IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Image Reranking Using Multimodal Sparse Coding

Mohammadi Aiman[1]*, Ruksar Fatima[2]

[1]M. Tech Student KBNCE, Kalaburagi
[2]CSE Dept., KBNCE, Kalaburagi

Available online at: www.ijcseonline.org

Accepted: 16/Jan/2019, Published: 31/Jan/2019

Abstract-- Image reranking is effective for improving the performance of a text-based image search. However, existing reranking algorithms are limited for two main reasons: 1) the textual meta-data associated with images is often mismatched with their actual visual content and 2) the extracted visual features do not accurately describe the semantic similarities between images. Recently, user click information has been used in image reranking, because clicks have been shown to more accurately describe the relevance of retrieved images to search queries. However, a critical problem for click-based methods is the lack of click data, since only a small number of web images have actually been clicked on by users. Therefore, we aim to solve this problem by predicting image clicks. We propose a multimodal hypergraph learning-based sparse coding method for image click prediction, and apply the obtained click data to the reranking of images. We adopt a hypergraph to build a group of manifolds, which explore the complementarily of different features through a group of weights. Unlike a graph that has an edge between two vertices, a hyperedge in a hypergraph connects a set of vertices, and helps preserve the local smoothness of the constructed sparse codes. An alternating optimization procedure is then performed, and the weights of different modalities and the sparse codes are simultaneously obtained. Finally, a voting strategy is used to describe the predicted click as a binary event (click or no click), from the images' corresponding sparse codes. Thorough empirical studies on a large-scale database including nearly 330K images demonstrate the effectiveness of our approach for click prediction when compared with several other methods. Additional image reranking experiments on real world data show the use of click prediction is beneficial to improving the performance of prominent graph-based image reranking algorithms.

Keywords: *Image reranking, click, manifolds, sparse codes.*

I. INTRODUCTION

One major problem impacting performance is the mismatches between the actual content of image and the textual data on the web page [4]. One method used to solve this problem is image re-ranking, in which both textual and visual information is combined to return improved results to the user. The ranking of images based on a text-based search is considered a reasonable baseline, albeit with noise. Extracted visual information is then used to re-rank related images to the top of the list.

Most existing re-ranking methods use a tool known as pseudo-relevance feedback (PRF), where a proportion of the top-ranked images are assumed to be relevant, and subsequently used to build a model for re-ranking. This is in contrast to relevance feedback, where users explicitly provide feedback by labeling the top results as positive or negative. In the classification-based PRF method, the top-ranked images are regarded as pseudo-positive and low-ranked images regarded as pseudo-negative examples to train a classifier, and then re-rank. Hsu et al. also adopt this pseudo-positive and pseudo-negative image method to develop a clustering-based re-ranking algorithm.

The problem with these methods is the reliability of the obtained pseudo-positive and pseudo-negative images is not guaranteed. PRF has also been used in graph-based re-ranking and Bayesian visual re-ranking. In these methods, low-rank images are promoted by receiving reinforcement from related high-rank images. However, these methods are limited by the fact that irrelevant high-rank images are not demoted. Therefore, both explicit and implicit re-ranking methods suffer from the unreliability of the original ranking list, since the textual information cannot accurately describe the semantics of the queries.

Instead of related textual information, user clicks have recently been used as a more reliable measure of the relationship between the query and retrieved objects [5], [6], since clicks have been shown to more accurately reflect the relevance [7]. Joachims et al. conducted an eye-tracking

experiment to observe the relationship between the clicked links and the relevance of the target pages, while Shokouhi et al. [8] investigated the effect of reordering web search results based on click through search effectiveness.

In the case of image searching, clicks have proven to be very reliable [7]; 84% of clicked images were relevant compared to 39% relevance of documents found using a general web search. Based on this fact, Jain et al. [9] proposed a method which utilizes clicks for query-dependent image searching. However, this method *only* takes clicks into consideration and neglects the visual features which might improve the retrieved image relevance to the query. In another study, Jain and Varma [10] proposed a Gaussian regression model which directly concatenates the clicks and various visual features into a long vector. Unfortunately the diversity of multiple visual features was not taken into consideration. According to commercial search engine analysis reports, only 15% of web images are clicked by web users. This lack of clicks is a problem that makes effective click-based re-ranking challenging for both theoretical studies and real-world implementation. In order to solve this problem, we adopt sparse coding to predict click information for web images.

Sparse coding is a popular signal processing method and performs well in many applications, e.g. signal reconstruction, signal decomposition, and signal denoising. Although orthogonal bases like Fourier or Wavelets have been widely adopted, the latest trend is to adopt an overcomplete basis, in which the number of basis vectors is greater than the dimensionality of the input vector. A signal can be described by a set of overcomplete bases using a very small number of nonzero elements. This causes high sparsity in the transform domain, but many applications need this compact representation of signals. In computer vision, signals are image features, and sparse coding is adopted as an efficient technique for feature reconstruction –. It has been widely used in many different applications, such as image classification, face recognition, image annotation, and image restoration.

In this paper, we formulate and solve the problem of click prediction through sparse coding. Based on a group of web images with associated clicks (known as a codebook), and a new image without any clicks, sparse coding is utilized to choose as few basic images as possible from the codebook in order to linearly reconstruct a new input image while minimizing reconstruction errors. A voting strategy is utilized to predict the click as a binary event (click or no click) from the sparse codes of the corresponding images. The overcomplete characteristic of the codebook guarantees the sparsity of the reconstruction coefficients.

II. RELATED WORK

People regularly interact with different representations of Web pages. A person looking for new information may initially find a Web page represented as a short snippet rendered by a search engine. When he wants to return to the same page the next day, the page may instead be represented by a link in his browser history. Previous research has explored how to best represent. Web pages in support of specific task types, but, consistency in representation across tasks is also important.

The related work [2] is all about exploring how different representations are used in a variety of contexts and present a compact representation that supports both the identification of new, relevant Web pages and the refinding of previously viewed pages. The visual snippet generation process involves four steps:

1. Cropping and scaling the salient image. The image is cropped manually along one dimension to an aspect ratioof 4x3 and scaled to 120x90. If no salient image is identified, a snapshot of the page is used instead, appropriately scaled.

2. Scaling the logo. The logo is scaled to fit within a 120x45 rectangle while preserving its original aspect ratio. The logos scale is chosen so that it either falls half of the height or the full width of the visual snippet. If no logo is available, it is omitted.

3. Cropping the title. 30-39 letters to be necessary to provide medium quality.

To provide satisfying summarized search result, they [3] a two-step ranking process. Considering both relevance and diversity in ranking object categories and the object layout was considered while selecting the most representative image for each category. The authors also believed that focusing on object queries is a promising direction for further advancing image search reranking and they envision the work in the future as follows: First, they will systematically classify queries into different domains regarding the possibility of image search reranking, and then develop algorithms to solve them respectively. Second,

motivated by the object bank image representation they may combine the object vocabulary discovered for the query and the objects from the collection to seek a more comprehensive representation of images and queries. Finally, identify and address the system challenges so as to most efficiently integrate this algorithm into a real-world image search engine.

Web image ranking is a tedious task because of the huge number of images in web and sparse click logs. Click logs [1] are used to know the relevancy of images under a query based on the number of clicks. Click logs of images are said to be sparse as users usually prefer clicking on web images. Thus, the very first point is to enrich the click logs by finding images that has similar features with that of existing images in the click log. Secondly, using sparse coding scores, the images are ranked. Finally, from the ranked image's metadata unique keywords are extracted and used for query recommendation. Image reranking is effective for improving the performance of a text-based image search. However, existing reranking algorithms are limited for two main reasons:

1) The textual meta-data associated with images is often mismatched with their actual visual content and

2) The extracted visual features do not accurately describe the semantic similarities between images.

Recently, user click information has been used in image reranking, because clicks have been shown to more accurately describe the relevance of retrieved images to search queries. However, a critical problem for click-based methods is the lack of click data, since only a small number of web images have actually been clicked on by users. Therefore, the aim to solve this problem by predicting image clicks

III. PROPOSED ALGORITHM

We present definitions of multimodal hypergraph learning-based sparse coding for click prediction, and define important notations used in the rest of the paper. Capital letters e.g. **X**, represent the database of web images. Lower case letters, e.g. **x**, represent images and $\mathbf{x}_i$ is the i th image of **X**. Superscript (i), e.g. $\mathbf{X}_{(i)}$ and $\mathbf{x}_{(i)}$, represents the web image's feature from the i th modality. A multimodal image database with n images and m representations can be represented as: $\mathbf{X} = \{\mathbf{X}_{(i)} = \{\mathbf{x}_{(i)1}, \ldots, \mathbf{x}_{(i)n}\} \in \mathbf{R}^{m_i \times n}\}_{i=1}^{t}$. Fig. 2 illustrates the details of the proposed framework. First, multiple features are extracted to describe web images. Second, from these features, we construct multiple hypergraph Laplacians, and perform sparse coding based on the integration of multiple features. Meanwhile, the local smoothness of the sparse codes is preserved by using manifold learning on the hypergraphs. The sparse codes of images, and the weights for different hypergraphs, are obtained by simultaneous optimization using an iterative two-stage procedure. A voting strategy is adopted to predict the click as a binary event (click or no click) from the obtained sparse codes. Specifically, the non-zero positions in sparse code represent a group of images, which are used to reconstruct the images. If more than 50% of the images have clicks, then the image is predicted as clicked. Otherwise, the image is predicted as not clicked. Finally, a graph-based schema is conducted with the predicted clicks to achieve image re-ranking. Some important notations are presented in Table I.

A. Definition of Hypergraph-Based Sparse Coding

Given an image $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{R}^d\}$, and web image bases with associated clicks as

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s,] \mid \mathbf{A} \in \mathbf{R}^{d \times s},$$

sparse coding can build a linear reconstruction of a given image **x** by using the bases in

$$\mathbf{A}: \mathbf{x} = c_1\mathbf{a}_1 + c_2\mathbf{a}_2 + \cdots + c_s\mathbf{a}_s = \mathbf{Ac}.$$

The reconstruction coefficient vector **c** for click prediction is sparse, meaning that only a small proportion of entries in **c** are non-zero. $\|\mathbf{c}\|_0$ can be denoted as the number of nonzero entries for vector **c**, and sparse coding can be described as: $\min \|\mathbf{c}\|_0$ s.t. $\mathbf{x} = \mathbf{Ac}$.

However, the minimization of this problem is NP-hard. It has been proven in that the minimization of l_1-norm approximates the sparsest near-solution. Therefore, most studies normally describe the sparse coding problem as the minimization of l_1-norm of the reconstruction coefficients. The objective of sparse coding can be defined as:

$$\min_{\mathbf{c}} \|\mathbf{x} - \mathbf{Ac}\|_2 + \alpha \|\mathbf{c}\|_1.$$

The reconstruction error is represented by the first term in (1), and the second term is adopted to control the sparsity of sparse codes **c**. α is the tuning parameter used to coordinate sparsity and reconstruction error. By using the sparse coding

method, the web images are represented independently, and similar web images can be described as totally different sparse codes. One reason for this is the loss of the locality information in equation (1).

IV. IMPLEMENTATION

We use real-world Web Queries dataset, which contains 200 diverse representative queries collected from the query log of a commercial search engine. In total, it contains 330,665 images. Table II provides details of the real-world web query datasets including the query number for each category and some examples.We select this dataset to assess our method for click prediction for two main reasons. First, the web queries and their related images originate directly from the internet, and the queries are mainly 'hot' (i.e. current) queries that have appeared frequently over the past six months. Second, this dataset contains real click data, making it easy to evaluate whether our method accurately predicts clicks on web images. The labels of images in the dataset are assigned according to their click counts. The images are categorized into two categories: the images of which the click count is larger than zero and the images of which the click count is zero. We represent each image by extracting five different visual features from the images including: block-wise color moments (CM), the HSV color histogram (HSV), color autocorrelogram (CO), wavelet texture (WT) and face feature.

B. Experiment Configuration

To evaluate the performance of the proposed method for click predication, we compare the following seven methods, including the proposed method:

1. Multimodal hypergraph learning-based sparse coding (MHL). Parameters α and β in (9) are selected by fivefold cross validation. The neighborhood size k in the hyperedge generation process and the value of z in (15) are tuned to the optimal values.

2. Multimodal graph learning-based sparse coding (MGL). Following the framework of (9), we adopt a simple graph to replace the hypergraph. The parameters α and β in (9) are determined using five-fold cross validation. The neighborhood size k and the value z are tuned to optimal values.

3. Single hypergraph learning-based sparse coding (SHL). The framework in (7) is adopted for each single visual feature separately. The average performance of SHL-SC is reported and we name it SHL(A). In addition, we concatenate visual features into a long vector and conduct SHL-SC on it. The results are denoted as SHL(L). The parameters in this method are tuned to optimal values.

4. Single graph learning with sparse coding (SGL). We adopt a simple graph to replace the hypergraph in (7). The performance of SGL(A) and SGL(L) are recorded. The parameters are tuned to their optimal values.

5. Regular sparse coding (SC). The sparse coding is directly conducted on each visual feature separately using Lasso. The average performance of SC is reported, and denoted as SC(A). In addition, we conduct sparse coding on the integrated long vector and record the results as SC(L).

6. The k-nearest neighbor algorithm (KNN). To provide the baseline performance for the experiment, we adopt KNN for each visual feature. This is a method that classifies a sample by finding its closest samples in the training set. In this experiment, each parameter is tuned to the optimal value. The KNN(A) and KNN(L) are reported.

7. The Gaussian Process regression [10]: This method identifies a group of clicked images and conducts dimensionality reduction on concatenated visual features. A Gaussian Process regressor is trained on the set of clicked images and is then used to predict click counts for all images. This method is named "GP" in the experimental results.

We randomly select images to form the image bases and test images. Since different queries contain a different number of images, it would be inappropriate to find a fixed number setting for different queries. Therefore, we choose different percentages of images to form the image bases. Specifically, the experiments are separated into two stages: the size of test image set is fixed at 5%, and the size of image base is varied from among [10%, 30%, 50%, 70%, 90%]; the size of image base is fixed at 75%, and the size of the test image is varied from among [5%, 10%, 15%, 20%, 25%]. Besides, we conduct experiments to show the effects of different parameters. For all methods, we independently repeat the experiments five times with randomly selected image bases and report the averaged results.

Algorithm 1 Implementations of sparse codes

Input: Basis matrix $\mathbf{A}$; kth image : $\mathbf{x}_k$; sparse codes $\mathbf{C}$; the weights $\sigma = [1/t, \ldots, 1/t]$; the $\hat{\mathbf{L}}$, α and β .

Output: Sparse codes for image $\mathbf{x}_k$: $\mathbf{c}_k$. $\mathbf{c}_k$ is the kth column in $\mathbf{C}$.

Step 1: For the vector $\mathbf{c}_k$, ϑ , $\hat{\mathbf{L}}_k$ and $\Upsilon_{\mathbf{c}_k}$, we adopt $\mathbf{c}_k^r$, ϑ_r , $\hat{\mathbf{L}}_k^r$ and $\Upsilon_{\mathbf{c}_k}^r$ to describe the rth entries.

Step 2: We initialize $\vartheta := \text{sign}(\mathbf{c}_k)$, and find the active set of $\mathbf{c}_k$: $(\mathbf{c}_k)_{active} := \text{Find}(\mathbf{c}_k \neq 0)$. $\vartheta_r \in \{-1, 0, 1\}$ is obtained by $\text{sign}(\mathbf{c}_k^r)$, which is the sign of the r th entry of $\mathbf{c}_k$.

Step 3: From zero coefficients of $\mathbf{c}_k$, we select $p := \arg\max_r \left| \Upsilon_{\mathbf{c}_k}^r \right|$:

3.1. If $\Upsilon_{\mathbf{c}_k}^p > \alpha$, we set $\vartheta_p := -1$, active set:= $\{p\} \cup$ active set .

3.2. If $\Upsilon_{\mathbf{c}_k}^p < -\alpha$, we set $\vartheta_p := 1$, active set:= $\{p\} \cup$ active set .

Step 4: The details for this step are:

4.1 Denote $\overline{\mathbf{A}}$ and $\overline{\Upsilon}_{\mathbf{c}_k \mathbf{c}_k}$ be the submatrix of $\mathbf{A}$ and $\Upsilon_{\mathbf{c}_k \mathbf{c}_k}$ that preserve the columns corresponding to the active set.

4.2 Denote $\overline{\mathbf{c}}_k$ and $\overline{\vartheta}$ be the subvectors of $\mathbf{c}_k$ and ϑ corresponding to the active set.

4.3 Obtain the solution by solving the unconstrained QP problem: $\min_{\overline{\mathbf{c}}_k} Q(\overline{\mathbf{c}}_k) + \alpha \overline{\vartheta}^T \mathbf{c}_k$. The analytical solution is:

$$\overline{\mathbf{c}}_{k\,new} = \left(\overline{\Upsilon}_{\mathbf{c}_k \mathbf{c}_k}\right)^{-1} \left(2\overline{\mathbf{A}}^T \mathbf{x}_k - 2\beta \left(\mathbf{C}_{-k} \hat{\mathbf{L}}_{k,-k} \right)_{active} - \alpha \overline{\vartheta} \right).$$

4.4 A discrete line search is conducted on the closed line segment from $\overline{\mathbf{c}}_k$ to $\overline{\mathbf{c}}_{k\,new}$:

- Check the objective value at $\overline{\mathbf{c}}_{k\,new}$ and all points where any coefficient changes sign.
- Update $\overline{\mathbf{c}}_k$ (and the corresponding entries in $\mathbf{c}_k$) to the point with the lowest objective value.

4.5 The zero coefficients are removed from the active set, and $\vartheta := sign(\mathbf{c}_k)$ is updated.

Step 5: The optimality conditions are justified:

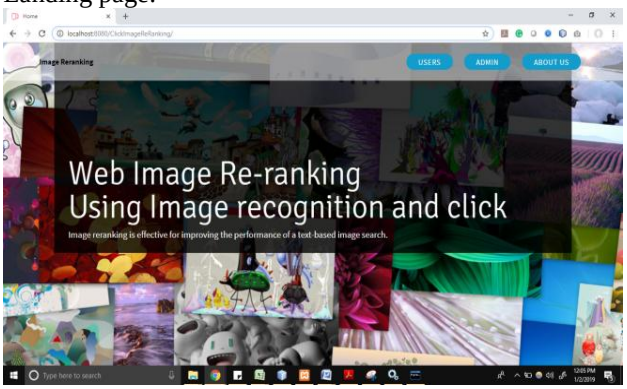
5.1 Check the optimality condition for non-zero coefficients: $\Upsilon_{\mathbf{c}_k}^j + \alpha sign(\mathbf{c}_k^j) = 0, \ \forall \ \mathbf{c}_k^j \neq 0$. If the condition in 5.1 is not met, Jump to Step 5; else check condition 5.2.

5.2 Check the optimality condition for zero coefficients: $\left| \Upsilon_{\mathbf{c}_k}^j \right| < \alpha, \ \forall \mathbf{c}_k^j = 0$. If condition in 5.2 is not satisfied, jump to step 4; otherwise return $\mathbf{c}_k$ as the solution, and update the sparse codes $\mathbf{C}$.

Fig. 3. Algorithm details of implementations for sparse codes.

V. OUTPUT

Landing page:



Upload Image for recognition:

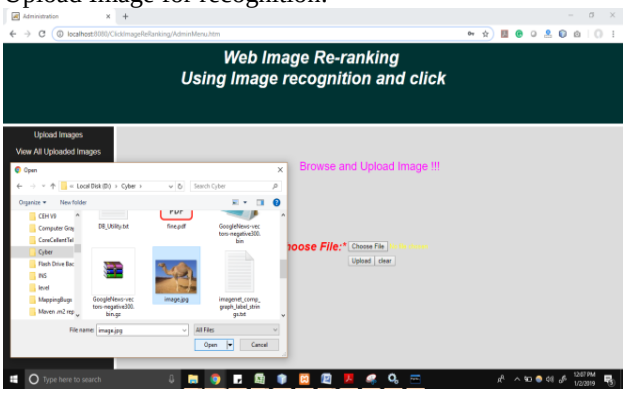
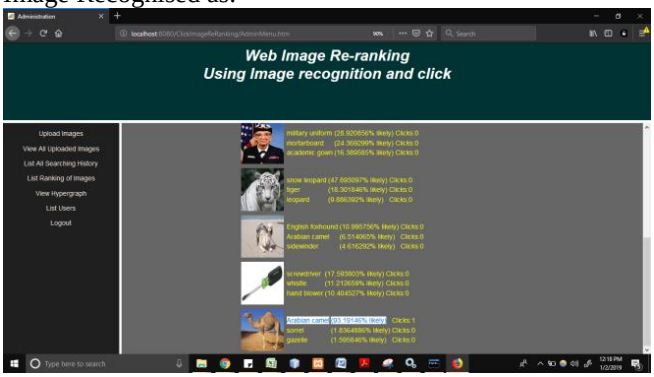
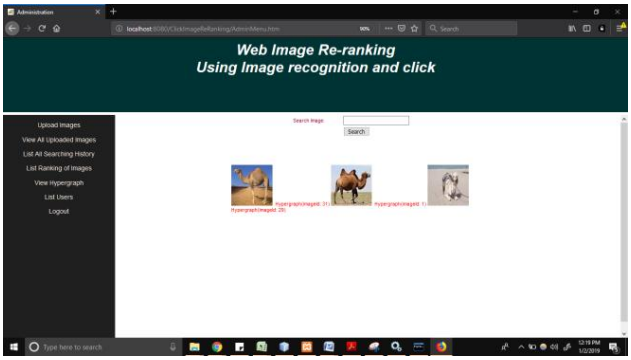


Image Recognised as:



View Hypergraph:



VI. CONCLUSION

In this paper, we have reviewed an Internet based image search approach. After a review of existing techniques related to web image re-ranking, we point out that these methods are not powerful enough to retrieve images efficiently by its including semantic concepts. Based on our findings we proposed and implemented Click based web image re-ranking technique which is efficient and effective. We have also used machine learning approaches to recognize images correctly and increase the rank of images based on the users interaction with searched image.

REFERENCES

[1] Xiaogang Wang, Ke Liu et.al, "Web Image Re-Ranking UsingQuery-Specific Semantic Signatures", IEEE Transactions on Pattern Analysis and Machine Intelligence Volume: 36 , Issue: 4 April 2014.

[2] Xinmei Tian, Dacheng Tao et.al, "Active Re-ranking for Web Image Search", IEEE Transactions on Image Processing, Vol. 19, No. 3, March 2010.

[3] J.Cui, F. Wen, et.al, "Real time Google and live image search reranking", The 16th ACM international conference on Multimedia, Pages 729-732, 2008.

[4] X. Tang, K. Liu, J. Cui, et. a, "Intent Search: Capturing User Intention for One-Click Internet Image Search", IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 34, No.7 pages 1342 – 1353, July 2012.
[5] Y. Rui, T. S. Huang, M. Ortega, et.al, "Relevance feedback: a power tool for interactive Content-based image retrieval", IEEE Transactions on Circuits and Systems for Video Technology, 1998.
[6] N. Rasiwasia, P. J. Moreno, et.al, "Bridging the gap: Query by semantic example", IEEE Transactions. On Multimedia, vol. 9, no. 5, pages.923 -938, August 2007.
[7] Xin Jin, JieboLuo,Jie Yu et. al, "Reinforce Similarity Integration in Image Rich Information Network", IEEE Transactions on Knowledge & Data Engineering, vol.25, Issue No.02, Feb 2013.
[8] E. Bart and S. Ullman. Single-example learning of novel classes using representation by similarity. In Proc. BMVC, 2005.
[9] D. Tao, X. Tang, X. Li, and X. Wu. "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval", IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006.
[10] A.W.M. Smeulders, M. Worring, S. Santini, et. al, "Content-Based Image Retrieval," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349-1380, Dec. 2000.

