International Journal Of Computer Science And Applications Vol. 1, No. 1, June 2008
ISSN 0974-1003

Web Page Classification Based on Document Structure without Negative Examples

Mrs. M .A.Shah
Lecturer,C.S.E. Dept.
Walchand College of Engineering,
Sangli
+91-9423872296
shah.medha@gmail.com

Dr. S.M. Deshpande
Professor,C.S.E. Dept.
Walchand College of Engineering,
Sangli
+91-9820028645
s_m_deshpande@yahoo.com

ABSTRACT
We demonstrate the usefulness of the PEBL framework for Web page classification which eliminates the need for manually collecting negative training examples in preprocessing. We extracted specially used and clearly defined features of web pages in our experiment. The PEBL framework applies an algorithm, called Mapping-Convergence (M-C), to achieve high classification accuracy (with positive and unlabeled data) as high as that of a traditional SVM (with positive and negative data). M-C runs in two stages: the mapping stage and convergence stage. In the mapping stage, the algorithm uses a weak classifier that draws an initial approximation of "strong" negative data. Based on the initial approximation, the convergence stage iteratively runs an internal classifier (e.g., SVM) which maximizes margins to progressively improve the approximation of negative data. Thus, the class boundary eventually converges to the true boundary of the positive class in the feature space. We implemented M-C algorithm, our experiments show that, given the same set of positive examples, the M-C algorithm performs almost as accurate as the traditional SVM gives 98.00 percentage Precision –Recall value .

Categories and Subject Descriptors:

Information storage and retrieval]: Web page Classification, document structure, feature vector.

General Terms: Algorithms, Performance, Design.

Keywords: Web page classification, Web mining, document classification, Mapping-Convergence (M-C)algorithm, SVM (Support Vector Machine).

1 INTRODUCTION
Through the billions of Web pages created with HTML and XML, or generated dynamically by underlying Web database service engines, the Web captures almost all aspects of human endeavor and provides a fertile ground for data mining. However, searching, comprehending, and using the semi-structured information stored on the Web poses a significant challenge because this data is more sophisticated and dynamic than the information that commercial database systems store. To supplement keyword-based indexing, which forms the cornerstone for Web search engines, researchers have applied data mining to Web-page classification. In this context, data mining helps Web search engines find high-quality Web pages.

While Web page classification has been actively studied, most previous approaches assume a multiclass framework, in contrast to the one-class binary classification problem that we focus on. These multiclass schemes (e.g., [1], [2]) define mutually exclusive classes a priori, train each class from training examples, and choose one best matching class for each testing data. However, mutual-exclusion between classes is often not a realistic assumption because a single page can usually fall into several categories. Moreover, such predefined classes usually do not match users' diverse and changing search targets. Researchers have realized these problems and proposed the classifications of user-interesting classes such as "call for papers," "personal homepages," etc. [3]. This approach involves binary classification techniques that distinguish Web pages of a desired class from all others.

Binary classifier is an essential component for Web mining because identifying Web pages of a particular class from the Internet is the first step of mining interesting data from the Web. A binary classifier is a basic component for building a type specific engine [4] or a multiclass classification system [5], [6]. When binary classifiers are considered independently in a multiclass classification system, an item may fall into none, one, or more than one class, which relaxes the mutual-exclusion assumption between classes [7].

However, traditional binary classifiers for text or Web pages require laborious preprocessing to collect positive and negative training examples. For instance, in order to construct a "homepage" classifier, one needs to collect a sample of homepages (positive training examples) and a sample of non-homepages (negative training examples). Collecting negative training examples is especially delicate and arduous because 1) negative training examples must uniformly represent the universal set excluding the positive class (e.g., sample of a non-homepage should represent the Internet uniformly excluding the homepages), and 2) manually collected negative training examples could be biased because of human's unintentional prejudice, which could be detrimental to classification accuracy.

To eliminate the need for manually collecting negative training examples in the preprocessing, we have used a framework, called Positive Example Based Learning (PEBL) [8]. Using a sample of the universal set as unlabeled data, PEBL learns from a set of positive data as well as a collection of unlabeled data. A traditional learning framework learns from labeled data which contains manually classified, both positive and negative examples. Unlabeled data indicates random samples of the universal set for which the class of each sample is arbitrary and uncorrelated. For example, samples of homepages and non-homepages are labeled data because we know the class of the samples from manual classification, whereas random sampling of the Internet provides unlabeled data because the classes of the samples are unknown. In many real-world learning problems including Web page classification, unlabeled and positive, data are widely available whereas acquiring a reasonable sampling of the negative is impossible or expensive because the negative data set is just the complement of the positive one, and, thus, its probability distribution can hardly be approximated [8], [9], [10].

Our goal is to achieve classification accuracy from positive and unlabeled data as high as that from fully labeled (positive and negative) data. Here, we assume that the unlabeled data is unbiased. There are two main challenges in this approach: 1) collecting unbiased unlabeled data from a universal set which can be the entire Internet or any logical or physical domain of Web pages, and 2) achieving classification accuracy from positive and unlabeled data as high as that from labeled data. To address the first issue, we assume it is sufficient to use random sampling to collect unbiased unlabeled data. Random sampling can be done in most databases, warehouses, and search engine databases (e.g., DMOZ) or it can be done independently directly from the Internet.

In this paper, we focus on the second challenge, achieving classification accuracy as high as that from labeled data. The PEBL framework applies an algorithm, called Mapping-Convergence (M-C), which uses the SVM (Support Vector Machine) techniques [11]. In particular, it leverages the marginal property of SVMs to ensure that the classification accuracy from positive and unlabeled data will converge to that from labeled data. We present the details of the SVM properties in Section 4. Our experiment uses the universal set, the entire Internet. Experiment shows that the PEBL framework is able to achieve the classification accuracy as high as using a fully labeled data.

The paper is organized as follows: Section 2 describes different web page classification approaches. Section 3 describes related work on PEBL. Section 4 reviews the marginal properties of SVMs. Section 5 presents the M-C algorithm and data flow during algorithm. Section 6 reports the result of a systematic experimental comparison.

2 WEB PAGES CLASSIFICATION APPROACHES:

Several attempts have been made to categorize the web pages with varying degree of success. The major classifications can be classified into the following broad categories.

1. Manual classification by domain specific experts.
2. Clustering approaches
3. META tags (which server the purpose of document indexing)
4. A combination of document content and META tags
5. Solely on document content
6. Link and content analysis
7. Structure based approach

3 RELATED WORK

H. Yu, J. Han, and K.C. Chang have proposed a PEBL framework and considered commonly used and clearly defined web-based features of web pages for classifying web page into user interesting classes such as “call for papers”, “personal home pages”. PEBL framework applies Mapping-Convergence algorithm[13] to achieve high classification accuracy. They have compared two different methods TSVM and PEBL. TSVM gave 88.11 percentage P-R and PEBL gave 85.89 percentage P-R with seven number of iterations to converge.

4 SVM OVERVIEW:

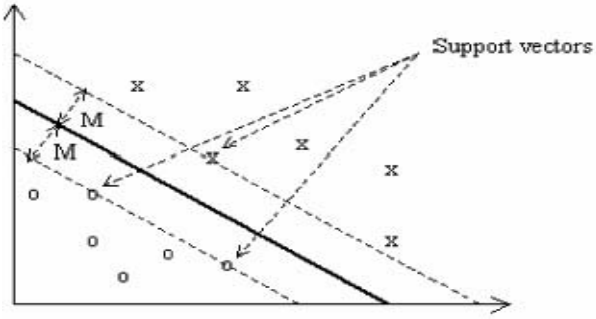

Figure 1: A linear SVM

A linear SVM is a hyper plane that separates a set of positive data from a set of negative data with maximum margin in the feature space. The margin (M) indicates the distance from the hyper plane (class boundary) to the nearest positive and negative data in the feature space. Fig. 1 shows an example of a simple two-dimensional problem that is linearly separable. Each feature corresponds to one dimension in the feature space. The distance from the hyperplane to a data point is determined by the strength of each feature of the data.

5 MAPPING-CONVERGENCE (M-C) ALGORITHM:

Notation: a feature $x_i \in \{0,1\}$, v is a subset of $\{x_1, ..., x_n\}$
f_p^i: frequency of feature x_i in positive class
f_u^i: frequency of feature x_i in unlabeled data set
θ: decision threshold (normally set to 1 but can be adjusted)
Input: POS, U
Output: output $h = \sum_{x_i \in v} x_i$
Algorithm: $v = null$
for $i = 0$ to n do
if $\frac{f_p^i}{f_u^i} > \theta$ then $v = v \cup x_i$
return $h = \sum_{x_i \in v} x_i$

Figure 2: Monotone-disjunction learning for the algorithm Ψ1

Input: positive training examples POS, unlabeled data set U
Output: hypothesis at each iteration $h'_1, ..., h'_k$

Ψ_1: an algorithm identifying "strong negatives" from U
Ψ_2: a supervised learning algorithm that maximizes margin (e.g. SVM)

Algorithm:
1. Use Ψ_1 to construct a classifier h from POS and U which classifies only "strong negatives" as negative and the others as positive
2. Classify U by h
   * N_1:= examples from U classified as negative by h
   * P_1:= examples from U classified as positive by h
3. Set $NEG := \emptyset$ and $i := 1$
4. Do loop until $N_i = \emptyset$
   4.1. $NEG := NEG \cup N_i$
   4.2. Use Ψ_2 to construct h'_i from POS and NEG
   4.3. Classify P_i by h'_i
      * N_{i+1}:= examples from P_i classified as negative by h'_i
      * P_{i+1}:= examples from P_i classified as positive by h'_i
   4.4. $i := i + 1$
5. Return $(h'_1, ..., h'_k)$

Figure 3: Mapping-Convergence Algorithm (M-C)

Figure 2. presents monotone disjunction algorithm which is mapping stage of M-C algorithm. M-C algorithm presented in Figure 3. M-C runs in two stages: the mapping stage and convergence stage. In the mapping stage, the algorithm uses a weak classifier (e.g., a rule-based learner) that draws an initial approximation of "strong" negative data. Based on the initial approximation, the convergence stage runs in iteration using a second base classifier (e.g., SVM) that maximizes margin to make progressively better approximation of negative data. Thus, the class boundary eventually converges to the true boundary of the positive class in the feature space. We present the conceptual data flow in M-C algorithm in Figure 4.

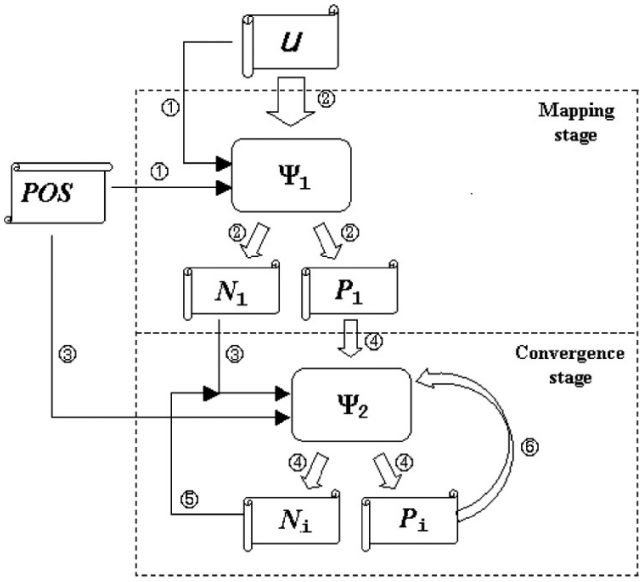

Figure 4. Data flow diagram of the mapping convergence (M-C) algorithm

6 RESULTS:

6.1 Data Sets and Experimental Methodology:

The universal set in our experiment is the Internet. To collect random samples of Internet pages, we used DMOZ[1] which is a free open directory of the Web containing millions of Web pages. A random sampling of a search engine database such as DMOZ[2] is sufficient to construct an unbiased sample of the Internet. We randomly selected 1,878 pages from DMOZ to collect unbiased unlabeled data. We also manually collected 337 personal homepages to classify the corresponding class. We used 180 pages for training and the other 157 pages for testing. For testing negative data (for evaluating the classifier), we manually collected 291 non-homepages. (We collected negative data just for evaluating the classifier we construct. The PEBL does not require collecting negative data to construct classifier.)

We extracted features from different parts of a page —URL, title, headings, anchor-text, other text, Image. We have checked term"~" in URL, or a word "homepage" in title, "home" or "my" word in headings. Frequency of words "I","my" , "myself" , "like" in other text has been calculated. The words like "About Me","photo galleary","personal interests" are compared with anchor text. The information in <img> tag is used to get the size of the image. Generally a photo of a person in the web page is more important information about a personal home page class. By extracting image feature in the web page results are encouraging. In our experiment we omitted meta-tags feature because of contents of feature could not contribute in major sense to the web page classification . Thus by applying mapping-convergence algorithm[13] by taking into consideration the enhanced feature vector we achieved 98.00 percentage P-R for both TSVM and PEBL. As feature vector is more perfect, more number of strong negatives are extracted at first time. Likewise over the iterations negatives are extracted and boundary converged to true boundary at fifth iteration. For SVM implementation, we used .NET version of LIBSVM. We used Gaussian kernels because of its high accuracy. Both TSVM and M-C show better performance with Gaussian kernels.

We report the result with precision-recall breakeven point (P-R), a standard measure for binary classification. Precision and recall are defined as:

$$\text{Precision} = \frac{\#\ \textit{Of correct positive prediction}}{\#\ \textit{of positive prediction}}$$

$$\text{Recall} = \frac{\#\ \text{Of correct positive prediction}}{\#\ \text{of positive prediction}}$$

The precision-recall breakeven point (P-R) is defined as the precision and recall value at which the two are equal. We calculated precision-recall breakeven point by averaging precision and recall value for each iteration [12].

6.2 Results Analysis:

We compare two different methods: TSVM and PEBL. (See Table 1 for the full names.). We first constructed an SVM from positive (POS) and unlabeled data (U) using PEBL. On the other hand, we manually classified the unlabeled data (U) to extract unbiased negatives from them, and then built a TSVM (Traditional SVM) trained from POS and those unbiased

[2] http://www.dmoz.org

negatives. We tested the same testing documents using those two methods. Table 1 shows the P-R (precision-recall breakeven points) of each method, and also the number of iterations to converge in the case of the PEBL. In most cases, PEBL without negative training data performs almost as well as TSVM with manually labeled training data. When we use PEBL without doing the manual classification, it gives 98.00 percent P-R. Figure 5 and 6 show the details of convergence (of the induced negative training data and corresponding P-R) at each iteration in the experiment of the universal set, the Internet. The number of induced negatives at the first iteration is around 599 (shown in the graph), and the P-R of the SVM trained from positive and those 599 negatives is 0.67 (shown in the graph). At the second iteration, the number of induced negatives is around 1280, and the SVM trained from positive and those 1280 negatives gives 0.70 P-R. Likewise, at the fifth iteration, the number of induced negatives is almost the same as the number of real unbiased negatives.

Table 1:

Precision-Recall Breakeven Points (P-R) Showing Performance of PEBL (Positive Example-Based Framework) and TSVM (Traditional SVM Trained from Manually Labeled Data

U	Class	TSVM	PEBL
Internet	Personal Home Page	98.00	98.00 (6)

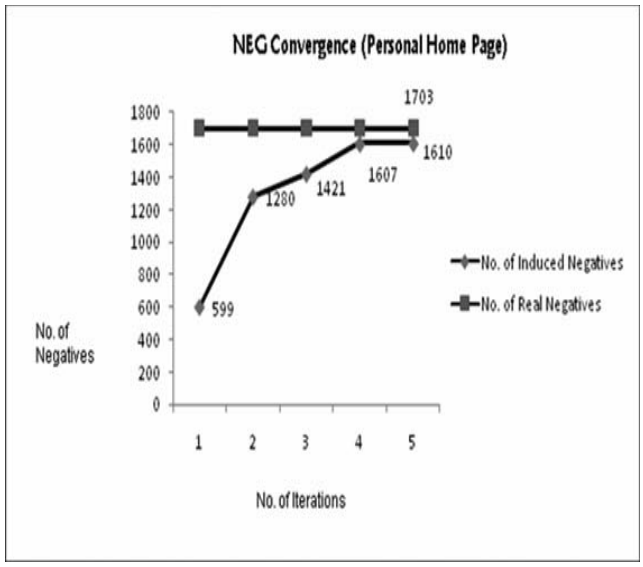

Figure 5. NEG convergence

Comment:

At the last iteration number of induced negatives reached to the level of real unbiased negatives manually extracted from unlabelled data to train TSVM. This graph proved that PEBL framework without negative training data perform as well as TSVM with manually labeled negative data.

Comment:

At the last iteration, when there were no negative values predicted by SVM, the P-R value is 0.79 which is high. The SVM constructed by this iterative method gives 98.00 P-R after testing. This is same as, TSVM constructed from manually labeled negative data.

7 CONCLUSION

The PEBL framework as a classifier that does not rely on negative labeled data, will be easily deployable, which makes it applicable in many practical applications like focused crawler, pattern recognition etc.

Exploiting structural information in web page such as size and location of image can contribute to better accuracy. In our implementation, based on size of image some assumptions have been made. By applying face detection algorithm to captured images, more accuracy can be achieved. Also the information about location of links can be exploited.

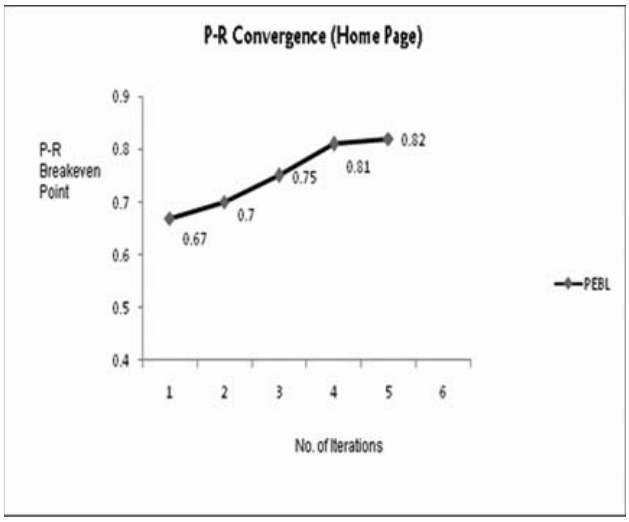

Figure 6. P-R convergence

8 REFERENCES

[1] H. Chen, C. Schuffels, and R. Orwig, "Internet Categorization and Search: A Machine Learning Approach," J. Visual Comm. and Image Representation, vol. 7, pp. 88-102, 1996.

[2] H. Mase, "Experiments on Automatic Web Page Categorization for IR System," technical report, Stanford Univ., Stanford, Calif.,1998.

[3] E.J. Glover, G.W. Flake, and S. Lawrence, "Improving Category Specific Web Search by Learning Query Modifications," Proc. 2001Symp. Applications and the Internet (SAINT '01), pp. 23-31, 2001.

[4] A. Kruger, C.L. Giles, and E. Glover, "Deadliner: Building a New Niche Search Engine," Proc. Ninth Int'l Conf. Information and Knowledge Management (CIKM '00), pp. 272-281, 2000.

[5] E.N. Mayoraz, "Multiclass Classification with Pairwise Coupled Neural Networks or Support Vector Machines," Proc. Int'l Conf.Artificial Neural Network (ICANN '01), pp. 314-321, 2001.

[6] E.L. Allwein, R.E. Schapire, and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin

Classifiers," J. Machine Learning Research, vol. 1, pp. 113-141, 2000.

[7] S. Dumais and H. Chen, "Hierarchical Classification of Web Content," Proc. 23rd ACM Int'l Conf. Research and Development in Information Retrieval (SIGIR '00), pp. 256-263, 2000.

[8] H. Yu, J. Han, and K.C.-C. Chang, "PEBL: Positive-Example Based Learning for Web Page Classification Using SVM," Proc. Eighth Int'l Conf. Knowledge Discovery and Data Mining (KDD '02), pp. 239-248, 2002.

[9] F. Letouzey, F. Denis, and R. Gilleron, "Learning from Positive and Unlabeled Examples," Proc. 11th Int'l Conf. Algorithmic Learning Theory (ALT '00), 2000.

[10] F. DeComite, F. Denis, and R. Gilleron, "Positive and Unlabeled Examples Help Learning," Proc. 11th Int'l Conf. Algorithmic Learning Theory (ALT '99), pp. 219-230, 1999.

[11] C. Cortes and V. Vapnik, "Support Vector Networks," Machine Learning, vol. 30, no. 3, pp. 273-297, 1995.

[12] Using SVMs for Text Categorization by Susan Dumais Decision Theory and Adaptive Systems Group Microsoft Research