)	0.28(0.003)	0.20(0.006)	0.18(0.005)
	1 - ERR	0.56(0.014)	0.36(0.007)	0.27(0.006)	0.26(0.007)
	1 - NDCG	0.63(0.015)	0.31(0.006)	0.22(0.005)	0.21(0.006)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.47(0.005)	0.32(0.002)	0.21(0.007)	0.20(0.007)
	1 - ERR	0.68(0.014)	0.42(0.007)	0.29(0.009)	0.27(0.008)
	1 - NDCG	0.64(0.015)	0.37(0.006)	0.24(0.007)	0.23(0.007)

ever, *q-SVM* outperforms *kNN-SVM* with the largest improvement of 30.4% in Scenarios III, where the selected 15 queries for *q-SVM* and *kNN-SVM* are "oversized". It suggests that *q-SVM* is more robust against the misspecification of the neighborhood due to the adaptively assigned weights.

More practically, to mimic the dependence of multiple retrieved documents for the same query, we simulate the data based on $\mathbf{x}_{ij} = \epsilon_i + \epsilon_j$ in Scenario III of Example 1, where $\epsilon_i, \epsilon_j \sim N(\mathbf{0}_b, 0.1\boldsymbol{I}_b)$. As indicated in Table 3, the proposed method consistently outperforms its competitors in both independent and dependent scenarios with a similar amount of improvement.

To further examine the effect of the construction of query features, we set $\epsilon = 0.1, 0.3, \cdots, 0.9$ in Scenario III with $\gamma = 0.0$, and the performance of *q-SVM* and *kNN-SVM* are summarized in Figure 1. Clearly, the accuracies of both *q-SVM* and *kNN-SVM* decrease as ϵ increases, and *q-SVM* consistently outperforms *kNN-SVM* with various level of noise in the feature vector.

As for the computational cost of each method, we fix $\lambda = 10^{-8}$ in Scenario IV with different number of queries and documents, and the running times for

Table 3
The averaged $\widehat{MRE}$, 1-ERR and 1-NDCG and their standard errors (in parenthesis) of various methods in Scenario III of Example 1, with either independent or dependent retrieved documents.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
Independent					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.30(0.002)	0.30(0.001)	0.23(0.004)	0.16(0.003)
	1 - ERR	0.52(0.008)	0.52(0.007)	0.37(0.008)	0.26(0.007)
	1 - NDCG	0.47(0.007)	0.47(0.006)	0.32(0.006)	0.21(0.003)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.30(0.002)	0.34(0.002)	0.28(0.004)	0.21(0.003)
	1 - ERR	0.51(0.006)	0.54(0.008)	0.44(0.006)	0.32(0.006)
	1 - NDCG	0.48(0.005)	0.50(0.007)	0.40(0.006)	0.28(0.005)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.31(0.001)	0.38(0.001)	0.33(0.004)	0.28(0.005)
	1 - ERR	0.52(0.007)	0.58(0.006)	0.50(0.007)	0.38(0.006)
	1 - NDCG	0.49(0.006)	0.55(0.006)	0.46(0.004)	0.28(0.005)
Dependent					
$\gamma = 0.0$	$\widehat{\text{MRE}}$	0.38(0.002)	0.34(0.002)	0.25(0.003)	0.15(0.003)
	1 - ERR	0.66(0.006)	0.62(0.006)	0.41(0.007)	0.22(0.006)
	1 - NDCG	0.62(0.006)	0.58(0.005)	0.36(0.005)	0.18(0.004)
$\gamma = 0.1$	$\widehat{\text{MRE}}$	0.38(0.003)	0.37(0.002)	0.28(0.003)	0.17(0.003)
	1 - ERR	0.64(0.005)	0.65(0.005)	0.44(0.007)	0.23(0.009)
	1 - NDCG	0.60(0.005)	0.62(0.002)	0.39(0.006)	0.20(0.007)
$\gamma = 0.2$	$\widehat{\text{MRE}}$	0.42(0.004)	0.41(0.001)	0.34(0.004)	0.26(0.004)
	1 - ERR	0.68(0.007)	0.70(0.006)	0.51(0.008)	0.35(0.008)
	1 - NDCG	0.65(0.006)	0.67(0.006)	0.47(0.007)	0.31(0.006)

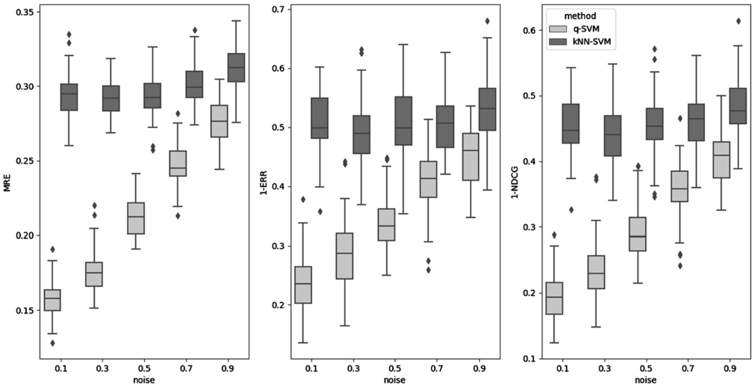

Fig 1. *The side-by-side box-plot for $\widehat{MRE}$, 1 - ERR and 1 - NDCG of q-SVM and kNN-SVM in Scenario III of Example 1 with various noise levels in constructing the query features.*

Table 4
The running time of various methods in scenario IV with different number of queries and documents.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
$n = 50$	$N = 20$	0.134	0.146	0.458	0.432
	$N = 50$	1.013	0.579	1.030	0.989
	$N = 100$	7.656	1.316	1.657	1.620
$n = 200$	$N = 20$	0.709	0.233	1.377	1.334
	$N = 50$	16.863	2.115	3.558	3.829
	$N = 100$	96.857	5.270	5.544	5.553
$n = 500$	$N = 20$	5.027	0.344	3.309	2.892
	$N = 50$	76.486	5.166	9.200	8.137
	$N = 100$	400.585	14.772	21.670	21.628

Table 5
Comparison of the ranking function and the scoring function in Scenario II of both examples with $\gamma = 0.0$.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
Example 1					
Scoring	$\widehat{\text{MRE}}$	0.27(0.001)	0.30(0.002)	0.09(0.002)	0.09(0.002)
	1 - ERR	0.45(0.007)	0.52(0.005)	0.12(0.005)	0.12(0.005)
	1 - NDCG	0.40(0.005)	0.47(0.004)	0.09(0.001)	0.09(0.003)
Ranking	$\widehat{\text{MRE}}$	0.28(0.002)	0.34(0.002)	0.18(0.003)	0.18(0.003)
	1 - ERR	0.44(0.007)	0.54(0.006)	0.27(0.008)	0.28(0.007)
	1 - NDCG	0.38(0.006)	0.49(0.006)	0.23(0.006)	0.23(0.006)
Example 2					
Scoring	$\widehat{\text{MRE}}$	0.27(0.003)	0.22(0.006)	0.13(0.007)	0.12(0.006)
	1 - ERR	0.41(0.010)	0.32(0.012)	0.18(0.007)	0.17(0.014)
	1 - NDCG	0.34(0.009)	0.26(0.010)	0.15(0.004)	0.13(0.011)
Ranking	$\widehat{\text{MRE}}$	0.28(0.004)	0.24(0.005)	0.13(0.006)	0.13(0.006)
	1 - ERR	0.41(0.010)	0.37(0.007)	0.21(0.008)	0.19(0.011)
	1 - NDCG	0.35(0.008)	0.30(0.006)	0.15(0.007)	0.15(0.009)

each method are summarized in Table 4. All experiments are conducted on a PC with 8-core Intel Xeon CPU with 32GB RAM. As expected, the running time of *rank-SVM* grows fast both in n and N, whereas *kNN-SVM* and *q-SVM* are comparable with *indv-SVM* due to the fact that both methods are trained through parallel computing.

Table 5 further compares the performance of the ranking and scoring functions as discussed in Section 3.2. It suggests that the scoring function yields similar ranking performance than the ranking function in both examples, even though the theoretical properties of the scoring function remains unclear. Due to the significant reduction of computational cost by using the scoring function, it is recommended to be used in large-scale real applications.

5.2. Application to the Yahoo! challenge

Yahoo! Labs led a learning to rank challenge in 2010 based on a real dataset that is used for training the Yahoo! search engine. The dataset contains 19,944 queries which are randomly selected from the query logs of the search engine, and 473,134 documents retrieved from different external search engines and various internal ranking functions. The average number of documents per query is about 24, and some queries have less than 5 or more than 100 documents. Note that the provided dataset is already processed, and the query-document pairs are represented by 591 numerical features. These features are provided regarding each query-document pair, where the largest category of features are the textual relationship between the document and query, including counts of occurrences in the document, average of the query terms, and BM25. Other categories of features include document statistics, document classifier, topical matching, click and external references. Furthermore, the relevance score ranges from 0 (`irrelevant`) to 4 (`perfectly relevant`), which was judged by professional editors following some specific guidelines [9].

For illustration, we randomly select 150 queries and about 4,000 documents, where each query has at least 20 documents. In text mining, a common practice

Table 6

The averaged $\widehat{MRE}$, 1-ERR, 1-NDCG and their standard errors (in parenthesis), and running time of various methods in the Yahoo! challenge dataset.

		rank-SVM	*indv-SVM*	*kNN-SVM*	*q-SVM*
Overall	$\widehat{\text{MRE}}$	0.376(0.003)	0.356(0.004)	0.328(0.005)	0.322(0.004)
	1 - ERR	0.669(0.003)	0.662(0.003)	0.644(0.003)	0.644(0.003)
	1 - NDCG	0.284(0.003)	0.266(0.003)	0.243(0.003)	0.239(0.003)
Observed	$\widehat{\text{MRE}}$	0.375(0.005)	0.341(0.004)	0.323(0.005)	0.318(0.005)
	1 - ERR	0.667(0.004)	0.654(0.003)	0.640(0.004)	0.640(0.004)
	1 - NDCG	0.286(0.003)	0.257(0.003)	0.241(0.003)	0.237(0.003)
Novel	$\widehat{\text{MRE}}$	0.391(0.010)	0.484(0.008)	0.369(0.011)	0.360(0.011)
	1 - ERR	0.680(0.008)	0.731(0.007)	0.675(0.008)	0.680(0.007)
	1 - NDCG	0.269(0.006)	0.343(0.008)	0.255(0.007)	0.253(0.006)
Time (sec)		276.818	35.417	183.911	199.057

is to map the textural queries and documents into a high-dimensional numerical space through the Word2vec model [5]. However, the Yahoo! dataset has been pre-processed, and the raw queries and documents are unavailable to public. Among the 591 processed numerical features, 6 of them remain constant for all the associated documents for each query, which are used as the query features in the analysis. The remaining 586 features then serve as the document features. The similarity between queries is computed by univariate Gaussian kernel, and its bandwidth h is set as the median of the pairwise Euclidean distances within the observed queries. For each query, its associated documents are randomly split into the training, validation and testing sets with the ratio 60%: 20%: 20%, and we also reserve $\gamma = 0.1$ proportion of queries and all of their associated documents into the testing set to mimic the novel queries.

We compare the proposed *q-SVM* with *rank-SVM*, *indv-SVM*, and *kNN-SVM*. For all SVM-based methods, a grid search is conducted on the validation set to find the optimal tuning parameters, where the grid for all methods is set as $\{10^{(\nu-31)/10}; \nu = 1, \cdots, 61\}$. For *q-SVM*, we truncate the weights to include only the top 3 queries for the observed queries, but the top 20 queries for novel queries. For *kNN-SVM*, $k = 3$ is set for observed queries and $k = 20$ is set for novel queries. The experiment is replicated 50 times, and the averaged performance measures and the running times are summarized in Table 6.

As suggested in Table 6, *q-SVM* yields better ranking performance than *rank-SVM*, *indv-SVM* and *kNN-SVM*, with 14.4%, 9.6% and 1.8% improvement on $\widehat{\text{MRE}}$, respectively. To scrutinize their performance, we also report their ranking accuracies on the observed and novel queries separately. It is interesting to note that *indv-SVM* yields accurate ranking for the observed queries, but fails to provide reasonable ranking for the novel queries. The accuracies of *rank-SVM*, *kNN-SVM* and *q-SVM* are deteriorated from the observed queries to the novel queries, but *q-SVM* and *kNN-SVM* consistently outperform *rank-SVM* in terms of $\widehat{\text{MRE}}$. The superior performance of *q-SVM* on both of the observed and novel queries indicates that the query-dependent method with kernel weighting provides an effective way to tackle the ranking problem.

6. Summary

This paper develops a general query-dependent ranking formulation, which admits different ranking functions for different queries and incorporates neighborhood structure among queries. The neighborhood structure not only helps to improve the ranking performance, but also enables the formulation to produce rankings for novel queries that are absent from the training set. The resultant optimization task is implemented via a scalable inexact ADMM algorithm. The asymptotic properties of the query-dependent ranking formulation are established to support its advantage against the existing competitors. Although the proposed formulation is formulated as a pairwise ranking method, it can be extended to the pointwise and listwise methods as well.

Acknowledgment

This research is supported in part by HK RGC Grants GRF-11302615, GRF-11331016 and GRF-11303918. The authors are grateful to the Associate Editor and reviewers for their insightful comments and suggestions which have improved the manuscript significantly.

Appendix: technical proofs

Proof of Lemma 1. Since $\mathrm{MRE}(f_{q_0}) = P\big((Y-Y')f_{q_0}(D,D') \leq 0 \big| Q = q_0\big)$, we have

$$f^*_{q_0}(D,D') = \operatorname*{argmin}_{f_{q_0}} \mathbb{E}_{D,D'}\mathbb{E}_{Y,Y'}(I(\mathrm{sign}(Y-Y')f_{q_0}(D,D') < 0)|D,D',Q=q_0),$$

where $I(\cdot)$ is an indicator function. It then suffices to consider the point-wise minimization for each pair (d,d'). Note that

$$\begin{aligned}&\mathbb{E}_{Y,Y'}\big(I(\mathrm{sign}(Y-Y')f_{q_0}(d,d') < 0)\big|D=d,D'=d',Q=q_0\big)\\ &\qquad = \Phi(d,d',q_0)I(f_{q_0}(d,d')<0) + (1-\Phi(d,d',q_0))I(f_{q_0}(d,d')>0),\end{aligned}$$

where $\Phi(d,d',q_0) = P(Y \geq Y'|D=d,D'=d',Q=q_0)$. Therefore, as a minimizer of $\mathrm{MRE}(f_{q_0})$, $f^*_{q_0}(d,d')$ must satisfy that $\mathrm{sign}(f^*_{q_0}(d,d')) = \mathrm{sign}(\Phi(d,d',q_0) - 1/2)$. This completes the proof of Lemma 1. □

Proof of Lemma 2. The proof mainly bases on the Taylor expression of $l^\pi(f)$. To this end, we first establish the upper bound for $l(f,q)$, $\nabla_q l(f,q)$ and $\nabla^2_q l(f,q)$, where $l(f,q) = \mathbb{E}\big(L(\mathrm{sign}(Y-Y')f(D,D'))|Q=q\big)$. For simplicity, we let $f^*_{q_0}(d,d') = \mathrm{sign}(P(Y \geq Y'|Q=q_0,D=d,D'=d') - 1/2)$. Note that the value of $f^\pi_{q_0}$ must be in $[-1,1]$, otherwise a truncation of $f^\pi_{q_0}$,

$$f^t_{q_0}(d,d') = \begin{cases} \mathrm{sign}\left(f^\pi_{q_0}(d,d')\right), & \text{if } |f^\pi_{q_0}(d,d')| \geq 1,\\ f^\pi_{q_0}(d,d'), & \text{if } |f^\pi_{q_0}(d,d')| < 1,\end{cases}$$

provides a smaller value of $l^\pi(f_{q_0})$, which contradicts with the fact that $f^\pi_{q_0}$ minimizes $l^\pi(f_{q_0})$. In this light, we can conclude that both $f^\pi_{q_0}$ and $f^*_{q_0}$ are contained in $\mathcal{F} = \{f : \sup_{d,d'} |f(d,d')| \leq 1\}$.

Then we have $\sup_{f\in\mathcal{F}} L(\text{sign}(Y-Y')f(D,D')) \leq 1+\sup_{f\in\mathcal{F}} \sup_{d,d'} |f(d,d')| \leq 2$, which implies that $\sup_{f\in\mathcal{F},q\in\mathcal{Q}_\eta} l(f,q)$, $\sup_{f\in\mathcal{F},q\in\mathcal{Q}_\eta} \|\nabla_q l(f,q)\|_\infty$ as well as $\sup_{f\in\mathcal{F},q\in\mathcal{Q}_\eta} \|\nabla^2_q l(f,q)\|_{\max}$ are all bounded based on Assumption A, where $\|\cdot\|_\infty$ and $\|\cdot\|_{\max}$ denote the vector infinity norm and matrix max norm respectively. It further concludes that $T(q) = l(f,q)\phi_Q(q)$ and its first and second derivatives are all bounded for any $f \in \mathcal{F}$, and $q \in \mathcal{Q}_\eta$.

Using similar technique for multivariate kernel density estimation [16], it follows from Assumption B that there exists a constant c_0 such that

$$\begin{aligned} l^\pi(f) &= \mathbb{E}_Q\big(\pi(Q,q_0)l(f,Q)\big) = \int_{\mathcal{R}^p} \pi(q,q_0)T(q)dq = \int_{\mathcal{R}^p} \mathcal{W}(u)T(q_0+hu)du \\ &= \int_{\mathcal{R}^p} \mathcal{W}(u)\big(T(q_0) + u^T\nabla T(q_0) + c_0h^2\big)du = T(q_0) + O(h^2), \end{aligned} \tag{6.1}$$

where $\nabla T(q_0)$ is the gradient of T evaluated at $q = q_0$. Therefore, for any given η, we have

$$\begin{aligned} &\sup_{q_0\in\mathcal{Q}_\eta} \Big(\text{MRE}(f^\pi_{q_0}) - \text{MRE}(f^*_{q_0})\Big) \leq \sup_{q_0\in\mathcal{Q}_\eta} \Big(l(f^\pi_{q_0},q_0) - l(f^*_{q_0},q_0)\Big) \\ &= \sup_{q_0\in\mathcal{Q}_\eta} \bigg(\Big(l(f^\pi_{q_0},q_0) - \frac{l^\pi(f^\pi_{q_0})}{\phi_Q(q_0)}\Big) - \Big(l(f^*_{q_0},q_0) - \frac{l^\pi(f^*_{q_0})}{\phi_Q(q_0)}\Big) \\ &\qquad\qquad + \frac{1}{\phi_Q(q_0)}\big(l^\pi(f^\pi_{q_0}) - l^\pi(f^*_{q_0})\big)\bigg) \\ &\leq \sup_{f\in\mathcal{F},q_0\in\mathcal{Q}_\eta} 2\Big|l(f,q_0) - \frac{l^\pi(f)}{\phi_Q(q_0)}\Big| + \sup_{q_0\in\mathcal{Q}_\eta} \frac{1}{\phi_Q(q_0)}\big(l^\pi(f^\pi_{q_0}) - l^\pi(f^*_{q_0})\big) \\ &\leq \sup_{f\in\mathcal{F},q_0\in\mathcal{Q}_\eta} 2\Big|l(f,q_0) - \frac{l(f,q_0)\phi_Q(q_0)+c_0h^2}{\phi_Q(q_0)}\Big| \leq \sup_{q_0\in\mathcal{Q}_\eta} 2\Big|\frac{c_0h^2}{\phi_Q(q_0)}\Big| \leq 2\eta^{-1}c_0h^2, \end{aligned} \tag{6.2}$$

where the third inequality follows from (6.1) and the fact that $f^\pi_{q_0}$ minimizes of $l^\pi(f_{q_0})$. The desired result follows immediately after (6.2). □

Proof of Theorem 1. We first introduce some notations. Let $L(f_{q_0}, z_{ij}, z_{il}) = L\big(\text{sign}(y_{ij} - y_{il})f_{q_0}(d_{ij}, d_{il})\big)$, $z_{ij} = (d_{ij}, y_{ij})$, $z_{il} = (d_{il}, y_{il})$, and $l^\pi_n(f_{q_0}) = \frac{1}{n}\sum_{i=1}^n \frac{\pi_i}{N(N-1)} \sum_{j\neq l} L(f_{q_0}, z_{ij}, z_{il})$. Since $\hat{f}_{q_0}$ minimizes $l^\pi_n(f_{q_0}) + \lambda_n\|f_{q_0}\|^2_{\mathcal{H}_K}$, we have $l^\pi_n(\hat{f}_{q_0}) + \lambda_n\|\hat{f}_{q_0}\|^2_{\mathcal{H}_K} \leq l^\pi_n(0) = \bar{\pi}$, where $\bar{\pi} = n^{-1}\sum_{i=1}^n \pi_i$. Furthermore, by the Law of Large Number and similar treatment in (6.2), there exists a constant c_1 such that $\bar{\pi} \leq 2\mathbb{E}\big(\pi(Q,q_0)\big) \leq 2\phi_Q(q_0) + O(h_n^2) \leq c_1^2$ with probability tending to 1 when h_n is sufficiently small and n is sufficiently large. Hence, $\|\hat{f}_{q_0}\|_{\mathcal{H}_K} \leq c_1\lambda_n^{-1/2}$ and $\|f^*_{q_0}\|_{\mathcal{H}_K} \leq c_1\lambda_n^{-1/2}$ when λ_n is sufficiently small. Therefore, both $f^*_{q_0}$ and $\hat{f}_{q_0}$ belong to $\mathcal{H}_K(\lambda_n) = \{f \in \mathcal{H}_K : \|f\|_{\mathcal{H}_K} \leq c_1\lambda_n^{-1/2}\}$.

We now are ready to bound $\mathrm{MRE}(\hat{f}_{q_0}) - \mathrm{MRE}(f^*_{q_0})$,

$$\begin{aligned}
&\sup_{q_0 \in \mathcal{Q}_\eta} \big(\mathrm{MRE}(\hat{f}_{q_0}) - \mathrm{MRE}(f^*_{q_0})\big) \leq \sup_{q_0 \in \mathcal{Q}_\eta} \big(l(\hat{f}_{q_0}, q_0) - l(f^*_{q_0}, q_0)\big) \\
&= \sup_{q_0 \in \mathcal{Q}_\eta} \Big(\big(l(\hat{f}_{q_0}, q_0) - \frac{l^\pi(\hat{f}_{q_0})}{\phi_Q(q_0)}\big) - \big(l(f^*_{q_0}, q_0) - \frac{l^\pi(f^*_{q_0})}{\phi_Q(q_0)}\big) \\
&\qquad\qquad + \frac{1}{\phi_Q(q_0)}\big(l^\pi(\hat{f}_{q_0}) - l^\pi(f^*_{q_0})\big)\Big) \\
&\leq \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} 2\Big|l(f, q_0) - \frac{l^\pi(f)}{\phi_Q(q_0)}\Big| + \sup_{q_0 \in \mathcal{Q}_\eta} \frac{1}{\phi_Q(q_0)}\big(l^\pi(\hat{f}_{q_0}) - l^\pi(f^*_{q_0})\big) \\
&\leq \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} 2\Big|l(f, q_0) - \frac{l^\pi(f)}{\phi_Q(q_0)}\Big| + \sup_{q_0 \in \mathcal{Q}_\eta} \frac{1}{\phi_Q(q_0)}\Big(l^\pi_n(\hat{f}_{q_0}) - l^\pi_n(f^*_{q_0})\Big) \\
&\quad + \sup_{q_0 \in \mathcal{Q}_\eta} \frac{1}{\phi_Q(q_0)}\big(l^\pi(\hat{f}_{q_0}) - l^\pi_n(\hat{f}_{q_0}) - l^\pi(f^*_{q_0}) + l^\pi_n(f^*_{q_0})\big) \\
&\leq \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} 2\Big|l(f, q_0) - \frac{l^\pi(f)}{\phi_Q(q_0)}\Big| + \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} \frac{2}{\phi_Q(q_0)}|l^\pi_n(f) - l^\pi(f)| \\
&\quad + \sup_{q_0 \in \mathcal{Q}_\eta} \frac{1}{\phi_Q(q_0)}\big(\lambda_n\|f^*_{q_0}\|^2_{\mathcal{H}_K} - \lambda_n\|\hat{f}_{q_0}\|^2_{\mathcal{H}_K}\big) \\
&\leq \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} 2\Big|l(f, q_0) - \frac{l^\pi(f)}{\phi_Q(q_0)}\Big| \\
&\quad + \sup_{f \in \mathcal{H}_K(\lambda_n), q_0 \in \mathcal{Q}_\eta} 2\eta^{-1}|l^\pi_n(f) - l^\pi(f)| + \eta^{-1}\lambda_n\|f^*_{q_0}\|^2_{\mathcal{H}_K} \\
&= M_1 + M_2 + \eta^{-1}\lambda_n\|f^*_{q_0}\|^2_{\mathcal{H}_K},
\end{aligned} \tag{6.3}$$

where the third last inequality follows from the fact that $\hat{f}_{q_0}$ is minimizer for $l^\pi_n(f_{q_0}) + \lambda_n\|f_{q_0}\|^2_{\mathcal{H}_K}$. Therefore, to bound $\mathrm{MRE}(\hat{f}_{q_0}) - \mathrm{MRE}(f^*_{q_0})$, it suffices to bound M_1 and M_2, separately.

Step 1: Similar as in proof of Lemma 2, we need to bound

$$\sup_{f \in \mathcal{H}_K(\lambda_n), q \in \mathcal{Q}_\eta} l(f, q), \qquad \sup_{f \in \mathcal{H}_K(\lambda_n), q \in \mathcal{Q}_\eta} \|\nabla_q l(f, q)\|_\infty \qquad \text{and}$$
$$\sup_{f \in \mathcal{H}_K(\lambda_n), q \in \mathcal{Q}_\eta} \|\nabla^2_q l(f, q)\|_{\max}.$$

Specifically,

$$\begin{aligned}
\sup_{f \in \mathcal{H}_K(\lambda_n), q \in \mathcal{Q}_\eta} l(f, q) &= \sup_{f \in \mathcal{H}_K(\lambda_n), q \in \mathcal{Q}_\eta} \int L(f, z, z')\phi(z, z'|q)dzdz' \\
&\leq \sup_{f \in \mathcal{H}_K(\lambda_n)} (1 + \sup_{d, d'}|f(d, d')|) \leq 1 + c_2\lambda_n^{-1/2},
\end{aligned}$$

where $\phi(z, z'|q)$ is the conditional density of (Z, Z') given $Q = q$, the last inequality follows from the fact that there exists a constant c_2 such that

$\sup_{d,d'} |f(d,d')| \leq c_2\lambda_n^{-1/2}$ [26], if $\|f\|_{\mathcal{H}_K(\lambda_n),} \leq c_1\lambda_n^{-1/2}$. Next, for each component $q^{(j)}, j = 1, \cdots, p$, there exists a constant c_3 such that

$$\begin{aligned}\sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} \frac{\partial l(f,q)}{\partial q^{(j)}} &= \sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} \int L(f,z,z')\frac{\partial\phi(z,z'|q)}{\partial q^{(j)}}dzdz' \\ &\leq c_3 \sup_{f\in\mathcal{H}_K(\lambda_n)} (1+\sup_{d,d'}|f(d,d')|) \leq c_3(1+c_2\lambda_n^{-1/2}),\end{aligned}$$

where the second inequality follows from Assumption A. Similarly, there exists a constant c_4 such that $\sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} \|\nabla_q^2 l(f,q)\|_{\max} \leq c_4(1+c_2\lambda_n^{-1/2})$. Therefore, for a constant c_5, we have

$$\begin{aligned}\max\Big\{ \sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} l(f,q), \sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} \|\nabla_q l(f,q)\|_\infty, \\ \sup_{f\in\mathcal{H}_K(\lambda_n),q\in\mathcal{Q}_\eta} \|\nabla_q^2 l(f,q)\|_{\max}\Big\} \leq c_5\lambda_n^{-1/2}.\end{aligned}$$

Hence, by mimicking (6.2), there exists a constant c_6 such that

$$M_1 \leq \sup_{f\in\mathcal{H}_K(\lambda_n),q_0\in\mathcal{Q}_\eta} 2\Big| l(f,q_0) - \frac{l(f,q_0)\phi_Q(q_0)+c_5\lambda_n^{-1/2}h_n^2}{\phi_Q(q_0)}\Big| \leq c_6\lambda_n^{-1/2}h_n^2\eta^{-1}, \tag{6.4}$$

suggesting that the bias due to kernel smoothing can be controlled by h_n and λ_n.

Step 2: Denote $\mathbb{G}_n = \sqrt{n}\big(\frac{1}{n}\sum_{i=1}^n g(q_i, z_{i1}, \cdots, z_{iN}) - \mathbb{E}\big(g(Q, Z_1, \cdots, Z_N)\big)\big)$. For any $r_n > 0$, we have

$$\begin{aligned}\mathbb{P}^*(M_2 \geq r_n) &= \mathbb{P}^*(\|\mathbb{G}_n\|_\mathcal{G} \geq \eta\sqrt{n}r_n/2) \\ &= \mathbb{P}^*\Big(\big\|\mathbb{G}_n\big\|_{\mathcal{G}/(h_n^{-(p+1)}\lambda_n^{-1/2})} \geq \eta\sqrt{n}h_n^{p+1}\lambda_n^{1/2}r_n/2\Big),\end{aligned} \tag{6.5}$$

where $\mathbb{P}^*$ is the outer probability, $\|\mathbb{G}_n\|_\mathcal{G} = \sup_{g\in\mathcal{G}}|\mathbb{G}_n|$, and the functional space $\mathcal{G}$ is defined as $\mathcal{G} = \Big\{g(Q, Z_1, \cdots, Z_N) = \frac{\pi(Q,q_0)}{N(N-1)}\sum_{j\neq l} L(f, Z_j, Z_l) : \ f \in \mathcal{H}_K(\lambda_n), \ q_0 \in \mathcal{Q}_\eta\Big\}$.

For any $q_1, q_2 \in \mathcal{Q}_\eta$ and $f_1, f_2 \in \mathcal{H}_K(\lambda_n)$, there exists a constant c_7 depending on the Lipschitz constant for $\mathcal{W}(\cdot)$ such that

$$\begin{aligned}\|g_1 - g_2\|_\infty &= \Big\|\frac{\pi(Q,q_1)}{N(N-1)}\sum_{j\neq l} L(f_1, Z_j, Z_l) - \frac{\pi(Q,q_2)}{N(N-1)}\sum_{j\neq l} L(f_2, Z_j, Z_l)\Big\|_\infty \\ &\leq \Big\|\frac{\pi(Q,q_1)}{N(N-1)}\sum_{j\neq l}\big(L(f_1, Z_j, Z_l) - L(f_2, Z_j, Z_l)\big)\Big\|_\infty \\ &\qquad + \Big\|\frac{\pi(Q,q_1)-\pi(Q,q_2)}{N(N-1)}\sum_{j\neq l} L(f_2, Z_j, Z_l)\Big\|_\infty\end{aligned}$$

$$
\begin{aligned}
&\le h_n^{-p}\|f_1 - f_2\|_\infty + c_5\lambda_n^{-1/2}\|\pi(Q,q_1) - \pi(Q,q_2)\|_\infty \\
&\le h_n^{-p}\|f_1 - f_2\|_\infty + c_7\lambda_n^{-1/2}h_n^{-(p+1)}\|q_1 - q_2\|_\infty, \qquad (6.6)
\end{aligned}
$$

where the last inequality follows from the Lipschitz continuity of the kernel function, and the second last inequality follows from the Lipschitz continuity of hinge loss.

It then suffices to consider the covering number of $\mathcal{H}_K(\lambda_n)$. By using Example 4 in [43], there exists a constant c_8 depending only on b such that

$$\log N\big(\epsilon, \mathcal{H}_K(\lambda_n)/(c_1\lambda_n^{-1/2}), \|\cdot\|_\infty\big) \le c_8\big(\log(1/\epsilon)\big)^{2b+1}.$$

For any given η, $\mathcal{Q}_\eta$ is a bounded set by Assumption A. Together with (6.6), for a constant c_9 depending on η, the covering number of $\mathcal{G}$ is,

$$\log N(\epsilon,\mathcal{G}/(h_n^{-(p+1)}\lambda_n^{-1/2}), \|\cdot\|_\infty) \le c_8\big(\log(1/\epsilon)\big)^{2b+1} + c_9\log(1/\epsilon).$$

An application of Theorem 2.14.10 in [33] with a constant c_{10} depending on b, yields that

$$
\begin{aligned}
\mathbb{P}^*(M_2 \ge r_n) &= \mathbb{P}^*\Big(\big\|\mathbb{G}_n\big\|_{\mathcal{G}/(h_n^{-(p+1)}\lambda_n^{-1/2})} \ge \eta\sqrt{n}h_n^{p+1}\lambda_n^{1/2}r_n/2\Big) \\
&\le c_{10}\exp(-n\eta^2 h_n^{2(p+1)}\lambda_n r_n^2). \qquad (6.7)
\end{aligned}
$$

We are now ready to derive the probability bounds for (6.3). Together with (6.4), and letting $r_n = t_n - c_6\eta^{-1}\lambda_n^{-1/2}h_n^2 - \eta^{-1}\lambda_n\|f_{q_0}^*\|_{\mathcal{H}_K}^2$ in (6.7), for any $t_n > 0$, we have

$$
\begin{aligned}
\mathbb{P}\Big(\sup_{q_0\in\mathcal{Q}_\eta}\big(\mathrm{MRE}(\hat{f_{q_0}}) - \mathrm{MRE}(f_{q_0}^*)\big) \ge t_n\Big) &\le \mathbb{P}(M_1 + M_2 + \lambda_n\eta^{-1}\|f_{q_0}^*\|_{\mathcal{H}_K}^2 \ge t_n) \\
&\le \mathbb{P}^*(M_2 \ge r_n) \le c_{10}\exp(-n\eta^2 h_n^{2(p+1)}\lambda_n r_n^2).
\end{aligned}
$$

The desired result then follows immediately. □

References

[1] Shivani Agarwal. Learning to rank on graphs. *Machine Learning*, 81:333–357, 2010. MR3108184

[2] Mayer Alvo and Philip Yu. *Statistical methods for ranking data*. Springer, 2014. MR3308929

[3] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. MR2268032

[4] Luis Antonio Belanche Muñoz and Marco Villegas. Kernel functions for categorical variables with application to problems in the life sciences. In *Artificial intelligence research and development: proceedings of the 16 International Conference of the Catalan Association of Artificial Intelligence*, pages 171–180, 2013.

[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[6] Jiang Bian, Tie-Yan Liu, Tao Qin, and Hongyuan Zha. Ranking with query-dependent loss for web search. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 141–150. ACM, 2010.

[7] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[8] Clément Calauzenes, Nicolas Usunier, and Patrick Gallinari. On the (non-) existence of convex, calibrated surrogate losses for ranking. In *Advances in Neural Information Processing Systems*, pages 197–205, 2012.

[9] Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. In *Yahoo! Learning to Rank Challenge*, pages 1–24, 2011.

[10] Tianle Chen, Yuanjia Wang, Huaihou Chen, Karen Marder, and Donglin Zeng. Targeted local support vector machine for age-dependent classification. *Journal of the American Statistical Association*, 109:1174–1187, 2014. MR3265689

[11] Stéphan Clémençon, Gabor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of U-statistics. *Annals of Statistics*, 36:844–874, 2008. MR2396817

[12] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94. ACM, 2008.

[13] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability.* CRC Press, 1996. MR1383587

[14] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum. Query dependent ranking using k-nearest neighbor. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–122. ACM, 2008.

[15] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[16] Wolfgang Härdle. *Smoothing techniques: with implementation in S.* Springer Science & Business Media, 2012. MR1140190

[17] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT, 2000.

[18] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446, 2002.

[19] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30:81–

93, 1938. MR0138175

[20] Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet comparisons. In *Advances in Neural Information Processing Systems*, pages 6807–6817, 2017.

[21] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, pages 571–580. ACM, 2010.

[22] Hang Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 7:1–121, 2014.

[23] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. Learning to model relatedness for news recommendation. In *Proceedings of the 20th International Conference on World Wide Web*, pages 57–66. ACM, 2011.

[24] Horia Mania, Aaditya Ramdas, Martin J Wainwright, Michael I Jordan, Benjamin Recht, et al. On kernel methods for covariates that are rankings. *Electronic Journal of Statistics*, 12(2):2537–2577, 2018. MR3843387

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] Sayan Mukherjee and Ding-Xuan Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7:519–549, 2006. MR2274377

[27] Deborah Nolan and David Pollard. U-processes: rates of convergence. *Annals of Statistics*, 15(2):780–799, 1987. MR0888439

[28] Wojciech Rejchel. On ranking and generalization bounds. *Journal of Machine Learning Research*, 13:1373–1392, 2012. MR2930642

[29] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems*, pages 937–944, 2002.

[30] Xiaotong Shen, George C Tseng, Xuegong Zhang, and Wing Hung Wong. On ψ-learning. *Journal of the American Statistical Association*, 98:724–734, 2003. MR2011686

[31] Yoshikazu Terada and Ulrike Luxburg. Local ordinal embedding. In *International Conference on Machine Learning*, pages 847–855, 2014.

[32] Kazuki Uematsu and Yoonkyung Lee. On theoretically optimal ranking functions in bipartite ranking. *Journal of the American Statistical Association*, 112(519):1311–1322, 2017. MR3735379

[33] Aad W Van Der Vaart and Jon A Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996. MR1385671

[34] Grace Wahba. *Spline models for observational data*. Siam press, 1990. MR1045442

[35] Grace Wahba et al. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87, 1999.

[36] Huahua Wang and Arindam Banerjee. Bregman alternating direction

method of multipliers. In *Advances in Neural Information Processing Systems*, pages 2816–2824, 2014.

[37] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of NDCG type ranking measures. In *Conference on Learning Theory*, pages 25–54, 2013.

[38] Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006. MR2172729

[39] Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007. MR2411659

[40] Yichao Wu and Yufeng Liu. Adaptively weighted large margin classifiers. *Journal of Computational and Graphical Statistics*, 22(2):416–432, 2013. MR3173722

[41] Lan Xue. Consistent variable selection in additive models. *Statistica Sinica*, 19:1281–1296, 2009. MR2536156

[42] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278. ACM, 2007.

[43] Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18:739–767, 2002. MR1928805

[44] Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14:185–205, 2012. MR2137897