

Learning Query and Image Similarities with Ranking Canonical Correlation Analysis

Ting Yao [†], Tao Mei [†], and Chong-Wah Ngo [‡]

[†] Microsoft Research, Beijing, China

[‡] City University of Hong Kong, Kowloon, Hong Kong

{tiyao, tmei}@microsoft.com, cscwngo@cityu.edu.hk

Abstract

One of the fundamental problems in image search is to learn the ranking functions, i.e., similarity between the query and image. The research on this topic has evolved through two paradigms: feature-based vector model and image ranker learning. The former relies on the image surrounding texts, while the latter learns a ranker based on human labeled query-image pairs. Each of the paradigms has its own limitation. The vector model is sensitive to the quality of text descriptions, and the learning paradigm is difficult to be scaled up as human labeling is always too expensive to obtain. We demonstrate in this paper that the above two limitations can be well mitigated by jointly exploring subspace learning and the use of click-through data. Specifically, we propose a novel Ranking Canonical Correlation Analysis (RCCA) for learning query and image similarities. RCCA initially finds a common subspace between query and image views by maximizing their correlations, and further simultaneously learns a bilinear query-image similarity function and adjusts the subspace to preserve the preference relations implicit in the click-through data. Once the subspace is finalized, query-image similarity can be computed by the bilinear similarity function on their mappings in this subspace. On a large-scale click-based image dataset with 11.7 million queries and one million images, RCCA is shown to be powerful for image search with superior performance over several state-of-the-art methods on both keyword-based and query-by-example tasks.

1. Introduction

Similarity function plays a key role in Web image search. Given a textual query, the objective is to retrieve the most relevant images and rank them by their degrees of relevance to the query. The relevance between the query and image can be viewed as a kind of similarity.

As textual queries and images are of two different views,

they cannot be directly compared. As a result, existing search engines to date highly rely on the surrounding texts associated with images. The similarity between a query and an image is then defined based on their textual feature vectors. The relevance models, including Vector Space Model [24], BM25 [21], and Language Models [28], can all be used as similarity functions. However, the text description may not precisely describe salient visual content, not to mention that some images do not even associate with any text. Consequently, the similarity from the feature-based vector model may suffer from robustness problem. Another solution of similarity measure is to learn image rankers on query-image pairs which are usually labeled by human experts. However, human labeling is always too expensive to obtain, making it hard to scale up. Even so called “experts” often find it hard to judge query-image relevance, resulting in noisy labeled training data.

Our similarity learning method addresses the aforementioned two issues. First, we consider the cross-view (i.e., text to image) similarity by learning a common latent subspace that allows direct comparison of textual queries and visual images in a low-dimensional space. The image representations are visual features extracted directly from the images, rather than textual features. By learning two linear mappings, the similarity between queries and images in the original two incomparable different spaces can be directly computed in the shared subspace. Moreover, the dimensionality of the latent subspace is significantly reduced compared with that of any original views, leading to saving in memory cost for search systems.

Second, the click-through data, which can be viewed as the footprints of user searching behavior, is explored as an effective means of understanding both the query and the user’s intent for image search [12]. As most image search engines display results as thumbnails, the user can browse the entire image search results before clicking on a specific image. As such, users predominantly tend to click on images that are relevant to their query. Therefore, the click-through data can serve as a reliable and implicit feedback for im-

age search. More importantly, *relative* similarity between different images and a common query is also manifested in the click-through data, which is further taken into account to learn similarity function here.

By jointly integrating subspace learning and click-through data, this paper presents a novel Ranking Canonical Correlation Analysis (RCCA) approach for similarity learning, as shown in Figure 1. Specifically, a bipartite graph between queries and images is constructed based on the image click-through data from a real image search engine. The query and image spaces are then formed, where a corresponding link between a query and an image is established, if the users who issue the query clicked the image. Next, Canonical Correlation Analysis (CCA) is performed for mapping the two views, represented by visual and textual features, into a common subspace where the correlation between the two views is maximized. Furthermore, as click-through data conveys relative relevance judgements indicated by different click counts on images in response to an identical query, a bilinear similarity function is learnt simultaneously while the subspace is fine tuned to respect these preference relations. Finally, the query and image similarity is measured by this bilinear similarity function. It is also worth noticing that the image-image and query-query similarities can be defined as dot products on the final subspace.

The remaining sections are organized as follows. Section 2 describes related work on similarity learning. Section 3 presents our ranking canonical correlation analysis similarity learning method. Section 4 provides empirical evaluations, followed by the conclusions in Section 5.

2. Related Work

Similarity learning is a fundamental problem in Web search and information retrieval. The research in this direction has proceeded along two dimensions: feature-based [2, 13, 20, 29] and learning-based [1, 7, 10, 18, 19, 23, 26].

Feature-based methods make use of features extracted from objects to measure similarity. Vector Space Model [24], BM25 [21], and Language Models [28] are three classical retrieval models for computing query-document similarity on term or n-gram feature vectors. In [29], queries are represented as n-grams and then the cosine similarity is utilized as the similarity function. Similar in spirit, Broder *et al.* proposed calculating query similarity with term and n-gram features enriched with a taxonomy of semantic classes [2]. Moreover, recent works on image representations are by encoding local descriptors, such as the vector of locally aggregated descriptors (VLAD) [13] and Fisher vector (FV) [20]. Any standard distance computed on the representations is further considered as image similarity.

Different from feature-based methods, learning-based approaches aim to directly learn the similarity between pairs of objects, particularly, in a shared common sub-

space. Canonical correlation analysis (CCA) [10], a classical and successful technique, explores the mapping matrices by maximizing the correlation between the projections in the subspace. As a nonlinear extension of CCA, Kernel CCA (KCCA) is to provide nonlinear mappings such that the correlation between two objects is maximized [7]. An alternative scheme to KCCA is Kernel Principal Component Analysis with CCA (KPCA-CCA), which was proposed by Nakayama *et al.* in [18]. Instead of directly learning the nonlinear mappings in KCCA, KPCA-CCA embeds the nonlinear metrics via KPCA and generates the new input for CCA. Recently, Gong *et al.* further incorporated a third view in CCA framework by minimizing the distances in the resulting common space between each pair of views of the same data [8]. Similarly, Partial Least Squares (PLS) also aims to model the relations between two or more sets of data by projecting them into the latent subspace [23]. The difference between CCA and PLS is that CCA utilizes cosine as the similarity function while PLS learns dot product. Later in [1], polynomial semantic indexing (PSI) is performed by learning two low-rank mapping matrices in a learning to rank framework, and then a polynomial model is considered to measure the relevance between query and document.

In addition, by further leveraging the click-through data for similarity learning, Wu *et al.* extended the PLS to Multi-view PLS by combining multiple features for learning query-document similarity on a click-through bipartite graph [26]. In another work by Yao *et al.* [27], by combining click-through and video document features for deriving a latent subspace, the dot product of the mappings in the latent subspace is taken as the similarity between videos. Recently, Pan *et al.* formulated image search as a click-through-based cross-view problem by learning a common subspace, in which the l_2 distance between query and image mappings is minimized and the structures in original spaces are preserved [19].

Our work belongs to learning-based similarity approaches. Different from the aforementioned learning-based works, our proposed method integrates the learning of the shared subspace and the bilinear similarity defined in the subspace simultaneously, which we show can better measure the similarity between views.

3. Similarity Learning From Click-through

The basic idea of this work is to facilitate similarity learning between query and image from click-through data by constructing a common latent subspace. In this way, the original incomparable textual query and visual image could be directly compared in the shared subspace. Moreover, the learning of the subspace and bilinear similarity in the subspace is integrated into an overall optimization problem simultaneously to better reflect the preference relations implicit in the click-through data. After we obtain the la-

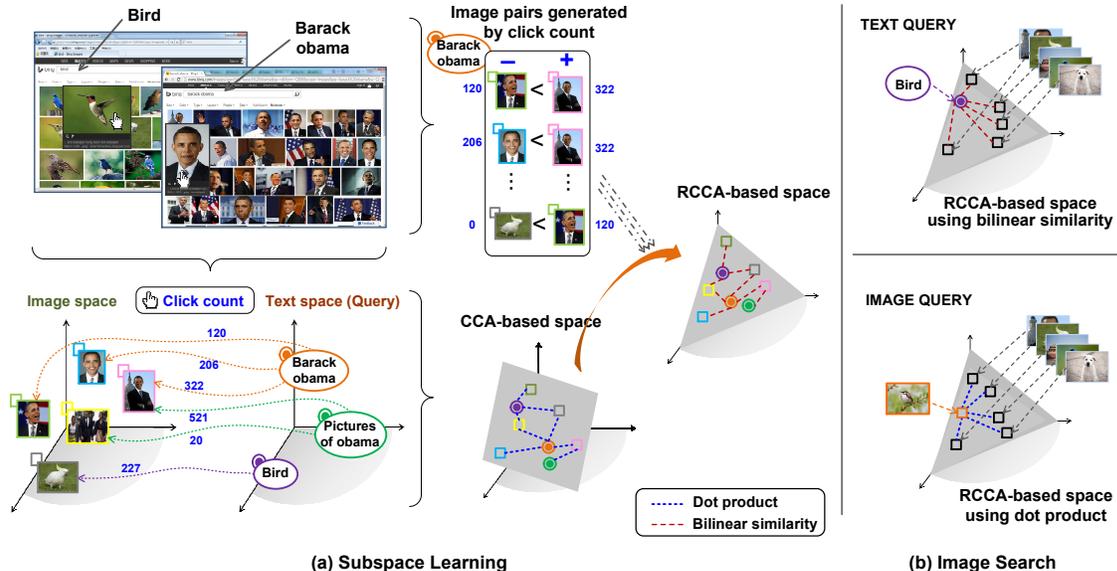


Figure 1. Ranking canonical correlation analysis based image search framework. (a) Latent subspace learning between textual query and visual image spaces. A CCA-based space is first learnt on the query-image correspondences exist in the click-through bipartite graph extracted from image search logs. Then a RCCA-based space is formed to adjust the CCA-based space to preserve the preference relations implicit in the click-through data, and simultaneously a bilinear similarity function is learnt to measure the query-image similarity in this space. (b) With the learnt RCCA-based space, keyword-based and query-by-example image search tasks can be directly implemented on the similarities measured between the projections on this subspace. For better viewing, please see original color pdf file.

tent subspace and bilinear similarity function, the similarity between query and image is then measured by using this bilinear function on their mappings. The approach overview is demonstrated in Figure 1.

We begin this Section by presenting the click-through bipartite graph that naturally encodes user actions in the query log, followed by the learning of an initial subspace using standard canonical correlation analysis (CCA). Then our ranking canonical correlation analysis (RCCA) is proposed by simultaneously adjusting the initial subspace and learning a bilinear similarity function in the subspace, so as to model the preference relations in the click-through data. Finally, the algorithm for both keyword-based and query-by-example image search is presented.

3.1. Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a click-through bipartite graph. $\mathcal{V} = Q \cup V$ is the set of vertices, which consists of a query set Q and an image set V . \mathcal{E} is the set of edges between the query and image vertices. The number associated with an edge represents the number of times that an image is clicked given a query. Suppose there are n triads $\{q_i, v_i, c_i\}_{i=1}^n$ generated from the click-through bipartite in total, where c_i is the click counts of image v_i in response to query q_i . Let $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}^T \in \mathbb{R}^{n \times d_q}$ and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}^T \in \mathbb{R}^{n \times d_v}$ denote the query and image feature matrix, where \mathbf{q}_i and \mathbf{v}_i are the textual and visual feature of query q_i and image v_i , and d_q and d_v are the feature dimensionality, respectively.

3.2. Canonical Correlation Analysis

From the click-through bipartite, each query establishes a link with each clicked image, which makes a natural correspondence between query and image spaces. The similarity in between, nevertheless, could not be directly computed since the representations of query and image are absolutely heterogenous. One solution, that we pursue in this work, is to rely on subspace learning, which assumes that a low-dimensional common subspace exists for the representations of query and image. Subspace learning methods typically produce linear transformations which are easy to implement and deploy. Inspired by the effectiveness of CCA [10] in cross-modal retrieval or multi-view embedding [8], we choose it to learn a shared common subspace between query and image spaces.

CCA is a technique for learning a shared subspace which reflects the correlations between the heterogeneous representations across two (or more) original spaces. The linear mapping function can be derived from this subspace by

$$f(\mathbf{q}_i) = \mathbf{q}_i \mathbf{W}_q^0, \text{ and } f(\mathbf{v}_i) = \mathbf{v}_i \mathbf{W}_v^0, \quad (1)$$

where d is the dimensionality of the common subspace, and $\mathbf{W}_q^0 \in \mathbb{R}^{d_q \times d}$ and $\mathbf{W}_v^0 \in \mathbb{R}^{d_v \times d}$ are the transformation matrices that project the query textual semantics and image content into the common subspace, respectively.

To learn the two linear projections \mathbf{W}_q^0 and \mathbf{W}_v^0 , the objective of CCA is to make $(\mathbf{Q} \mathbf{W}_q^0, \mathbf{V} \mathbf{W}_v^0)$ maximally cor-

related as

$$\begin{aligned} (\mathbf{W}_q^0, \mathbf{W}_v^0) &= \operatorname{argmax}_{\mathbf{w}_q^0, \mathbf{w}_v^0} \operatorname{corr}(\mathbf{Q}\mathbf{w}_q^0, \mathbf{V}\mathbf{w}_v^0) \\ &= \operatorname{argmax}_{\mathbf{w}_q^0, \mathbf{w}_v^0} \frac{\langle \mathbf{Q}\mathbf{w}_q^0, \mathbf{V}\mathbf{w}_v^0 \rangle}{\|\mathbf{Q}\mathbf{w}_q^0\| \|\mathbf{V}\mathbf{w}_v^0\|}. \end{aligned} \quad (2)$$

Let \mathbf{C}_{QQ} and \mathbf{C}_{VV} represent the empirical covariance matrices for query and image space respectively, while let \mathbf{C}_{QV} denote the cross-covariance. The above optimization problem can be rewritten as

$$(\mathbf{W}_q^0, \mathbf{W}_v^0) = \operatorname{argmax}_{\mathbf{w}_q^0, \mathbf{w}_v^0} \frac{\|\mathbf{W}_q^{0\top} \mathbf{C}_{QV} \mathbf{W}_v^0\|}{\sqrt{\|\mathbf{W}_q^{0\top} \mathbf{C}_{QQ} \mathbf{W}_q^0\| \|\mathbf{W}_v^{0\top} \mathbf{C}_{VV} \mathbf{W}_v^0\|}}. \quad (3)$$

The optimization of Eq.(3) can be solved by the following generalized eigenvalue problem

$$\begin{pmatrix} \mathbf{Q}^\top \mathbf{Q} & \mathbf{Q}^\top \mathbf{V} \\ \mathbf{V}^\top \mathbf{Q} & \mathbf{V}^\top \mathbf{V} \end{pmatrix} \begin{pmatrix} w_q^0 \\ w_v^0 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{Q}^\top \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^\top \mathbf{V} \end{pmatrix} \begin{pmatrix} w_q^0 \\ w_v^0 \end{pmatrix}, \quad (4)$$

where w_q^0 and w_v^0 is a column of \mathbf{W}_q^0 and \mathbf{W}_v^0 , respectively. The size of this problem is $(d_q + d_v) \times (d_q + d_v)$.

In order to obtain a d -dimensional subspace, we form the projection matrices \mathbf{W}_q^0 and \mathbf{W}_v^0 from the top d eigenvectors corresponding to each w_q^0 and w_v^0 , respectively. Once the projection matrices are learnt, the respective projection matrix is applied to each original space separately.

3.3. Ranking Canonical Correlation Analysis

The correspondence of each query and image pair linked in click-through bipartite is considered equally in learning CCA, regardless of different image click counts in response to an identical query. On the other hand, the preference relations like ‘‘for query q , image v_a should be ranked higher than image v_b ,’’ are conveyed in the click-through that image v_a receives higher click counts than v_b in answering the query q and have been proved to be effective in learning search functions [5, 14]. Inspired by the idea of jointly learning the subspace and decision function in classification [16], we develop a ranking canonical correlation analysis (RCCA) model, which aims to adjust the subspace learnt by CCA to further preserve the preference relations implicit in the click data. Moreover, to better reflect the relative preference relations in terms of query-image similarity, a bilinear similarity function is simultaneously learnt. With this, query-image similarity can be calculated by the bilinear similarity function on their mappings in the subspace.

Formally, given a query q and an image v , we wish to learn two mappings, which can map query q from query space and image v from image space into a common latent space. Meanwhile, a query-image similarity function $s(q, v, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W})$ is presented to measure the similarity of v given q in the latent space. We consider a parametric similarity function that has a bilinear form as

$$s(q, v, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) = (q\mathbf{W}_q)\mathbf{W}(v\mathbf{W}_v)^\top, \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the bilinear similarity matrix, $\mathbf{W}_q \in \mathbb{R}^{d_q \times d}$ and $\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$ are the transformation matrices that project the query and image space into the common subspace, respectively.

The overall objective function of our RCCA is as

$$\begin{aligned} \operatorname{argmin}_{\mathbf{W}_q, \mathbf{W}_v, \mathbf{W}} \mathcal{L}(s(q, v, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W})) + \frac{\mu}{2} \|\mathbf{W}\|^2 \\ + \left(\frac{\gamma}{2} \|\mathbf{W}_q - \mathbf{W}_q^0\|^2 + \frac{\eta}{2} \|\mathbf{W}_v - \mathbf{W}_v^0\|^2 \right), \end{aligned} \quad (6)$$

where $\mathcal{L}(\bullet)$ is a general loss function, \mathbf{W}_q^0 and \mathbf{W}_v^0 are the initial transformation matrices learnt by CCA presented in Section 3.2, μ , γ , and η are tradeoff parameters.

In particular, the objective function is composed of three components. The first term is to minimize the ranking loss on the click-through bipartite. Specifically, we can easily obtain a set of triplets \mathcal{T} from our click-through data, where each tuple (q, v^+, v^-) consists of a query q , an image v^+ with higher clicks and a lower clicked image v^- . To preserve these preference relations in the triplets, we aim to optimize \mathbf{W}_q , \mathbf{W}_v and \mathbf{W} which makes $s(q, v^+, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) > s(q, v^-, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W})$, i.e., image v^+ is assigned a higher similarity score to query q than v^- . It is worth noticing that it is a good choice by involving some images not clicked by query q as v^- in the triplets, enforcing the projections of images with different semantics become far away in the learnt subspace. Thus, the similarities between the mappings of images in the subspace will be capable of distinguishing images with different semantics. The margin ranking loss [11] which has been used in information retrieval [3, 14] is employed and defined as

$$\begin{aligned} \mathcal{L}(s(q, v, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W})) = \\ \sum_{\mathcal{T}} \max(0, 1 - s(q, v^+, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) + s(q, v^-, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W})). \end{aligned} \quad (7)$$

In order to avoid overfitting, it is necessary to add the other two regularization terms, i.e., the last two terms in Eq. (6). The term $\|\mathbf{W}\|^2$ is to explicitly penalize overly complex matrix, while the last term results in better generalization of RCCA approach by seeking the new subspace that is close to the subspace learnt by using CCA.

To address the optimization problem in Eq.(6), we use stochastic gradient descent in this work due to its efficiency and capability of applying to highly scalable problems. Readers can refer to [3] for details. After the optimization of \mathbf{W}_q , \mathbf{W}_v and \mathbf{W} , we can obtain the similarity function defined in Eq.(5). Next, given a test query and image pair (\hat{q}, \hat{v}) , we compute the similarity between the pair as

$$s(\hat{q}, \hat{v}, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) = (\hat{q}\mathbf{W}_q)\mathbf{W}(\hat{v}\mathbf{W}_v)^\top. \quad (8)$$

This value reflects how relevant the given image could be in answering a query, with higher score indicating higher relevance. Thus, given a textual query, a rank list of images is produced by sorting the scores of query-image pairs.

Algorithm 1 RCCA for Image Search

- 1: **Input:** Click-through bipartite $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Query feature q and image feature v .
 - 2: Generate a set of triplets (q, v^+, v^-) as labeled data based on the click-through. Initialize the matrices \mathbf{W}_q and \mathbf{W}_v using a normal distribution with mean zero and standard deviation one. Initialize the matrix \mathbf{W} with the identity matrix \mathbf{I} . Set the learning rate α , and three tradeoff parameters μ, γ and η .
 - 3: **for** all the triplets **do**
 - 4: $\mathbf{W} = (1 - \alpha\mu)\mathbf{W}$
 $\mathbf{W}_q = (1 - \alpha\gamma)\mathbf{W}_q + \alpha\gamma\mathbf{W}_q^0$
 $\mathbf{W}_v = (1 - \alpha\eta)\mathbf{W}_v + \alpha\eta\mathbf{W}_v^0$
 - 5: **if** $1 - s(q, v^+, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) + s(q, v^-, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) > 0$ **then**
 - 6: $\mathbf{W} = \mathbf{W} + \alpha\mathbf{W}_q^T q^T (v^+ - v^-) \mathbf{W}_v$
 $\mathbf{W}_q = \mathbf{W}_q + \alpha q^T (v^+ - v^-) \mathbf{W}_v \mathbf{W}^T$
 $\mathbf{W}_v = \mathbf{W}_v + \alpha (v^+ - v^-)^T q \mathbf{W}_q \mathbf{W}$
 - 7: **end if**
 - 8: **end for**
 - 9: **Output:**
 Query-image similarity function:
 $\forall \hat{q}, \hat{v}, s(\hat{q}, \hat{v}, \mathbf{W}_q, \mathbf{W}_v, \mathbf{W}) = (\hat{q}\mathbf{W}_q)\mathbf{W}(\hat{v}\mathbf{W}_v)^T$
 Image-image similarity function:
 $\forall v', \hat{v}, g(v', \hat{v}, \mathbf{W}_v) = (v'\mathbf{W}_v)(\hat{v}\mathbf{W}_v)^T$
-

Moreover, when given an image example as query, the similarity between an image pair (v', \hat{v}) is computed as

$$g(v', \hat{v}, \mathbf{W}_v) = (v'\mathbf{W}_v)(\hat{v}\mathbf{W}_v)^T. \quad (9)$$

Therefore, query-by-example image search can also be performed in a similar fashion, by sorting similarity scores to produce an image rank list. The additional consideration of preference relations provides better measurement in the subspace, such that visually similar but semantically different images will receive lower similarity scores. Note that query-query relations, which are useful for applications, such as query suggestion, query expansion and query rewriting tasks, can also be computed. Algorithm 1 summarizes the major steps in RCCA for image search.

3.4. Complexity Analysis

The complexity of our proposed RCCA approach is $O(|\mathcal{T}| \times d_q \times d_v \times d)$, where $|\mathcal{T}|$ represents the number of the training triplets. The training of 1.5 million triplets with $d_q = 50,000$, $d_v = 1,000$ and $d = 80$ in our experiments can be finished within five days on one server. More importantly, the training complexity is linear to the number of triplets, which makes the incremental update with new triplets very fast. For online search, RCCA takes less than one second to finish computing the similarities for 1,000 query-image pairs on a regular PC (Intel dual-core 3.5GHz CPU and 16 GB RAM). In other words, computing the similarity of each query-image pair only takes 1.0 millisecond. Clearly, the speed is fast enough for instant response.

cardinal logo	red fox	sun moon	leaf
 25	 983	 20	 673
 13	 306	 13	 518
 2	 12	 5	 1
 1	 1	 2	 1

Figure 2. Examples in Clickture dataset. Each row lists the click counts of images in response to the query shown in the first row.

4. Experiments

We conducted our experiments on the Clickture dataset [12] and evaluated our approach on both keyword-based and query-by-example image search.

4.1. Dataset

The dataset, Clickture, is a large-scale click based image dataset [12]. It was collected from one year click-through data of one commercial image search engine. The dataset comprises two parts, i.e., the training and development (dev) sets. The training set consists of 23.1 million $\{query, image, click\}$ triads, where *query* is a textual word or phrase, *image* is a base64 encoded JPEG image thumbnail, and *click* is an integer of value no less than one. The training set contains 11.7 millions of distinct queries and 1.0 million unique images. Figure 2 shows a few exemplary queries with their clicked images and click counts in the Clickture. For example, the first image receives 25 clicks when retrieved by the query “cardinal logo.” We can easily find that images with higher clicks are more relevant to the query than that with lower clicks. The dev dataset contains 79,926 $\langle query, image \rangle$ pairs generated from 1,000 queries. The relevancy of each image to query was manually annotated on a three point ordinal scale: Excellent, Good, and Bad.

Each query is represented as a vector of words, and each word is weighted by term frequency (*tf*). Words are stemmed and stop words are removed. In our experiments, we use the top 50,000 most frequent words as the word vocabulary. Inspired by the success of deep convolutional neural networks (DCNN) [4, 17], we take the output of 1000-way fc8 classification layer by using DeCAF [6] as the image representation in this work, which is a 1000-dimensional feature vector.

4.2. Keyword-based Image Search

We first investigate our RCCA method on keyword-based image search task. The task is to estimate the similarity of each query-image pair in the dev set, and then for each query, we order the images based on their similarities.

Compared Approaches. We compare the following approaches for performance evaluation:

- (1) N-Gram SVM (NGS) trains a SVM model for each

query in the training set, by treating the clicked images as positive examples. Negative samples for classifier learning are randomly drawn. The result for a given test query is obtained by linearly fusing the rank lists of the classifiers whose queries share common n-grams with the test query.

(2) Graph-based Label Propagation (*GLP*) employs the nearest neighbors search [25] on an image similarity graph constructed by visual representations for finding Top K similar training images to the test image. The queries in the training set, which have clicks on these found training images, are then aggregated to predict the relevance of the test query-image pair.

(3) Passive-Aggressive Model [9] (*PA*) measures the match between a query-image pair by projecting the query into the image space. The learning of the mapping matrix from query to image space is performed by adapting the Passive-Aggressive algorithm.

(4) Polynomial Semantic Indexing [1] (*PSI*) first chooses a low dimensional representation space for query and image. A polynomial model is then discriminatively learnt for mapping between query-image pair and relevance score.

(5) Canonical Correlation Analysis [8, 10] (*CCA*) aims to maximize the correlation between the projections of images and queries in the subspace. Its two variants, i.e., Kernel CCA (*KCCA*) [7] and Kernel PCA with CCA (*KPCA-CCA*) [18] are also compared. The former directly learns the non-linear projections in CCA, while the latter first maps the inputs via KPCA and then feeds the new inputs into CCA.

(6) Click-through-based Cross-view Learning [19] (*CCL*) learns the subspace by jointly minimizing the distance between the mappings of query and image in the latent subspace while preserving the structure in original space.

(7) Ranking Canonical Correlation Analysis (*RCCA*) is our proposed approach described in Algorithm 1.

Parameter Settings. *NGS* and *GLP* are two baselines, which predict the relevance score on the original visual feature. *PA* directly projects query into the image space. The final feature dimensionality is thus equal to that of visual feature. For the other six subspace learning methods, the dimensionality of the latent subspace is in the range of {40, 60, 80, 100}. We set the learning rate $\alpha = 0.07$, and three tradeoff parameters $\mu = \gamma = \eta = 1.0$ in our *RCCA* algorithm by using a validation set.

Evaluation Metrics. For the evaluation of image search, we adopted Normalized Discounted Cumulative Gain (*NDCG*) as the performance metric. Given an image ranked list, the *NDCG* score at the depth of d in the ranked list is defined by: $NDCG@d = Z_d \sum_{j=1}^d \frac{2^{r^j} - 1}{\log(1+j)}$, where $r^j = \{Excellent = 3, Good = 2, Bad = 0\}$ is the manually judged relevance for each image with respect to the query. Z_d is a normalizer factor to make the score for d Excellent results 1. The final metric is the average of *NDCG@d* for all the queries in the dev set.

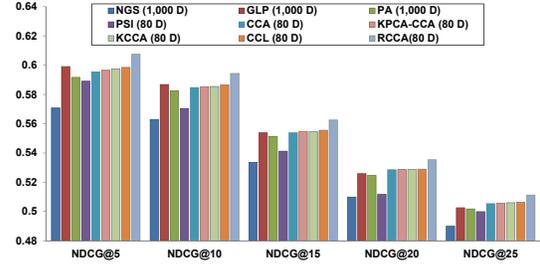


Figure 3. The *NDCG* of different approaches for keyword-based image search. The numbers in the brackets represent the feature dimension used in each approach.

Performance Comparison. Figure 3 shows the *NDCG* performances on image search of nine runs averaged over 1,000 queries in Clickture dev dataset. It is worth noting that the predictions of *NGS*, *GLP* and *PA* are performed on the original image visual features of 1,000 dimensions and for other six methods, the performances are given by choosing 80 as the dimensionality of the latent subspace.

Overall, our proposed *RCCA* consistently outperforms the other runs across different depths of *NDCG*. In particular, the *NDCG@25* of *RCCA* can achieve 0.5112, making the improvement over *NGS* model by 4.3%, which is so far the highest performance reported on Clickture dataset. More importantly, by learning a low-dimensional latent subspace, the dimension of the mapped textual query and visual image is reduced by several orders of magnitude. *GLP*, which performs nearest neighbor search, is effective in finding the top relevant images. However, the performance gain of *GLP* against *NGS* is gradually decreased when going deeper into the list. Furthermore, *RCCA* by additionally incorporating the preference relations leads to a performance boost against *CCL*, *CCA*, *KPCA-CCA*, and *KCCA*. *RCCA* outperforms *PSI* and *PA*. Although the three runs involve the utilization of the preference relations, different strategies are used for learning the mapping matrices. The learning of *PSI* and *PA* solely depends on the relative relations, while *RCCA* additionally preserves the correlations between two views making it more generalizable. Moreover, the similarity functions are different in the way that *PSI* and *PA* both use dot product, and *RCCA* is by learning a bilinear similarity function. The result basically indicates the advantage of learning the subspace and similarity function simultaneously by preserving the preference relations in the click-through data.

There is a performance gap between *PA* and *PSI*. Although both runs attempt to learn linear mapping projections based on the preference relations, the defined common subspaces are different that *PSI* learns a new common subspace, while *PA* directly projects the query into image space and considers the image space as the common space. Moreover, *CCL* by further preserving the structure in original spaces is superior to *CCA*, *KPCA-CCA* and *KCCA*, both of which extract nonlinear relations, slightly improve *CCA*,



(a) 1967 mustang restomod



(b) ryan good

Figure 4. Examples showing the top ten image search results by different methods of queries “1967 mustang restomod,” and “ryan good” (better viewed in color). The relevance scale is provided at the top left corner for each image.

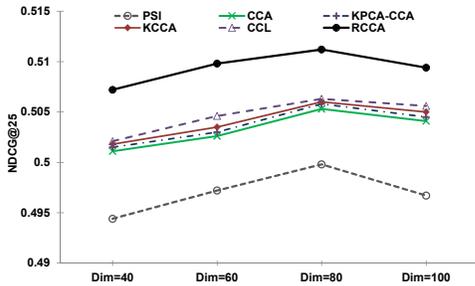


Figure 5. The $NDCG@25$ performance with different dimensionalities of the latent subspace.

but the performances are still lower than *RCCA*. The result indicates that *RCCA* is benefited from the utilization of relative preference, and capable of differentiating visually similar images in response to an identical query. Another observation is that the performance gain of *RCCA* is almost consistent when going deeper into the list. This further confirms the effectiveness of *RCCA*. In addition, to verify that the performance of different approaches is not by chance, we conducted significance test using the randomization test [22]. The number of iterations used in the randomization is 100,000 and at 0.05 significance level. *RCCA* is found to be significantly better than others.

Figure 4 shows the top ten image search results by different approaches for the query “1967 mustang restomod,” and “ryan good.” We can see the proposed *RCCA* method gets the most satisfying ranking results. Specifically, for the query “1967 mustang restomod,” *RCCA* retrieves eight excellent images in the returned top ten results, which is significantly better than other baselines.

Effect of the Dimensionality of the Latent Subspace.

In order to show the relationship between the performance and the dimensionality of the latent subspace, we compared the results of the dimension in the range of 40, 60, 80, and 100. Note that only the six subspace learning methods are included in this comparison. The results are shown in Figure 5. Compared to the other five runs, performance im-

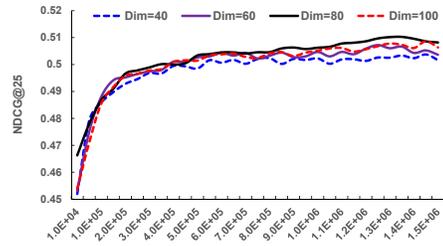


Figure 6. The $NDCG@25$ performance of *RCCA* with the increase of the number of training triplets.

provement is consistently observed at each dimensionality of the latent subspace by our proposed *RCCA* method. Furthermore, *RCCA* achieves the best result at the latent subspace dimensionality of 80, and the results at other dimensionalities are close to the best one. This observation basically verifies that *RCCA* has a good property of being not sensitive to the change of dimensionality of the latent space. In addition, the performance trends of $NDCG$ at other depths are similar with that of $NDCG@25$.

Varying the number of training triplets. Next, we conducted experiments to evaluate the performance of *RCCA* by varying the number of training triplets from 10K to 1.5 millions. Note that the training time grows linearly with the number of triplets. $NDCG@25$ performances with the increase of the number of training triplets are reported by using different dimensionalities of the latent subspace in Figure 6. Not surprisingly, we can observe that the performance is consistently improved with the increase of triplets at each dimensionality of the subspace. Furthermore, after learning a number of triplets (1.0 million in our case), the performances of *RCCA* change very smoothly.

4.3. Query-by-Example Image Search

The second experiment was conducted on query-by-example image search task. In particular, an image from dev set is taken as an image example to search semantically similar images in the training set. For each image example,

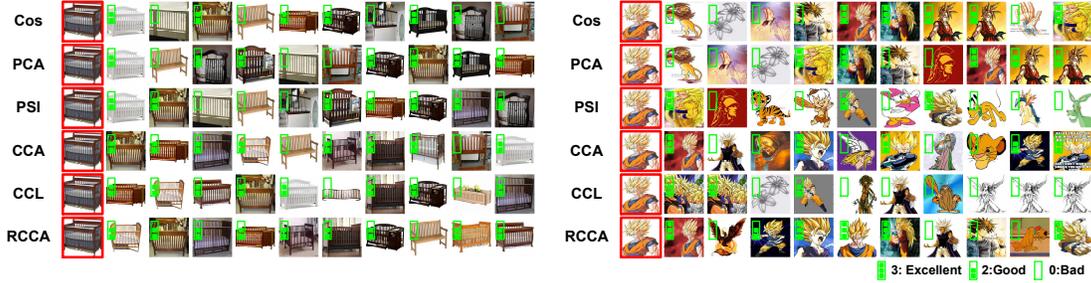


Figure 7. Examples showing the top ten image search results by different methods in response to two query images (better viewed in color). In each row, the first image with a red bounding box is the query image.

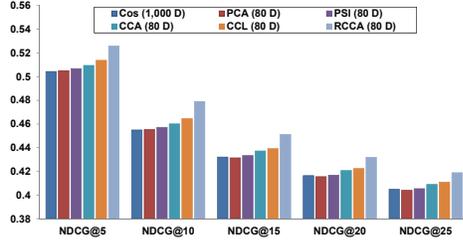


Figure 8. The NDCG of different approaches for query-by-example image search. The numbers in the brackets represent the feature dimension used in each approach.

we sort the images by their similarities to the query image.

Ground Truth. For objective evaluation, ground truth is generated directly from click-through data. Specifically, 1K unique images that are annotated as “Excellent” to their respective queries in the dev set are randomly selected as the test image examples. For each image example, the images clicked by the same query in the training set are taken as “Excellent” ones to the image example. In addition, for training queries that share more than one common noun phase with the query of the test image example, their clicked images are regarded as “Good” ones. The other images in the training set are all treated as “Bad” ones. Note that all the query terms are already stemmed and stop words are removed. The averaged $NDCG$ over 1K test image examples are finally reported.

Compared Approaches. As NGS , GLP and PA are all based on original image visual features, we report the three as one, called Cos , which measures the Cosine similarity between original visual features. In addition to PSI , CCA and CCL , PCA which first performs Principal Component Analysis [15] (PCA) on visual features and then computes Cosine similarity between the principle components is further considered as another baseline.

Performance Comparison. Figure 8 shows the performance of different approaches in terms of $NDCG$ at different depths. Overall, $RCCA$ consistently exhibits better performance than other approaches. Compared to Cos , $RCCA$ raises the $NDCG@10$ from 0.455 to 0.48 while reducing the feature dimension by more than ten times. CCA , as a cross-view version of PCA , outperforms PCA . This somewhat reveals that by maximizing the query and image cor-

relations in CCA , the similarities between image mappings could better reflect their semantics relations. By further incorporating preference relations in click-through data, $RCCA$ is capable of separating the images with different semantics and thus leads to a better performance. Similar to the observations in keyword-based image search, CCL exhibits better performance than CCA , but shows worse performance than $RCCA$. Moreover, $RCCA$ again shows statistically better performance than the others according to randomization test [22]. Figure 7 further illustrates the top ten image search results by different methods in response to two query images. We can clearly see that the images retrieved by our $RCCA$ approach are more similar in semantics with the query images. Take the first query image as an example, $RCCA$ could better distinguish the images of “crib” from “chairs” which are visually very similar.

5. Conclusions

In this paper, we have investigated the similarity learning between textual query and visual image by leveraging both click data and subspace learning techniques. We have proposed a novel ranking canonical correlation analysis method for similarity learning. Specifically, two linear projections of query and image spaces to a common subspace are initially learnt by maximizing their correlation. The two projections are further adjusted and simultaneously a bilinear similarity function is learnt in the subspace by preserving the preference relations implicit in the click-through data. Then the learnt similarity in the subspace is taken as the query-image and image-image similarity and evaluated in the context of both keyword-based and query-by-example image search. Our future works are as follows. First, the learnt query-query similarity in the subspace will be explored for applications such as query expansion, query suggestion, and query rewriting. Furthermore, we will investigate the kernel version of our method.

Acknowledgments

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 120213).

References

- [1] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, C. Cortes, and M. Mohri. Polynomial semantic indexing. In *NIPS*, 2009.
- [2] A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, D. Metzler, L. Riedel, and J. Yuan. Online expansion of rare queries for sponsored search. In *WWW*, 2009.
- [3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.
- [4] C. F. Cadieu, H. Hong, D. Yamins, N. Pinto, N. J. Majaj, and J. J. DiCarlo. The neural representation benchmark and its evaluation on brain and machine. In *ICLR*, 2013.
- [5] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2011.
- [6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [7] K. Fukumizu, F. R. Bach, and A. Gretton. Statistical consistency of kernel canonical correlation analysis. *JMLR*, pages 361–383, 2007.
- [8] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, (106):210–233, 2014.
- [9] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Trans. on PAMI*, 2008.
- [10] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.
- [11] R. Herbrich, T. Graepel, and K. Obermayer. Advances in large margin classifiers, chapter large margin rank boundaries for ordinal regression. *MIT Press, Cambridge, MA*, 2000.
- [12] X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li. Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines. In *ACM MM*, 2013.
- [13] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. on PAMI*, 34:1704–1716, 2012.
- [14] T. Joachims. Optimizing search engines using clickthrough data. In *ACM KDD*, 2002.
- [15] I. Jolliffe. *Principal component analysis*. Springer verlag, 2002.
- [16] T. Kobayashi. Low-rank bilinear classification: Efficient convex optimization and extensions. *IJCV*, pages 308–327, 2014.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [18] H. Nakayama, T. Harada, and Y. Kuniyoshi. Evaluation of dimensionality reduction methods for image auto-annotation. In *BMVC*, 2010.
- [19] Y. Pan, T. Yao, T. Mei, H. Li, C. W. Ngo, and Y. Rui. Click-through-based cross-view learning for image search. In *SIGIR*, 2014.
- [20] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.
- [21] R. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC*, 1994.
- [22] J. P. Romano. On the behavior of randomization tests without a group invariance assumption. *Journal of the American Statistical Association*, 85(411):686–692, 1990.
- [23] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. *Subspace, Latent Structure and Feature Selection*, pages 34–51, 2006.
- [24] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
- [25] J. Wang, J. Wang, G. Zeng, R. Gan, S. Li, and B. Guo. Fast neighborhood graph search using cartesian concatenation. In *ICCV*, 2013.
- [26] W. Wu, H. Li, and J. Xu. Learning query and document similarities from click-through bipartite graph with metadata. In *ACM WSDM*, 2013.
- [27] T. Yao, T. Mei, C.-W. Ngo, and S. Li. Annotation for free: Video tagging by mining user search behavior. In *ACM MM*, 2013.
- [28] C. X. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. on Information Systems*, 22(2):179–214, 2004.
- [29] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *WWW*, 2006.