GE-CKO: A method to optimize composite kernels for Web page classification

Jian-Tao Sun[1], Ben-Yu Zhang[2], Zheng Chen[2], Yu-Chang Lu[1], Chun-Yi Shi[1], Wei-Ying Ma[2]

[1]Department of Computer Science
TsingHua University
Beijing 100084, P.R.China
sjt@mails.tsinghua.edu.cn {lyc,scy}@tsinghua.edu.cn

[2]Microsoft Research Asia
5F, Sigma Center, 49 Zhichun Road
Beijing 100080, P.R.China
{byzhang,zhengc,wyma}@microsoft.com

Abstract

Most of current researches on Web page classification focus on leveraging heterogeneous features such as plain text, hyperlinks and anchor texts in an effective and efficient way. Composite kernel method is one topic of interest among them. It first selects a bunch of initial kernels, each of which is determined separately by a certain type of features. Then a classifier is trained based on a linear combination of these kernels. In this paper, we propose an effective way to optimize the linear combination of kernels. We proved that this problem is equivalent to solving a generalized eigenvalue problem. And the weight vector of the kernels is the eigenvector associated with the largest eigenvalue. A support vector machine (SVM) classifier is then trained based on this optimized combination of kernels. Our experiment on the WebKB dataset has shown the effectiveness of our proposed method.

1. Introduction

In recent years, the exponential growth of WWW has brought a revolutionary change to human life. At the same time, to facilitate user browsing such a large knowledge pool has become an increasingly great challenge to information retrieval. Some websites manually maintain a hierarchical structure, such as Yahoo! (http://www.yahoo.com/) and the Open Directory Project (http://dmoz.org/) do. While it provides high accuracy, such manual maintenance is very expensive. To automatically include new emerging pages into website structures, web page classification becomes a hot research topic in Information Retrieval (IR).

Web page classification is more than typical text based classification methods. The main reason is that Web pages have far richer structures, which result in heterogeneous features other than plain text, such as hypertext tags, metadata and hyperlinks. Moreover, these features may play different roles in predicting the page's category. Many methods have been proposed to make use of these heterogeneous features for classification, which largely fall into three categories:

- Applying traditional text classification methods on the concatenation of all words, such as pure text, metadata, in-link anchor text, etc [10, 9, 17]. This method is simple but information like hyperlink relations is not fully exploited.
- Using separate features to train individual classifiers, whose outputs are combined to give a final category for each page[3, 20]. However, this method has to estimate the performance of each classifier before it can give a final judgment.
- Exploiting hyperlink topologies between Web pages. Such algorithms include composite kernel method [13], iterative labeling method [5] and relational learning method [19],etc.

In this paper, we focus on the composite kernel method for Web page classification. [14] create two kernels based on text and hyperlink features respectively, and linearly combine them into a composite kernel. An SVM

Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)
0-7695-2100-2/04 $ 20.00 IEEE

classifier trained using the final kernel has good generalization performance. However, [14] has two requisitions before it can achieve better generalization performance than single kernel method: assume each candidate kernel is used to construct an SVM classifier, then first, these classifiers should have approximately the same performance individually, second, their Support Vectors are different. Under these pre-requisitions, the individual kernels are equally treated, i.e. they are assigned with the same weight.

In this paper, the problem we are attacking is how to optimize composite kernel combinations for the Web page classification task. A more general composition kernel method called GE-CKO (Generalized Eigenvalue based Composite Kernel Optimization) was proposed for classifying Web pages. We do not require the candidate kernels to have similar performance. We proved that by solving a generalized eigenvalue problem and using the eigenvector associated with the largest eigenvalue as the weight vector of the initial kernels, the combination can be optimized. The state-of-the-art SVM classifier is used because it promises a very good generalization performance. It is valuable to highlight some points here:

1. In our proposed method, the individual kernel's weight can automatically be tuned. No prior knowledge on feature types is required.
2. We investigated the composite kernel optimization problem and converted it to a generalized eigenvalue problem based on a theoretical analysis.
3. The dimension of the matrix in the generalized eigenvalue problem is equal to the number of individual kernels, thus our method can efficiently handle many types of features.
4. Our GE-CKO is simple to implement and can automatically adapt itself to heterogeneous features.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to kernel and kernel alignment methods. In Section 3, we describe the problem and propose our theoretical analysis. Meanwhile, the GE-CKO algorithm is presented. In Section 4, the experimental results on the WebKB dataset are shown as well as some discussions are presented. Section 5 describes the related works. The conclusions and future work are discussed in Section 6.

2. Kernel Methods

In this section, we give a brief introduction to kernel methods and kernel alignment. Kernel based methods have been widely used for their simplicity and good generalization performance [6, 22]. One key problem of these methods is the kernel construction for a particular task. To measure the fitness between a kernel and the learning task, kernel alignment is proposed which is also used in our paper as an optimization criterion.

2.1. Kernel and Kernel Combination

Kernel methods use nonlinear transformations to embed data of input space $\mathcal{X}$ into high dimensional feature space $\mathcal{F}$ [6, 22]. Algorithms like SVM that make use of inner products in $\mathcal{F}$ use kernel functions to directly calculate the inner products in $\mathcal{X}$. According to Mercer's theorem, if a function produces symmetric and positive semi-definite matrices for any finite set, it is a valid kernel [6]. For kernel methods, an important step is to construct the kernel matrix (or Gram matrix), because all the information needed by the learning machine is contained in the kernel matrix.

One approach to construct a complex kernel is to combine a bunch of individual ones. In particular, let k_i,$i = 1 \cdots t$, be kernels, the linear combination $\sum_{i=1}^{t} \alpha_i k_i, \alpha_i \geq 0$ is also a valid kernel [6]. For example, in [14], individual kernels based on plain text and citation relation are constructed respectively for Web page classification. The kernel for text is created based on the BOW (bag-of-words) representation for the text content. When the co-citation kernel is constructed, each page is represented as a BOL (bag-of-links) vector. In both cases, the vectors are re-weighted to better reflect the similarity measure.

2.2. Kernel Alignment

In [7], alignment was proposed as a method for measuring the fitness of agreement between a kernel and the learning task.

Definition 1 *The empirical alignment between kernel k_1 and kernel k_2 with respect to the training set S is the quantity*

$$A(k_1, k_2) = \frac{\langle K_1, K_2 \rangle}{\|K_1\|_F \|K_2\|_F} \tag{1}$$

where K_i is the kernel matrix for the training set S using kernel function k_i, $\|K_i\|_F = \sqrt{\langle K_i, K_i \rangle_F}$,$\langle K_i, K_j \rangle_F$ is the Frobenius inner product between K_i and K_j. $S = \{(x_i, y_i) \,|x_i \in \mathcal{X}, y_i \in \{-1, +1\}, i = 1, \cdots, m\}$, $\mathcal{X}$ is the input space, y is the target vector.

Let $K_2 = yy'$,then the empirical alignment between kernel k and target vector y is:

$$A\left(k, yy'\right) = \frac{\langle K, y'Ky \rangle_F}{\|K\|_F \|yy'\|_F} = \frac{y'Ky}{m\|K\|_F} \tag{2}$$

It has been shown that if a kernel is well aligned with the target information, there exists a separation of the data with a low bound on the generalization error [7]. Thus, we can optimize the kernel alignment based on training set information to improve the generalization performance on the test set.

3. Problem Formulation

In this section, we focus our discussion on linearly combining individual kernels to achieve an optimized composite one. After the theoretical analysis, we propose a GE-CKO algorithm for common classification tasks.

3.1. Composite Kernel Alignment Optimization

Here we consider the linear combination of kernels:

$$k(\alpha) = \sum_{i=1}^{p} \alpha_i k_i \tag{3}$$

Individual kernels $k_i, i = 1, \cdots p$ are given in advance. Our purpose is to tune α to maximize $A(\alpha, k, yy')$, the empirical alignment between $k(\alpha)$ and the target vector y. Here we do not constrain α to be non-negative. The reason is that, first, it is possible for the combined kernel to be positive semi-definite when some coefficients are negative [16]; second, even $k(\alpha)$ is not a Mercer kernel, we can still apply the generalized SVM algorithms that don't require the kernel matrix to be positive semi-definite [21].

Hence, we have:

$$\begin{aligned}
\hat{\alpha} &= \arg_\alpha \max \left(A\left(\alpha, k, yy'\right)\right) \\
&= \arg_\alpha \max \left(\frac{\langle \sum_i \alpha_i K_i, yy' \rangle}{m\sqrt{\left\langle \sum_i \alpha_i K_i, \sum_j \alpha_j K_j \right\rangle}} \right) \\
&= \arg_\alpha \max \left(\frac{\sum_i \alpha_i \langle K_i, yy' \rangle}{m\sqrt{\sum_{i,j} \alpha_i \alpha_j \langle K_i, K_j \rangle}} \right) \\
&= \arg_\alpha \max \left(\frac{(\sum_i \alpha_i u_i)^2}{m^2 \sum_{i,j} \alpha_i \alpha_j v_{ij}} \right) \\
&= \arg_\alpha \max \left(\frac{1}{m^2} \cdot \frac{\alpha^T U \alpha}{\alpha^T V \alpha} \right)
\end{aligned}$$

where $u_i = \langle K_i, yy' \rangle$,$U_{ij} = u_i u_j$,$V_{ij} = v_{ij} = \langle K_i, K_j \rangle$.

Let

$$J(\alpha) = \frac{\alpha^T U \alpha}{\alpha^T V \alpha} \tag{4}$$

In mathematical physics, $J(\alpha)$ is the generalized Rayleigh quotient [8]. To obtain $\hat{\alpha}$, it is equivalent to solving a generalized eigenvalue problem:

$$U\alpha = \lambda V \alpha \tag{5}$$

And $\hat{\alpha}$ corresponds with the eigenvector which has the largest absolute eigenvalue.

Input:
Training set (x_1, y_1),$\cdots$,(x_m, y_m), $y_i \in \{-1, +1\}$, $i = 1 \cdots m$. Each example x_i in the training set is represented using p set of features: $x_i^j \in f^j$,$j = 1, \cdots, p$. And p kernels k_1,$\cdots$,k_p are given corresponding with the p set of features respectively.
Output:
The classification model.
Algorithm:
1. Create kernel matrices $K_1, \cdots, K_p$. K_j is constructed using the jth set of features and the jth kernel respectively, $j = 1, \cdots p$.
2. Compute the dot product between each pair of matrices in the set $\{K_1, \cdots K_p, yy'\}$. Construct the $p \times p$ matrices U and V. Let $u_i = \langle K_i, yy' \rangle$, $U_{ij} = u_i u_j$, $V_{ij} = \langle K_i, K_j \rangle$.
3. Solve the generalized eigenvalue problem: $U\alpha = \lambda V \alpha$. λ_t is the eigenvalue with the largest absolute value. The eigenvector associated with λ_t is saved in $\hat{\alpha}_i$, $i = 1, \cdots, p$.
4. If $\hat{\alpha}$ is non-negative, go to step 5. Otherwise let $s = \sum_{i=1}^{p} exp(\hat{\alpha}_i)$, $\hat{\alpha}_i = exp(\hat{\alpha}_i)/s$, $i = 1, \cdots, p$.
5. Let $K = \sum_{i=1}^{p} \hat{\alpha}_i K_i$, solve the quadratic programming problem based on matrix K and output the classifier model.

Figure 1: The GE-CKO algorithm

3.2. GE-CKO Algorithm

Based on the theoretical analysis in Section 3.1, we proposed an algorithm named GE-CKO(Generalized Eigenvalue based Composite Kernel Optimization) for Web page classification. Figure 1 shows the outline of the algorithm.

Note in step 4 of the GE-CKO algorithm. If $\hat{\alpha}$ contains negative elements, it is transformed to be in [0,1] using equation (6):

$$\hat{\alpha}_i = \frac{exp(\hat{\alpha}_i)}{\sum_{k=1}^{p} exp(\hat{\alpha}_k)}, i = 1, \cdots, p \tag{6}$$

In fact, the combination of kernels with negative coefficients may still result in a positive semi-definite kernel [16]. In this paper, because we use the SVM^{light} [12] package for training, and it requires the kernel matrix to be positive semi-definite, thus the step 4 in GE-CKO is necessary. If the generalized SVM algorithms that do not solve quadratic programming problems are used, step 4 should be removed.

4. Experiments

4.1. Data Set

To demonstrate the performance of the GE-CKO algorithm presented, we use the WebKB dataset [2, 19] collected at CMU for evaluation. This corpus consists of the Web pages from computer science department of various universities. We consider three binary classification tasks in the data set, identifying Course, Faculty and Student homepages. There are a total of 8260 pages that are not duplicated. The distribution of pages across the categories is shown in Table 1.

Category	Percentage	Number
Course	11.2%	926
Faculty	19.8%	1637
Student	13.6%	1124
Other	55.4%	4573

Table 1: Distribution of WebKB pages

4.2. Evaluation Measure

We employ the standard F_1 measure to evaluate the Web page classification performance [23]. Precision (P) is the proportion of correct positive documents among all the predicted positive documents by the classifier. Recall (R) is the proportion of predicted correct positive documents among all the correct positive documents. F_1 measure is defined below:

$$F_1 = 2 \times P \times R/(P + R) \quad (7)$$

To evaluate the global performance in multi-category case, two ways of averaging over all categories exist. Macro-averaging gives equal weights to every category and micro-averaging gives equal weights to every document [23]. Due to the space limitation, only the macro-averaging is used in our experiments. For each category, 10 fold cross-validation iis calculated.

4.3. Results and Discussions

4.3.1. Baseline We extracted features from different parts of a page: title, plain text, link and in-link anchor text. Based on each type of features, individual kernels are constructed and the standard SVM classifier is trained using them respectively. We name the individual SVM algorithm according to the features it uses. For example, SVM based on text kernel is called Text-SVM. In addition, if the step 2-4 in GE-CKO are removed and $\hat{\alpha}$ is assigned with equal weight, we get a composite kernel SVM algorithm where different kernels are equally treated. In this paper, we call this algorithm EQU-SVM. The above algorithms are applied as baseline and are compared with GE-CKO.

The TFIDF-weighted BOW method is used to represent plain text, title and anchor text. The stop-words are removed and the remaining words are stemmed by the Porter algorithm [1]. The BOL vectors are represented using the in-links and out-links respectively. The dimension of the vectors is equal with the number of documents in the dataset. All the feature vectors are normalized to have unit L_2-norm. In the following experiments, the quadratic programming problem is implemented using the SVM^{light} package [12]. In order to allow comparisons between different algorithms, we have used the linear kernel function for all the individual kernels.

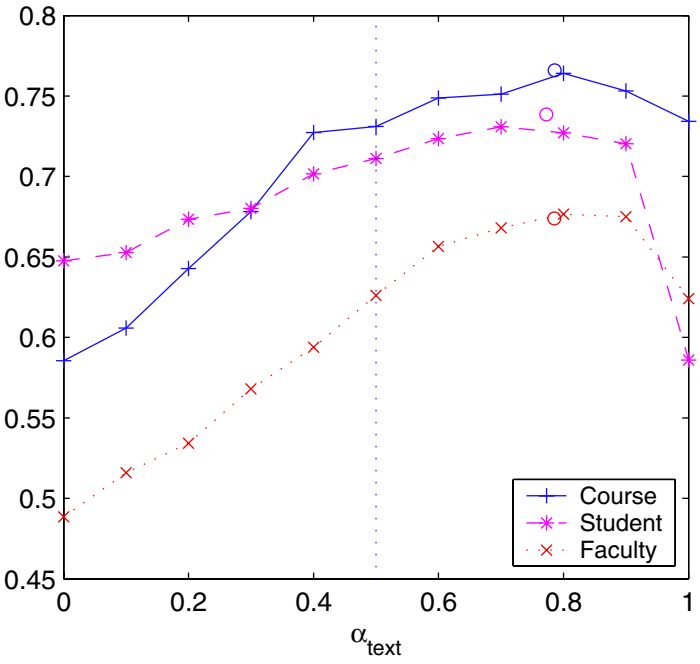

Figure 2: F_1 performance vs. α_{text}. Features used are text and in-link. α_{text} is the weight parameter associated with text features in equation (3).

4.3.2. Does GE-CKO improve generalization? We first conduct an experiment to examine the performance of GE-CKO compared with the baseline algorithms. In this experiment, plain text and in-link features are used. Figure 2 presents the result. In this figure, α_{text} is the weight parameter in equation (3), when α_{text} is equal to 0, only in-link kernel is used. The bigger α_{text} is, the more weight is assigned to the text kernel. When α_{text} reaches 1, the classifier is trained base on text kernel only. The dashed line indicates the output of the EQU-SVM, and the circles correspond with the results of the GE-CKO. Because α_{text} is in-

	Text-SVM	InLink-SVM	EQU-SVM	GE-CKO
Macro F_1	0.648	0.573	0.688	0.726

Table 2: Macro F_1 performance. Features used are text and in-link.

	Text-SVM	InLink-SVM	OutLink-SVM	EQU-SVM	GE-CKO
Course	0.734	0.585	0.325	0.698	0.764
Student	0.586	0.647	0.308	0.713	0.735
Faculty	0.624	0.488	0.299	0.571	0.674
MacroF_1	0.648	0.573	0.311	0.661	0.724

Table 3: Macro F_1 performance. Features used are text, in-link and out-link.

	Text-SVM	EQU-SVM using FC5	GE-CKO using FC5
Macro F_1	0.648	0.701	0.765

Table 4: Macro F_1 performance. All five types of features: text, in-link, out-link, title and in-link anchor text are used .

creased in step 0.1, the circles are not exactly on the three curves.

It is apparent that the generalization performance of SVM has high dependency on α_{text} for all the three tasks. For the category Course and Faculty, results of Text-SVM are more promising, while for the Student category results of InLink-SVM are better. For Course and Faculty tasks, the EQU-SVM and the Text-SVM are approximately equal in F_1 measure. While for other cases, the EQU-SVM outperforms the standard SVM on either type of kernel. The GE-CKO shows promising results. For the three tasks, the GE-CKO beats all the baseline algorithms. Table 2 shows the macro-averaging F_1 result. The GE-CKO achieves an improvement of 12.0% and 5.5% relative to the Text-SVM and EQU-SVM respectively.

4.3.3. The performance of GE-CKO when more kernels are available. In this sub section, we study whether our GE-CKO algorithm is robust if more kernels exist. First we apply this algorithm to combine text kernel, in-link kernel and out-link kernel. Table 3 presents the experiment results. From Table 3, we found that the OutLink-SVM performs worst. When it is combined with the other two kernels, the performance of EQU-SVM becomes worse than the Text-SVM for Student and Faculty tasks. In contrast, the GE-CKO outperforms each baseline algorithm. It obtains an improvement of 11.7%, 26.4% and 9.5% compared with Text-SVM, InLink-SVM and EQU-SVM respectively.

In order to further examine the generalization performance of GE-CKO with respect to the number of kernels, we use all five types of features extracted for experiment. Figure 3 shows the results on five different feature combination schemes(FC1-FC5), each combination has one more type of features than its proceeding one. For example, FC1 includes only text features, FC2 contains text and in-link features. The features that are added in turn are indicated under the stair-step graph. For Course and Faculty tasks, it is observed that the EQU-SVM's generalization performance varies irregularly when the new features are added. However, the GE-CKO algorithm achieves a steady improvement for all the tasks. As illustrated in Table 4, when all kernels are combined the macro-F_1 reaches the maximum, with an increase of 18.1% and 9.1% relative to Text-SVM and EQU-SVM respectively.

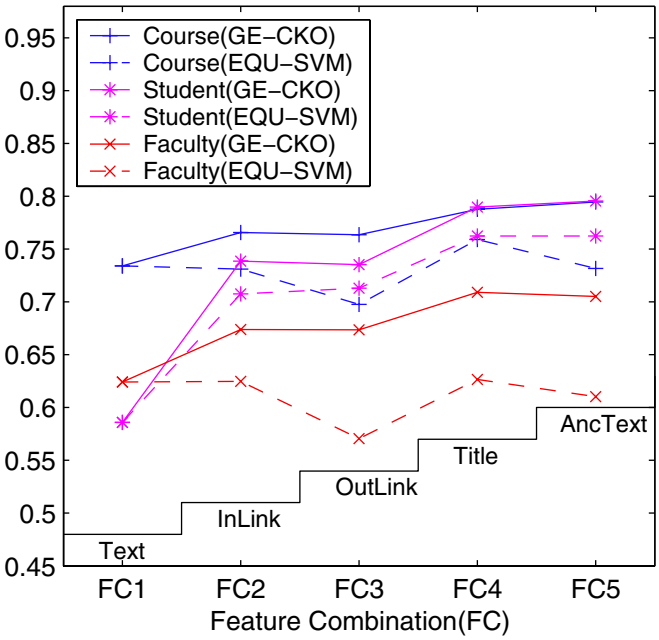

Figure 3: Macro F_1 performance vs. feature combination scheme.

4.4. Discussion

In the experiments above, we found that the generalization performance of the composite kernel method has high dependency on the weights of individual kernels. It is shown that the algorithm can automatically adapt itself with kernels based on different types of features. In our opinion, the success is due to the kernel optimization step. In equation (4), the $J(\alpha)$ we try to maximize is a global measure of correlations among the kernels and the fitness between kernels and the classification task. When the individual kernels have different contributions to the classification task, there is no guarantee of performance improvement if they are equally combined. This is confirmed by the fact that the GE-CKO algorithm outperforms all the baseline algorithms in the experiments. In addition, the EQU-SVM is quite unstable because it takes no steps to optimize the composite kernel.

5. Related Work

Web page classification has been extensively studied. As Web pages usually contain rich heterogeneous features, many researchers resort to utilizing more information to improve the generalization performance. Furnkranz et al. [9] and Glover et al. [11] used anchor text and hyperlink-surrounding text to classify Web pages. Chakrabarti et al. [5], Oh et al. [17] and Calado et al. [4] combined link and content information using a unified model. Joachims et al. [14] used combined kernels for Web page classification. According to the experiment results reported, the conclusions do not agree with each other. One potential reason for this is that these experiments use different documents or categories. Yang et al. believe that the regularities in the dataset and the choice of hypertext representation are crucial for Web page classification [10].

Kernel methods like SVM are used for Web page classification and show promising results [22]. Sun et al. combined different heterogeneous features and apply SVM for Web page classification [20]. The performance improvement is significant. Although the SVM algorithm has good generalization performance, there is seldom work that deeply exploit SVM to deal with heterogeneous features. In [18], Pavlidis et al. investigated combining heterogeneous data for gene functional classification. He compared the performance of the SVM algorithm using three methods to combine different types of features. However, the questions mentioned in Section 1 are not answered.

In this paper, we propose a general composite kernel optimization method to combine individual kernels. Our theoretical analysis is based on the kernel alignment theory. It is proved that if a kernel has a perfect alignment with the classification task, there is a low bound on the generalization error [16]. In [16, 7, 15], different kernel alignment techniques are proposed. In these works, the kernel alignment problem is different from the one studied in this paper. In their works, the purpose is to adapt one kernel matrix to align well with the target information. However, what we are attacking is how to optimize several individual kernels for the learning task.

6. Conclusions and Future Work

In this paper, we propose a general method named GE-CKO to optimize composite kernels for classifying Web pages. We use linear kernel with fixed parameters on each type of features, and combine them using an optimized linear combination algorithm by solving the generalized eigenvalue problem. An SVM classifier is then constructed based on the resulted combination of kernels. We compared our method with the individual kernel SVM method and the composite kernel SVM without optimization. Experiment results show that our method has a better generalization performance and is capable of automatically adapting to heterogeneous features. It is unnecessary to distinguish which type of features are important or to clarify the relations between them.

Although the GE-CKO algorithm is proposed for Web page classification task, it is easy to be extended for regression case. In the future, we will try to include kernel selection problem for individual kernels into the composite kernel optimization framework.

Acknowledgements

We thank Xin-Jing Wang and Gui-Rong Xue for comments on the paper and useful discussions.

References

[1] The porter stemming algorithm. http://www.tartarus.org/ martin/PorterStemmer.

[2] The webkb dataset. http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data.

[3] P. N. Bennett, S. T. Dumais, and E. Horvitz. Probabilistic combination of text classifiers using reliability indicators: models and results. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 207–214. ACM Press, 2002.

[4] P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, and M. A. Goncalves. Combining link-based and content-based methods for web document classification. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 394–401. ACM Press, 2003.

[5] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In L. M. Haas and A. Tiwary, editors, *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US, 1998. ACM Press, New York, US.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[7] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Neural Information Processing Systems*, pages 367–373, 2001.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification, 2nd edition*. Wiley, N.Y., 2001.

[9] J. Furnkranz. Exploiting structural information for text classification on the WWW. In *Intelligent Data Analysis*, pages 487–498, 1999.

[10] R. Ghani, S. Slattery, and Y. Yang. Hypertext categorization using hyperlink patterns and meta data. In C. Brodley and A. Danyluk, editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 178–185, Williams College, US, 2001. Morgan Kaufmann Publishers, San Francisco, US.

[11] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake. Using Web structure for classifying and describing Web pages. In *Proceedings of WWW-02, International Conference on the World Wide Web*, 2002.

[12] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[13] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[14] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite kernels for hypertext categorisation. In C. Brodley and A. Danyluk, editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 250–257, Williams College, US, 2001. Morgan Kaufmann Publishers, San Francisco, US.

[15] J. Kandola, J. Shawe-Taylor, and N. Cristianini. On the extensions of kernel alignment. Technical Report NC-TR-02-120, Neural Networks and Computational Learning Theory, 2002.

[16] J. Kandola, J. Shawe-Taylor, and N. Cristianini. Optimizing kernel alignment over combinations of kernels. Technical Report NC-TR-02-121, Neural Networks and Computational Learning Theory, 2002.

[17] H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext catergorization method using links and incrementally available class information. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271. ACM Press, 2000.

[18] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the fifth annual international conference on Computational biology*, pages 249–255. ACM Press, 2001.

[19] S. Slattery and M. Craven. Discovering test set regularities in relational domains. In P. Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 895–902, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

[20] A. Sun, E.-P. Lim, and W.-K. Ng. Web classification using support vector machine. In *Proceedings of the fourth international workshop on Web information and data management*, pages 96–99. ACM Press, 2002.

[21] V. Tresp. Scaling kernel-based systems to large data sets. *Data Mining and Knowledge Discovery*, 5(3):197–211, 2001.

[22] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[23] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.