Online Learning to Rank for Content-Based Image Retrieval*

Ji Wan[1,2,3], Pengcheng Wu[2], Steven C. H. Hoi[2], Peilin Zhao[4], Xingyu Gao[1,2,3], Dayong Wang[5], Yongdong Zhang[1], Jintao Li[1]

[1] Key Laboratory of Intelligent Information Processing of CAS, ICT, CAS, China
[2] Singapore Management University [3] University of the Chinese Academy of Sciences
[4] Institute for Infocomm Research, A*STAR, Singapore [5] Michigan State University, MI, USA
{wanji,gaoxingyu,zhyd,jtli}@ict.ac.cn, {pcwu,chhoi}@smu.edu.sg,
zhaop@i2r.a-star.edu.sg, dywang@msu.edu

Abstract

A major challenge in Content-Based Image Retrieval (CBIR) is to bridge the semantic gap between low-level image contents and high-level semantic concepts. Although researchers have investigated a variety of retrieval techniques using different types of features and distance functions, no single best retrieval solution can fully tackle this challenge. In a real-world CBIR task, it is often highly desired to combine multiple types of different feature representations and diverse distance measures in order to close the semantic gap. In this paper, we investigate a new framework of learning to rank for CBIR, which aims to seek the optimal combination of different retrieval schemes by learning from large-scale training data in CBIR. We first formulate the problem formally as a learning to rank task, which can be solved in general by applying the existing batch learning to rank algorithms from text information retrieval (IR). To further address the scalability towards large-scale online CBIR applications, we present a family of online learning to rank algorithms, which are significantly more efficient and scalable than classical batch algorithms for large-scale online CBIR. Finally, we conduct an extensive set of experiments, in which encouraging results show that our technique is effective, scalable and promising for large-scale CBIR.

1 Introduction

Content-based image retrieval (CBIR) has been extensively studied for many years in multimedia and computer vision communities. Extensive efforts have been devoted to various low-level feature descriptors [Jain and Vailaya, 1996] and different distance measures defined on some specific sets of low-level features [Manjunath and Ma, 1996]. Recent years also witness the surge of research on local feature based representations, such as the bag-of-words models [Sivic *et al.*, 2005] using local feature descriptors (e.g., SIFT [Lowe, 1999]).

*This work was supported by Singapore MOE tier 1 research grant (C220/MSS14C003) and the National Nature Science Foundation of China (61428207).

Although CBIR has been studied extensively for years, it is often hard to find a single best retrieval scheme, i.e., some pair of feature representation and distance measure, which can consistently beat the others in all scenarios. It is thus highly desired to combine multiple types of diverse feature representations and different kinds of distance measures in order to improve the retrieval accuracy of a real-world CBIR task. In practice, it is however nontrivial to seek an optimal combination of different retrieval schemes, especially in web-scale CBIR applications with millions or even billions of images. Besides, for real-world CBIR applications, the optimal combination weights for different image retrieval tasks may vary across different application domains. Thus, it has become an urgent research challenge for investigating an automated and effective learning solution for seeking the optimal combination of multiple diverse retrieval schemes in CBIR.

To tackle the above challenge, in this paper, we investigate a machine learning framework of learning to rank algorithms in seeking the optimal combination of multiple diverse retrieval schemes for CBIR by learning from large-scale training data automatically. In particular, we first formulate the problem as a learning to rank task, which thus can be solved in general by applying the existing batch learning to rank algorithms in text IR. However, to further improve the efficiency and scalability issues, we present a family of online learning to rank algorithms to cope with the challenge of large-scale learning in CBIR. We give theoretical analysis of the proposed online learning to rank algorithms, and empirically show that the proposed algorithms are both effective and scalable for large-scale CBIR tasks.

In summary, our main contributions of this paper include: i) We conduct a comprehensive study of applying learning to rank techniques to CBIR, aiming to seek the optimal combination of multiple retrieval schemes; ii) We propose a family of efficient and scalable online learning to rank algorithms for CBIR; iii) We analyze the theoretical bounds of the proposed online learning to rank algorithms, and also examine their empirical performances extensively.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents our problem formulation and a family of online learning to rank algorithms for CBIR, and Section 4 gives theoretical analysis. Section 5 discusses our experiments and Section 6 concludes this work.

2 Related Work

2.1 Learning to Rank and CBIR

Learning to rank has been extensively studied in text Information Retrieval (IR) [Qin *et al.*, 2010]. In general, most existing approaches can be grouped into three major categories: (i) pointwise, (ii) pairwise, and (iii) listwise approaches. We briefly review related work in each category below.

The first group, the family of pointwise learning to rank approaches, simply treats ranking as a regular classification or regression problem by learning to predict numerical ranking values of individual objects. For example, in [Cooper *et al.*, 1992; Crammer and Singer, 2001; Li *et al.*, 2007], the ranking problem was formulated as a regression task in different forms. In addition, [Nallapati, 2004] formulated the ranking problem as a binary classification of relevance on document objects, and solved it by applying some discriminative models such as SVM.

The second group of learning to rank algorithms, the family of pairwise approaches, treats the pairs of documents as training instances and formulates ranking as a task of learning a classification or regression model from the collection of pairwise instances of documents. A variety of pairwise learning to rank algorithms have been proposed by applying different machine learning algorithms [Joachims, 2002; Burges *et al.*, 2005; Tsai *et al.*, 2007]. The well-known algorithms include: SVM-based approaches such as RankSVM [Joachims, 2002], neural networks based approaches such as RankNet [Burges *et al.*, 2005], and boosting-based approaches such as RankBoost [Freund *et al.*, 2003], etc. This group is the most widely explored research direction of learning to rank, in which many techniques have been successfully applied in real-world commercial systems. In general, our proposed approaches belong to this category.

The third group, the family of listwise learning to rank approaches, treats a list of documents for a query as a training instance and attempts to learn a ranking model by optimizing some loss functions defined on the predicted list and the ground-truth list. There are two different kinds of approaches in this category. The first is to directly optimize some IR metrics, such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2000]. Example algorithms include AdaRank [Xu and Li, 2007] and SVM-MAP by optimizing MAP [Yue *et al.*, 2007], and SoftRank [Taylor *et al.*, 2008] and NDCG-Boost [Valizadegan *et al.*, 2009] by optimizing NDCG, etc. The other is to indirectly optimize the IR metrics by defining some listwise loss function, such as ListNet [Cao *et al.*, 2007] and ListMLE [Xia *et al.*, 2008].

Unlike the extensive studies in text IR literature, learning to rank has been seldom explored in CBIR, except some recent study in [Faria *et al.*, 2010; Pedronette and da S Torres, 2013] which simply applied some classical batch learning to rank algorithms. Unlike their direct use of the batch learning to rank algorithms that are hardly scalable for large-scale CBIR applications, we propose a family of efficient and scalable online learning to rank algorithms and evaluate them extensively on a comprehensive testbed. Finally, we note that our work is also very different from a large family of diverse existing studies in CBIR [He *et al.*, 2004; Hoi *et al.*, 2006; Chechik *et al.*, 2010] that usually to apply machine learning techniques (supervised or unsupervised learning) to learn a good ranking function on a single type of features or some combined features. Such existing techniques potentially could be incorporated as one component of our scheme, which is out of scope of the discussions in this work.

2.2 Online Learning

Online learning is a family of efficient and scalable machine learning algorithms [Rosenblatt, 1958; Crammer *et al.*, 2006] extensively studied in machine learning for years. In general, online learning operates in a sequential manner. Consider online classification, each time step, an online learner processes an incoming example by first predicting its class label; after that, it receives the true class label from the environment, which is then used to measure the loss between the predicted label and the truth label; at the end of each time step, the learner is updated whenever the loss is nonzero. Typically, the goal of an online learning task is to minimize the cumulative mistakes over the entire sequence of predictions.

In literature, a variety of algorithms have been proposed for online learning [Hoi *et al.*, 2014]. The most well-known example is the Perceptron algorithm [Rosenblatt, 1958]. In recent years, various algorithms have been proposed to improve Perceptron [Li and Long, 1999; Crammer *et al.*, 2006], which usually follow the criterion of maximum margin learning principle. A notable approach is the family of Passive-Aggressive (PA) learning algorithms [Crammer *et al.*, 2006], which updates the classifier whenever the online learner fails to produce a large margin on the current instance. These algorithms are often more efficient and scalable than batch learning algorithms. In this work, we aim to extend the existing online learning principle for developing new learning to rank algorithms. In addition, we note that our work is also very different from another study in [Grangier and Bengio, 2008] which focuses on text-based image retrieval by applying PA algorithms. By contrast, our CBIR study focuses on image retrieval based on the visual similarity. Finally, the proposed online learning to rank is based on linear models and is thus more scalable than the kernel-based similarity learning approaches [Xia *et al.*, 2014].

3 Online Learning to Rank for CBIR

In this section, we present the problem formulation and the proposed online learning to rank algorithms for CBIR.

3.1 Problem Formulation

Let us denote by $\mathcal{I}$ an image space. Each training instance received at time step t is represented by a triplet instance (q_t, p_t^1, p_t^2), where $q_t \in \mathcal{I}$ denotes the t-th query in the entire collection of queries, $p_t^1 \in \mathcal{I}$ and $p_t^2 \in \mathcal{I}$ denote a pair of images for ranking prediction w.r.t. the query q_t.

We also denote by $y_t \in \{+1, -1\}$ the true ranking order of the pairwise instances at step t such that image p_t^1 is ranked before p_t^2 if $y_t = +1$; otherwise p_t^1 is ranked after p_t^2. We introduce a mapping function

$$\phi : \mathcal{I} \times \mathcal{I} \mapsto \mathbb{R}^n,$$

which creates a n-dimensional feature vector from an image pair. For example, consider $\phi(q,p) \in \mathbb{R}^n$, one way to extract one of the n features is based on different similarity measures on different feature descriptors.

The goal of a learning to rank task is to search for the optimal ranking model $\mathbf{w} \in \mathbb{R}^n$ with the following target ranking function for any triplet instance (q_t, p_t^1, p_t^2):

$$\begin{aligned} f(q_t, p_t^1, p_t^2) &= \mathbf{w}^\top \phi(q_t, p_t^1, p_t^2) \\ &= \mathbf{w}^\top (\phi(q_t, p_t^1) - \phi(q_t, p_t^2)). \end{aligned}$$

By learning an optimal model, we expect the prediction output by the function $f(q_t, p_t^1, p_t^2)$ will be positive if an image p_t^1 is more similar to the query q_t than another image p_t^2, and negative otherwise.

In particular, for a sequence of T triplet training instances, our goal is to optimize the sequence of ranking models $\mathbf{w}_1, \ldots, \mathbf{w}_T$ so as to minimize the prediction mistakes during the entire online learning process. Below we present a family of online learning algorithms to tackle the learning to rank tasks. We note that we mainly explore the first-order online learning techniques due to their high efficiency, but the similar idea could also be extended by exploring second-order online learning techniques [Dredze *et al.*, 2008].

3.2 Online Perceptron Ranking (OPR)

The Online Perceptron Ranking (OPR) follows the idea of Perceptron [Rosenblatt, 1958], a classical online learning algorithm. In particular, given any training instance (q_t, p_t^1, p_t^2) and true label y_t, at step t, OPR makes the following update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_t(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)), \quad (1)$$

whenever $y_t \mathbf{w}_t{}^\top(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)) < 0$; otherwise, the ranking model remains unchanged.

3.3 Online Passive Aggressive Ranking (OPAR)

The Online Passive Aggressive Ranking (OPAR) follows the idea of the online passive-aggressive (PA) learning [Crammer *et al.*, 2006] to tackle this challenge. In particular, we first formulate the problem as an optimization task (OPAR-I):

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell(\mathbf{w}; (q_t, p_t^1, p_t^2), y_t), \quad (2)$$

where $\ell(\mathbf{w})$ is a hinge loss defined as

$$\ell(\mathbf{w}) = \max(0, 1 - y_t \mathbf{w}^\top(\phi(q_t, p_t^1) - \phi(q_t, p_t^2))),$$

and $C > 0$ is a penalty cost parameter. We can also formulate this problem as another variant (OPAR-II):

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell(\mathbf{w}; (q_t, p_t^1, p_t^2), y_t)^2. \quad (3)$$

The above two optimizations trade off two major concerns: (i) the updated ranking model should not be deviated too much from the previous ranking model $\mathbf{w}_t$, and (ii) the updated ranking model suffers a small loss on the triplet training instance (q_t, p_t^1, p_t^2). The tradeoff is essentially controlled by the penalty cost parameter C. Finally, we can derive the following proposition for the closed-form solutions to the above optimizations.

Proposition 1. *The optimizations in (2) and (3) have the following closed-form solution:*

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \lambda_t y_t(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)), \quad (4)$$

where λ_t for (2) *is computed as:*

$$\lambda_t = \min(C, \frac{\ell_t(\mathbf{w}_t)}{\|\phi(q_t, p_t^1) - \phi(q_t, p_t^2))\|^2}), \quad (5)$$

and λ_t for (3) *is computed as:*

$$\lambda_t = \frac{\max(0, 1 - \mathbf{w}_t{}^\top y_t(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)))}{\|\phi(q_t, p_t^1) - \phi(q_t, p_t^2))\|^2 + \frac{1}{2C}}.$$

The above proposition can be obtained by following the similar idea of passive aggressive learning in [Crammer *et al.*, 2006]. We omit the details here due to the space limitation. From the results, we can see that the ranking model remains unchanged if $\mathbf{w}_t{}^\top y_t(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)) \geq 1$. That is, we will update the ranking model whenever the current ranking model fails to rank the order of p_t^1 and p_t^2 w.r.t. query q_t correctly at a sufficiently large margin.

3.4 Online Gradient Descent Ranking (OGDR)

The Online Gradient Descent Ranking (OGDR) follows the idea of Online Gradient Descent [Zinkevich, 2003] to tackle our problem. When receiving a training instance (q_t, p_t^1, p_t^2) and its true label y_t at each time step t, we suffer a hinge loss as Eq. (5). Then we update the ranking model based on the gradient descent of the loss function:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \ell(\mathbf{w}; (q_t, p_t^1, p_t^2), y_t), \quad (6)$$

where η is the learning rate. More specifically, whenever the loss $\ell(\mathbf{w}; (q_t, p_t^1, p_t^2), y_t)$ is nonzero, OGDR makes the following update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta y_t(\phi(q_t, p_t^1) - \phi(q_t, p_t^2)). \quad (7)$$

4 Theoretical Analysis

In this section, we analyze the performance of the proposed online algorithms. We firstly present a lemma to disclose the relationship between the cumulative loss and an IR performance measure, i.e., mean average precision (MAP).

Lemma 1. *For one query q_t and its related images, the MAP is lower bounded as follows:*

$$MAP \quad \geq \quad 1 - \frac{\gamma_{MAP}}{T} \sum \ell(\mathbf{w}; (q_t, p_t^1, p_t^2), y_t),$$

where $\gamma_{MAP} = 1/m$, and m is the number of relevant documents.

Proof. Using the essential loss idea defined in [Chen *et al.*, 2009], from Theorem 1 of [Chen *et al.*, 2009] we could see the essential loss is an upper bound of measure-based ranking errors; besides, the essential loss is the lower bound of the sum of pairwise square hinge loss, using the properties of square hinge loss, which is non-negative, non-increasing and satisfy $\ell(0) = 1$. □

The above lemma indicates that if we could prove bounds for the online cumulative hinge loss compared to the best ranking model with all data beforehand, we could obtain the cumulative IR measure bounds. Fortunately there are strong theoretical loss bounds for the proposed online learning to rank algorithms. Therefore, we can prove the MAP bounds for each of the proposed algorithms as follows.

Theorem 1. *Assume* $\max_t \|\phi(q_t,p_t^1)-\phi(q_t,p_t^2)\|^2 \leq X$, *the MAP of the Online Perceptron Ranking algorithm is bounded as*

$$MAP \quad \geq \quad 1-\frac{\gamma_{MAP}X}{2}-\frac{\gamma_{MAP}}{T}\{\frac{1}{2}\|\mathbf{w}\|^2+\sum \ell_t(\mathbf{w})\}.$$

Proof. Define $\Delta_t = \|\mathbf{w}_t-\mathbf{w}\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}\|^2$, it is not difficult to see:

$$\sum_{t=1}^{T}\Delta_t = \|\mathbf{w}\|^2-\|\mathbf{w}_{T+1}-\mathbf{w}\|^2 \leq \|\mathbf{w}\|^2.$$

In addition, according to the update rule, we have

$$\begin{aligned}&\Delta_t = -2y_t(\mathbf{w}_t-\mathbf{w})\cdot(\phi(q_t,p_t^1)-\phi(q_t,p_t^2))\\&-\|\phi(q_t,p_t^1)-\phi(q_t,p_t^2)\|^2\\&\leq 2\ell_t(\mathbf{w}_t)-2\ell_t(\mathbf{w})-X\end{aligned}$$

Combining the above two inequalities results in

$$\|\mathbf{w}\|^2 \geq \sum_t[2\ell_t(\mathbf{w}_t)-2\ell_t(\mathbf{w})-X].$$

Re-arranging the above inequality gives

$$\sum \ell_t(\mathbf{w}_t) \leq \frac{1}{2}\|\mathbf{w}\|^2+\sum[\ell_t(\mathbf{w})+\frac{1}{2}X].$$

Plugging the above inequality into the Lemma 1 concludes the proof. □

Theorem 2. *Assume* $\max_t \|(\phi(q_t,p_t^1)-\phi(q_t,p_t^2))\|^2 \leq X$, *the MAP of the OPAR-I algorithm is bounded as follows:*

$$\begin{aligned}MAP \quad &\geq \quad 1-\frac{\gamma_{MAP}C^2X}{\lambda_*}\\&- \quad \frac{\gamma_{MAP}}{T\lambda_*}\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum \ell_t(\mathbf{w})\},\end{aligned}$$

where $\lambda_* = \min_{\lambda_t>0}\lambda_t$, *while the MAP of the OPAR-II algorithm is bounded as:*

$$\begin{aligned}MAP \geq 1-\frac{\gamma_{MAP}(X+1/(2C))}{\ell_* T}\{\|\mathbf{w}\|^2\\+2C\sum \ell_t(\mathbf{w})^2\},\end{aligned}$$

where $\ell_* = \min_{\ell_t>0}\ell_t$.

Proof. Firstly, for the OPAR-I algorithm, it is not difficult to show that

$$\sum_{t=1}^{T}\Delta_t = \|\mathbf{w}\|^2-\|\mathbf{w}_{T+1}-\mathbf{w}\|^2 \leq \|\mathbf{w}\|^2,$$

where $\Delta_t = \|\mathbf{w}_t-\mathbf{w}\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}\|^2$. In addition, using the relation between $\mathbf{w}_t$ and $\mathbf{w}_{t+1}$ gives

$$\begin{aligned}\Delta_t =& -2\lambda_t y_t(\mathbf{w}_t-\mathbf{w})^\top[\phi(q_t,p_t^1)-\phi(q_t,p_t^2)]\\&-\lambda_t^2\|(\phi(q_t,p_t^1)-\phi(q_t,p_t^2))\|^2\\ \leq& 2\lambda_t\ell_t(\mathbf{w}_t)-2\lambda_t\ell_t(\mathbf{w})-\lambda_t^2X.\end{aligned}$$

Combining the above two inequalities gives

$$\|\mathbf{w}\|^2 \geq \sum\{2\lambda_t\ell_t(\mathbf{w}_t)-2\lambda_t\ell_t(\mathbf{w})-\lambda_t^2X\}.$$

Denote $\lambda_* = \min_{\lambda_t>0}\lambda_t$ and re-arranging the above inequality, we can get

$$\sum \ell_t(\mathbf{w}_t) \leq \frac{1}{\lambda_*}\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum \ell_t(\mathbf{w})+\frac{1}{2}C^2XT\}.$$

Plugging the above inequality into Lemma 1 concludes the first part of this theorem.

Similarly, assume $\ell_* = \min_{\ell_t(\mathbf{w}_t)>0}\ell_t(\mathbf{w}_t)$ for OPAR-II algorithm, we can prove

$$\sum_t \ell_t(\mathbf{w}_t) \leq \frac{(X+1/(2C))}{\ell_*}\{\|\mathbf{w}\|^2+2C\sum \ell_t(\mathbf{w})^2\}.$$

Combining the above inequality with Lemma 1 concludes the second part of this theorem. □

Theorem 3. *Assume* $\max_t \|(\phi(q_t,p_t^1)-\phi(q_t^,p_t^2))\|^2 \leq X$, *the MAP of the online gradient descent ranking algorithm is bounded as:*

$$MAP \geq 1-\frac{\gamma_{MAP}}{T}\{\frac{1}{2\eta}\|\mathbf{w}\|^2+\frac{\eta XT}{2}+\sum_{t=1}^{T}\ell_t(\mathbf{w})\}.$$

Proof. Firstly, according to equation (6), we have

$$\begin{aligned}&\|\mathbf{w}_{t+1}-\mathbf{w}\|^2 = \|\mathbf{w}_t-\eta\nabla\ell_t(\mathbf{w}_t)-\mathbf{w}\|^2\\&=\|\mathbf{w}_t-\mathbf{w}\|^2-2\langle\mathbf{w}_t-\mathbf{w},\eta\nabla\ell_t(\mathbf{w}_t)\rangle+\|\eta\nabla\ell_t(\mathbf{w}_t)\|^2.\end{aligned}$$

The above equality can be reformulated as follows:

$$\begin{aligned}&\langle\mathbf{w}_t-\mathbf{w},\nabla\ell_t(\mathbf{w}_t)\rangle\\&= \frac{1}{2\eta}[\|\mathbf{w}_t-\mathbf{w}\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}\|^2+\|\eta\nabla\ell_t(\mathbf{w}_t)\|^2]. \quad (7)\end{aligned}$$

Secondly, $\ell_t(\cdot)$ is convex, so

$$\ell_t(\mathbf{w}) \geq \ell_t(\mathbf{w}_t)+\langle\nabla\ell_t(\mathbf{w}_t),(\mathbf{w}-\mathbf{w}_t)\rangle.$$

Reformulating this inequality and plugging it into Eq. (7) results in

$$\begin{aligned}&\ell_t(\mathbf{w}_t)-\ell_t(\mathbf{w}) \leq \langle\nabla\ell_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}\rangle\\&\frac{1}{2\eta}[\|\mathbf{w}_t-\mathbf{w}\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}\|^2+\|\eta\nabla\ell_t(\mathbf{w}_t)\|^2].\end{aligned}$$

Summing the above inequality over t, we get

$$\begin{aligned}&\sum_{t=1}^{T}\ell_t(\mathbf{w}_t)-\sum_{t=1}^{T}\ell_t(\mathbf{w})\\&\leq \sum_{t=1}^{T}\frac{1}{2\eta}[\|\mathbf{w}_t-\mathbf{w}\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}\|^2+\|\eta\nabla\ell_t(\mathbf{w}_t)\|^2]\\&=\frac{1}{2\eta}[\|\mathbf{w}_1-\mathbf{w}\|^2-\|\mathbf{w}_{T+1}-\mathbf{w}\|^2]+\sum_{t=1}^{T}\frac{1}{2\eta}\|\eta\nabla\ell_t(\mathbf{w}_t)\|^2\\&\leq \frac{1}{2\eta}\|\mathbf{w}\|^2+\sum_{t=1}^{T}\frac{\eta}{2}\|y_t(\phi(q_t,p_t^1)-\phi(q_t^,p_t^2))\|^2.\end{aligned}$$

Re-arranging the above inequality results in

$$\sum_{t=1}^{T}\ell_t(\mathbf{w}_t) \leq \frac{1}{2\eta}\|\mathbf{w}\|^2+\frac{\eta XT}{2}+\sum_{t=1}^{T}\ell_t(\mathbf{w}).$$

Plugging the above inequality into the Lemma 1 concludes the proof. □

5 Experiments

We conduct an extensive set of experiments for benchmark evaluations of varied learning to rank algorithms for CBIR tasks, including both batch and online learning algorithms.

5.1 Testbeds for Learning to Rank

Table 1 shows a list of image databases in our testbed. For each database, we randomly split it into five folds, in which one fold is used for test, one is for validation, and the rest are for training. Besides, to test the scalability of our technique for large-scale CBIR, we also include a large database ("ImageCLEFFlickr"), which includes ImageCLEF as a ground-truth subset and 1-million distracting images from Flickr.

Table 1: List of image databases in our testbed.

Datasets	#images	#classes	#train-instances
Holiday	1,491	500	200,000
Caltech101	8,677	101	200,000
ImageCLEF	7,157	20	200,000
Corel	5,000	50	200,000
ImageCLEFFlickr	1,007,157	21	3,000,000

To generate training data of query-dependent descriptors, for each query in a dataset, we involve all positive/relevant images and sample a subset of negative/irrelevant images. The feature mapping $\phi(q, p) \in \mathbb{R}^n$ is computed over 9 different features with 4 similarity measurements, which results in 36-dimensional feature representation. Due to the low efficiency of the existing batch learning to rank algorithms, we design two different experiments. The first aims to evaluate different learning to rank algorithms on all the standard databases, in which we can only sample a total of 200,000 training instances as training data set to ensure that all the batch learning to rank algorithms can be completed. The second aims to examine if the proposed technique can cope with large amount of training data, in which a total of 3-million training instances were generated in the training data set. For validation and test data sets, we randomly choose 300 validation images and 150 test images from each fold.

5.2 Setup and Compared Algorithms

To conduct a fair evaluation, we choose the parameters of different algorithms via the same cross validation scheme in all the experiments. To evaluate the retrieval performance, we adopt the mean Average Precision (mAP), a metric widely used in IR, which is calculated based on the Average Precision (AP) value of all the queries, where the value of AP is the area under precision-recall curve for a query.

To evaluate the efficacy of our scheme, we compare the proposed family of online learning to rank algorithms, including OPR, OPAR-I, OPAR-II and OGDR, against several representative batch learning to rank algorithms in text IR, including RankNet [Burges *et al.*, 2005], Coordinate Ascent("C-Ascent") [Metzler and Croft, 2007], RankSVM [Herbrich *et al.*, 2000] and LambdaMART("λ-MART") [Wu *et al.*, 2010]. Besides, we also evaluate two straightforward baselines: (i) "Best-Fea": it selects the best query-dependent descriptor for ranking via cross validation; and (ii) "Uni-Con": it uniformly combines all the query-dependent descriptors for ranking.

5.3 Evaluation on Standard Datasets

We first evaluate the algorithms on the standard datasets. Table 2 shows average MAP performance on five standard datasets. Several observations can be drawn as follows.

Table 2: Evaluation of the average MAP performance.

Algorithm	Holiday	Caltech101	ImageCLEF	Corel
Best-Fea	0.4892	0.2664	0.5777	0.1846
Uni-Con	0.5175	0.2594	0.6174	0.2990
RankNet	0.6292	0.2753	0.6326	0.3133
C-Ascent	0.6373	0.3193	**0.6803**	0.3406
RankSVM	**0.6429**	0.3270	0.6585	0.3366
λ-MART	0.6230	**0.3650**	0.6796	**0.3683**
OPR	0.6219	**0.3285**	0.6555	0.3292
OPAR-I	0.6329	0.3070	0.6556	0.3340
OPAR-II	0.6283	0.3157	**0.6632**	**0.3389**
OGDR	**0.6368**	0.3024	0.6626	0.3228

First, we observe that all the learning to rank algorithms outperform the two heuristic baselines ("Best-Fea" and "Uni-Con") for most cases. This clearly demonstrates that the proposed learning to rank framework can effectively combine different feature representation and distance measures for improving image retrieval performance. Second, comparing different batch learning to rank algorithms, we observe that no single method can beat the others on all datasets, which is consistent to some previous empirical study in text IR, and λ-MART tends to perform slightly better which attained the best performance among 2 out of 4 datasets. Third, by examining the proposed online learning to ranking algorithms, we found that their average mAP performance is fairly comparable to the batch algorithms, which indicates that the online algorithms are at least as effective as the existing batch algorithms in terms of the retrieval efficacy.

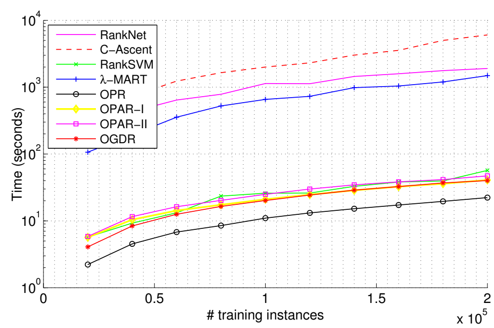

Figure 1: Cumulative time cost on Corel w/ 200k instances.

To evaluate the efficiency and scalability, we measure the time cost taken by different learning to rank algorithms given different amounts of training data. Figure 1 shows the evaluation of CPU time cost on the Corel dataset on different amounts of training instance streams from a total of 200,000 training instances. The online learning algorithms take only tens of seconds for training while batch learning algorithms are much slower, e.g. C-Ascent take around 2 hours. It is clear to see that the proposed online algorithms are considerably more efficient and scalable than most of the existing batch algorithms.

5.4 Evaluation on the Large-scale Dataset

In this experiment, we evaluate the proposed family of online learning to rank algorithms on the large-scale dataset, i.e., the ImageCLEFFlickr data set with over 1-million images and 3-million training instances. For the batch algorithms, we can only evaluate the RankSVM since the other algorithms are too computationally intensive to run on this data set.

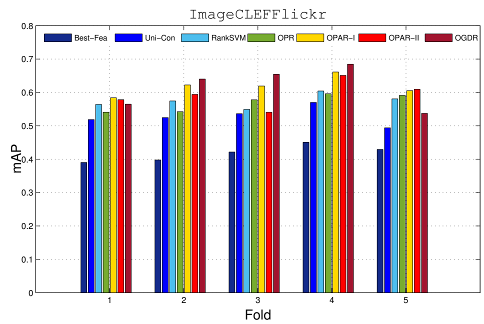

Figure 2: Evaluation of the MAP performance on the ImageCLEFFlickr dataset with over 1-million images.

Figure 2 shows the evaluation of mAP performance on five different folds and Table 3 shows the evaluation of running time cost on 3-million training instances. We can draw several observations from the results. First, the online learning to rank algorithms generally outperform the baseline algorithms without learning to rank significantly. Furthermore, our proposed algorithms achieve better or at least comparable accuracy performance than the state-of-the-art batch learning to rank approaches. Finally, the online learning to rank algorithms are generally more efficient than the batch algorithm.

Table 3: Running time(s) on 3-million training instances.

RankSVM	OPR	OPAR-I	OPAR-II	OGDR
4737	1154	1370	1708	2307

5.5 Evaluation on the Large Scale Online CBIR

We now simulate a real-world online CBIR system by assuming training data arrive sequentially. This is a more realistic setting, especially for web image search engines where user query log data collected from click actions often arrive sequentially. At each iteration, after receiving a query image, we first apply the former learned model for CBIR, and then assume the top 50 retrieved images would be labeled, e.g., via a relevance feedback scheme interactively. After that, the newly received labeled data are adopted to update the model which will predict the next query image. Because batch learning algorithms require all the labeled data (including the earlier iterations) available for training, we employ a reservoir scheme that cached all the labeled data for RankSVM.

Specifically, we use ImageCLEFFlickr as the database set, and randomly select 2,000 images for sequential queries. Fig 3 shows the improvement of NDCG@50 for different algorithms compared to the Uni-Con baseline, whose average NDCG@50 over 2,000 queries is about 0.80.

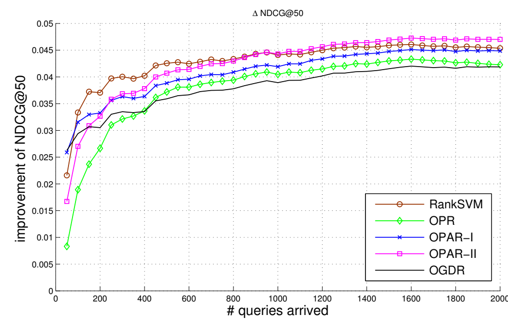

Figure 3: Online cumulative retrieval performance.

We also measure the cumulative CPU time cost taken by different algorithms shown in Figure 4. The batch learning algorithm RankSVM take about a few of hours for re-training while all online learning methods just take several seconds. It is clear to observe that batch algorithms are impractical for this application, meanwhile the proposed online algorithms are significantly more efficient and scalable.

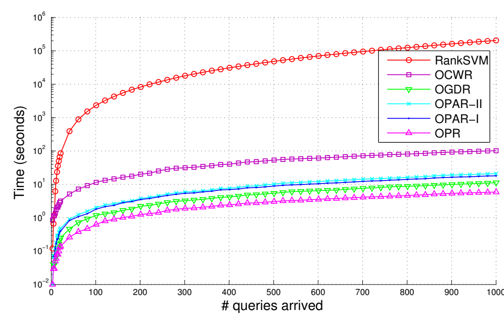

Figure 4: Online Cumulative Time Cost.

6 Conclusions

This paper investigates a new framework of efficient and scalable learning to rank for CBIR, which aims to learn an optimal combination of multiple feature representations and different distance measures. We formulate the problem as a learning to rank task, and explore online learning to solve it. To overcome the drawbacks of existing batch learning to rank techniques, we present a family of efficient and scalable online learning to rank algorithms, which are empirically as effective as the batch algorithms for CBIR, but significantly more scalable by avoiding re-training. Finally, we note that our technique is rather generic, which could be extended for solving many other types of multimedia retrieval tasks.

References

[Burges *et al.*, 2005] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[Cao *et al.*, 2007] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

[Chechik *et al.*, 2010] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010.

[Chen *et al.*, 2009] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhiming Ma, and Hang Li. Ranking measures and loss functions in learning to rank. In *NIPS*, 2009.

[Cooper *et al.*, 1992] William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. Probabilistic retrieval based on staged logistic regression. In *SIGIR*, 1992.

[Crammer and Singer, 2001] Koby Crammer and Yoram Singer. Pranking with ranking. In *NIPS*, pages 641–647, 2001.

[Crammer *et al.*, 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, 2006.

[Dredze *et al.*, 2008] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *ICML*, pages 264–271, 2008.

[Faria *et al.*, 2010] Fabio F. Faria, Adriano Veloso, Humberto M. Almeida, Eduardo Valle, Ricardo da S. Torres, Marcos A. Gonçalves, and Wagner Meira, Jr. Learning to rank for content-based image retrieval. In *MIR*, 2010.

[Freund *et al.*, 2003] Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.

[Grangier and Bengio, 2008] David Grangier and Samy Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE TPAMI*, 30(8):1371–1384, 2008.

[He *et al.*, 2004] Jingrui He, Mingjing Li, Hong-Jiang Zhang, Hanghang Tong, and Changshui Zhang. Manifold-ranking based image retrieval. In *ACM MM*, pages 9–16. ACM, 2004.

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

[Hoi *et al.*, 2006] Steven CH Hoi, Wei Liu, Michael R Lyu, and Wei-Ying Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, volume 2, pages 2072–2078. IEEE, 2006.

[Hoi *et al.*, 2014] Steven C.H. Hoi, Jialei Wang, and Peilin Zhao. Libol: A library for online learning algorithms. *JMLR*, 15:495–499, 2014.

[Jain and Vailaya, 1996] Anil K. Jain and Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.

[Järvelin and Kekäläinen, 2000] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR*, 2000.

[Joachims, 2002] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[Li and Long, 1999] Yi Li and Philip M. Long. The relaxed online maximum margin algorithm. In *NIPS*, pages 498–504, 1999.

[Li *et al.*, 2007] Ping Li, Christopher J. C. Burges, and Qiang Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In *NIPS*, 2007.

[Lowe, 1999] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[Manjunath and Ma, 1996] B. S. Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE TPAMI*, 18(8):837–842, 1996.

[Metzler and Croft, 2007] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, 2007.

[Nallapati, 2004] Ramesh Nallapati. Discriminative models for information retrieval. In *SIGIR*, 2004.

[Pedronette and da S Torres, 2013] Daniel Carlos Guimarães Pedronette and Ricardo da S Torres. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 2013.

[Qin *et al.*, 2010] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.*, 13(4):346–374, 2010.

[Rosenblatt, 1958] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.*, 7:551–585, 1958.

[Sivic *et al.*, 2005] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering objects and their location in images. In *CVPR*, 2005.

[Taylor *et al.*, 2008] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM*, 2008.

[Tsai *et al.*, 2007] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. Frank: a ranking method with fidelity loss. In *SIGIR*, pages 383–390, 2007.

[Valizadegan *et al.*, 2009] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *NIPS*, 2009.

[Wu *et al.*, 2010] Qiang Wu, Christopher J. Burges, Krysta M. Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Inf. Retr.*, 13(3):254–270, June 2010.

[Xia *et al.*, 2008] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *ICML*, 2008.

[Xia *et al.*, 2014] Hao Xia, Steven CH Hoi, Rong Jin, and Peilin Zhao. Online multiple kernel similarity learning for visual search. *IEEE TPAMI*, 36(3):536–549, 2014.

[Xu and Li, 2007] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR*, pages 391–398, 2007.

[Yue *et al.*, 2007] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *SIGIR*, 2007.

[Zinkevich, 2003] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.