Proceedings of the IASTED International Conference
DATABASES AND APPLICATIONS
February 17-19, 2004, Innsbruck, Austria

Classification of Websites as Sets of Feature Vectors

Hans-Peter Kriegel
Institute for Computer Science
University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
kriegel@dbs.informatik.uni-muenchen.de

Matthias Schubert
Institute for Computer Science
University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
schubert@dbs.informatik.uni-muenchen.de

ABSTRACT
The world wide web is the largest source for all kind of information currently available. Due to its enormous size retrieving relevant information is a difficult task for which users often rely on directory services. A directory service provides a huge topic tree containing links for each topic. Due to the generality of the topics most links direct to websites or domains, instead of single webpages. For maintaining a directory service, automatic classification of new websites into the topics of the tree would be very beneficial. Therefore, this paper introduces a new approach to website classification that is based on sets of feature vectors. Compared to previous approaches our new method requires no preprocessing, but provides high accuracy in efficient time.

KEY WORDS
web mining, website classification, sets of feature vectors.

1 Introduction

The world wide web is the largest source for all kinds of information that is available to a very high number of people all over the world. Due to the vast amount of information and the distributed organization, tracking topic specific information can be a challenging problem. To solve this problem two kinds of services have emerged helping to retrieve information: search engines like Google [1] and directory services like DMOZ [2] or yahoo! [3].

Whereas a search engine basically consists of a large database of words with respect to the web documents they occur in, a directory service consists of a large topic tree organizing a huge, predefined set of possible query topics. For each topic the directory service provides a webpage containing web content specific to the topic. Directory services might link to single webpages, but more often they link to complete websites. A website is a set of webpages published by the same person, group or institution referring to a common topic. For example, most companies offer a website consisting of multiple webpages that describe different information like services, vacancies or locations. Thus, the information presented on a website might belong to various subtopics, but follows the common purpose to describe the company.

Finding websites instead of single webpages has various applications. Companies might look for potential customers, suppliers or competitors. Another example is a user that wants to buy a new computer and then tries to find a computer retailer. Afterwards he can look for offers on computers within the sites of computer retailers only. Let us note that these queries might also be solved by search engines, but the precision of the search results tends to be very low, because the search engine retrieves any page containing the name of the wanted computer or the terms used to describe it.

Since directory services are usually maintained manually or semi-automatically, extending a directory service is rather time-consuming. Clearly, a system that automatically determines the correct class of a given website would significantly speed up the maintainance of a directory service . Thus, a classifier that determines the most likely class for a website is an important improvement.

In [4], we introduced a first approach on website classification. This approach derived a so-called topic frequency vector for each website and afterwards used this vector for classification. However, this former approach had an important drawback: generating training data in the space of topic frequency vectors is very time consuming and requires a lot of manual interaction.

In this paper, we therefore introduce a new approach that is based on describing a website as a set of feature vectors. By choosing this representation, we avoid the effort spent on deriving topic frequency vectors and still achieve high classification accuracy. We employ kNN-classification and use clustering to increase efficiency and performance.

The rest of the paper is organized as follows. Section 2 summarizes some work on the classification of texts and webpages. Section 3 formally introduces the problem of website classification and describes our former approach. Section 4 presents our new approach to website classification based on kNN-classification and sets of feature vectors. The next section provides the results of our experimental evaluation comparing both approaches in terms of classification accuracy and efficiency. The last section summarizes the paper and gives some directions for future research.

2 Related Work

In this section, we briefly review related work on text classification, in particular classification of webpages. Text classification has been an active area of research for many years. All methods of text classification require several steps of preprocessing. First, any non-textual information such as HTML-tags and punctuation is removed from the documents. Then, stopwords such as "I", "am", "and" etc. are also removed. Typically, the terms are reduced to their basic stem applying a stemming algorithm. Most text classification algorithms rely on the so-called vector-space model. In this model, each text document is represented by a vector of frequencies of the remaining terms. The term frequencies may be weighted by the inverse document frequency, i.e. a term occurring in fewer documents obtains a larger weight. Finally, the document vectors are normalized to unit length to allow comparison of documents of different lengths. The vector-space has a very high dimensionality since even after preprocessing there are typically still several thousands of terms. Due to the high dimensionality, most frequencies are zero for any single document and many of the standard classification methods perform poorly.

However, methods that do not suffer so much from high dimensionalities have been very successful in text classification, such as naive Bayes [5] and support vector machines [6, 5]. Another approach to text classification is centroid based k-nearest neighbor classification [7], which is related to our new approach of website classification. The idea is to summarize each class by the centroid of its training documents. Thus, kNN-classification is considerably faster and more accurate. However, [7] discusses classification of single text documents and only mentions the possibility to cluster the training set for multiple centroids in its conclusions.

Another task in text classification is to determine the subset of a topic hierarchy that is contained in a single document [8]. Though this approach bears a similar name to the website classifier described in the next section and employs a "topic frequency representation" at some stage, it is only applicable for the classification of single text documents and not sets of webpages, i.e. websites.

While most of the above methods have been applied to pure text documents, an increasing number of publications especially deals with the classification of webpages. Several authors have proposed methods to exploit the hyperlinks to improve better classification accuracy. [9] introduces several methods of relational learning considering the existence of links to webpages of specific classes. [10] presents techniques for using the class labels and the text of neighboring (i.e., linked) webpages. However, all these methods aim at classifying single webpages, not complete websites.

3 Website Classification and the Topic Frequency Vector Approach

As mentioned before a website is a set of webpages that is published by the same person, group or institution and usually serves a common purpose, e.g. to present an complete organization or company. Since it is difficult to exactly identify websites based on this very general definition, in this paper, we equate a website with the set of webpages belonging to a domain (e.g. "www.lmu.de"). This easier-to-process definition holds for most real world websites. Only very large websites are spread over more than one domain and most relevant organizations rent their own domain name for better identification. Let us note that our classification methods are not bound to this definition, but can handle other more complex definitions too, as long as there is a set of webpages representing a website.

To classify complete websites, we described several methods in [4]. The simplest approach summarizes all webpages within a website into one single feature vector and afterwards classifies this vector. However, this approach performed very poorly and thus, classifying a website in the same way as a single text document is an inadequate approach to website classification. The approach offering the best results is named *topic frequency vector* approach (TFV-approach). To employ the TVF-approach for website classification, several preprocessing steps are needed. First of all, a set of training websites has to be acquired. Employing a directory service this is easy, since there are already leaves in the topic tree mostly linking to relevant sites. To attain the webpages representing these websites, one could download all pages, but it is usually enough to restrict the number of webpages to a maximum of 120. During classification this restriction can be achieved by an incremental classifier as described below.

The next step is to find a proper set of page classes for each website class. The set of page classes should contain classes that describe webpages being typical to occur in a website of a certain class. For example, "computer repair service" might be a meaningful page class for the site class "computer retailer". To distinguish all websites belonging to some other class, we employ a global "other" site class that is not specified any further. This "other" site class is described by one "other" page class only. After definition, we have to generate training examples for each of the page classes. Therefore, we examine the pages within the training websites and manually label the pages with the proper page class. Thus, we receive the training set for the page classes, additionally to the needed set of training websites. Both of these steps tend to be very time consuming, since there is no approved general way for automatic generation of classes and training examples. For example, determining page classes and labelling enough training pages took about 3 days for the data used in the evaluation of [4].

After generating the training set for the page classes, we train a naive Bayes classifier, that is capable to label new unknown webpages with the most likely page class. To

classify a website, we derive the so-called topic frequency vectors (TFVs).

The vector-space of topic frequencies is spanned by the set of page classes. The idea is to count the occurrences of each page class in a site. Thus, a TFV gives an brief overview of the page classes that occur within a site. For classification, we employ a second naive Bayes classifier trained on the TFVs derived from the training websites. Classification of an unknown website is achieved in two steps. First, we classify the webpages of the website and therefore, derive the TVF. Afterwards we employ the second classifier to predict a site class from this TFV. In [4], we additionally described a variant for incremental classification that does not employ all webpages of a website. The idea is to read a page from the homepage –the page given by the domain name only–, follow the links and measure the quality of each path leading away from it. If this quality is not "relevant" enough, we prune the path. After all pathes are pruned, the treated portion of the website is usually a good representation for the complete site. In this incremental variant, the prediction of the second classifier is calculated during the traversal of the website and is used to decide the quality of each path to be pruned. For more details please read [4].

4 Classification of Websites as Sets of Feature Vectors

4.1 The K-Nearest Neighbor Classifier

Turning away from the concept of page classes leaves us without an appropriate feature transformation for websites. Thus, there is no feature space that most of the well-established classification methods like Bayes classifiers or SVMs require. Therefore, we adopt the paradigm of k-nearest neighbor (kNN) classification that only assumes a pairwise distance function. For the classification of an unknown object, a basic kNN-classifier performs a kNN-query on the training database and returns the majority class of the returned k nearest neighbors of the given object. The key to the effectiveness of kNN-classification is an intuitive distance function. Since the content of each single page $p \in W$ can be represented by a feature vector of term frequencies, a whole website is represented by a set of feature vectors. Several distance measures for sets of vectors in a metric space have been introduced in the literature [11, 12]. From these distance measures, the Sum of Minimum Distances (SMD) [11] most adequately reflects the intuitive notion of similarity between two websites. In the context of website classification, it can be defined in the following way. Let W_1,W_2 be two websites and let $f : P \rightarrow N^d$ be a feature transformation that returns the feature vector of a page $p \in P$ where P is the set of all webpages. Furthermore, let $d(x, y)$ be a distance measure on feature vectors. The SMD of W_1 and W_2 is given by:

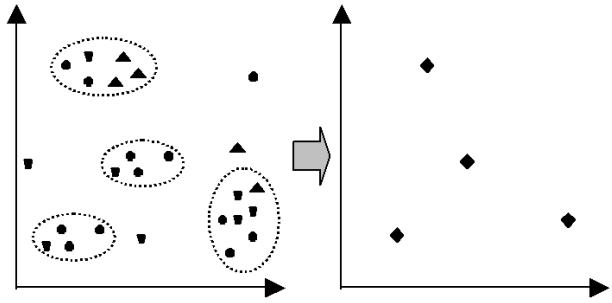

Figure 1. Centroid set of a sample website class.

$$SMD(W_1, W_2) = \frac{\sum_{v_i \in W_1} \min_{w_j \in W_2} d(f(v_i), f(w_j)) + \sum_{w_j \in W_2} \min_{v_i \in W_1} d(f(v_i), f(w_j))}{|W_1| + |W_2|}$$

The idea of SMD is to map every element of both sets to the closest element in the other set. This means that several pages belonging to the website W_1 might be mapped to the same page in the other site W_2 or vice versa. This is quite adequate for websites since the number of different webpages describing the same information may vary among different websites belonging to the same class. Furthermore, sites of very related content but varying size will become very similar w.r.t. SMD since the cardinality of the set is not considered. The SMD is reflexive and symmetric, but does not fulfill the triangle inequality, i.e. it is not a metric. The SMD distance calculation for a pair of websites W_1 and W_2 has a quadratic runtime complexity $O(w^2)$ where w denotes the maximum of the numbers of webpages of W_1 and W_2. As distance measure between the single feature vectors, we use the cosine coefficient which is well-established for text data.

4.2 Improving Efficiency Using Centroid Sets

Though the basic kNN-classifier is very accurate, the computational cost for classifying a website is very high. The standard approach of speeding-up kNN-queries by using a multi-dimensional index structure such as the X-tree [13] is infeasible because SMD is not metric. An alternative to speed-up classification is to reduce the size of the training database to one representative per website class.

Therefore, we introduce centroid sets to summarize and represent a website class. The idea is that each website class provides several groups of webpages that are somehow related and can be summarized by one common representative. Let us note that these groups of pages are somewhat similar to the manually assigned page classes of the TFV approach, but refer to one site class only and are derivable without manual interaction. In [7], the authors show

that the centroid of several text documents is a useful representative for a complete class in terms of kNN- classification. Furthermore, the paper mentions clustering to treat multi-modal classes in its conclusion chapter. However, to our knowledge the authors did not follow this direction any further. To speed up, website classification and find meaningful descriptions of websites, we take up this idea.

Given some groups of related elements, we calculate one mean vector for the training pages of each group and define the centroid set for a website class as the set of all such mean vectors: Let S be a set of sets s_i with vectors v_{j,s_i} and let $\pi_l(s_i) = \{v | g(v) = l \; \forall v \in s_i\}$ be the restriction of s_i to group l where g is a mapping from a vector v to a group $l \in G$, the set of all groups. Then the centroid set CS of S is defined as:

$$CS(S) = \left[c_j \mid \forall j \in G, c_j = \frac{1}{|\bigcup_{\forall i} \pi_l(s_i)|} \cdot \sum_{x \in \bigcup_{\forall i} \pi_l(s_i)} x \right]$$

Figure 1 illustrates the centroid set for a sample website class using a two-dimensional feature space for the webpages. The remaining problem is now to determine the grouping within the training pages of a website class. Fortunately, the task of identifying similar groups of instances within a database of feature vectors is known as clustering. Though there are many established clustering algorithms, the choice of a suitable one for our problem is limited by two requirements. First, the number of clusters should be determined by the clustering algorithm. Since there is no apriori-knowledge about the groups within a site class, we are unable to input the number of clusters. Second, the cluster algorithm should be able to deal with noise. In our context, noise represents webpages that are uncommon for the class of websites they occur in. To provide relevant generalization, noise should not be considered within the constructed centroid set. We choose GDBSCAN [13] to group the training pages within each website class, because of its ability to find an arbitrary number of clusters and to filter out noise. The parameters of GDBSCAN are used to adjust the number of centroids per class and to control how much noise is eliminated. To conclude, the centroid set for a website class C_i is derived as follows :

1. Join the (feature vectors of the) webpages found in the training websites of class C_i into one set.
2. Determine clusters in this set of feature vectors using GDBSCAN.
3. For each cluster, calculate the centroid and insert it into the centroid set of class C_i.

4.3 Incremental Classification

When using this compact representation of website classes, the website classifier has to calculate the SMD between a test website and all website classes, each represented by a centroid set. Now the following problem occurs, especially when classifying objects belonging to the obligatory "other" class that is trained on a random mix of different types of websites not further distinguished. Consider a website consisting of a few webpages only that even can be part of the training set. The portion of the SMD that sums up the distance of the webpages of this website to the centroid set will be rather small, which is intended, since it is part of the training set. However, the second sum consisting of the distances of all centroids to the few specialized pages in the website will be very large, since we might have many other topics in the representative object that are not obligatory for websites belonging to the site class. This contradicts our intuition because an instance belonging to a class should be very similar to the class representative. To avoid this effect, we replace the SMD by the half-SMD (HSMD) as a more adequate distance measure for calculating the distance of test websites to centroid sets. Let W be a website, let C be a centroid set and let $f : P \rightarrow N^d$ be a transformation that returns the feature vector of $p \in P$ where P is the set of all pages and centroids respectively. Furthermore, let $dist(x, y)$ be a distance measure on feature vectors. The HSMD of W to C is given by:

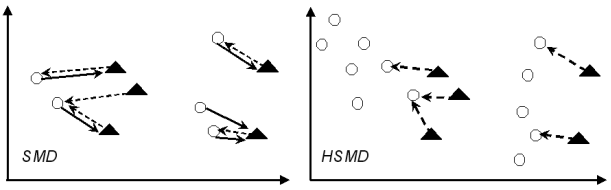

Figure 2. Illustration of SMD(left) and HSMD(right).

$$HSMD(W, C) = \frac{\sum_{v_i \in W} \min_{w_j \in C} d(f(v_i), f(w_j))}{|W|}$$

A further advantage of HSMD is a faster calculation, especially for incremental classification. Classifying a website incrementally using the introduced methods of NN-classification is basically possible for all variants mentioned above. Since the view of a website as a set of pages allows us to treat the already retrieved part as a website itself, the distances can be calculated for each subset, too. However, the variant using centroid sets and HSMD is suited best for incremental classification, due to the following reasons. By limiting the training set to just one instance per site class, we can store the HSMD values of the subset retrieved so far with each centroid set. Since the HSMD only considers the distance from the page to its nearest neighbor in the representative object, the distance can be summed up during the traversal. Thus, the effort for extending the classification to an additional webpage is limited to one NN-query for each class. Pruning the website to a representative is done as in [4], by limiting the path length from the home page.

	TFV		5-NN		Cent.Cl.	
class	pre.	rec.	pre.	rec.	pre.	rec.
busin.sch.	0.74	0.98	0.75	0.89	0.87	0.96
horse deal.	0.80	0.86	0.95	0.78	0.95	0.78
game retail.	0.75	0.75	0.92	0.60	0.77	0.85
ghosts	0.50	0.92	0.60	0.75	0.90	0.75
astron.	0.63	0.92	0.79	0.88	0.88	0.88
snowboard	0.61	0.75	0.86	0.60	0.93	0.70
Acc. 7-Cl.	0.65		0.76		0.81	

Table 1. Comparison of precision and recall. Last line: Accuracy for the 7-class problem.

5 Experimental Evaluation

For our experiments, we employed the yahoo! [3] hierarchy to provide classes and corresponding training sites. In our test bed we chose 6 different website classes and built an additional "other"-class from a randomly chosen mixture of other yahoo! classes. Our training database consisted of 86 websites for the category "other" and between 12 and 47 example sites for the 6 classes. The total number of sites was 234 comprising a total of about 18,000 single webpages. The classifiers were implemented in Java and tested on a workstation equipped with 2 Pentium 4 processors (2,4 GHz) and 4 Gb main memory.

The first set of experiments tested precision and recall for each of the 6 website classes for the two-class case only. The comparison partners included a multinomial naive Bayes classifier on TFVs as in [4] and a basic 5-NN classifier using SMD (5-NN for short). Furthermore, we tested an incremental 1-NN-classifier employing centroid sets and HSMD (centroids classifier). Without having appropriate page classes and training pages, we used the site classes also as page classes to generate TFVs.

The topics of the single webpages were determined by another naive Bayes classifier. Note that only the TFV and the centroids classifier employed incremental classification, using only a reduced portion of the website as shown in [4]. A second set of experiments investigated the ability of the above three classification methods to handle more than one class, by giving the complete training set to the classifier as a 7-class problem. Both experiments used 10-fold cross validation. The results displayed in table 1 document the ability of the basic 5-NN classifier to provide good precision and recall without using page classes. The TFV-classifier using the provisional TFVs still shows acceptable results, but the 5-NN-classifier achieves a better trade-off between precision and recall in most of the cases. The incremental centroids classifier provided very good accuracy and outperformed the other two classifiers. Let us note that the results of the incremental centroids classifier displayed in table table 1 do not belong to the parameter setting offering the best accuracy, but to the setting with the best trade-off between classification time and accuracy.

website class	5-NN	Cent.Cl.	TFV
business school	39.16	0.37	0.12
horse dealer	22.40	0.28	0.02
game retailer	22.27	0.34	0.09
ghosts	28.67	0.38	0.03
astronomy	36.99	0.42	0.24
snowboarding	31.59	0.36	0.04

Table 2. Classification time in seconds per website for the two class problems.

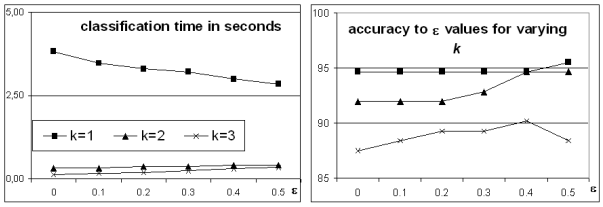

Figure 3. Accuracy and classification time depending on the parameter setting for GBDBSCAN for the Astronomy example.

The accuracies achieved for the 7-class problem, listed in the last line of table 1, follow the trend observed in the 2-class problems and underline the capability of the centroids classifier to handle larger classification problems. The accuracy, i.e. the percentage of correctly classified instances with respect to all tested instances, is used to measure this experiment, to give an compact view to the all-over performance of the classifier.

In table 2, we display the average time spent on the classification of one website for the 2-class problems. The results clearly show that the basic 5-NN classifier takes a considerable amount of time for classification. On the other hand, the incremental centroids classifier performed pretty well compared to the extremely fast TFV-approach and offered a speed-up of about 100 compared to the basic 5-NN approach. This enormous speed up is due to the small average number of centroids (about 180 per centroid set) and the use of incremental classification considering only few pages of a website (about 20) for very accurate classification. Summarizing the centroids classifier offered remarkable classification accuracy in efficient time.

A third experiment investigated the effects of the parameter setting of GDBSCAN [13], the clustering algorithm used to derive the centroid sets. For the astronomy example, Figure 3 shows the dependency of accuracy and classification time on the number k of neighbors needed to define a core point and the radius ϵ. The shape of the graph indicates that the influence of the radius is very stable within the interval from 0 to 0.5 which is half of the possible target interval of the cosine coefficient. The influence of the number of neighbors k on the other hand, shows an obvious decrease of accuracy for $k = 3$ and no signif-

icant efficiency gain for $k < 2$. Therefore, setting $k = 2$ and $\epsilon = 0.4$ offered a good trade-off between classification time and accuracy.

6 Conclusion

In this paper, we treat website classification as an emerging possibility to automatically maintain directory services. We therefore treat websites as sets of feature vectors. This new representation is much easier to derive and does not require defining page classes and manually labelling of training examples as it was demanded in previous approaches. We employ kNN-classification to predict site classes and use the sum of minimum distances (SMD) to relate the websites to each other. To further increase classification time and accuracy, we derive so-called centroid sets to represent a class of websites. Furthermore, we restrict the SMD to the half-SMD, which is more adequate to compare websites to centroid sets and is easier to compute during incremental classification. In our experimental evaluation, we demonstrate the capability of our new method to produce high classification accuracy in efficient time.

In our future work, we want to implement a topic specific directory service that is maintained automatically. Thus, new websites are classified automatically to the set of classes they belong to. Therefore, we plan to adapt methods of hierarchical classification and evaluate website classification on a much broader basis. Furthermore, we plan to grow the class tree automatically. To achieve this extension a leaf node of the tree has to be split into a set of new subclasses after a certain number of entries is reached. To find the set of new subclasses we plan to cluster the websites into a meaningful grouping. Thus, we can keep the transparency of a node at an acceptable level.

References

[1] Google: web search engine. (http://www.google.com)

[2] DMOZ: open directory project. (http://dmoz.org)

[3] Yahoo!: web directory service. (http://www.yahoo.com)

[4] Ester, M., Kriegel, H.P., Schubert, M.: "Website Mining : A new way to spot Competitors, Customers and Suppliers in the World Wide Web", *Proc. 8th ACM SIGKDD 02* , Edmonton, Alberta, CA, 2002, 249–258.

[5] Yang, Y., Liu, X.: "A Re-Examination of Text Categorization Methods", *Proc. 22nd ACM SIGIR*, Berkley, US, 1999,42–49.

[6] Joachims, T.: "Text Categorization with Suport Vector Machines: Learning with Many Relevant Features", *Proc. 10th ECML* , Chemnitz, Germany., Volume 1398 of LNCS., 1998, 137–142

[7] Han, E.H., Karypis, G.: "Centroid-Based Document Classification: Analysis and Experimental Results", *Proc. 4th PKDD'00*, Lyon, France, 2000, 424–431

[8] Gelbukh, A., Sidorov, A., Guzman-Arenas: "Use of a weighted topic hierarchy for document classification", *TSD.99*, Lecture Notes in AI N1692, Springer-Verlag, 1999, 130–135

[9] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: "Learning to Construct Knowledge Bases from the World Wide Web", *Art. Int.* **118**, 1999, 69–113

[10] Chakrabarti, S., Dom, B., Indyk, P.: "Enhanced hypertext categorization using hyperlinks", *Proc. 17th ACM SIGMOD*,New York, US, 1998, 307–318

[11] Eiter, T., Mannila, H.: "Distance Measures for Point Sets and Their Computation", *Acta Informatica*, **34**, 1997, 103–133

[12] Ramon, J., Bruynooghe, M.: "A polynomial time computable metric between points sets", *Acta Informatica*, **37**, 2001, 765–780

[13] Sander, J., Ester, M., Kriegel, H.P., Xu, X.: "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications", *Data Mining and Knowledge Discovery*, **2**, 1998, 169–194