Web Image Classification for Information Extraction

Martin Labský[1], Miroslav Vacura[1], Pavel Praks[2]

[1] Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
{labsky,vacuram}@vse.cz
[2] Department of Mathematics and Descriptive Geometry,
Department of Applied Mathematics, Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
pavel.praks@vsb.cz

Abstract. We describe an approach to classifying images found on the WWW for the purpose of information extraction (IE). Among features used for classification are image sizes, colour histograms, and the similarity of the classified image's content to images in a training collection. Our content similarity metric is based on the latent semantic index. Results are presented on a collection of 1624 image occurrences found on bicycle shop websites, and the task is to distinguish bicycle images from the rest.

1 Introduction

The task of image classification has a wide range of applications, including retrieval from image collections, iris recognition, pornography blocking and information extraction (IE). In the IE domain, image classification enables extraction of objects that are presented partially or entirely by graphical means. We experimented with extraction of product offers, which consisted of several textual fields and of an image of the product. Results are presented for the area of bicycles, however the classifier is domain-independent.

In section 2 we describe the image collection used in our experiments. Section 3 lists features used for classification and discusses the importance of each feature. Section 4 presents classification error rates. Finally, related and further work is discussed in section 5.

2 Image Collection

All images used in our experiments come from a collection of 133 HTML documents chosen from the Google Directory *Sports-Cycling-BikeShops-Europe-UK-England*. Each document contains from 1 to 50 bicycle offers, and about 61% of offers include a bicycle picture. There are typically 3–4 documents from the same

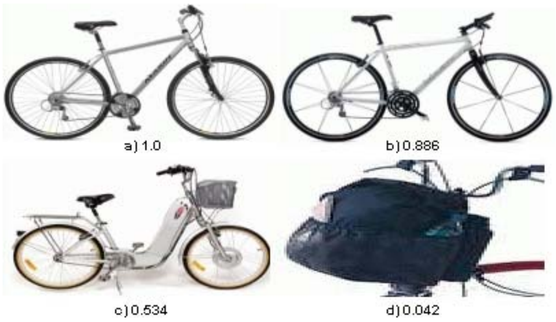

Fig. 1. Samples images sorted by similarity to the first image

shop in the data. Together there are 1624 occurrences of 900 unique images[3]. Sample images are shown in Figure 1.

In the following, results are presented based on image occurrences, not on unique images. This method of evaluation seems to be more appropriate for IE applications, since extraction depends separately on each occurrence being correctly classified. We define an image occurrence by the statement that image with a particular *URL* occurs in document *D*. We therefore treat multiple occurrences of the same image inside a single document as a single occurrence, while the same image appearing in two different documents would be considered twice.

In our data, most repeating images are advertisement banners and images used for page layout. Of the 1624 image occurrences, there were 598 bicycle images (positive examples) and the remaining 1026 images were considered negative. Positive examples of bicycle images include those that were not part of any bicycle offer labeled for extraction.

3 Features Used For Classification

In the following we describe each feature used for classification. Some of the utilised features are similar to those used by Nakahira [5]. We attempt to evaluate the contribution of each feature to the overall classification performance by evaluating simple classifiers that use only that particular feature. All results are measured using 10-fold cross-validation, where images are split so that no images from a single HTML document appear both in training and test data.

3.1 Image similarity using Latent Semantic Indexing (LSI)

We employed a *latent semantic* approach to measuring image similarity, described in [6] and [7]. Latent Semantic Indexing (LSI) is an information retrieval strategy that has originally been used for semantic analysis and retrieval from

[3] The image collection is available from `http://rainbow.vse.cz`.

large amounts of text documents. LSI can be viewed as a variant of the vector space model with its term-document matrix approximated via dimension reduction methods such as the Singular Value Decomposition (SVD). Numerical experiments pointed out that dimension reduction methods, when applied to original data, bring two main advantages to information retrieval: (1) automatic noise filtering and (2) natural clustering of documents with similar characteristics. In our approach, a raster image is coded as a sequence of pixels. Image is thus represented as a vector in m-dimensional space, where m denotes the number of pixels (keywords). Prior to applying our method, all images are rescaled to the same size and converted to gray scale. The term-document matrix A is then an $m \times n$ matrix containing information about m pixels (keywords) in n images (documents).

The Singular Value Decomposition (SVD) of document matrix. Let the symbol A denote the $m \times n$ data matrix, i.e., the document matrix. The aim of SVD is to compute decomposition

$$A = USV^T \tag{1}$$

where S is an $m \times n$ diagonal matrix with nonnegative diagonal elements called the singular values, U and V^T are $m \times m$, and $n \times n$ orthogonal matrices, i.e., the following conditions hold: $U^T = U^{-1}$, $V^T = V^{-1}$. The columns of matrices U and V^T are called the left singular vectors and the right singular vectors, respectively.

The SVD decomposition can be computed so that the singular values are sorted by decreasing order. For large real problems, the full SVD decomposition is memory and time consuming operation. Moreover, our experiments show that computation of very small singular values and associated singular vectors can damage retrieval results. Due to these facts, only the k largest singular values of A and the corresponding left and right singular vectors are computed and stored in memory in practice.

In this way a multi-dimensional space is reduced to k-dimensional vector space according to:

$$A_k = U_k S_k V_k^T \tag{2}$$

where the symbol U_k denotes $m \times k$ matrix derived from the U matrix by the selection of its k first columns, S_k is $k \times k$ diagonal matrix with the diagonal including the first k singular values, and V_k is $n \times k$ matrix acquired by the selection of the k first columns of the V matrix. The columns of the matrix V_k^T contain transformed (i.e. filtered) documents of the original data collection.

In other words, the SVD allows the approximation of the matrix A with respect to the column vectors. The k-approximation (A_k) of the A matrix rank is acquired by choosing only the k first singular values of the matrix S, while the other ones are neglected. The LSI algorithm was implemented in Matlab[4]. For the computation of few singular values and vectors of the matrix A we used the

[4] The Math Works, Ltd.

standard Matlab command $svds(A, k)$.
There is no exact routine for the selection of the optimal number of computed singular values and vectors [1]. For this reason, the number of singular values and associated singular vectors used for SVD calculation was estimated experimentally.
The SVD of any realistic term-document matrix is a very memory and time consuming operation. Analysing the original LSI [3] and using observations from linear algebra, a new SVD-free LSI procedure was derived. The derived LSI algorithm replaces the expensive SVD of the non-square matrix A by the partial eigenproblem of $A^T A$ ("covariance matrix"). Using a Lanczos-based iterative method, solution to this partial symmetric eigenproblem can be obtained very effectively. In addition, the size of the eigenproblem does not depend on the number of pixels (keywords). Our numerical experiments proved that the derived SVD-free LSI is suitable for both image and text retrieval [6, 7].

The computation of similarity coefficients between transformed documents. For image classification, the reduced matrix is searched for image vectors that are most similar to the (also reduced) vector of the image to be classified. There is a lot of possibilities how to calculate similarity between two vectors. In this paper, we use the well known *cosine similarity*, which measures the cosine of an angle between two vectors in vector space. Intuitively, the *distance* of two images is proportional to the angle Φ between their reduced vectors. The *cosine similarity* $cos(\Phi_j)$ of a reduced query image q to a reduced image I_j from collection is calculated as

$$cos(\Phi_j) = \frac{q^T I_j}{\sqrt{q^T q}\sqrt{I_j^T I_j}} \tag{3}$$

where $1 \leq i \leq n$. For the cosine similarity it holds: $-1 \leq cos(\Phi_j) \leq 1$. Decreasing the angle Φ_j between vectors causes the absolute value of the $cos(\Phi_j)$ similarity to increase. We chose $abs(cos(\Phi_i))$ as the ultimate similarity measure. Sample similarity values are shown in Figure 1, relative to the upper left image.

For image classification, we built the reduced term-document matrix solely from positive examples of bicycle images. The image q to be classified is used as a query and K most similar images are retrieved along with their similarities. These K image-to-image similarities are then averaged to compute $sim_C(q)$, the similarity of q to the *collection* of images C.

$$sim_C(q) = \frac{\sum_{K\ best\ images\ j \in C} sim(q, j)}{K} \tag{4}$$

Experimentally, we set $K = 20$, since lower values of K lead to a decrease in the similarity's robustness[5] and higher values did not bring further improvement. We

[5] With low values of K, $sim_C(q)$ became too sensitive to individual images j with misleading values of $sim(q, j)$.

tested a simple classifier with $sim_C(q)$ being the only attribute[6] and achieved an error rate of 26.4% on our document collection. The classifier decided according to a similarity threshold estimated on a subset of each training fold.

3.2 Image size

Image width and height proved to be simple yet powerful features for our task. We modelled the size of bicycle images using a 2-dimensional normal distribution, only estimated from positive training examples. The width and height of a new image q are first evaluated using the estimated normal density N. The density value is then normalised to interval (0,1) using the density's maximum value N_{max}.

$$siz(q) := \frac{N(width, height)}{N_{max}} \tag{5}$$

Within our document collection, the similarity score $siz(q)$ appeared to be the best single predictor with an error rate of 6.2%. Again, the classifier was based on a decision threshold for $siz(q)$ and was evaluated analogously to the similarity classifier. When we added the original image width and height as two additional attributes, and used the PART decision list from Weka [9], the error rate dropped to 3.2%. We speculate this might be due to the ability of the decision list to capture exact standardised sizes of both product and non-product images (e.g. banners). However, this optimistic result is mainly due to our collection being limited to relevant product catalogues only. When dealing with heterogeneous data, e.g. when choosing from multiple product types, the content-related features will become more important.

3.3 HSV Histogram

Another method that we employed in attempt to classify web images was based on colour information. Images found on web pages are usually in JPG, GIF or PNG format and use RGB colour space. This colour space has some properties that make certain kinds of image analysis inefficient. This is why we first transformed images to the HSV colour space, which represents every colour by hue and amount of lightness and saturation.

In the HSV colour space, the metrical distance of codes of two different colours is in tight relation to their similarity as perceived by humans. Since colour similarity is held well by the HSV model, we quantised its colour space to discrete intervals containing similar colours. We have used standard quantisation – 18 discrete values for hue and 3 values for lightness and saturation [8]. The resulting quantised histograms contained 162 different colour groups.

[6] Similarity scores were computed for each of the 10 cross-validation folds based on term matrices computed from the remaining training folds. The classifier used the same training and test folds.

For each image, the quantised histogram represented the numbers of pixels that had colour associated with a given colour group. Let c_i be the number of pixels associated with colour group i, then histogram of image q is the vector $h_q =< c_1, c_2, \ldots, c_{162} >$. This histogram is then normalised to $\hat{h}_q =< \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{162} >$ where (using $k = 10000$):

$$\hat{c}_n = \frac{c_n \cdot k}{\sum_{i=1}^{162} c_i} \tag{6}$$

Normalised histograms were used as input to various classifiers available in the Weka system. The best error rate was again achieved by the PART decision list – 5.2%.

4 Results

Finally, we built a classifier that used all of the features described above: the similarity score $sim_C(q)$, size score $siz(q)$, width and height, and the 162-dimensional HSV histogram vector. We also tried adding the image's occurrence count within the same HTML document, since count above 1 seemed to be a good predictor of the image not depicting a product. However, this feature was dominated by the other features and did not bring further improvement.

The best performing classifier we found was again the PART decision list, with an error rate of 2.6%. Results for all single-feature classifiers and the combined classifier are summarised in Table 1[7].

Table 1. Image classification results

	Similarity	Size	Size2	Histogram	Combined
Error rate (%)	26.4	6.2	3.2	5.2	2.6

For the IE task, we used a variant of the PART decision list that could refuse classification if it was not sure. This was to allow the extractor component to give more weight to the context (nearby text and other extracted fields) that surrounded the image within the HTML page. Finally, the extractor was able to achieve 86.9% precision and 89.1% recall for bicycle image extraction [4].

5 Related and Future Work

Nakahira [5] describes a similar project aiming at web image classification into several domain–independent classes based on the image's function in the web page. Yanai [10] presents a trainable image recognition system capable of classifying novel images based on their content into semantic classes such as animal or

[7] Size2 includes the original image width and height as attributes.

product names. The Crossmarc project [2] deals with cross-lingual product IE from the web. Crossmarc uses an image processing algorithm that extracts text from encountered images, which is further used by textual IE. For image classification, this text can be taken as another feature, as well as other text contained in the HTML document and related to the image. Such text especially includes the *alt* and *title* attributes, the image URL, and the URL and *title* attribute of a containing link.

We plan to test the image classifiers on different domains; one application will involve determining the type of depicted products. Some of the presented methods may be used for generalised web image classification task, based on a domain ontology. These ideas are being implemented as part of an ontology-driven web service based architecture of our long-term project Rainbow [4].

The research is partially supported by grant no.201/03/1318 of the Grant Agency of the Czech Republic, "Intelligent analysis of the WWW content and structure".

References

1. Berry W.M,, Dumais S. T., O'Brien G. W.: Using linear algebra for intelligent information retrieval. In: SIAM Review, 37 (1995), pp. 573-595.
2. Grover C., McDonald S., Gearailt D., Karkaletsisy V., Farmakiotouy D., Samaritakisy G., Petasis G., Pazienza M., Vindigni M., Vichotz F., Wolinskiz F.: Multilingual XML-Based Named Entity Recognition for E-Retail Domains. In: LREC Conference, Las Palmas, 2002.
3. Grossman D. A., Frieder O.: Information retrieval: Algorithms and heuristics. Kluwer Academic Publishers, 2nd ed., 2000.
4. Labský M., Svátek V., Šváb O.: Information Extraction from HTML Product Catalogues: from Source Code and Images to RDF. Accepted for: Int. Conference on Web Intelligence, Compiegne, 2005.
5. Nakahira, K., Yamasaki, T., Aizawa, K.: Accuracy Enhancement of Function-Oriented Web Image Classification. In: Proc. of The 14th Int. WWW Conference, Chiba, 2005.
6. Praks P., Dvorský J., Snášel V.: Latent semantic indexing for image retrieval systems. In: Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg, 2003.
7. Praks P., Dvorský J., Snášel V. and Černohorský J.: On SVD-free Latent Semantic Indexing for Image Retrieval for application in a hard industrial environment. In: Proceedings of the IEEE Int. Conference on Industrial Technology, Maribor, 2003.
8. Smith, J. R., Chang S. F.: Tools and techniques for color image retrieval. In: Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV, volume 2670, San Jose, 1996.
9. Witten, I. H., Frank, E.: Generating Accurate Rule Sets Without Global Optimization. In: Proc. of the 15th Int. Conference on Machine Learning, Morgan Kaufmann, CA, 1998.
10. Yanai, K.: Web Image Mining toward Generic Image Recognition. In: Proc. of The 12th Int. WWW Conference, Budapest, 2003.