Online Image Classifier Learning for Google Image Search Improvement*

Yuchai Wan, Xiabi Liu and Jie Bing
School of Computer Science and Technology
Beijing Institute of Technology
Beijing 100081, China
{wanyuchai, liuxiabi, bingjie3908}@bit.edu.cn

Yunpeng Chen
The Middle School Attached to Northern Jiaotong University
Beijing 100081, China
qw.2080@yahoo.com.cn

Abstract **- This paper proposes a content based method to improve image search results from Google search engine. The images returned by Google are used to learn a statistical binary classifier for measuring their relevance to the query. The learning process includes three stages. In the first stage, positive and negative examples are selected from the images by using k-medoids clustering technique. In the second stage, an initial classifier is obtained by performing the Expectation-Maximization (EM) algorithm on positive examples. In the third stage, the Max-Min posterior Pseudo-probabilities (MMP) learning method with dynamic data selection is applied to refine the classifier iteratively. When the classifier learning is completed, all the images are re-ranked in descending order of their posterior pseudo-probabilities. The experimental results show that the proposed approach can bring better image retrieval precisions than original Google results, especially at top ranks. Thus it is helpful to reduce the user labor of browsing the ranking in depth for finding the desired images.**

Index Terms - Image search engine, Content-based image retrieval (CBIR), Google, Image classifier learning, Online learning.

1. INTRODUCTION

The number of images on the web is increasing at a surprising rate. For example, a four-year old online photo sharing website, Flickr, has more than 40 million monthly visitors and 2 billion photos uploaded; in fact, in a single day, a few million photos are uploaded [1]. Facing such huge and fast-expanded web image database, we need more effective image retrieval techniques to find out the desired images.

Existing image retrieval methods can be divided into two categories: text based (TBIR) and content based (CBIR). The currently main image search engines such as Google rely almost purely on TBIR techniques. TBIR indexes and retrieves images according to only textural information related to images. Since the image content is ignored, the returned results are not always satisfactory. It is widely expected to improve image search by introducing CBIR techniques. However, there are enormous categories of images for searching on the web. It is difficult, if not impossible, to manually collect and label images for learning the classifiers for all categories. Recently, people have realized that web image search results could be very useful for image classifier learning and the learned classifier can be used to improve the search results in turn.

Berg and Forsyth [2] collected the animal images based on Google text search. They combined word and image information to determine whether an image in the web page belongs to the given animal category. The exemplar images for each category is selected based on text based topic discovery and user feedback. Then the set of exemplar images is expanded and re-ranked by the combination of two modalities. Li et al. [3] presented an incremental approach to web image collection and category model learning. The two procedures alternate to get more image data and more accurate category model via incremental model learning strategy. Fergus et al. [4] proposed an extended probabilistic Latent Semantic Analysis (pLSA) method to learn the object category from Google image search result and tested it for object recognition and Google search improvement. Li et al. [5] developed a system called Word2Image to produce sets of high quality, precise, diverse and representative images to visually translate a given word based on the web image collections. Schroff et al. [6] automatically generate a large number of images for a specified object class based on text based web search results. They combine text/meta-data and visual features to achieve a completely automatic ranking of the images. The images are firstly re-ranked using a Bayes posterior estimator trained on the text surrounding the image and meta data features. Then A SVM visual classifier is learnt to improve the ranking further. Kennedy et al. [7] compares advantages and disadvantages of two ways of acquiring training data for image classification: web search and human annotation. They explored the trade-off between the two ways and developed a system for recommending the suitable way of training data acquirement for a given visual concept.

Besides image classification based approach, there are some other approaches to improve text based web image search by exploiting image content, such as user interactive methods [8-9], image similarity based methods [10-15], multiple search engines based method [16], and etc. Gao et al. [8] proposed a method to filter out junk images from keyword-based Google search results by using kernel based image clustering technique, where an incremental learning

* This work is partially supported by National Natural Science Foundation of China (Grant #60973059) and Program for New Century Excellent Talents in University of China (Grant #NCET-10-0044) to X. Liu and National Undergraduate Innovational Experimentation Program (Grant #091000734) to J. Bing.

algorithm is developed to refine the clustering results according to the user interactions. Cui et al. [9] re-ranked text based image search results in an interactive manner. After query by keyword, user can click on one image, indicating this is the query image. Then all the returned images are re-ranked according to their similarities with the query. Zhou et al. [10] considered images as visual block sets and analyzed the authority of images by exploring the underlying visual block link structure via Latent Semantic Analysis (LSA). The idea of Page-Rank is leveraged to re-rank visual blocks and then re-rank images. Wang et al. [11] proposed a method called ContextRank to re-rank images by building Markov random process of a surfer jumping across visual words within and among images. Zitouri et al. [12] presented similarities of all images in a graph structure and find the densest component that corresponds to the largest set of most similar subset of images. Then the results are re-ranked based on the densest component. Jhuo and Lee [13] boosted the visual similarity measure associated with image relevance and performed a random walk over a similarity graph for re-ranking. Popescu et al. [14] introduced a lightweight re-ranking method that compares each result not only to the other query results but also to an external, contrastive class of items. Yao et al. [15] proposed the idea of co-reranking for image search, by coupling two random walks for visual and textual information, while reinforcing the mutual exchange and propagation of information relevancy across different modalities. Liu et al. [16] presented CrowdReranking which is characterized by mining relevant visual patterns from image search results of multiple search engines.

In this paper, we propose a content based method to improve Google image search. The foundation of the proposed method is that the relevance of an image to the query is measured by using posterior pseudo-probability [17], an imitation of posterior probability. The images from Google search result are employed to learn a posterior pseudo-probability function, based on which the images are re-ranked. The function learning includes three stages. Firstly, the *k*-medoids clustering [18] is used to select positive and negative examples in the images returned for a given query by Google search engine. Secondly, the Expectation-Maximization algorithm [19] is performed on positive examples to get an initial posterior pseudo-probability function. Thirdly, the discriminative learning algorithm of Max-Min posterior Pseudo-probabilities (MMP) [6] is called iteratively to refine the function. In each cycle of MMP learning, the images are re-ranked in descending order of their posterior pseudo-probabilities and the training examples are dynamically updated accordingly. The image ranking in the last cycle of the MMP learning is outputted as the final result. Fig. 1 shows the flow chart of the proposed algorithm.

We conducted 10 rounds of experiments by submitting each of 10 query keywords to Google search engine in Taiwan and testing the proposed method on the returned images. The experimental results show that our method is effective and promising. Compared with the original Google results, the mean average precision is improved from 72.09% to 76.53%, and the average precisions at top ranks of 10, 20, 30, 50 and 100 are increased by 12.99%, 13.33%, 12%, 11.11%, 2.9%, respectively, which shows that our method can help the user to find more relevant images early, i.e. reduce the user labor of browsing the ranking in depth for finding the desired images.

The rest of this paper is organized as follows. Section 2 describes the posterior pseudo-probability function for measuring the relevance of an image to the query. Section 3 presents the online learning method of posterior pseudo-probability function based on *k*-medoids clustering, EM algorithm, and MMP algorithm with dynamic data selection. Section 4 reports the experimental results. We conclude in Section 5.

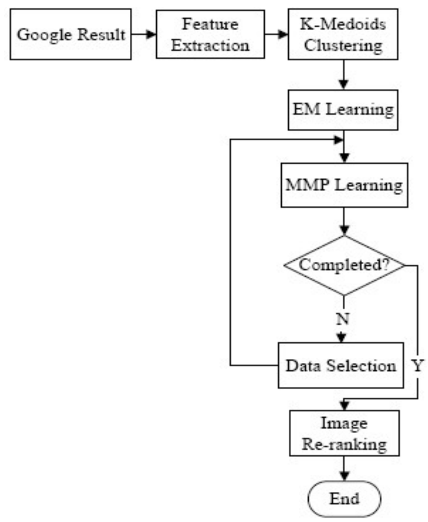

Fig. 1 The flow chart of the proposed image re-ranking algorithm.

2. Posterior Pseudo-Probability

As described above, we adopt the posterior pseudo-probability [17], an imitation of the posterior probability, to measure the relevance of an image to the given query. Let $\boldsymbol{x}$ be the feature vector extracted from an image, ω be the query keyword, then the posterior pseudo-probability is computed as

$$f(\boldsymbol{x}|\omega) = 1 - \exp\left(-\lambda p^{\mu}(\boldsymbol{x}|\omega)\right), \tag{1}$$

where $p(\boldsymbol{x}|\omega)$ is the class-conditional probability density, λ and μ are two positive numbers. According to Eq. (1), the posterior pseudo-probability is in direct proportional to the class-conditional probability density, so the classification decision made by posterior pseudo-probabilities is consistent with that by traditional Bayesian counterpart which assumes the prior probabilities of all classes are equal. However, a posterior pseudo-probability takes values in $[0, 1]$, so it is a natural and suitable measurement of relevance degree.

Before the use of Eq. (1), the image representation and the form of $p(\boldsymbol{x}|\omega)$ should be provided. We represent an image as an 80-D feature vector which consists of 9-D color moments and 71-D Gabor based texture features. As for the form of $p(\boldsymbol{x}|\omega)$, we assume it as Gaussian mixture model (GMM). The GMM is a general model for estimating an unknown probability density function and under regular conditions it may approximate any continuous function having a finite number of discontinuities [20]. Let K be the number of components in GMM, w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ be the weight, the mean, and the covariance matrix of the k-th Gaussian component, respectively. w_k satisfies $\sum_{k=1}^{K} w_k = 1$. Then we have

$$p(\boldsymbol{x}|\omega) = \sum_{k=1}^{K} w_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

where

$$\begin{aligned} &N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= (2\pi)^{-d/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k)\right). \end{aligned} \tag{3}$$

$\boldsymbol{\Sigma}_k$ is further assumed to be diagonal for feasible computation, i.e., $\boldsymbol{\Sigma}_k = (\sigma_{kj})_{j=1}^{80}$.

By substituting Eq. (2) into (1), we get the function of measuring the relevance of an image to the query keyword as

$$f(\boldsymbol{x}; \boldsymbol{\Lambda}) = 1 - \exp\left(-\lambda\left(\sum_{k=1}^{K} w_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)^{\mu}\right), \tag{4}$$

where $\boldsymbol{\Lambda}$ denotes the set of unknown parameters:

$$\boldsymbol{\Lambda} = \{\lambda, \mu, w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}, k = 1, \cdots, K. \tag{5}$$

Some parameters in Eq. (5) must satisfy certain constraints, which are transformed to unconstrained domain for easier implementation. The constraints and transformation of parameters are listed in Table I. A tiny variance value in covariance matrices of the GMM will lead to the computational instability of class-conditional probability density function. So we impose a positive minimum limit on variance value, which is denoted as τ in Table I. Consequently, the transformed parameter set is

$$\tilde{\boldsymbol{\Lambda}} = \{\tilde{\lambda}, \tilde{\mu}, \tilde{w}_k, \boldsymbol{\mu}_k, \tilde{\boldsymbol{\Sigma}}_k\}, k = 1, \cdots, K. \tag{6}$$

We use the learning method described in Section 3 to estimate $\tilde{\boldsymbol{\Lambda}}$ and then transform it into the original $\boldsymbol{\Lambda}$.

TABLE I
CLASSIFICATION FUNCTION PARAMETER TRANSFORMATION

Original parameters and constrains	Transformation of parameters
$\lambda > 0$	$\lambda \to \tilde{\lambda}: \ \lambda = \exp(\tilde{\lambda})$
$\mu > 0$	$\mu \to \tilde{\mu}: \ \mu = \exp(\tilde{\mu})$
$\sigma_{kj} > \tau$	$\sigma_{kj} \to \tilde{\sigma}_{kj}: \ \sigma_{kj} = \exp(\tilde{\sigma}_{kj}) + \tau$
$\sum w_k = 1$	$w_k \to \tilde{w}_k: \ w_k = e^{\tilde{w}_k} / \sum e^{\tilde{w}_k}$

3. POSTERIOR PSEUDO-PROBABILITY FUNCTION LEARNING

3.1 Function Initialization

Positive and negative examples should be selected from the image search result for estimating the posterior pseudo-probability function. There are two categories of images returned by search engine: relevant to the query or not. Therefore, we firstly apply k-medoids clustering technique [18] to divide the images in the search result into two groups. Considering that the visual appearance of relevant images is similar while that for irrelevant images is diverse, we take the images in the more stable cluster as positive examples and others as negative examples. Here the cluster stability is measured by the average distance of the data to the cluster center. The less the average distance is, the more stable the cluster is.

The EM algorithm [19] is performed on the positive examples to get the Maximum Likelihood Estimation (MLE) of parameters in GMM, and set λ and μ through careful experiments. Thus we get an initial posterior pseudo-probability function for re-ranking images.

3.2 Function Refinement

In this section, we use a discriminative learning algorithm, called Max-Min posterior Pseudo-probabilities (MMP) [17], on all the examples including positive and negative examples to refine the initial posterior pseudo-probability function obtained by using the EM algorithm.

3.2.1 MMP learning

The main idea behind MMP learning is to optimize the classifier performance through maximizing posterior pseudo-probabilities towards 1 for each class and its positive examples, while minimizing those towards 0 for each class and its negative examples. More formally, Let $\hat{\boldsymbol{x}}_i$ be the feature vector of arbitrary positive example, $\bar{\boldsymbol{x}}_i$ be the feature vector of arbitrary negative example, m and n be the number of positive and negative examples, respectively. Then the objective for the MMP learning is designed as

$$F(\tilde{\boldsymbol{\Lambda}}) = \frac{1}{m}\sum_{i=1}^{m}\left[f(\hat{\boldsymbol{x}}_i; \tilde{\boldsymbol{\Lambda}}) - 1\right]^2 + \frac{1}{n}\sum_{i=1}^{n}\left[f(\bar{\boldsymbol{x}}_i; \tilde{\boldsymbol{\Lambda}})\right]^2. \tag{7}$$

$F(\tilde{\boldsymbol{\Lambda}}) = 0$ means the perfect classification performance on the training data. Consequently, we can obtain the optimum parameter set $\tilde{\boldsymbol{\Lambda}}^*$ in the posterior pseudo-probability function by minimizing $F(\tilde{\boldsymbol{\Lambda}})$:

$$\tilde{\boldsymbol{\Lambda}}^* = \arg\min_{\tilde{\boldsymbol{\Lambda}}} F(\tilde{\boldsymbol{\Lambda}}). \tag{8}$$

The gradient descent algorithm is employed to optimize the parameter set $\tilde{\boldsymbol{\Lambda}}^*$ according to Eq. (8), i.e. the following iterative equation is used to update the parameters:

$$\tilde{\boldsymbol{\Lambda}}_{t+1} = \tilde{\boldsymbol{\Lambda}}_t - \alpha_t \nabla F(\tilde{\boldsymbol{\Lambda}}_t), \tag{9}$$

until convergence or a prefixed maximum number of iterations is reached. Let ε be an infinitesimal, the convergence condition is

$$\sqrt{\sum\left[\nabla F\left(\tilde{\Lambda}_t\right)\right]^2} \le \varepsilon \, . \tag{10}$$

In Eq. (9)-(10), $\tilde{\Lambda}_t$ and α_t are the parameters and the step size in the *t*-th iteration respectively, and $\nabla F\left(\tilde{\Lambda}_t\right)$ is the partial derivatives of $F\left(\tilde{\Lambda}_t\right)$ with respect to the parameters in $\tilde{\Lambda}_t$.

3.2.2 Dynamic Selection of Training Data

There are noises in training data obtained by *k*-medoids clustering. The reason is that the *k*-medoids clustering utilizes the similarities between images, but ignores the category information. The posterior pseudo-probability function is a statistical model of the given category, by using which the category of an image can be determined more accurately. So we update the positive and negative examples according to the posterior pseudo-probabilities of images and perform MMP learning iteratively with updated training data. In the first cycle of MMP learning, we use positive and negative examples selected by *k*-medoids clustering, as described in Section 3.1. In the following cycles, the posterior pseudo-probability for each image is computed and used to re-select the training examples. Actually, we arrange the images in descending order of their posterior pseudo-probabilities. Then the top *N* images are selected as positive examples and the last *N* images as negative examples.

Algorithm 1 summarizes the process of our iterative MMP learning with dynamic data selection.

Algorithm 1. The iterative MMP learning with dynamic data selection

Input: Google image search result; positive examples and negative examples from *k*-medoids clustering; initial posterior pseudo-probability function.

Repeat

Step1. Perform MMP learning on current positive and negative examples to update the parameters in the posterior pseudo-probability function.

Step2. Use the updated function to compute the posterior pseudo-probability of each image.

Step3. Arrange images in descending order of their posterior pseudo-probabilities.

Step4. Select the top *N* images as positive examples and the last *N* images as negative examples.

Until the image order is unchanged or the number of iteration times reaches prefixed maximum value.

Output: the final posterior pseudo-probability function as well as re-ranked images.

4. Experiments

4.1 Experimental Setup

We conducted total 10 rounds of Google image search improvement. In each round, a query keyword was submitted to Google search engine in Taiwan[1]. We received top 200 images returned by Google search engine and manually labeled them as relevant to the query or not. Then the experiments of image re-ranking were performed on these images. A few images in Google search results can not be downloaded successfully. So the actual number of images used in the rounds of experiments could be less than 200. Table Ⅱ lists 10 query keywords and the corresponding number of successfully downloaded images. It should be noted that if you repeat our experiments, the returned images could be slightly different from those in our experiments since Google search are updated with time to time.

In dynamic data selection procedure for MMP learning, we select images ranked at top 30% as positive examples and lowest 30% images as negative examples.

Table Ⅱ
Queries submitted to Google search engine in Taiwan and the number of successfully downloaded images

ID	Query	Image Number
1	filament lamp	183
2	lump sugar	197
3	pepper	195
4	ground	200
5	tibetan antelope	193
6	basket	196
7	strawberry	194
8	cherry	199
9	morning glory	189
10	branch	200

The performance of the proposed re-ranking algorithm is evaluated by two widely used measures: the precision at position η ($p@\eta$) and the mean average precision (MAP). Let q be the query keyword, i be the image returned at rank i , $rel(q,i)$ is the function indicating whether i is relevant to q :

$$rel(q,i) = \begin{cases} 1, i is relevant to q \\ 0, others \end{cases} . \tag{11}$$

Then $p@\eta$ is

$$p@\eta(q) = \frac{1}{\eta}\sum_{i=1}^{\eta} rel(q,i) . \tag{12}$$

The average precision (AP) for a given query is

$$\overline{p}(q) = \frac{1}{N_R}\sum_{\eta=1}^{N} p@\eta(q) \times rel(q,\eta) , \tag{13}$$

where N and N_R are the number of returned images and the number of returned relevant images, respectively.

The MAP is the mean of the average precision over all the queries. Let W be the set of query keywords, then

$$MAP = \frac{1}{|W|}\sum_{q \in W} \overline{p}(q) . \tag{14}$$

We further record the precisions at five top ranks including 10, 20, 30, 50, and 100. The data is useful to evaluate the method ability for allowing the user to find their desired image early, or in other words, reducing their labor of browsing the

[1] http://images.google.com.tw

ranking in depth.

4.2 Experimental Results

The AP for each query and the overall MAP are listed in Table Ⅲ, where AP_Google and AP_Ours means the average precision for initial Google results and re-ranked results by using our method, respectively, and IRate means the increase rate brought by our method.

The comparisons of the average $p@\eta$ at each of five top ranks over all the queries between initial Google search results and our re-ranked results are shown in Fig. 2, where it can be clearly seen that our algorithm provides the user more desired results early. And the topper the rank is, the more obvious the improvement of precision is. In fact, the average $p@\eta$ is increased by 12.99%, 13.33%, 12%, 11.11%, 2.9% for $\eta = 10, 20, 30, 50, 100$, respectively.

Fig. 3 shows some examples that our algorithm can bring better retrieval results compared with initial Google search results, where Fig. 3(a) and (b) are top 10 images in initial Google results and our re-ranked results, respectively. In Fig. 3, the images in red box are irrelevant images and others are relevant images, and the query keywords corresponding with the results are given in the leftmost of each row.

There are two queries, i.e. 9^{th} and 10^{th} query, for which our algorithm results in worse APs than initial Google result. For the 9^{th} query, our algorithm leads to better $p@\eta$ for $\eta = 10, 20, 30, 50$. Only $p@100$ is decreased. Since the user wants to find more relevant images early, we think this decrease in AP can be accepted. For the 10^{th} query, not only AP but also five $p@\eta$ are deteriorated. We compare the top 20 images returned for the 10^{th} query by Google and our algorithm in Fig. 4, where we can see that the appearance of branch images is diverse in color and texture. We think this is the main reason behind the deterioration. More sophisticated features will be considered in the future work to improve our algorithm.

4.3 Experiments for Data Selection

The training data is one of key factors in the classifier learning. We have given an automatic data selection schema in Section 3, which includes two steps: data initialization by clustering and data updating by posterior pseudo-probabilities. In order to investigate the influence of data selection on the performance of the proposed method, we conducted another three kinds of image re-ranking experiments by configuring our algorithm framework accordingly.

The purpose of the first kind of experiments is to choose the better clustering method between *k*-mediods and commonly used *k*-means. Either of two clustering methods was tested under the proposed framework, respectively. The corresponding APs are listed in Table Ⅳ, where AP_Medoids and AP_Means denote the results for *k*-mediods and *k*-means clustering, respectively. Obviously, our algorithm with *k*-mediods is totally same as that tested (AP_Ours) in Section 4.2. The data in Table Ⅳ shows that the performance of *k*-mediods is slightly better than *k*-means.

The purpose of the second kind of experiments is to evaluate the effectiveness of dynamic data selection procedure for MMP learning. So we removed this procedure from the algorithm, i.e. the MMP is performed just once on the training data obtained by *k*-mediods clustering. The result is also shown in Table Ⅳ, denoted as "AP_NotDS". As shown in Table Ⅳ, the AP is increased from 75.50% to 76.53% by introducing dynamic data selection procedure.

The purpose of the third kind of experiments is to test our automatic data selection schema as a whole. So we compare it with manual labeling schema. In manual labeling schema, both clustering and dynamic data selection procedure are removed from the algorithm framework. The images labeled as relevant are taken as positive examples and others as negative examples. Then the EM algorithm and the MMP algorithm are performed on these perfect training examples. The corresponding AP is listed in Table Ⅳ as "AP_Manual". By comparing AP_Manual with AP_Mediods, we can see that a perfect data selection schema such as manual labeling can further strengthen the power of the proposed method. Since the manual labeling is not practical in real applications, the more effective automatic data selection strategy will be explored in the future. We also observed an interesting phenomenon that although the manual labeling leads to better APs for most of queries, some of APs are really decreased such as 5^{th}, 7^{th}, 8^{th} and 10^{th}. We guess this phenomenon is caused by unsatisfactory image features.

Table Ⅲ

THE AVERAGE PRECISION FOR EACH QUERY OBTAINED BY INITIAL GOOGLE RESULT (AP_GOOGLE) AND OUR RE-RANKED RESULT (AP_OURS), WHERE IRATE MEANS THE INCREASE RATE OF AP BROUGHT BY AP_OURS COMPARED WITH AP_GOOGLE.

Query ID	AP_Google	AP_Ours	IRate
1	72.46%	88.84%	16.38%
2	64.84%	76.80%	11.96%
3	68.01%	77.36%	9.35%
4	74.16%	78.69%	4.53%
5	81.61%	87.57%	5.96%
6	66.28%	68.37%	2.09%
7	80.92%	85.45%	4.53%
8	70.60%	72.30%	1.7%
9	77.09%	73.86%	-3.23%
10	64.87%	56.08%	-8.79%
MAP	72.09%	76.53%	4.45%

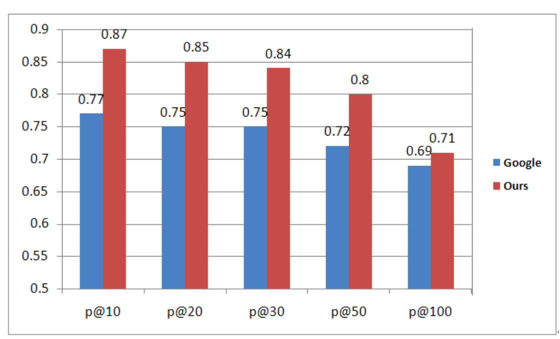

Fig. 2 The comparisons of average $p@\eta$ at each of five top ranks between initial Google search results and our re-ranked results.

Fig. 3 Examples of top 10 images in initial Google search results and our improved results, where the red box indicate irrelevant images and others are relevant and the leftmost text is the query keyword: (a) initial Google results; (b) our improved results.

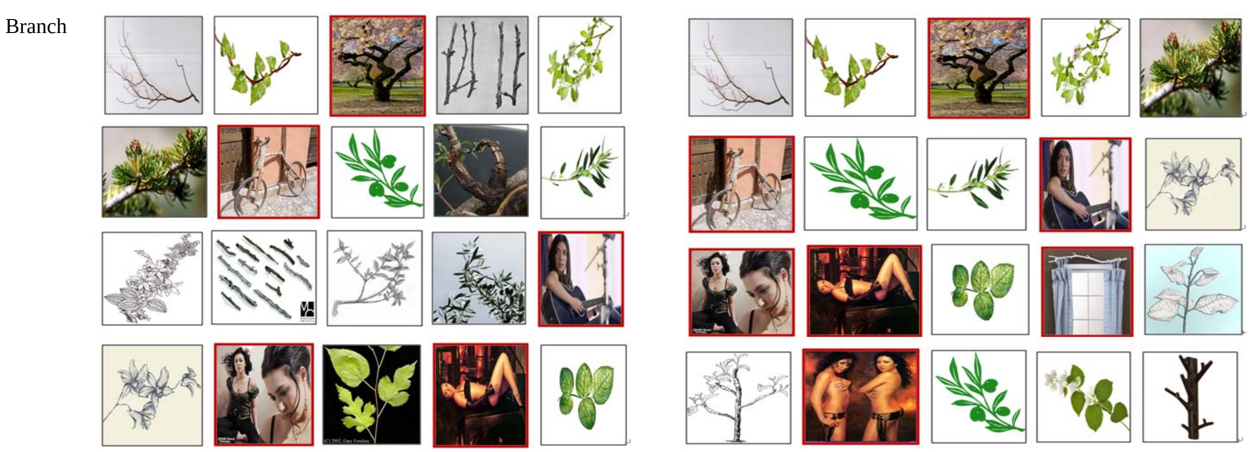

Fig. 4 The illustration of deteriorated $p@20$ for the query "branch", where the red box indicate irrelevant images and others are relevant: (a) initial Google result; (b) our re-ranked result.

Table Ⅳ

THE AVERAGE PRECISIONS FOR REFLECTING THE INFLUENCE OF DATA SELECTION ON THE PERFORMANCE OF THE PROPOSED METHOD, WHERE AP_MEDIODS, AP_MEANS, AP_NOTDS, AP_MANUAL MEANS THE RESULTS FROM OUR ALGORITHM WITH *K*-MEDIODS CLUSTERING, *K*-MEANS CLUSTERING, WITHOUT DYNAMIC DATA SELECTION, AND WITH MANUAL LABELING, RESPECTIVELY

Query ID	AP_ Mediods	AP_ Means	AP_ NotDS	AP_ Manual
1	88.84%	86.89%	83.72%	92.06%
2	76.80%	74.64%	75.44%	85.44%
3	77.36%	77.87%	76.32%	80.51%
4	78.69%	79.01%	76.93%	84.24%
5	87.57%	85.50%	86.49%	84.81%
6	68.37%	70.34%	68.88%	74.80%
7	85.45%	83.36%	85.78%	84.47%
8	72.30%	71.47%	70.60%	70.30%
9	73.86%	76.61%	73.86%	84.80%
10	56.08%	55.02%	56.97%	55.28%
Mean	76.53%	76.07%	75.50%	79.67%

5. CONCLUSIONS

This paper has proposed an approach to improve Google image search results by exploiting image content. The main contribution is an online learning method of posterior pseudo-probability function for image re-ranking, in which a key problem is how to automatically select the training data from Google search result. We tackle this problem by firstly using *k*-medoids clustering technique to obtain an initial training data set and then dynamically updated it in iterative discriminative learning process.

The experiments were preceded by inputting keywords into Google search engine in Taiwan and performed the proposed algorithm on search results. The experimental results show that our method can lead to better retrieval precisions, especially the precisions at top ranks. Compared with initial Google results, the mean average precision over the testing keywords is improved from 72.09% to 76.53%, and the precisions at top ranks 10, 20, 30, 50 and 100 are increased by 12.99%, 13.33%, 12%, 11.11%, 2.9%, respectively. The data shows that our method can reduce the user labor of browsing the ranking in depth for finding the desired images.

As demonstrated in the experiments in Section 4.3, the selection of positive and negative examples from the search results is important for the success of the proposed method. In the future, we will investigate more effective data selection schema for further improving image retrieval precisions. Furthermore, we only used global image features including color and texture, which could fail to capture the appearance characteristics of the images from a given category and discriminate them from other categories' images. As analyzed in the experimental results, this is a main reason behind the deteriorated re-ranking. We consider the local feature is a possible solution to this problem. However, the currently main local features based image representation such as bag-of-features [21] are time-consuming and unsuitable for real-time online applications. We expect to strengthen our method by exploring efficient local features and adding them into our algorithm framework in the future.

REFERENCES

[1] J. Z. Wang, D. Geman, J. Luo, and R. M. Gray, "Real-world image annotation and retrieval: an introduction to the special section," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1873-1876, 2008.

[2] T. L. Berg and D. A. Forsyth, "Animals on the web," in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 2006.

[3] L. J. Li, G. Wang, and F. F. Li, "OPTIMOL: automatic online picture collection via incremental model learning," in Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, USA, 2007.

[4] R. Fergus, F. F. Li, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 2005.

[5] H. Li, J. Tang, G. Li, and T.-S. Chua, "Word2Image: Towards Visual Interpreting of Words," in Proceedings of the 16th International Conference on Multimedia (MM'08), Vancouver, British Columbia, Canada, October 26–31, 2008, pp. 813-816.

[6] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting Image Databases from the Web," in Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, 2007, pp. 2036-2043.

[7] L. S. Kennedy, S. F. Chang, and I. V. Kozintsev, "To search or to label? Predicting the performance of search-based automatic image classifiers," in Proceeding of the 8th ACM Multimedia International Workshop on Multimedia Information Retrieval (MIR'06), Santa Barbara, California, USA, 2006.

[8] Y. Gao, J. Peng, H. Luo, D. A. Keim, and J. Fan, "An Interactive Approach for Filtering Out Junk Images from Keyword-based Google Search Results," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 12, pp. 1851-1865, Dec. 2009.

[9] J. Cui, F. Wen, and X. Tang, "Real Time Google and Live Image Search Re-ranking," in Proceedings of the 16th International Conference on Multimedia (MM'08), Vancouver, British Columbia, Canada, October 26–31, 2008.

[10] W. Zhou, Q. Tian, and H. Li, "Visual Block Link Analysis for Image Re-ranking," in Proceedings of the first International Conference on Internet Multimedia Computing and Service (ICIMCS'09), Kunming, Yunnan, China, Nov. 23-25, 2009, pp. 10-16.

[11] S. Wang, Q. Huang, S. Jiang, L. Qin, and Q. Tian, "Visual ContextRank for Web Image Re-ranking," in Proceedings of the first ACM workshop on Large-scale multimedia retrieval and mining (LS-MMRM'09), Beijing, China, Oct. 23, 2009, pp. 121-128.

[12] H. Zitouni, S. Sevil, D. Ozkan, and P. Duygulu, "Re-ranking of Web Image Search Results using a Graph Algorithm," in Proceedings of 19th International Conference on Pattern Recognition (ICPR'08), Tampa, FL, USA,. Dec. 8-11, 2008.

[13] I. H. Jhuo and D. T. Lee, "Boosting-based Multiple Kernel Learning for Image Re-ranking," in Proceedings of the 18th International Conference on Multimedia (MM'10), Firenze, Italy, Oct. 25–29, 2010, pp. 1159-1162.

[14] A. Popescu, P.-A. Moëllic, I. Kanellos, and R. Landais, "Lightweight Web Image Reranking," in Proceedings of the 17th International Conference on Multimedia (MM'09), Beijing, China, Oct. 19–24, 2009, pp. 657-660.

[15] T. Yao, T. Mei, and C.-W. Ngo, "Co-reranking by Mutual Reinforcement for Image Search," in Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'10), Xi'an, China, July 5-7, 2010, pp. 34-41.

[16] Y. Liu, T. Mei, and X.-S. Hua, "CrowdReranking: Exploring Multiple Search Engines for Visual Search Reranking," in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09), Boston, Massachusetts, USA, July 19–23, 2009, pp. 500-507.

[17] X. Liu, Y .Jia, X. Chen, Y. Deng, and H. Fu, "Image Classification

Using the Max-Min Posterior Pseudo-Probabilities Method," Technical Report, School of Computer Science and Technology, Beijing Institute of Technology, China, 2008. Available at http://isc.cs.bit.edu.cn/faculties/liuxiabi/papers/2008_1.pdf.

[18] J. Han and M. Kamber, *Data Mining: Concepts and Techniques (second edition)*, Beijing: China Machine Press, China, 2007.

[19] A. P. Dempster, N. M .Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM Algorithm," *J. R. Stat. Soc.*, vol. 39, pp. 1–38, 1977.

[20] N. Vlassis and A. Likas, "A Kurtosis-based dynamic approach to Gaussian mixture modeling," *IEEE Trans. Syst. Man Cybern*, *A*, vol. 29 pp. 393–399, 1999.

[21] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision,* vol. 73, no. 2, pp. 213–238, 2007.