



Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

RankCNN: When learning to rank encounters the pseudo preference feedback

Yuan Dong^{a,*}, Chong Huang^a, Wei Liu^b^a Beijing University of Posts and Telecommunications, 100876 PR China^b France Telecom Research & Development, Beijing 100190, PR China

ARTICLE INFO

Article history:

Received 10 September 2013

Received in revised form 15 October 2013

Accepted 27 October 2013

Available online xxxxx

Keywords:

Pseudo preference feedback

Convolutional neural networks

Learning to rank

RankCNN

ABSTRACT

Learning to rank has received great attentions in the field of text retrieval for several years. However, a few researchers introduce the topic into visual reranking due to the special nature of image presentation. In this paper, a novel unsupervised visual reranking is proposed, termed rank via the convolutional neural networks (RankCNN). This approach integrates deep learning with pseudo preference feedback. The optimal set of pseudo preference pairs is first detected from initial list by a modified graph-based method. Ranking is then reduced to pairwise classification in the architecture of CNN. In addition, Accelerated Mini-Batch Stochastic Dual Coordinate Ascent (ASDCA) is introduced to the framework to accelerate the training. The experiments indicate the competitive performance on the LETOR 4.0, the Paris and the *Francelandmark* dataset.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The content-based visual retrieval [1] has been extensively investigated and applied in a number of applications. While successful, the traditional content-based search sometimes fails with some irrelevant retrievals (Fig. 1). As a result, many methods are proposed to rerank the initial results, known as learning to rank. Learning to rank has been applied to document retrieval, collaborative filtering, expert finding, sentiment analysis, and product rating [2]. In conducting learning to rank, the major methods fall into three categories: (1) the pointwise approach, (2) the pairwise approach, and (3) the listwise approach. In the pointwise approaches, each training instance is associated with a rating. The learning is to find a model that can map instances into ratings that are close to their true ones. The listwise approaches use a list of ranked objects as training instances and learn to predict the list of objects. The pairwise approaches take document pairs as instances in learning, and formalize the problem of learning to rank as that of classification. Specifically, document pairs are collected from the ranking lists, and each pair is assigned a label representing the relative relevance of the two documents. A classification model is then developed based on the labeled data to rerank the list. Based on the Support Vector Machines (SVM), Boosting, and Neural Network classifiers, some known methods are developed such as RankSVM [3], RankBoost [4], and RankNet [5]. Considering the most existing methods are pairwise approaches, we choose the pairwise approaches as the basic conception in this paper.

It is interesting to note “learn to rank” has attracted great attention in text retrieval not for the visual ranking. This may be due to the major difference between the document representation and the image representation. For example, the dimension of visual feature is generally much greater than that of document feature; each dimension of document feature has real physical meaning but visual feature does not. Let image be presented as a (Bag of the Word) BoW vector, it will be up to million dimensions which truly challenges the classifiers such as SVM, Boosting or Neural Network.

Fortunately, convolutional neural networks (CNN) [5], one of variants of multilayer perception (MLP), have been proposed to handle the natural image classification problem effectively. According to [7], the current best error rate of CNN on the MNIST digit recognition task is less than 0.3% which is comparable to human performance [8]. CNN thus has the promise to be applied in the visual reranking if it is offered the correct preference pairs. During the reranking process, there is a need to detect the maximum likelihood preference pairs. A method proposed by [9], termed pseudo preference feedback (PPF), is demonstrated to be successful to automatically discover an optimal set of pseudo preference pairs. This method could potentially integrate with CNN as an unsupervised learning pipeline to handle visual reranking problem. While promising, one challenge in the field is the computation expenses. It is known that the efficient gradient descent method is able to accelerate the convergence, and obtain better local minimum. Yet, implementing the method in the parallel computing system is less touched.

In this paper, we propose an unsupervised learning-based visual reranking. Before training, a graph-based method [10] is improved to detect the pseudo preference pairs. During the learning, a simple probabilistic cost function is proposed, which reduces the ranking to the pairwise classification. This approach called RankCNN, motivated by

* Corresponding author.

E-mail addresses: yuandong@bupt.edu.cn (Y. Dong), huangchong661100@bupt.edu.cn (C. Huang), wei.liu@orange.com (W. Liu).

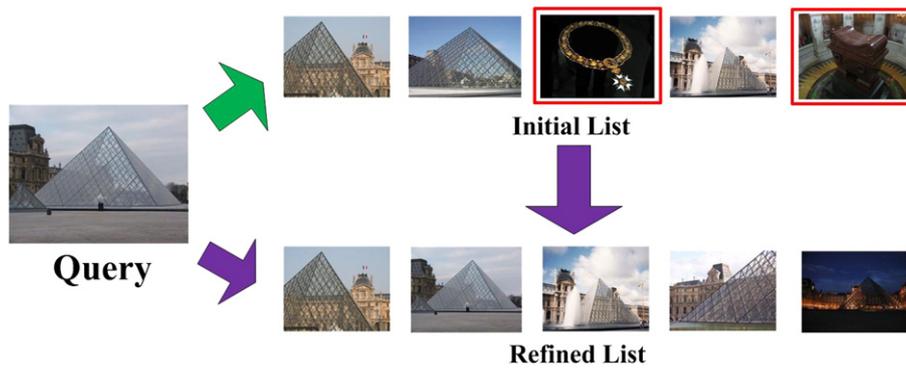


Fig. 1. Visual reranking. The pictures in the red boxes are false positive, which would be filtered in the refined list. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

RankNet, is implemented in the architecture of Convolutional Neural Networks to model the underlying ranking function. Fig. 2 outlines the overall framework for visual reranking.

The rest of the paper is organized as follows. Section 2 introduces related work. The detection of pseudo preference pairs is introduced in Section 3 and the RankCNN model is described in Section 4. Section 5 reports our experimental results followed by conclusion in Section 6.

2. Related work

In this section, several existing methods are presented. These methods were chosen due to their close relation with our approach. The spirit of RankNet is the key of our reranking method, and the convolutional neural networks are the basic architecture. Pseudo preference feedback and gradient descent are the focus for optimization. These methods will be discussed as follows.

2.1. RankNet

In the RankNet, the ranking problem is transformed into the classification of two categories, namely, high-rank and low-rank. This method is simple to train and gives good performance on a real world ranking

problem with large amounts of data. It is known that RankNet outperforms RankSVM and RankBoost for large dimension dataset [5]. In the RankNet, the cross entropy is employed as loss function as follows:

$$C_{ij} = -\bar{P}_{ij} \log P_{ij} - (1 - P_{ij}) \log (1 - \bar{P}_{ij})$$

where \bar{P}_{ij} is the desired target value for the posteriors probability that x_i ranks higher than x_j , and P_{ij} is the modeled posterior probability. Several existing methods could be used to solve the minimization problem such as gradient descent. While successful, to the best of our knowledge, there is little exploration on applying RankNet for visual ranking.

2.2. Convolutional neural networks

The convolutional neural networks (CNNs) have strong capability to present the image because of its controlled depth and breadth, and they could make strong and mostly correct assumptions about the nature of images [6]. Compared to standard feedforward neural network with similarly-sized layers (such as DBN), CNNs have fewer connections and parameters thus they are easier to train, though their theoretically-best performance may be slightly worse [11]. However, the CNNs still suffer from computation expense which limits its application to high-

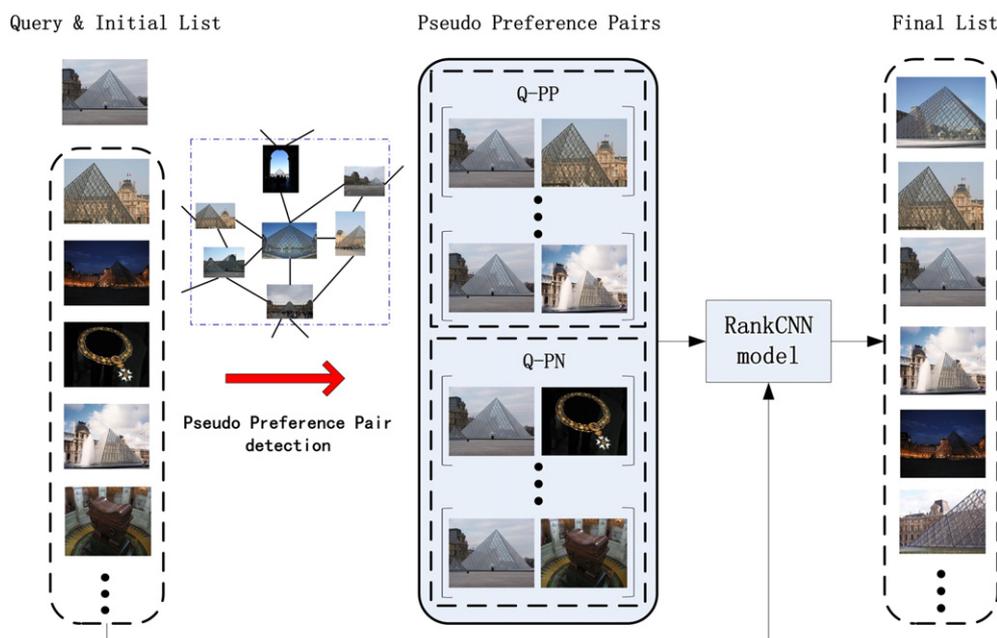


Fig. 2. The framework of our visual reranking.

resolution images. Fortunately, current GPUs, paired with a highly-optimized implementation of 2D convolution, are powerful enough to facilitate the training of interestingly-large CNNs. The most recent literature indicates that the CNNs have been successfully applied in some large datasets such as ImageNet, which contain 1.2 million high-resolution images [11].

2.3. Pseudo preference feedback

Labeled data is precondition for training. There are several methods proposed to detect an optimal set of pseudo preference pairs as the labeled data. [10] presents a learning-based approach to video search reranking by investigating the ranking order information. This is called pseudo preference feedback (PPF) as it does not rely on any user interaction. In this approach, considering that even relevant search results often have distinct appearances, all the samples in the initial ranked list are clustered into several categories in terms of visual appearance. These categories indicate the level of relevance. Thus the idea of PPF is to learn the preference relation in each category. In [10], the pairs of training samples include one pseudo-positive and one pseudo-negative sample (PP-PN), or two pseudo-positive samples (PP-PP). The Ranking Support Vector Machines (Ranking SVM) is used to learn a reranking model based on the selected pairs. The PPF-based method is fully automatic, without any auxiliary knowledge. It has been proven that this approach can effectively improve the performance of initial search result.

2.4. Gradient descent

Gradient descent is the most common method on solving minimization problem as follows.

$$P(w) = \frac{1}{n} \sum_{i=1}^n \Phi_i(w) + g(w)$$

Where ridge regression is obtained by setting $g(w) = \frac{\lambda}{2} \|w\|^2$ and $\Phi_i(w) = (w^T u_i - y_i)^2$. At the same time, the closely related methods are proposed such as stochastic gradient descent (SGD) [12–18], which has become popular for solving large scale supervised machine learning optimization problems such as SVM, due to their strong theoretical guarantees. Dual Coordinate Ascent (DCA) [19–21] solves the dual problem of the above equation. Specifically, for each i let $\phi_i^* : R \rightarrow R$ be convex conjugate of ϕ_i , namely, $\phi_i^* = \max_z (z^T u - \phi_i(z))$. The dual problem is $\max_{\alpha} D(\alpha)$

$$\text{where } D(\alpha) = \frac{1}{n} \sum_{i=1}^n -\phi_i^*(-\alpha_i) - g^*\left(\frac{1}{n} \sum_{i=1}^n \alpha_i\right).$$

Based on Dual Coordinate Ascent (DCA), a new technique called Stochastic Dual Coordinate Ascent (SDCA) [22] is presented to perform more efficient than SGD while keeping the strong theoretical guarantees. [23] considers an extension of SDCA under the minibatch setting that is often used in practice. This method not only provides a fast convergence rate for solving regularized loss minimization problems in machine, but also could be easily implemented over a parallel computing system.

In this research, we propose a novel unsupervised visual reranking method, termed rank via the convolutional neural networks (RankCNN). We integrate deep learning with pseudo preference feedback (See Section 3) into the initial ranking. CNN is then applied for the reranking. Accelerated Mini-Batch Stochastic Dual Coordinate Ascent is applied to accelerate the training for computationally affordable process (See Section 4).

3. Detection of pseudo preference pairs

In the framework, the pseudo preference pairs are detected using graph-based method instead of cluster-based. The pseudo preference pair includes query and a pseudo-negative sample (Q-PN), or query and a pseudo-positive sample (Q-PP). Let us first consider the detection of pseudo-positive sample.

Given that the perfect pseudo-positive samples are relevant to the query image, the detected pseudo-positive samples should be as precise as possible. The architecture of the previous work [9] is slightly modified to detect the pseudo-positive samples. Before detection, the database is translated into a weighted undirected graph $G = (W, V, E)$. It is reasonable to assume that the images containing the same view of the object being connected, thus the reciprocal neighbor relation [24,25] are used to weight the edge between images as follows:

$$w(i, i') = \begin{cases} \frac{N_k(i) \cap N_k(i')}{k} & \text{if } (i, i') \in R_k(i, i') \\ 0 & \text{else} \end{cases} \quad (1)$$

where $N_k(i)$ is the set of the k nearest neighbors of image i , and $R_k(i, i')$ is described as:

$$R_k(i, i') \square i' \in N_k(i) \wedge i \in N_k(i')$$

The object is to search a subgraph G' from G maintaining the maximum density in Eq. (2). Since there exists the direct or indirect reciprocal neighbor relation between query and other images in subgraph G' , we conclude that these images are pseudo-positive samples.

$$G' = \underset{G'=(W',V',E') \subset G, q \in V'}{\operatorname{argmax}} \frac{\sum_{(i,i')} w(i, i')}{|V'|} \quad (2)$$

The subgraph starts with query and other pseudo-positive samples are inserted into the subgraph successively. An approximate solution is adopted to solve the Eq. (2) as Algorithm 1.

Algorithm 1. Pseudo-positive samples detection

Input:

q , database $D = \{i_1, i_2, \dots, i_M\}$, parameter k , N_c

Initialization:

pseudo positive samples $I = \{q\}$,

edge nodes $E = \{\emptyset\}$,

reciprocal neighbor $R = \{r \in D \mid R_k(q, r)\}$,

outer nodes $O = \{D \setminus R\}$, $N_t = 1$

while $N_t \leq N_c$ **do**

$E \leftarrow \{E \cup R\}$

$$e = \operatorname{argmax}_{e \in E} \sum_{i \in I} w(i, e) \alpha^{\max(r_i, r_e)}$$

where α is a constant ranging from 0.5 to 0.8, and $r_i(r_e)$ is the initial ranking of $i(e)$.

$I \leftarrow \{I \cup \{e\}\}$

$E \leftarrow \{E \setminus \{e\}\}$

$R \leftarrow \{r \in O \mid R_k(e, r)\}$

$O \leftarrow \{O \setminus R\}$

$N_t ++$

end

Output: I

In experiment, the images from the complementary set of subgraph G' are sampled randomly as pseudo-negative samples.

4. Training RankCNN model

Assumed that the query has the same rank as the pseudo-positive sample (which here and below is written as $Q \approx P$), query would be higher than pseudo-negative sample ($Q > P$). Given the set of Q - PP and Q - PN , they can be described as $P(Q \approx P)$ and $P(Q > P)$. The training pairs need not to be complete, but all the pseudo preference pairs are supposed to maximize the above probability as far as possible.

We first consider the model $f: (R^d, R^d) \rightarrow R \in (0,1)$. More concretely, the model can be indicated as follows:

$$f(A, B) = \begin{cases} 1 & \text{if } A \in Q, B \in PN \\ 0 & \text{if } A \in Q, B \in PP \end{cases} \quad (3)$$

In this model, there exists an implicit function $h: R^d \rightarrow R$, which means that $h(A) > h(B)$ when $A > B$. In order to meet the above form, Eq. (3) is rewritten as follows:

$$f(A, B) = \frac{\exp(h(A)-h(B))}{1 + \exp(h(A)-h(B))} \quad (4)$$

In order to train the model maintaining maximal probability, the cost function is formulated as follows:

$$C_q = \alpha \sum_{d \in PN} (f(q, d)-1)^2 + (1-\alpha) \sum_{d \in PP} (f(q, d))^2 \quad (5)$$

where α is a coefficient weighting the pseudo preference pairs. Considering that the convolutional neural networks (CNN) have been proven successfully in visual classification, the CNN could be integrated with RankNet as the presentation of the function f . Similar to CNN, the system starts with image pixels as input, followed by three alternative convolutional layers and max pooling layers. The difference is that the input is an image pair and the last layer is only one neuron which serves as the output of C_q .

In our experiments, the image is transformed into grayscale image size of 128×128 . The first layer (Conv1) has 32 kernel size of 5×5 . The next layer (Pool1) has pooling size of 3×3 . The third layer (Conv2) has 32 kernel size of 5×5 . The fourth layer (Pool2) has pooling size of 3×3 . The fifth layer (Conv3) has 64 kernel size of 5×5 . The sixth layer (Pool3) has pooling size of 3×3 . The architecture of the RankCNN is shown in Fig. 3.

According to Eq. (5), cost function is convex function over g and h . In addition, the CNN reminds us of the existing methodologies on classification which can be directly applied, such as back propagation. In general, the cost function can be solved by using gradient descent method. Denote all the parameters in the model by $\theta = \{w_i, b_j; i = 1 \dots m, j = 1 \dots n\}$, we take the derivatives of cost function with respect to the parameters as:

$$\frac{\partial C_q}{\partial \theta} = 2\alpha \sum_{d \in PN} (f(q, d)-1) \frac{\partial f(q, d)}{\partial \theta} + 2(1-\alpha) \sum_{d \in PP} f(q, d) \frac{\partial f(q, d)}{\partial \theta} \quad (6)$$

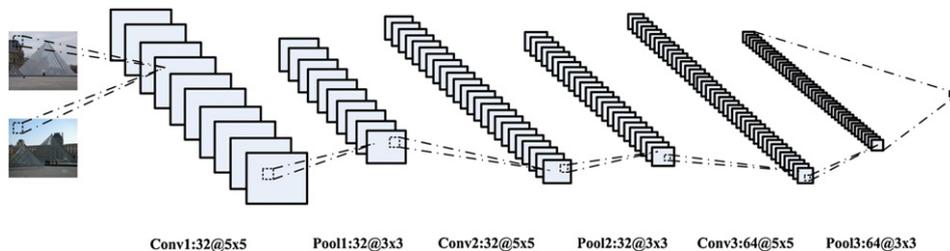


Fig. 3. An illustration of the RankCNN model.

$$\frac{\partial f(q, d)}{\partial \theta} = \frac{\exp(h(q)-h(d)) \left(\frac{\partial h(q)}{\partial \theta} - \frac{\partial h(d)}{\partial \theta} \right)}{(1 + \exp(h(q)-h(d)))^2} \quad (7)$$

Because the derivatives of h are similar to back propagation, interested readers please refer to CNN for details [5]. In order to avoid trained model being biased toward queries with more document pairs, the pseudo positive samples are divided into several batches. At each iteration, one batch is selected randomly as pseudo positive input and the pseudo negative samples are sampled as 5 times as that of pseudo positive input. Pseudo code for training the RankCNN model is given in Algorithm 2.

Algorithm 2. Training the RankCNN

```

Input:
pseudo positive set  $PP = \{p_1, p_2, \dots, p_M\}$ ,
pseudo negative set  $PN = \{n_1, n_2, \dots, n_N\}$ ,
 $q, Iter, Trial$ 
Initialization:
Set  $i=0$ 
while  $i \leq Iter$  do
     $i=i+1$ 
    Set  $t=0$ 
    while  $t \leq Trial$  do
         $t=t+1$ 
        Pick subset  $I \subset \{1, \dots, N\}$  and  $J \subset \{1, \dots, M\}$ 
        randomly. Construct the pseudo preference
        pairs with  $q$ 
        Make a gradient step to minimize Eqn.5
    end
    Update  $\theta$ 
end
Output:  $\theta$ 
    
```

Though the above algorithm could be solved successfully by the Stochastic Gradient Descent (SGD), the promise of a recently proposed Accelerated Mini-Batch Stochastic Dual Coordinate Ascent (ASDCA) motivates us to adopt ASDCA into our framework. Specifically, our goal is to solve $\min_{x \in R^d} P(\theta)$ where

$$P(\theta) = \frac{1}{N} \sum_{i=1}^N \phi_i(\theta) + \frac{1}{M} \sum_{j=1}^M \varphi_j(\theta) + g(\theta) \quad (8)$$

$$\begin{aligned} \phi_i(\theta) &= (f(q, d_i; \theta) - 1)^2 & d_i \in PN \\ \varphi_j(\theta) &= (f(q, d_j; \theta))^2 & d_j \in PP \end{aligned}$$

Similarly, the dual problem is to $\max_{\alpha, \beta} D(\alpha, \beta)$

$$\text{where } D(\alpha, \beta) = \frac{1}{N} \sum_{i=1}^N -\phi_i^*(-\alpha_i) + \frac{1}{M} \sum_{j=1}^M -\varphi_j^*(-\beta_j) - g^* \left(\frac{1}{N} \sum_{i=1}^N \alpha_i + \frac{1}{M} \sum_{j=1}^M \beta_j \right).$$

Algorithm 3 lists the pseudo code of the ASCDA on the proposed RankCNN model.

Algorithm 3. Accelerated training the RankCNN

Input:

pseudo positive set $PP = \{p_1, p_2, \dots, p_M\}$,
 pseudo negative set $PN = \{n_1, n_2, \dots, n_N\}$,
 $q, Iter, \gamma \in [0, 1]$

Initialization:

Set $t=0, \theta^{(0)} = 0$
 $\alpha_1^{(0)} = \alpha_2^{(0)} = \dots = \alpha_N^{(0)} = \bar{\alpha}^{(0)} = 0,$
 $\beta_1^{(0)} = \beta_2^{(0)} = \dots = \beta_M^{(0)} = \bar{\beta}^{(0)} = 0$

while $t \leq Iter$ **do**

$$u^{(t-1)} = (1 - \gamma)\theta^{(t-1)} + \gamma \nabla g^*(\bar{\alpha}^{(t-1)} + \bar{\beta}^{(t-1)})$$

Pick subset $I \subset \{1, \dots, N\}$ and $J \subset \{1, \dots, M\}$ randomly. Then update the dual variables in I and J

$$\alpha_i^{(t)} = (1 - \gamma)\alpha_i^{(t-1)} + \gamma \nabla \phi_i(u^{(t-1)}) \text{ for } i \in I$$

$$\beta_j^{(t)} = (1 - \gamma)\beta_j^{(t-1)} + \gamma \nabla \varphi_j(u^{(t-1)}) \text{ for } j \in J$$

$$\bar{\alpha}^{(t)} = \bar{\alpha}^{(t-1)} + N^{-1} \sum_{i \in I} (\alpha_i^{(t)} - \alpha_i^{(t-1)})$$

$$\bar{\beta}^{(t)} = \bar{\beta}^{(t-1)} + M^{-1} \sum_{j \in J} (\beta_j^{(t)} - \beta_j^{(t-1)})$$

$$\theta^{(t)} = (1 - \gamma)\theta^{(t-1)} + \gamma \nabla g^*(\bar{\alpha}^{(t)} + \bar{\beta}^{(t)})$$

end

Output: θ

After learning the model, the parameters of the RankCNN model are obtained. Each image in the initial list is taken as input of h , and then is ranked according to the output in descending order.

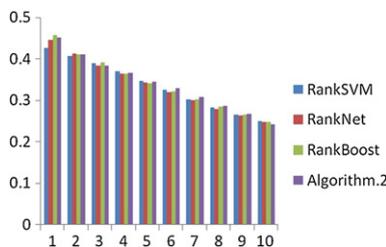


Fig. 4. The $p@n$ on LETOR4.0 (MQ2008) for various methods.

Table 1
The MAP (in %) on the Paris dataset.

INIT	QEB	MWD	Rank CNN
56.38	63.19	70.52	71.47

5. Experiment

5.1. Dataset

Before introducing the system in visual field, we first test Algorithm 2 in text retrieval. LETOR 4.0 dataset is chosen as the benchmark.

Following the same experimental setting as in [26], we evaluate the ability of visual reranking on the two databases – only the Paris database, and the Paris + Francelandmark. Francelandmark include some images crawled from Flickr, Bing and Google using queries of famous 78 France landmarks and 24 artworks.

The LETOR 4.0 [27] contains 8 datasets for four ranking settings derived from the two query sets and the Gov2 web page collection. The 5-fold cross validation strategy is adopted and the 5-fold partitions are included in the package. In each fold, there are three subsets for learning: training set, validation set and testing set. The precision at top n ($p@n$) is taken as measurement.

The Paris [28] includes 6391 images collected from Flickr by searching for particular Paris landmarks. There are 55 images extracted from dataset as the query. The retrieval performance is measured by mAP(mean Average Precision). In addition, $p@n$ is also taken as the measurement of ranking ability.

The Francelandmark contains 86,717 images in total, which get closer to the authentic application. The performance is evaluated by the precision at top n candidates. We choose two groups of queries to simulate the real conditions:

- Low Precision (LP): 25 queries where the precision at top 25 candidates is lower than 30%.
- High Precision (HP): 25 queries where the precision at top 25 candidates is high than 70%.

5.2. Comparison methods

In text retrieval, our algorithm is slightly modified. The documents with the largest relevance label are taken as queries. The pseudo

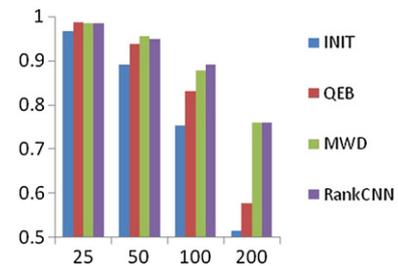


Fig. 5. The $p@n$ on the Paris dataset for various methods.

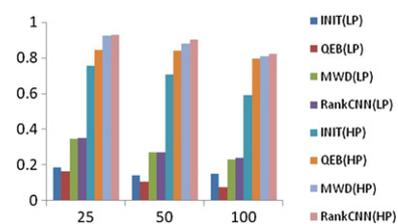


Fig. 6. The $p@n$ on the Francelandmark dataset.

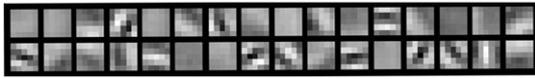


Fig. 7. The kernels of the first convolutional layers.

positive set and pseudo negative set come from the documents with the different relevance label. Because our learning algorithm is based on pairwise model, we choose the RankSVM, RankNet, and Rankboost as the comparative test.

In visual retrieval, the initial result is generated as follows: the Harris Laplace and SIFT is taken as the detector and descriptor respectively. Next, 40 M descriptors are selected randomly as training set of 1 M codebook, which is generated using Approximate K-means (AKM). All images are represented by the BoW vector, and are ranked according to the negative Euclidean distance between these images and query in descending order. In order to evaluate the reranking performance, several existing reranking methods are compared as follows.

5.2.1. Query Expansion Baseline (QEB)

This method [29] assumes the high-rank images in the initial list to be relevant. The images are ranked according to their average similarities with the top candidates.

5.2.2. Maximizing Weighted Density (MWD)

This method [10] reranks the images according to their time of insertion into subgraph while maximizing weighted density.

5.2.3. RankCNN

This method is introduced generally in the above sections. The RankCNN is built based on a Python Library Theano [30].

5.3. Evaluation

5.3.1. LETOR 4.0

In this dataset, the performance is shown as follows:

As shown in Fig. 4, compared with other text retrieval methods, Algorithm 2 constructed randomly pseudo preference pairs to minimize the cost function, which could efficiently solve the problem that the number of generated document pairs varies largely from query to query. Thus the introduction of (Q-PN) contributes to obtaining the more robust model than other pairwise approaches which only concern about positive–negative pairs.

5.3.2. Paris

Table 1 and Fig. 5 show the results for the Paris dataset, the performances of all methods are better against the initial rank.

5.3.3. Francelandmark

The performance of several methods on different initial list is shown in Fig. 6.

Figs. 5 and 6 demonstrate that RankCNN has been successfully proven in visual reranking. The QEB could provide better reranking results when the initial list has high precision on the top candidates, but it suffers when the high-ranking images include much false positive. MWD outperforms QEB, however it encounters low recalling problem

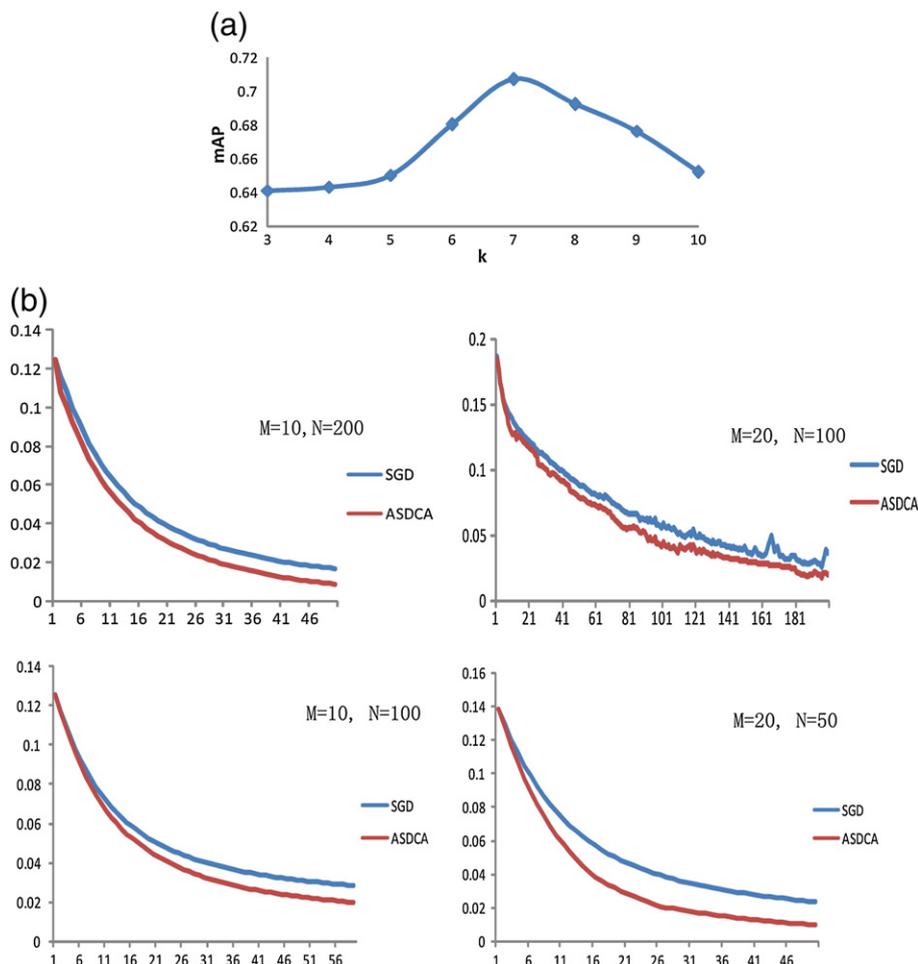


Fig. 8. (a) The mAP curves of RankCNN in terms of the parameter k on the Paris. (b) The convergence curve of SGD and ASDCA with different number of pseudo positive and pseudo negative sets.



Fig. 9. The query is in the red box; the top 20 candidates are shown in the initial result (in the blue box) and the refine result (in the black box). From top to bottom: The Louvre (a), Louvre (b), Eiffel Tower, and triumphal arch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for certain query. RankCNN is the best performers on *Francelandmark* dataset.

In Fig. 7 32 convolutional kernel size of 5×5 learned by the first convolutional layers are shown. From the pattern of these images, they are similar with the kernels of the CNN. They both indicate the depiction of the texture of images.

We note in RankCNN, parameters k and N_c would impact the performance of the algorithm. The size of k controls is the strict degree about choosing the reciprocal neighbors. If the reciprocal neighbors are selected too rigorously, the image and its reciprocal neighbors look pretty much the same leading to lose the augmentability. On the other hand, if k is set to be too loose, the algorithm may render large number of noisy neighbor. In terms of N_c , its role is similar with k , and the difference is that N_c is related to sufficiency and precision of the confident samples. To fully explore the applicability of RankCNN on visual reranking, additional experiments are conducted to access the impact of the parameter setting on the performance of RankCNN. Fig. 8(a) shows that the parameters k and N_c have great influence on the performance of proposed RankCNN.

Fig. 8(b) illustrates that when solving the loss minimization problems, the ASDCA outperforms the SGD in our RankCNN model. At the same time, it is found that ASDCA has more smooth convergence curve than SGD. This improvement could be explained in the following aspects. First, the ASDCA absorbs the benefit of SGD by using random minibatch. In this way, the processing time of a mini-batch of size m is much smaller than m times the processing time of one example (mini-batch of size 1). In the practical training of neural networks with SGD or ASDCA, it is more efficient to perform matrix–matrix multiplications over a mini-batch than an equivalent amount of matrix–vector multiplication operations. This benefit is notable especially when GPU is used. Second, as stated in [22] said, the convergence rate of ASDCA is significantly better than that of the SGD because the ASDCA has a clear stopping criterion and it does not tend to be too aggressive at the beginning of the optimization process, especially when λ is very small. Generally, the convergence of ASDCA becomes faster when we are interested in more accurate solutions while SGD reaches a moderate accuracy quite fast. According to Fig. 6, when the number of pseudo positive set and pseudo negative set is 20 and 50 respectively, the solution is most accurate.

Fig. 9 shows the initial result and refined result for different query. The top 20 candidates are only given in score order. As for the ‘Louvre (a)’ and ‘Louvre (b)’, they produce two totally different initial results. In the query ‘Louvre (a)’, the precision of top 20 candidates is less than 25% mainly because of the noisy descriptors from query. Our reranking method could model the latent target. As a result, the refined outcome is accurate in the top 20 candidates while the top one is not the most similar to the query. In the query ‘Louvre (b)’, the precision of top 20 candidates are higher than 75%. According to the refined result, the rerank method still work. In the query ‘Eiffel Tower’, due to the cluster of text, the initial result includes several false positives. In the query ‘triumphal arch’, the affined deformation makes it difficult for image retrieval. Still, we observe that the search target could be detected accurately.

6. Conclusion

In this paper, an unsupervised learning-based visual reranking approach is proposed, where the training pairs are obtained by pseudo preference feedback. From the experiment, pseudo preference feedback lays the groundwork for our unsupervised learning, and RankCNN model has been proven successful for different datasets. Moreover, it broadens the outlook of visual reranking by integrating the pseudo preference feedback with class-based approach. In future work, more sophisticated model would be explored to implement efficient online learning.

Acknowledgments

This work is sponsored by collaborative Research Project (SEV011 00474) between Beijing University of Posts and Telecommunications and

France Telecom R&D, and the National High Technology Research and Development Program of China (863 Program, No.2012AA012505).

References

- [1] W. Xing-yuan, C. Zhi-feng, Y. Jiao-jiao, An effective method for color image retrieval based on texture, *Comput. Stand. Interfaces* 34 (1) (January 2012) 31–35.
- [2] C. Zhe, Q. Tao, L. Tie-Yan, T. Ming-Feng, L. Hang, Learning to Rank: From Pairwise Approach to Listwise Approach, *Proc. ICML, NY, USA, 2007*, pp. 129–136.
- [3] R. Herbrich, T. Graepel, K. Obermayer, Support Vector Learning for Ordinal Regression, *Proc. ICANN, 1999*, pp. 192–201.
- [4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to Rank Using Gradient Descent, *Proc. ICML, NY, USA, 2005*, pp. 89–96.
- [5] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preference, *J. Mach. Learn. Res.* (2003) 933–969.
- [6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (November 1998) 2278–2324.
- [7] <http://yann.lecun.com/exdb/mnist/>.
- [8] D. Ciresan, U. Meier, J. Schmidhuber, Multi-Column Deep Neural Networks for Image Classification, *Arxiv preprint arXiv:1202.2745* 2012.
- [9] L. Yuan, M. Tao, H. Xian-Sheng, T. Jinhui, W. Xiuqing, L. Shipeng, Learning to Video Search Rerank via Pseudo Preference Feedback, *ICME2008*. 297–300.
- [10] Shaoting Zhang, Ming Yang, Timothee Cour, Kai Yu, Dimitris N. Metaxas, Query Specific Fusion for Image Retrieval, *Proc. ECCV, Florence, Italy, 2012*, pp. 660–673.
- [11] A. Krizhevshy, I. Sutskever, G.E. Hinton, Imagenet Classification With Deep Convolutional Neural Networks, *Proc. NIPS 2012, Lake Tahoe, Nevada, USA, 2012*.
- [12] H. Robbins, S. Monro, A stochastic approximation method, *Ann. Math. Stat.* (1951) 400–407.
- [13] N. Murata, A Statistical Study of On-Line Learning, *Online Learning and Neural Networks*, Cambridge University Press, Cambridge, UK, 1998.
- [14] L.B.Y. Le Cun, Large Scale Online Learning, *Advances in Neural Information Processing Systems 16, Proceedings of the 2003 Conference*, vol. 16, MIT Press, 2004, p. 217.
- [15] T. Zhang, Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms, *Proceedings of the Twenty-First International Conference on Machine Learning, 2004*.
- [16] L. Bottou, O. Bousquet, The Tradeoffs of Large Scale Learning, *NIPS, 2008*, pp. 161–168.
- [17] S. Shalev-Shwartz, Y. Singer, N. Srebro, Pegasos: Primal Estimated sub-Gradient Solver for SVM, *ICML, 2007*, pp. 807–814.
- [18] M. Riedmiller, Advanced supervised learning in multi-layer perceptrons – from backpropagation to adaptive learning algorithms, *Comput. Stand. Interfaces* 16 (3) (July 1994) 265–278.
- [19] M. Lou Padgett, W.J. Karplus, S. Deiss, R. Shelton, Computational intelligence standards: motivation, current activities and progress, *Comput. Stand. Interfaces* 16 (3) (July 1994) 185–203.
- [20] J.C. Platt, Fast training of Support Vector Machines Using Sequential Minimal Optimization, in: B. Schölkopf, C. Burges, A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.
- [21] T. Joachims, Making Large-Scale Support Vector Machine Learning Practical, in: B. Schölkopf, C. Burges, A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.
- [22] Shai Shalev-Shwartz, Tong Zhang, Stochastic dual coordinate ascent methods for regularized loss minimization, *J. Mach. Learn. Res.* 14 (Feb 2013) 567–599.
- [23] Shai Shalev-Shwartz, Tong Zhang, Accelerated mini-batch stochastic dual coordinate ascent, *arXiv:1305.2581* 2013.
- [24] C. Schmid, H. Harzallah, J. Verbeek, Accurate Image Search Using the Contextual Dissimilarity Measure, *IEEE Transactions, PAMI, Jan. 2010*.
- [25] Qin Danfeng, Stephan Gammeter, Lukas Bossard, Till Quack, Luc VanGool, Hello Neighbor: Accurate Object Retrieval With k-Reciprocal Nearest Neighbors, *Proc. CVPR, CO, USA, 2011*, pp. 777–784.
- [26] H. Chong, D. Yuan, B. Hongliang, W. Lezi, Z. Nan, C. Shusheng, Z. Jian, An Efficient Graph-Based Visual Reranking, *ICASSP, YVR, CA2013*.
- [27] <http://research.microsoft.com/en-us/um/beijing/projects/letor/>.
- [28] <http://www.robots.ox.ac.uk/vgg/data/parisbuildings>.
- [29] J. Sivic, A. Zisserman, A Text Retrieval Approach to Object Matching in Videos, *Proc. ICCV, Nice, FR, 2003*, pp. 1470–1477.
- [30] <http://deeplearning.net/software/theano/>.



Dong Yuan is an associate professor at Beijing University of Posts and Telecommunications, China. He is also invited as “France Telecom – Orange Expert on Solution of Content Service” of France Telecom R&D Global. He received his Ph.D degree in Shanghai Jiao Tong University at 1999, worked as R&D scientist at Nokia Research Center China from 1999 to 2001, worked as post doctoral research staff at Engineering Department Cambridge University UK from 2001 to 2003. His current research interests include semantic video indexing, video copy detection, and multimedia content search.



Huang Chong is a postgraduate student at Beijing University of Posts and Telecommunications, China. He received the B.S. degree in Information Engineering in Beijing University of Posts and Telecommunications at 2011. He is also an intern in the France Telecom R&D Beijing since 2011. His current research interests are copy detection, images retrieval, images classification and deep learning.



Liu Wei is an employee of France Telecom R&D Beijing as a researcher. He received his Master's degree in Beijing University of Posts and Telecommunication at 2011. His current research interests include video copy detection, face detection, facial landmark detection and face recognition.