Learning to Rank Using Localized Geometric Mean Metrics

Yuxin Su
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
yxsu@cse.cuhk.edu.hk

Irwin King
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
king@cse.cuhk.edu.hk

Michael Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
lyu@cse.cuhk.edu.hk

ABSTRACT

Many learning-to-rank (LtR) algorithms focus on query-independent model, in which query and document do not lie in the same feature space, and the rankers rely on the feature ensemble about query-document pair instead of the similarity between query instance and documents. However, existing algorithms do not consider local structures in query-document feature space, and are fragile to irrelevant noise features. In this paper, we propose a novel Riemannian metric learning algorithm to capture the local structures and develop a robust LtR algorithm. First, we design a concept called *ideal candidate document* to introduce metric learning algorithm to query-independent model. Previous metric learning algorithms aiming to find an optimal metric space are only suitable for query-dependent model, in which query instance and documents belong to the same feature space and the similarity is directly computed from the metric space. Then we extend the new and extremely fast global Geometric Mean Metric Learning (GMML) algorithm to develop a localized GMML, namely L-GMML. Based on the combination of local learned metrics, we employ the popular Normalized Discounted Cumulative Gain (NDCG) scorer and Weighted Approximate Rank Pairwise (WARP) loss to optimize the *ideal candidate document* for each query candidate set. Finally, we can quickly evaluate all candidates via the similarity between the *ideal candidate document* and other candidates. By leveraging the ability of metric learning algorithms to describe the complex structural information, our approach gives us a principled and efficient way to perform LtR tasks. The experiments on real-world datasets demonstrate that our proposed L-GMML algorithm outperforms the state-of-the-art metric learning to rank methods and the stylish query-independent LtR algorithms regarding accuracy and computational efficiency.

KEYWORDS

Learning to Rank, Distance Metric Learning, Local Metric Learning

ACM Reference format:
Yuxin Su, Irwin King, and Michael Lyu. 2017. Learning to Rank Using Localized Geometric Mean Metrics. In *Proceedings of SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan, ,* 10 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080828

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan
© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080828

1 INTRODUCTION

In many information retrieval systems, especially Web search, users expect to obtain the most relevant documents according to users' query phrase or document. This task is technically formulated as a ranking problem. Most of the Web search engines exploit this ranking task based on learning-to-rank (LtR) techniques [21]. In the LtR framework, a machine learning algorithm is typically employed to derive a ranking model about document collection from a training subset of documents with labels or partial order. After the supervised or semi-supervised learning procedures, the ranking model is expected to retrieval top-k (ordered) relevant documents from the candidate collection when a query is given.

In practice, search engines develop the LtR model in two stages: (i) candidate retrieval and (ii) candidate re-ranking [22]. In the first stage, search engine retrieves from the inverted document repository a sufficiently large set of relevant candidate documents $\mathcal{D}_q$ matching a user's query. It is used to avoid applying the ranking model to all documents possibly matching a user's query. This stage usually requires that the size of candidate set is much larger than the number of the relevant URLs to be included in the returned page. Based on the candidate document set $\mathcal{D}_q$ obtained in the first stage, Web search engines reformulate the documents with features extracted from the query-document pair and hide query features, then employ the LtR model without the dependency of query instance to score and re-rank the document collection $\mathcal{D}_q$. Finally, search engines return the top-k documents to the user.

In Web search engine, the time-budget of this two-stage framework is usually limited. Therefore, strongly motivated by the time budget consideration, the current two most efficient and the state-of-the-art methods are based on the additive ensemble of regression trees, namely Gradient-Boosted Regression Tree (GBRT) [11] and λ-MART [4]. These two kinds of methods are capable of meeting the time requirement with acceptable accuracy even when thousands of regression tree are evaluated for each document. However, one of the drawbacks of this line of methods is that when the input samples contain an enormous amount of non-informative features, many methods fail to identify the most relevant features. Therefore, researchers are still trying to devise techniques and strategies to find a better way of combining features extracted from query-document pairs through discriminative training to accelerate the training process for document ranking without losing in quality [12, 35].

Another perspective to the ranking problem is to seek the best similarity measurement and develop the corresponding efficient algorithm. These approach aims to optimize the accuracy in the first stage to find candidate documents or even directly return the top-k documents with an order. Concerning accuracy, the similarity-based models for a ranking problem can be classified into three categories from the formulation of the loss function: pointwise, pairwise and listwise loss functions [3]. Practically, the pairwise loss function tends to be more efficient for training and have been widely adopted by large Web search engines [3].

The pairwise similarity motivates that how to apply the classical metric learning or similar learning methods to the ranking problem [7]. The metric learning algorithms aim to find a better distance metric than Euclidean metric to measure the pairwise similarity. The advantage of such metric-learning-to-rank [24] framework has two folds: (1) the metrics often preserves the nearest neighborhood information, which is the perfect structure to conduct ranking; (2) a proper metric containing the structural information of the document collections in the document space is useful for reducing the over-fitting and improving the robustness to noise features [16]. Therefore, the metric-learning-to-rank methods [18, 19, 25] typically enjoy higher accuracy. Nevertheless, unfortunately, the disadvantage of metric-learning-to-rank also has two folds: (1) many metric learning algorithms [8, 32] are degraded by its extremely high computational expense; (2) the similarity measurement is not suitable for LtR because we can not estimate the similarity between query and documents with features extracted from other domain knowledge.

In this paper, we focus on improving the ranking accuracy at the second stage in the search engine and attempt to provide a new query-independent model for LtR task. Different from the existing research on how to combine features extracted from other domains, we try to learn an optimal representation of these features via metric learning algorithm. To adopt query-dependent metric learning framework to a query-independent model, we propose a concept called *ideal candidate document*, which represents a perfect match for a given query. With the help of this concept, we can quickly evaluate all candidate documents and sort them by calculating the distance based on the optimal metric space between the *ideal candidate document* and other documents. Same with the query-dependent model, the shorter distance leads to a higher relevance to the query.

Since features from different domains generate local structure on the whole feature space, in order to preserve more local information and avoid overfitting, we develop a novel local metric learning framework for ranking with high efficiency and accuracy. Our localized metric learning algorithm extends from the state-of-the-art global metric learning algorithm called Geometric Mean Metric Learning (GMML) [36], and we apply Weighted Approximate Rank Pairwise (WARP) loss to optimize the metric space around the ideal document from the combination of several anchor documents.

We summarize our main contributions as follows:

- To the best of our knowledge, we are the first to extend geometric mean metric learning algorithm to a local metric learning approach in order to capture the local structures for LtR problem.
- We propose a novel *ideal candidate document* concept to transform metric-learning-to-rank framework from query-dependent model to query-independent model, which brings wider applications for metric learning and also improves the accuracy of classical LtR task.
- We conduct extensive experiments to reveal that our method outperforms the state-of-the-art query-dependent metric-learning-to-rank algorithms and query-independent LtR methods both in the accuracy and the computational complexity.

2 PRELIMINARIES AND RELATED WORK

Since our approach employs local metric learning algorithm to conduct the LtR task, two sets of previous work relate to our work:

2.1 Learning to Rank

In the information retrieval setting, a search engine maintains a collection of candidate examples $\mathcal{D}$. Given a query q, the search engine returns the top ranked subset of documents $\left\{p \in \mathbb{R}^d\right\} \subset \mathcal{D}_q \subset \mathcal{D}$ from the collection with order, ranked by a specific ranking model $f_q(p)$.

According to the formulation of the loss function, the LtR methods are categories into three folds: (1) pointwise loss approach, (2) pairwise loss approach and (3) listwise loss approach.

For pointwise loss function, Li et al. [17] cast the ranking problem to a multi-class classification problem. The training process relies on enough labeled information, which is not always easy to satisfy. Pairwise loss approach such as RankNet [2], RankBoost [10] focus on the relative order, which is capable of being adapted to classification problem. In the listwise loss approach, a relevance label l related with the query for ground truth is usually bound to the document p. Cao et al. [5] first propose to find the optimal permutation to minimize the listwise loss function. McFee [25] proposes a similar objective, but the different solution from the metric learning methods.

The majority of LtR methods follows listwise loss function. Currently, the most popular methods [4, 9, 11] come from the combination of an ensemble of trees like random forest and the boosting-like methods [10]. Based on multiple decision trees, this kind of methods gains an accepted level of accuracy.

2.2 Metric Learning

The (squared) Mahalanobis distance, an extension of Euclidean distance, measures the distance between two points lie on the special linear space. It is defined as

$$d_{\mathbf{M}}\left(p_1, p_2\right) = \left(p_1 - p_2\right)^T \mathbf{M}\left(p_1 - p_2\right), \tag{1}$$

where $p_1, p_2 \in \mathbb{R}^d$ are input examples, $\mathbf{M}$ is a symmetric and positive semi-definite $d \times d$ matrix. When $\mathbf{M} = \mathbf{I}$, the Mahalanobis distance is equivalent to the Euclidean distance.

There are plenty of algorithms aiming at learning such metric by solving a semidefinite or a quadratic program [29, 32, 34]. Almost all the metric learning algorithms try to constrain the similar data points and to scatter those dissimilar data points. Early work like [34] formulates this problem as an optimization problem on the second-order cone, which is costly solvable. Davis et al. [8], Weinberger et al. [32] and Shen et al. [29] formulate different kinds of optimization problems, namely ITML, LMNN, BoostMetric respectively. However, the common issue that their solutions are computationally expensive. Very recently, Zadeh et al. [36] propose a new objective function and give the closed-form solution from the geometric domain. It is the most promising global metric learning method because of the computational speed several orders of magnitude faster than the widely used ITML and LMNN methods.

There are two different roadmaps to conduct the LtR tasks from the metric learning perspective: (1), McFee [25] and Lim et al. [18, 19] learn global metric with Positive Semi-Definite (PSD) constraint on the metric parameter. They belong to the application of the standard metric learning algorithm. (2), Chechik et al. [7] and Liu et al. [20] remove PSD constraint and employ the bilinear model to measure the similarity between two data points. Usually, without PSD constraints, the bilinear model easily leads to over-fitting. However, PSD constraint brings a tremendous amount of computation.

In most cases, global metric learning relies on a learned PSD matrix, which is not only computational expensive in high-dimensional case but not reasonable for retrieval ranking problem. In the LtR framework, the local similarity is far more important than the dissimilar information because we aim at ranking the relevant documents around a user's query. Therefore, several important local metric learning approaches are related to our work. Wang et al. [31] parameterize the weight function of each data point. The approach enhances the model complexity but brings the extra computation. Hauberg et al. [13] provide the theoretical analysis about the optimal weight function. However, the calculation of the geodesics is extremely expensive.

3 OUR PROPOSED METHODOLOGY

In the LtR problem, a ranked list of the relevant documents is returned for a specific user's query. In this situation, we can assume without losing generality that all relevant documents should be closer to an unreal document than other irrelevant documents. This unreal document should be related to the query. Therefore, although the query instance is not accessible in the document feature space, we can still construct this unreal candidate document to represent the query in the feature space of the document. In our paper, this unreal but perfect-matching document is named as the *ideal candidate document*.

Usually, the indexed documents are assumed to be static, and the set of queries considered as input testing data is dynamic. This assumption allows us to transform the training documents to an another static representation, and model *ideal candidate documents* for each query to a dynamic combination of static documents.

In our paper, we assume the documents including candidate documents and ideal documents lie on the surface of a Riemannian manifold. Then, we attempt to build the similarity measurement between documents on the geodesic lines in the Riemannian manifold. Very often, a single linear metric $\mathbf{M}$ can not describe the whole surface of Riemannian manifold adequately. It reveals the inability of a single metric to model the complexity of the LtR problem. Furthermore, the discriminative features vary between different neighborhoods on the surface of the manifold. To address this limitation, researchers try to learn a set of local metrics representing the various regions of the surface. In most cases, local metric learning algorithms will generate a local metric for each learning example [26]. The whole parameters of these kinds of the algorithm are prohibitively huge when the number of examples becomes large.

In our approach, we follow [31] to learn a local metric for a part of the feature space of documents, in which case the number of learned metrics m can be considerably smaller than n, the size of the examples collection.

Suppose we have learned m local metrics $\{\mathbf{M}_1, \ldots, \mathbf{M}_m\}$ and the associated anchor points $\{p_1, \ldots p_r, \ldots p_m\}$. The choice of anchor points and the computation of local metrics are described in Subsection 3.1. Then the similarity model $f(q, p)$ between two documents p_i and p_j is extended from Eq. (1) as follows:

$$f(p_i, p_j) = d_{\mathbf{M}(p_i)}(p_i, p_j) \tag{2}$$

$$\mathbf{M}(p_i) = \sum_{r=1}^{m} w_r(p_i)\mathbf{M}_r, \tag{3}$$

where $w_r(p_i) \geq 0$ is the weight of document p_i for local metric $\mathbf{M}_r$. The PSD constraints of $\mathbf{M}(p)$ is automatically satisfied if all local metric $\mathbf{M}_r$ are PSD matrices. These formulation includes m anchor documents and p_i should be close to these anchor documents [27]. It is clear that the *ideal candidate document* should be close to several high relevant documents. Therefore, we can employ these high relevant documents as anchor documents to construct the local metric space around the *ideal candidate document*.

With the above assumption and observation, the task of information retrieval precedes in the following steps:

(1) Given a candidate collection $\mathcal{D}_q$ for query q , we employ high/low relevant documents to compute a $\mathbf{M}$ and find a anchor point p_r to maximize the ranking scorer under the metric $\mathbf{M}$ by computing $(p_i - p_r)\,\mathbf{M}\,(p_i - p_r)^\top$.
(2) After sampling m candidate collection to find m anchor documents and m associated metrics, we can construct the *ideal candidate document* based on a combination of m anchor documents.
(3) We can build an evaluation function to measure the similarity between candidate document and *ideal candidate document*, then, sort these documents via the similarity to *ideal candidate document*.

3.1 Computation of Basis Metrics

Before constructing the local metrics in Eq. (3), we need to learn m local metrics. With the assumption that each local metric $\mathbf{M}_r$ represents a part of feature space, we can employ the classical single metric learning algorithm associated with a subset of the triplets from a part of examples space.

In this paper, we extend the state-of-the-art global metric learning algorithm GMML [36] into local metric forms. The extension consists of two parts:

(1) The local basis metric associated with the triplets set $\mathcal{D}_r$ is computed by the original GMML.
(2) The smooth weighting function $w_r(p)$ is computed from Eq. (11).

Given a subset of the triplets $\mathcal{T}_r = \left(p_i, p_j, p_k\right)$ such that p_i is more similar to p_j than to p_k, we can extract the similarity set S_r and

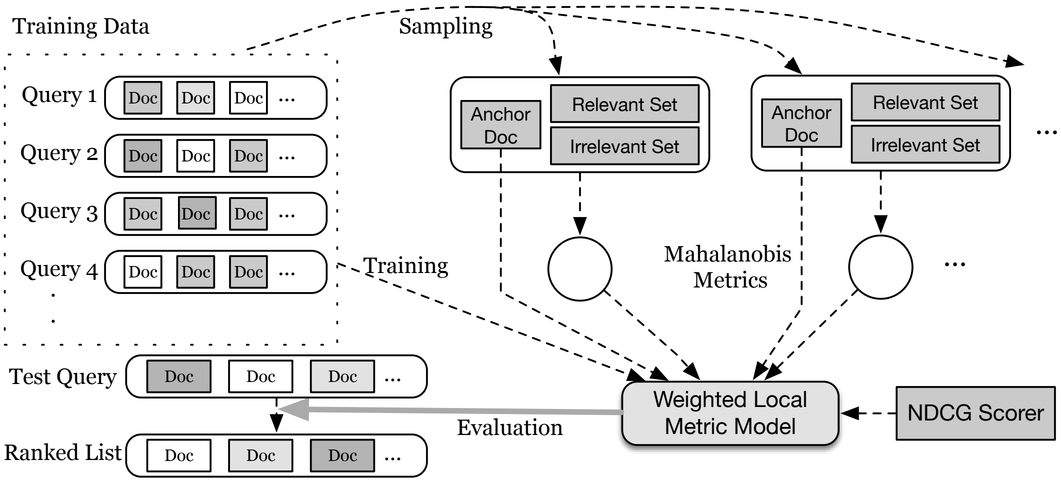

Figure 1: General framework of proposed L-GMML for ranking. Different gray levels in query test represent the relevant level of the document

the dissimilarity set D_r by following the instruction in Section 3.4. Then we construct two corresponding matrices:

$$\mathbf{S}_r = \sum_{(p_i,p_j)\in S_r} \left(p_i - p_j\right)\left(p_i - p_j\right)^\top \tag{4}$$

$$\mathbf{D}_r = \sum_{(p_i,p_k)\in D_r} \left(p_i - p_k\right)\left(p_i - p_k\right)^\top \tag{5}$$

Then, the basic optimization formulation of local metric $\mathbf{M}_r$ is defined as follows:

$$\min_{\mathbf{M}_r \succ 0} \quad h(\mathbf{M}_r) := \mathrm{tr}\left(\mathbf{M}_r\mathbf{S}_r\right) + \mathrm{tr}\left(\mathbf{M}_r^{-1}\mathbf{D}_r\right) \tag{6}$$

Equation (6) implies that GMML will return a single local metric $\mathbf{M}_r$ that minimize the sum of distances over all the similar pairs S_r and maximize the distance over all the dissimilar pairs D_r.

The closed-form solution of Eq. (6) is obtained by

$$\nabla h(\mathbf{M}_r) = \mathbf{S}_r - \mathbf{M}_r^{-1}\mathbf{D}_r\mathbf{M}_{\mathrm{r}}^{-1} \tag{7}$$

Taking $\nabla h(\mathbf{M}_r) = 0$, we obtain:

$$\mathbf{M}_r\mathbf{S}_r\mathbf{M}_r = \mathbf{D}_r \tag{8}$$

Equation (8) is a Riccati equation whose unique solution is [36]

$$\mathbf{M}_r = \mathbf{S}_r^{-1/2}\left(\mathbf{S}_r^{1/2}\mathbf{D}_r\mathbf{S}_r^{1/2}\right)^{1/2}\mathbf{S}_r^{-1/2} \tag{9}$$

In experiments, $\mathbf{M}_r$ is efficiently computed from Cholesky-Schur method [14].

3.2 Smoothing Weight Functions

Lots of researchers try to provide the insights of their local metric learning approaches [13, 31] by modeling their methods from the perspective of Riemannian metric. An important property about the Riemannian metric is that a Riemannian metric $M(p)$ on a manifold $\mathcal{M}$ is a smoothly varying inner product $\left\langle x_i, x_j \right\rangle_p = x_i^T\mathbf{M}(p)x_j$ in the tangent space $\mathcal{T}_p\mathcal{M}$ of each point $p \in \mathcal{M}$. From Lemma 1 in [13], when the weight function $w_r(p)$ is smooth with p, Eq. (3) will be a well-studied Riemannian metric. Therefore, any well-designed local metric methods should provide a smooth weight function.

Another important issue is that the weight function $w_r(p)$ should reflect the fitness of the local metric $\mathbf{M}_r$. Suppose $(p, p_r) \in S_r$, it indicates that $\mathbf{M}_r$ is the best local metric to measure the similarity between p_r and other examples, which means that Eq. (8) should be robust against the additive similar pair (p, p_r). Therefore, the weight function $w_r(p)$ should be in the opposite to $\mathbf{M}_r(p-p_r)(p-p_r)^T\mathbf{M}_r$.

Take the limit as an example, if $\mathbf{M}_r(p-p_r)(p-p_r)^T\mathbf{M}_r = \mathbf{0}$, then,

$$\mathbf{M}_r\left(\mathbf{S}_r + (p-p_r)\right)\left(\mathbf{S}_r + (p-p_r)\right)^T\mathbf{M}_r = \mathbf{D}_r \tag{10}$$

The solution of Eq. (10) is the same with Eq. (9), which indicates that $\mathbf{M}_r(p-p_r)(p-p_r)^T\mathbf{M}_r$ is a proper measurement whether the $\mathbf{M}_r$ is the optimal local metric for the document p.

By taking consideration about the above observation, we propose the smoothing weight functions [1] as:

$$w_r(p) = \exp\left(-\frac{\rho}{2}\|p-p_r\|_{\mathbf{M}_r}\right), \tag{11}$$

where, $\|\cdot\|_{\mathbf{M}_r}^2$ is the L2 norm with the metric $\mathbf{M}_r$.

$$\|p-p_r\|_{\mathbf{M}_r}^2 = \mathrm{tr}\left(\mathbf{M}_r(p-p_r)(p-p_r)^T\mathbf{M}_r\right) \tag{12}$$

From Eq. (12) and Eq. (10), we can easily know $\|p-p_r\|_{\mathbf{M}_r}$ is a proper measurement about the similarity between query p and the anchor point p_r associated with the local metric $\mathbf{M}_r$.

Therefore, our evaluation function is formulated as:

$$f_q(p, \Phi_q) = -\sum_{r=1}^{m}\Phi_q^{(r)} \cdot \exp\left(-\|p-p_r\|_{M_r}\right) \cdot \|p-p_r\|_{M_r}, \tag{13}$$

where, $\Phi_q \in \mathbb{R}^m$, $\Phi_q^{(r)} = \exp\left(\rho_q^{(r)}/2\right)$ is the key parameter we need to learn in order to find a better manifold structure. Higher $f_q(p, \Phi_q)$ means that p is closer to the *ideal candidate document.* In the next subsection, we will introduce our exploration to optimize Φ for a specific ranking problem.

3.3 Update of Φ

In the above subsection, we formulate a general local metric framework in Eq. (13) to represent the manifold structure. From the theoretical analysis in [27], the whole space of Φ keeps the learned manifold smooth. Therefore, we define our loss function under the popular Weighted Approximate Rank Pairwise (WARP) framework [33] and optimize the associated objective function to obtain an optimal solution for ranking task.

The WARP loss for a given set of candidate document $\mathcal{D}_q$ with query ID q is defined as:

$$\mathcal{L}(q) = \frac{1}{\left|\mathcal{D}_q^+\right|} \sum_{p \in \mathcal{D}_q^+} L\left(v_q\left(p^+\right)\right), \tag{14}$$

where $v_q(p^+)$ is the number of violators in $\mathcal{D}_q$ for positive p^+, defined as:

$$v_q(p^+) = \sum_{p^- \in \mathcal{D}_q^-} \mathbf{I}\left[f_q\left(p^-, \Phi_q\right) - f_q\left(p^+, \Phi_q\right)\right] \tag{15}$$

To obtain better NDCG score, $L(\cdot)$ is defined as:

$$L(k) = \sum_{i=1}^{k} \frac{1}{\log_2(i+1)} \tag{16}$$

In order to optimize Φ_q, we follows the methods in [18, 33] to approximate $L\left(v_q(p^+)\right)$ by a continuous formulation with hinge loss:

$$\sum_{p^- \in \mathcal{V}_{q,p^+}} L\left(\left|\mathcal{V}_{q,p^+}\right|\right) \frac{\left[\zeta - f_q\left(p^+, \Phi_q\right) + f_q\left(p^-, \Phi_q\right)\right]_+}{\left|\mathcal{V}_{q,p^+}\right|}, \tag{17}$$

where for a given q, p^+, ζ is the hinge loss margin. $\mathcal{V}_{q,p^+}$ is the set of violators with hinge loss:

$$\mathcal{V}_{q,p^+} = \left\{p^- \in \mathcal{X}_q^- \mid f_q(p^+, \Phi_q)\right\} \tag{18}$$

In order to obtain an unbiased estimation of the loss function in Eq. (17), we can randomly sample $q, p^+ \in \mathcal{D}_q$ and find an violator p^- such that $\zeta + f_q\left(p^-, \Phi_q\right) > f_q\left(p^+, \Phi_q\right)$. In this situation, the tuple of (q, p^+, p^-) has the following contribution to Eq. (17):

$$l\left(q, p^+, p^-\right) = L\left(\left|\mathcal{V}_{q,p^+}\right|\right)\left(\zeta - f_q\left(p^+, \Phi_q\right) + f_q\left(p^-, \Phi_q\right)\right) \tag{19}$$

From the WARP framework, $\left|\mathcal{V}_{q,p^+}\right|$ can be approximated by $\left\lfloor\left|\mathcal{D}_q^-\right| / N_q\right\rfloor$, where N_q is the number of less relevant documents p^- drawn with replacement from $\mathcal{D}_q^-$ until a violator is found.

Finally, the stochastic gradient descent for the parameter Φ can be easily conducted at iteration t as:

$$\begin{aligned}
&\Phi_q(t+1) \\
=&\Phi_q(t) - \mu \frac{\partial l\left(q, p^+, p^-\right)}{\partial \Phi_q(t)}, && (20) \\
=&\Phi_q(t) - \mu L\left(\left|\left\lfloor \frac{\left|\mathcal{D}_q^-\right|}{N_q} \right\rfloor\right|\right) \cdot \left[\frac{\partial f_q(p^-, \Phi_q(t))}{\partial \Phi_q(t)} - \frac{\partial f_q\left(p^+, \Phi_q(t)\right)}{\partial \Phi_q(t)}\right], && (21)
\end{aligned}$$

where $\frac{\partial f_q(p, \Phi_q)}{\partial \Phi_q} = \left[\frac{\partial f_q\left(p, \Phi_q^{(r)}\right)}{\partial \Phi_q^{(r)}}\right]_{r=1\ldots m}$. To avoid over-fitting, we project $\Phi_q^{(r)}$ to zero when Eq. (21) leads to negative value. We take derivation from Eq. (13) to obtain:

$$\frac{\partial f_q\left(p, \Phi_q^{(r)}\right)}{\partial \Phi_q^{(r)}} = \exp\left(-\|p - p_r\|_{M_r}\right) \cdot \|p - p_r\|_{M_r} \tag{22}$$

Overall, our proposed algorithm is illustrated in Figure 1 and summarized in Algorithm 2.

3.4 Sampling Strategy

Our approach will not iterate all triplets for $\mathcal{D}$ introduced in Section 3.1, because learning the global ranking model from all triplets is an NP-hard problem [23]. Hence, we choose to stochastically sample the triplets $\left(p_i, p_j, p_k\right)$ from candidate documents. p_i and p_j representing similar documents are sampled from high relevant documents, then p_k is sampled from the less relevant documents. In our implementation, we only sample p_k from the documents with zero relevant label.

For sampling procedure in Section 3.3, Φ_i and Φ_j are independent for two queries i and j. Therefore, the update can be computed in a highly parallel way.

ALGORITHM 1: Geometric Mean Metric Learning (GMML) [36]

```
Input: $\mathcal{D}^+$ : positive set of documents, $\mathcal{D}^-$ : negative set of documents, $\lambda$
    : regularization parameter
Output: $M \in \mathbb{S}_+^d$ : Mahalanobis metric;
$\mathbf{S} = \lambda\mathbf{I} + \sum_{p_i \neq p_j, p_i \in \mathcal{D}^+, p_j \in \mathcal{D}^+} \left(p_i - p_j\right)\left(p_i - p_j\right)^\top$;
$\mathbf{D} = \lambda\mathbf{I} + \sum_{p_i \in \mathcal{D}^+, p_j \in \mathcal{D}^-} \left(p_i - p_j\right)\left(p_i - p_j\right)^\top$;
$\mathbf{M} = \mathbf{S}^{-1/2}\left(\mathbf{S}^{1/2}\mathbf{D}\mathbf{S}^{1/2}\right)^{1/2}\mathbf{S}^{-1/2}$
```

4 EVALUATION

In this section, we discuss the implementation of our approach for the LtR problem and display extensive experiments evaluating our methodology in comparison to the state-of-the-art (R-MLR, GBRT, and λ-MART). Our design on experiments tackle the following questions:

- Do we develop a correct localized extension to the global GMML? To answer this question, we generate varied scale synthetic datasets to evaluate the performance gain against global metric learning algorithm when a different number of local metrics invoke in our L-GMML approach to prove the correctness.

ALGORITHM 2: L-GMML to Rank

```
Input: Candidate set for $c$ queries $\left\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_q, \ldots, \mathcal{D}_c\right\}$, $m$ :
       number of local metrics, $T$ : number of iteration, $\mu$ : step size, $\zeta$:
       hinge loss margin
Output: $\{(p_1, M_1), (p_2, M_2) \ldots, (p_m, M_m)\}$ : set of local metrics and
        associated anchor points, $p \in \mathbb{R}^d$, $M \in \mathbb{S}_+^d$, $\Phi \in \mathbb{R}^{c \times m}$ : weights
        for local metrics to model the *ideal candidate documents* for each
        queries
for $q \in [1, c]$ do
    Extract $\mathcal{D}_q^+$ and $\mathcal{D}_q^-$ from $\mathcal{D}_q$;
end
for $i \in [1, m]$ do
    Sample $\mathcal{D}_i^+$ and $\mathcal{D}_i^-$ from $\left\{\mathcal{D}_q\right\}_{q \in [1, c]}$;
    $M_i$ = GMML $\left(\mathcal{D}_i^+, \mathcal{D}_i^-\right)$;
    for $p \in \mathcal{D}_i^+$ do
        $\Gamma_p^{(i)}$ ←Sort $\mathcal{D}_i$ in ascending order by computing
          $\|p - d\|_{M_i}^2 \ \ \forall d \in \mathcal{D}_q$;
    end
    Find the anchor point $p_r$ with maximum NDCG score of $\Gamma_{p_r}^{(i)}$;
end
for $t = 1$ *to* $T$ do
    Sample a tuple $(q, p^+, p^-)$ from $\left\{\mathcal{D}_q\right\}_{q \in [1, c]}$ such that
    $\zeta + f_q\left(p^+, \Phi_q(t)\right) > f_q\left(p^-, \Phi_q(t)\right)$;
    $N_q$ ← the number of less relevant documents drawn with replacement
      from $\mathcal{D}_q^-$ until $p^-$ is found;
    $\Phi_q(t+1) =$
      $\left[\Phi_q(t) - \mu L\left(\left\lfloor \frac{|\mathcal{D}_q^-|}{N_q} \right\rfloor\right) \cdot \left[\frac{\partial f_q(p^-, \Phi_q(t))}{\partial \Phi_q(t)} - \frac{\partial f_q(p^+, \Phi_q(t))}{\partial \Phi_q(t)}\right]\right]_+$;
end
```

- Is our assumption on the existence of local structures reasonable? If reasonable, does our solution enjoy high computational efficiency and the good scalability for scaled datasets? We make comparisons with the state-of-the-art metric learning algorithms for ranking in the query-dependent model on scaled datasets. We attempt to demonstrate the improvements of our approach over other metric learning algorithms.
- Does our LtR algorithm have any amazing properties? We conduct experiments on real-world large-scale datasets to illustrate the enormous improvement of our approach on accuracy compared with the dominant ranking methods like GBRT and λ-MART in the query-independent framework.

4.1 Experiments Setting

In our experiments, we have implemented our local GMML (L-GMML) algorithm in Julia[1], the source code is released at Github[2]. To make a fair comparison against the state-of-the-art ranking methods, we also implement R-MLR, GBRT and λ-MART in Julia. We take RankLib[3], an open-source implementation of the GBRT and λ-MART algorithms and MLR[4] as references to implement these algorithms in Julia.

Table 1: Different kinds of song representation

	# of features	# of songs
Audio	1,024	5,419
Lyrics-128	128	2000
Lyrics-256	256	2000

Our program is executed on an Ubuntu 14.04 LTS server with 12 Intel Xeon E5-2620 cores and 128GB main memory. All baseline methods and our method are performed in a parallel way to fully utilize the computational resources. Our R-MLR implementation is based on the parallel MLR-ADMM [19]. GBRT and λ-MART come from RankLib.

The statistical tests in the following experiments are computed over the values for Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [15] at the top k retrieved documents denoted as NDCG@k. These two metrics are the most important and frequently used in information retrieval community to evaluate a given permutation of a ranked list using binary and multi-relevance order.

4.2 Datasets

For all real-world datasets, we split each of them into two components: 1), the training set is used to learn ranking models; 2), the test set is purely used to evaluate the performance of the learned ranking models.

All the datasets we use are freely available online for scientific purpose. Such datasets can be divided into two groups:

4.2.1 Query-dependent Dataset. We employ CAL10K [30] to make fair comparisons between our L-GMML and R-MLR. Because, in the original paper, R-MLR performs well on the CAL10K dataset. Following the experiments in [19], we use a subset of the CAL10K dataset, which is provided as ten 40/30/30 splits of a collection of 5419 songs.

4.2.2 Query-Independent Datasets. In this subsection, we employ two popular real-world large-scale datasets: Yahoo! and MSN to evaluate the competitive performance of proposed L-GMML against the state-of-the-art query-independent LtR methods.

Yahoo! datasets come from Yahoo! Learning to Rank Challenge [6]. The datasets consist of feature vectors extracted from query-url pairs along with relevance judgment labels.

In our experiments, we also employ the two set of MSN learning to rank[5] datasets: MSLR-10K and MSLS-30K, both of which consists of 136 features extracted from query-url pairs. The MSN datasets provide relevance judgment labels ranging from 0 (irrelevant) to 4 (perfect match). In experiments, each MSN dataset is partitioned into five subsets for five-fold cross validation.

The complete statistical information about these datasets are listed at Table 2.

[1] http://julialang.org/

[2] https://github.com/yxsu/LtR.jl

[3] http://sourceforge.net/p/lemur/wiki/RankLib/

[4] https://github.com/bmcfee/mlr

[5] https://www.microsoft.com/en-us/research/project/mslr/

Table 2: Characteristics of publicly available large-scale datasets for learning to rank

Name	# of Queries			# of Doc.			Rel. Levels	# of Features	Year
	Train	Vali.	Test	Train	Vali.	Test			
Yahoo! Set I	19,944	2,994	6,983	473,134	71,083	165,660	5	519	2010
Yahoo! Set II	1,266	1,266	3,798	34,815	34,881	103,174	5	596	2010
MSLR-WEB10K	6,000	2,000	2,000	723,412	235,259	235,259	5	136	2010
MSLR-WEB30K	31,531	6,306	6,306	3,771k	6,306	753k	5	136	2010

4.3 Evaluation of the Proposed Approach

In our L-GMML model, the most important hyper-parameter is the number of local metrics, which has significant influence on the overall model performance. We will evaluate the correction of our localized extension method from synthetic datasets, and reveal the impact of the metric numbers.

4.3.1 Global GMML vs Local GMML. In this subsection, we attempt to employ multi-class classification problem to verify the correction of the local metric learning algorithm. Because multi-class synthetic datasets certainly contain local structures around the center of each class. If the accuracy gain is observed, we can also address the objective that local metric learning approach is designed to extend the global metric learning method's ability of modeling complex data manifold.

We employ the normal distribution to generate synthetic datasets with multiple centers and 95% confidence interval. The datasets with {10,50,100} classes are denoted as Synthetic-10, Synthetic-50, Synthetic-100 respectively. In these synthetic datasets, we assign the index of class to the relevant label of the corresponding data point.

We report the performance gain of the proposed local GMML against the global GMML in Figure 2. We can easily find the fact that when the number of local metrics is approximate to the number of the real centers in Gaussian synthetic data distribution, the relative accuracy gain of local metrics is maximized. This observation meets the objective of local metric learning approach.

4.3.2 The Number of Local Metrics. In this subsection, we will evaluate the significance of the number of local metrics, which is typically the most important parameter in the field of local metric learning. A large number of local metrics will enhance the algorithm's ability to model the complex manifold structure. However the computational complexity increases linearly with the number of local metrics. In experiments, we need to carefully tune the number of local metrics to make the balance between model's ability and computational complexity.

Figure 3 displays the impact of the number of local metrics on all datasets used in our paper. For all datasets, localized method can compete with the corresponding global method with a single metric. This fact proves that our localized extension is reasonable. Another obvious observation is that the optimal number of local metrics varies dramatically among different datasets, since it is decided by the complexity of the manifold structure sealed in the data space.

4.3.3 Scalability. In our experiments, the synthetic datasets is primarily invoked to evaluate the scalability of our approach.

Due to the limited scalability of real-world datasets, we synthesize datasets with the feature dimensionality scaled from 10 to 1000. In this experiment, we fix the number of local metrics as 10 since we only concern about the computational complexity on different scaled dimensions instead of the optimal number of local metrics. Figure 4 illustrates the training time of our L-GMML on these datasets.

Compared with other local metric learning methods, the less training time come from two-fold issues: (1) the GMML in Algorithm 1 is very fast. (2) The update of weighting function in our approach is relatively simple and straightforward. It does not involve the huge computational resources to find the optimal form.

4.4 Comparison with R-MLR

The Robust Metric-Learning-to-Rank (R-MLR) [19] is the most competitive metric learning method for ranking. It retrieves relevant examples in response to a query instance. To make direct comparisons, we need to modify our approach by assigning all anchor points to the query instance. Because our approach is originally designed for the query-independent framework.

In this set of experiments, we evaluate our approach on the music similarity task, because the R-MLR method is verified to be successful in music similarity task compared with other metric-learning-to-rank methods such as MLR [25], $L1$-MLR [28]. For each song p_i, a relevant set $\mathcal{D}_i^+ \subset \mathcal{D}_{\text{train}}$ is defined as the subset of songs in the training set performed by the top 10 most similar artists to the performer of p_i, where the similarity between artists is measured by the number of shared users in a sample of collaborative filter data [24]. This top-10 thresholding results in the relevant sets in this data being asymmetric and non-transitive. Therefore, the traditional pairwise metric learning methods do not work in this situation. However, our approach based on the sampling on the relevant set is not necessary to obey the symmetric and transitive properties.

The experiments are conducted on two different kinds of song representation: audio and lyrics, whose details are listed in Table 1. We use recommended candidate hyper-parameters in the original paper to tune R-MLR on validation set and select the best parameter to evaluate the performance of the model.

Since the scalability of the original R-MLR is limited, the experiments of R-MLR employ the latent features compressed by PCA. Our approach has no such problem and is suitable to conduct the training process on the original 1,024 features.

Figure 5 illustrates the performance of three metric learning algorithms. We fix the number of local metrics in our L-GMML as 1 to obtain the global GMML algorithm. The motivation of making

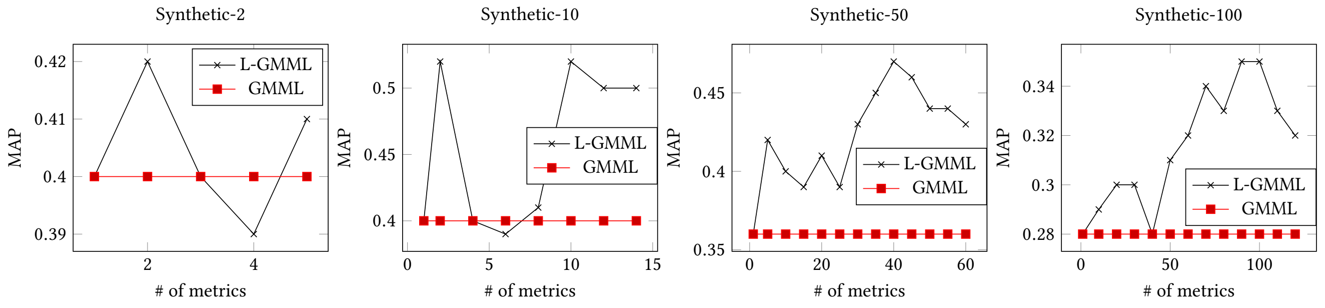

Figure 2: Comparisons between global GMML and local GMML on synthetic datasets. The performance is measured by MAP

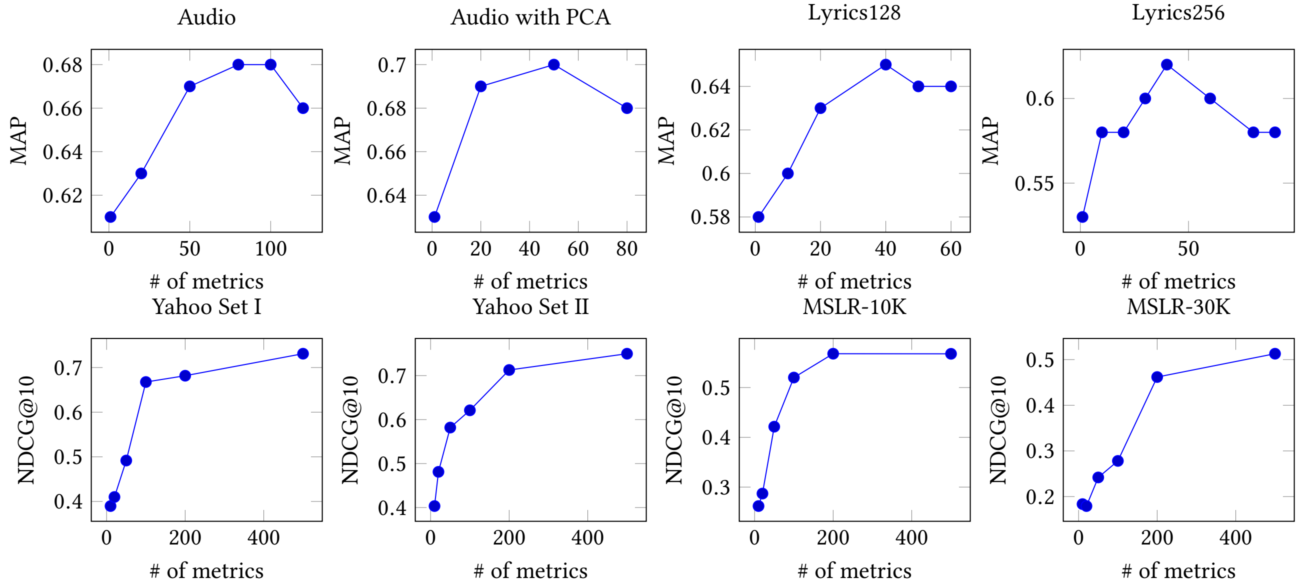

Figure 3: The variation of performance caused by different number of local metrics

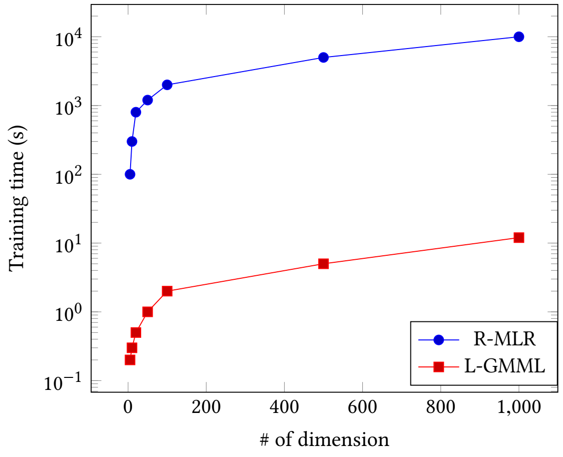

Figure 4: Training time of L-GMML on different scaled synthetic datasets

Table 3: Comparison on the training time of R-MLR and L-GMML. The number of local metrics in L-GMML is fixed as 50

Time (s)	R-MLR	L-GMML
Audio	N/A	38
Audio with PCA	607	4.7
Lyrics-128	302	2.6
Lyrics-256	1241	7.8

such comparison is that we attempt to demonstrate the different influence of the new GMML algorithm and our proposed L-GMML algorithm on the performance improvements.

Therefore, we can draw the conclusion from the experiments in this subsection that the proposed approach outperforms other metric learning algorithms for the ranking problem regarding accuracy and computational efficiency.

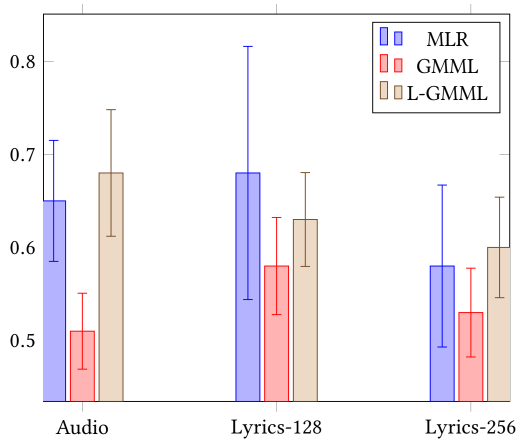

Figure 5: Music similarity performance of each algorithm on the three feature representation Audio, Lyrics-128 and Lyrics-256. Performance was measured by MAP and averaged across 10 folds.

4.5 Comparisons on Large-scale Real-world Datasets

We attempt to find amazing features of our method in the comparisons with two state-of-the-art ranking methods, Gradient-Boosted Regression Trees (GBRT) and λ-MART on Yahoo! Set I&II, MSLR-10K, and MSLR-30K. Because they have been proved to be the most effective in the Yahoo! learning to rank challenge and become the dominant methods in the LtR field.

For these four datasets, the feature domain varies dramatically. To avoid for challenging the floating point precision in complex mathematical computation, we preprocess these four datasets by normalizing each feature dimension with 2-norm. For the stochastic sampling procedure in Algorithm 2, to find the optimal model, we try different initial weight values $\Phi(1)$ ranging from 0.1 to 10, the hinge loss margin ζ ranging from 0.01 to 1.

The training time of GBRT and λ-MART is sensitive to the number of trees in both of the models. The number of local metrics also determines the training time of L-GMML. When we plan to make comparisons on the accuracy and training time of three methods, we fix the number of trees of GBRT and λ-MART as 5000 and the number of local metrics as 500. The motivation of these choices is that the performance of these two methods become stable on the four datasets. The comprehensive comparisons on a different measurement of the above three methods are illustrated in Table 4. From the table, we can draw a conclusion that our approach enjoys a huge advantage in accuracy compared with the state-of-the-art ranking methods.

Currently, the only disadvantage of our approach lies in scoring time. Table 5 displays the comparisons about the time of scoring documents. Our algorithm heavily relies on the scoring for each document in different stages, which is less efficient than GBRT and λ-MART. On the other hand, our approach is simple in structure, and GMML in the first stage is also efficient. Therefore, our method still has an advantage in computationally efficiency. The time-consuming comparison in Table 4 can prove this statement.

5 CONCLUSION

In this paper, we focus on improving the accuracy of LtR methods by utilizing the local structure of documents and degrading irrelevant features. We firstly developed a localized GMML algorithm for the query-independent ranking framework. Specifically, we proposed a concept called *ideal candidate document* to adopt metric learning for ranking algorithm from a query-dependent model to widely used query-independent model. In our approach, a well defined smooth weighting function is optimized by reducing the popular WARP loss, which is defined for the candidate document set of a given query. Then we can efficiently score document by calculating the distance between candidate documents and a nonexistent *ideal candidate document* from an optimized metric space. The experiments prove that our approach outperforms both of the state-of-the-art query-dependent algorithms and query-independent algorithms.

6 ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14208815 and No. CUHK 14234416 of the General Research Fund), and 2015 Microsoft Research Asia Collaborative Research Program (Project No. FY16- RES-THEME-005).

REFERENCES

[1] D. Broomhead, D. S. and Lowe. 1988. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems* 2 (1988), 321– 355.

[2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*. 89–96. DOI: http://dx.doi.org/10.1145/1102351.1102363

[3] Christopher J C Burges. 2010. From RankNet to LambdaRank to LambdaMART : An Overview. *Technical Report* (2010).

[4] Christopher J C Burges, Krysta M Svore, Paul N. Bennett, Andrzej Pastusiak, and Qiang Wu. 2011. Learning to Rank Using an Ensemble of Lambda-Gradient Models. *Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings* 14 (2011), 25–35.

[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank. In *Proceedings of the 24th international conference on Machine learning - ICML '07*. ACM Press, New York, New York, USA, 129–136. DOI: http://dx.doi.org/10.1145/1273496.1273513

[6] Olivier Chapelle. 2011. Yahoo ! Learning to Rank Challenge Overview. *JMLR: Workshop and Conference Proceedings* 14 (2011), 1–24.

[7] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large Scale Online Learning of Image Similarity Through Ranking. *The Journal of Machine Learning Research* 11 (3 2010), 1109–1135.

[8] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning - ICML '07*. ACM Press, New York, New York, USA, 209–216. DOI: http://dx.doi.org/10.1145/1273496.1273523

[9] Clebson C.A. de Sá, Marcos A. Gonçalves, Daniel X. Sousa, and Thiago Salles. 2016. Generalized BROOF-L2R. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*. ACM Press, New York, New York, USA, 95–104. DOI: http://dx.doi.org/10.1145/2911451.2911540

[10] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research* 4 (2003), 933–969. DOI: http://dx.doi.org/10.1162/jmlr.2003.4.6.933

[11] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29, 5 (2001), 1189–1232. DOI: http://dx.doi.org/DOI10.1214/aos/1013203451 arXiv:arXiv:1011.1669v3

Table 4: Performance of GBRT, λ-MART and proposed L-GMML on large-scale real-world datasets. Results of MSLR-WEB10K and MSLR-WEB30K are averaged from the 5 folds in the datasets.

		GBRT		λ-MART		L-GMML	
Dataset		Test Set	Time (min.)	Test Set	Time (min.)	Test Set	Time (min.)
Yahoo! Set I	NDCG@5	0.6529	41.2	0.6567	46.5	**0.6698**	28.1
	NDCG@10	0.6824	43.3	**0.7060**	48.0	0.6715	28.9
	NDCG@20	0.6912	41.5	**0.7091**	46.9	0.6934	28.8
Yahoo! Set II	NDCG@5	0.6731	37.6	0.6791	43.1	**0.7096**	26.5
	NDCG@10	0.6817	36.8	0.7062	43.3	**0.7264**	26.6
	NDCG@20	0.6954	37.4	0.7087	43.8	**0.7219**	26.4
MSLR-WEB10K	NDCG@5	0.4019 ± **0.0083**	49.4 ± 5.2	0.4417 ± 0.0131	58.3 ± 2.8	**0.4771** ± 0.0951	19.7 ± 2.1
	NDCG@10	0.4342 ± 0.0219	48.3 ± 2.1	0.4513 ± **0.0196**	57.6 ± 3.8	**0.5390** ± 0.0812	19 ± 3.1
	NDCG@20	0.4512 ± 0.0279	48.8 ± 3.8	0.4634 ± **0.0257**	57.1 ± 5.2	**0.551** ± 0.0728	19 ± 2.8
MSLR-WEB30K	NDCG@5	0.409 ± 0.0312	167 ± 28.6	0.3812 ± **0.0297**	182 ± 19.8	**0.4837** ± 0.0715	71.7 ± 2.7
	NDCG@10	0.4146 ± 0.0327	177 ± 30.1	0.409 ± **0.0232**	183 ± 17.9	**0.4976** ± 0.0619	71.9 ± 3.9
	NDCG@20	0.421 ± 0.361	167 ± 27.3	0.4112 ± **0.0240**	181 ± 10.7	**0.5038** ± 0.0718	72.5 ± 5.3

Table 5: Per-document scoring time of GBRT, λ-MART and L-GMML on Yahoo! and MSLR datasets. The scoring time is united in *ms*

	GBRT	λ-MART	L-GMML
Yahoo! Set I	276	302	421
Yahoo! Set II	218	286	421
MSLR-WEB10K	73	92	158

[12] Yasser Ganjisaffar, Rich Caruana, and Cristina Videira Lopes. 2011. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval - SIGIR '11*. ACM Press, New York, New York, USA, 85. DOI: http://dx.doi.org/10.1145/2009916.2009932

[13] SÃÿren Hauberg, Oren Freifeld, and Michael J. Black. 2012. A Geometric take on Metric Learning. In *Advances in Neural Information Processing Systems*. 2024–2032.

[14] Bruno Iannazzo. 2016. The geometric mean of two matrices from a computational viewpoint. *Numerical Linear Algebra with Applications* 23, 2 (3 2016), 208–229. DOI: http://dx.doi.org/10.1002/nla.2022

[15] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446. DOI: http://dx.doi.org/10.1145/582415.582418

[16] Brian Kulis. 2013. Metric Learning: A Survey. *Foundations and Trends® in Machine Learning* 5, 4 (2013), 287–364. DOI: http://dx.doi.org/10.1561/2200000019

[17] Ping Li, Qiang Wu, and Christopher J. Burges. 2008. McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In *Advances in Neural Information Processing Systems*. 897–904.

[18] Daryl Lim and Gert Lanckriet. 2014. Efficient Learning of Mahalanobis Metrics for Ranking. In *Proceedings of The 31st International Conference on Machine Learning*. 1980–1988.

[19] Daryl K H Lim, Brian Mcfee, and Gert Lanckriet. 2013. Robust Structural Metric Learning. *Proceeding of the 30th International Conference on Machine Learning (ICML '13)* 28 (2013), 615–623.

[20] Kuan Liu, AurÃľlien Bellet, and Fei Sha. 2015. Similarity Learning for High-Dimensional Sparse Data. *Aistats* 38 (2015), 1–14.

[21] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (6 2009), 225–331. DOI: http://dx.doi.org/10.1561/1500000016

[22] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. 2015. Speeding up Document Ranking with Rank-based Features. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*. ACM Press, New York, New York, USA, 895–898. DOI: http://dx.doi.org/10.1145/2766462.2767776

[23] Brian McFee. 2012. *More like this: machine learning approaches to music similarity.* Ph.D. Dissertation.

[24] Brian McFee, Luke Barrington, and Gert Lanckriet. 2012. Learning content similarity for music recommendation. *IEEE Transactions on Audio, Speech and Language Processing* 20, 8 (2012), 2207–2218. DOI: http://dx.doi.org/10.1109/TASL.2012.2199109

[25] Brian McFee and Gert R Lanckriet. 2010. Metric learning to rank. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010), 775–782.

[26] Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D Lee. 2010. Generative local metric learning for nearest neighbor classification. *Advances in Neural Information Processing Systems* 18 (2010), 417–424.

[27] Deva Ramanan and Simon Baker. 2011. Local Distance Functions A Taxonomy, New Algorithms, and an Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 4 (2011), 794–806.

[28] Rómer Rosales and Glenn Fung. 2006. Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. ACM Press, New York, New York, USA, 367. DOI: http://dx.doi.org/10.1145/1150402.1150444

[29] Chunhua Shen, Junae Kim, Lei Wang, and Anton van den Hengel. 2012. Positive Semidefinite Metric Learning Using Boosting-like Algorithms. *Journal of Machine Learning Research* 13 (2012), 1007–1036. http://dl.acm.org/citation.cfm?id=2343679

[30] Derek Tingle, Youngmoo E. Kim, and Douglas Turnbull. 2010. Exploring automatic music annotation with "acoustically-objective" tags. In *Proceedings of the international conference on Multimedia information retrieval - MIR '10*. ACM Press, New York, New York, USA, 55. DOI: http://dx.doi.org/10.1145/1743384.1743400

[31] Jun Wang, Alexandros Kalousis, and Adam Woznica. 2012. Parametric Local Metric Learning for Nearest Neighbor Classification. In *Advances in Neural Information Processing Systems*. 1601–1609.

[32] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research* 10 (2009), 207–244. DOI: http://dx.doi.org/10.1126/science.277.5323.215

[33] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: Learning to rank with joint word-image embeddings. *Machine Learning* 81, 1 (2010), 21–35. DOI: http://dx.doi.org/10.1007/s10994-010-5198-3

[34] Eric P. Xing, Michael I. Jordan, Stuart J. Russell, and Andrew Y. Ng. 2002. Distance Metric Learning with Application to Clustering with Side-Information. In *Advances in Neural Information Processing Systems*. 521–528.

[35] Zhixiang Eddie Xu, Kilian Q Weinberger, Olivier Chapelle, St Louis, and Olivier Chapelle Cc. 2012. The Greedy Miser: Learning under Test-time Budgets. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. 1175–1182.

[36] Pourya Habib Zadeh, Reshad Hosseini, and Suvrit Sra. 2016. Geometric mean metric learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML '16)*. 2464–2471.