A Focused Crawling for the Web Resource Discovery Using a Modified Proximal Support Vector Machines

YoungSik Choi, KiJoo Kim, and MunSu Kang

Department of Computer Engineering, Hankuk Aviation University,
200-1 Hwajun-Dong Dukyang-Gu
Kyounggi-Province, Koyang-City, Korea
{choimail, zgeniez, mskang}@hau.ac.kr
http://data-mining.hau.ac.kr

Abstract. With the rapid growth of the World Wide Web, a focused crawling has been increasingly of importance. The goal of the focused crawling is to seek out and collect the pages that are relevant to a predefined set of topics. The determination of the relevance of a page to a specific topic has been addressed as a classification problem. However, when training the classifiers, one can often encounter some difficulties in selecting negative samples. Such difficulties come from the fact that collecting a set of pages relevant to a specific topic is not a classification process by nature.

In this paper, we propose a novel focused crawling method using only positive samples to represent a given topic as a form of hyperplane, where we can obtain such representation from a modified Proximal Support Vector Machines. The distance from a page to the hyperplane is used to prioritize the visit order of the page. We demonstrated the performance of the proposed method over the WebKB data set and the Web. The promising results suggest that our proposed method be more effective to the focused crawling problem than the traditional approaches.

1 Introduction

The World Wide Web continues to grow rapidly at a rate of a million pages per day [1]. Searching a document from such enormous number of Web pages has brought about the need for a new ranking scheme beyond the traditional information retrieval schemes. Hyperlink analysis has been proposed to overcome such problems and showed some promising results [6][9]. Now, hyperlink analysis related technologies are somehow embedded into modern commercialized search engines, notably Google [9][11]. Although hyperlink based methods appear to be an effective approach showing reasonable retrieval results, there are still many problems yet to be solved. Such problems are mainly caused by the simple fact that there are too many pages out there to be crawled. Even the largest crawlers cover only 30 ~ 40% of the Web and the refreshes take up to a month [1][9]. Many ideas have been proposed in recent years to handle such problems; among them a focused crawling has gained much attention [1-5]. A focused crawling is to seek out and collect the pages that are relevant to specific topics so that it may be able to cover topical portions of the Web quickly without

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3480, pp. 186 – 194, 2005.

having to explore a whole Web [2][3]. One can view the focused crawling as a resource discovery process from the Web. For instance, the focused crawling has been used for constructing Knowledge Base from the Web [2].

The focused crawling is a process of selecting relevant pages from the Web. The selection criterion on relevant pages is generally based on the assumption of "topic locality" that child pages are likely to contain same topic to that of their parent pages [2][9]. Starting from seed pages, usually from a few representative pages on a given topic, the focused crawler explores pages that are relevant to the topic according to the predefined relevance measurement. The measurement of relevance affects the whole performance of the focused crawler. In general, the judgment of relevance has been interpreted as a classification process [1][9] where a classifier should be trained by using positive and negative examples. In such approach, we often encounter difficulties choosing negative samples. This is because selecting pages related to a specific topic is not a classification process by nature. Moreover, in a classification paradigm, there ought to be a hard decision making on the class membership, which causes sometimes to lose important pages. This can be alleviated by making soft decision [9] but still having to maintain only two classes, relative or irrelative. Considering the diversity of the Web content, one can hardly imagine that one classifier may be able to bisect the entire Web. Instead, it is more natural to represent a specific topic as a one-class and to measure relevance for that class.

The Proximal SVMs are well known for sustaining all the good features of Proximal SVMs including a powerful generalization capability out of training samples [7][8]. In this paper we present a modified Proximal SVM to seek out the proximal hyperplane that best represents the underlying distribution of a given set of positive samples. That is, our modified Proximal SVM only uses positive samples from a given topic so that one can use the distance from a page to the proximal hyperplane as a measurement of a page's relevance to a topic. This approach can be viewed as a traditional LS (Least Squares) Regression with regularization and therefore enjoys both a representation power from the LS Regression and a generalization power from the SVM [10]. In a training phase, we extract the proximal hyperplane over positive samples by using the modified Proximal SVMs. During the crawling, the distance to the proximal hyperplane from each crawled page is computed and quantized. According to the quantized distance, the offspring pages from each crawled page are put into one of the priority bins. Then, the focused crawler selects one page out of a non-empty highest priority bin for the next crawl.

This paper is organized as follows. In the next section, we present the general framework of the focused crawling using a modified Proximal SVM in Section 2. Section 3 discusses experimental results in detail. We make our conclusions and summaries in Section 4.

2 Framework for the Proposed Focused Crawling

2.1 Proximal Support Vector Machines

The linear PSVM (Proximal Support Vector Machine) seeks to find the "proximal planes" that best represent training samples while maximizing the distance between

the two proximal planes. The obtained two proximal planes comprise the optimal separating hyperplane [7].

The training samples are represented as a matrix $\mathbf{A} \in \Re^{m \times n}$, where m denotes the number of the training samples and n represents the dimensionality. Matrix D denotes an $m \times m$ diagonal matrix with the values {-1, +1} of each element and y represents a column vector whose element represents an error. Matrix $\mathbf{I}$ denotes an identity matrix.

In linearly separable cases, the optimal separating hyperplane can be represented as $\mathbf{x}^T\mathbf{w} - \gamma = 0$, where $\mathbf{x} \in \Re^n$. The objective function for the Proximal SVM can be stated as

$$\text{minimize } \nu \frac{1}{2}\|y\|^2 + \frac{1}{2}(\mathbf{w}^T\mathbf{w} + \gamma^2) \qquad (1)$$

$$\text{subject to } D(\mathbf{A}\mathbf{w} - \mathbf{e}\gamma) + y = \mathbf{e}$$

Note that v is a non-zero constant and e is the column vector of ones. The objective function in equation (1) can be restated as the "regularized least squares." Figure 1 shows that the positive proximal plane ($xw - \gamma = 1$) and the negative proximal plane ($xw - \gamma = -1$) are located in the middle of positive and negative training samples, respectively.

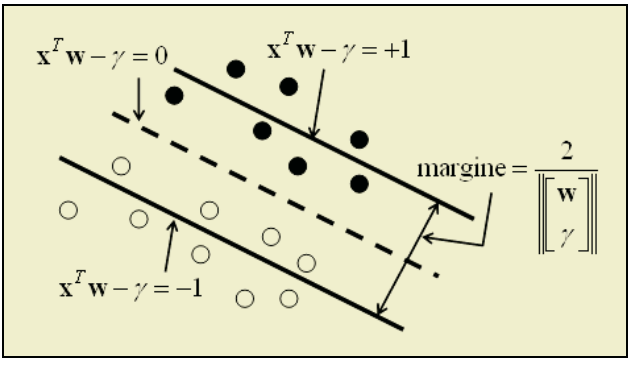


Fig. 1. Proximal Support Vector Machines: positive and negative proximal planes

One can solve the optimization problem in equation (1) using the Lagrangian multipliers $\boldsymbol{\alpha} \in \Re^m$ as follows.

$$L(\mathbf{w}, \gamma, y, \boldsymbol{\alpha}) = \nu \frac{1}{2}\|y\|^2 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{w}\\ \gamma\end{bmatrix}\right\|^2 - \boldsymbol{\alpha}^T(D(\mathbf{A}\mathbf{w} - \mathbf{e}\gamma) + y - \mathbf{e}) \qquad (2)$$

Taking the derivative of equation (2) with respect to $\mathbf{w}$, γ, y and setting them to zero, and do some algebra, the following equations can be obtained.

$$\boldsymbol{\alpha} = \left(\mathbf{I}/\nu + D\left(\mathbf{A}\mathbf{A}^T + \mathbf{e}\mathbf{e}^T\right)D\right)^{-1}\mathbf{e} = \left(\mathbf{I}/\nu + \mathbf{H}\mathbf{H}^T\right)\mathbf{e} \qquad (3)$$

$$\mathbf{H} = D[\mathbf{A} \quad -\mathbf{e}]$$

The linear decision function can be obtained as follows.

$$\mathbf{x}^T\mathbf{w} - \gamma = \mathbf{x}^T\mathbf{A}^T D\boldsymbol{\alpha} - \gamma = 0 \tag{4}$$

One can obtain the kernel version of Proximal SVM by replacing $\boldsymbol{A}$ by $\boldsymbol{K}$, where $\boldsymbol{K}$ is an abridged form of matrix $\mathbf{K}(\mathbf{A}, \mathbf{A}^T)$ whose elements are dot products of two samples in a given kernel space.

$$\begin{gathered} \upsilon = \left(\mathbf{I}/\nu + D(\mathbf{K}\mathbf{K}^T + \mathbf{e}\mathbf{e}^T)D\right)^{-1}\mathbf{e} = \left(\mathbf{I}/\nu + GG^T\right)^{-1}\mathbf{e}, \\ G = D[\mathbf{K} \quad -\mathbf{e}] \end{gathered} \tag{5}$$

$$f(x) = (\mathbf{K}(\mathbf{x}^T, \mathbf{A}^T)\mathbf{K}(\mathbf{A}, \mathbf{A}^T)^T + \mathbf{e}^T)D\upsilon = 0 \tag{6}$$

In [10], it was shown that the performance of the PSVM should be comparable with that of the SVM with fast computational time.

2.2 A Modified Proximal Support Vector Machine

If we set the diagonal matrix D to identity matrix in equation (1), the corresponding equation becomes the following.

$$\begin{gathered} \text{minimize } \nu\frac{1}{2}\|y\|^2 + \frac{1}{2}(\mathbf{w}^T\mathbf{w} + \gamma^2) \\ \text{subject to } \mathbf{A}\mathbf{w} - \mathbf{e}\gamma + y = \mathbf{e} \end{gathered} \tag{7}$$

In equation (2), we want to minimize the geometric distances from all samples to a hyperplane $\mathbf{w}\mathbf{x}$-γ=1 that is corresponding to a positive proximal hyperplane in Figure 1. Therefore the optimal hyperplane that satisfies equation (7) can be considered as a proximal hyperplane that best represents the underlying positive samples. We can obtain such a proximal hyperplane using Lagrangian multipliers as in Section 2.1.

$$L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\gamma^2 + \frac{\nu}{2}\|y\|^2 - \boldsymbol{\alpha}^T(\mathbf{A}\mathbf{w} - \gamma + y - \mathbf{e}) \tag{8}$$

Starting from equation (7), one can easily show the following formula for the modified Proximal SVM.

$$\boldsymbol{\alpha} = \left(\mathbf{A}\mathbf{A}^T + \mathbf{e}\mathbf{e}^T + \mathbf{I}/\nu\right)^{-1}\mathbf{e} \tag{9}$$

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} - \gamma = \mathbf{x}^T\mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T + \mathbf{e}\mathbf{e}^T + \mathbf{I}/\nu\right)^{-1}\mathbf{e} + \mathbf{e}^T\boldsymbol{\alpha} \tag{10}$$

From equation (10), the distance form proximal hyperplane is defined as follows.

$$\text{Distance} = |f(\mathbf{x}) - 1| \tag{11}$$

The kernel version of a modified Proximal SVM is defined as following, and the distance can be obtained using equation (11).

$$\boldsymbol{\alpha} = \left(\mathbf{K}(\mathbf{A}, \mathbf{A}^T) + \mathbf{e}\mathbf{e}^T + \mathbf{I}/\nu\right)^{-1}\mathbf{e} \tag{12}$$

$$f(\mathbf{x}) = \mathbf{K}\left(\mathbf{x}^T, \mathbf{A}^T\right)\left(\mathbf{K}\left(\mathbf{A}, \mathbf{A}^T\right) + \mathbf{e}\mathbf{e}^T + \mathbf{I}/\nu\right)^{-1}\mathbf{e} + \mathbf{e}^T\boldsymbol{\alpha} \tag{13}$$

2.3 Focused Crawling Using Proximal Hyperplane

The proximal hyperplane represents a distribution of data samples. In other words the proximal hyperplane is a maximal margin hyperplane where most training data samples reside [8]. Therefore, one can use the distance to a proximal hyperplane as a dissimilarity measurement for a class membership. We apply this idea to the focused crawling.

Algorithm for the Focused Crawling Using Proximal Hyperplane

A. Training Phase
1. Determine the Proximal Hyperplane over positive samples using equation (11)
2. Compute a mean value of the distances to Hyperplane from positive samples.

B. Crawling Phase
1. Set the number of Priority Bins to k
2. Get one URL from a non-empty highest Priority Bin
3. Download a corresponding page from the Web
   3.1 Compute distance from the downloaded page using equation (11)
   3.2 Quantize the distance as following
      If distance $\leq$ mean, then Priority = 1
      Else if mean < distance $\leq$ 0.5, then Priority = $2+\left\lfloor (k-2)\frac{\text{distance}-\text{mean}}{0.5-\text{mean}} \right\rfloor$
      Otherwise, Priority = k
   3.3 Extract all the URLs in this page and put them into a Priority Bin
4. Go to step 2.

Fig. 2. Proposed Focused Crawling Algorithm Using a Proximal Hyperplane

First, we determine the proximal hyperplane from a given positive samples. We also compute a mean value of the distances from the samples to the proximal plane. Then, we start to crawl with a few seed URLs. The seed URLs may be a set of representative URLs in a given topic, or can be a set of target hosts in which the crawler is confined to explore. After fetching a page of a selected URL, the crawler transforms the page into a vector, and extracts all the hyperlinks, URLs from the fetched page. Now, the crawler computes the distance from a transformed vector to the predetermined proximal hyperplane, and quantizes the distance so that the URLs extracted from the fetched page can be put into a corresponding priority bin. There are *k* priority bins and each bin corresponds to one quantized distance level. The focused crawler chooses one URL at a time from a non-empty highest priority bin so that it can crawl favorably pages closer to a proximal hyperplane. The overall focused crawling algorithm is shown in Fig. 2.

3 Results

3.1 Experimental Setup

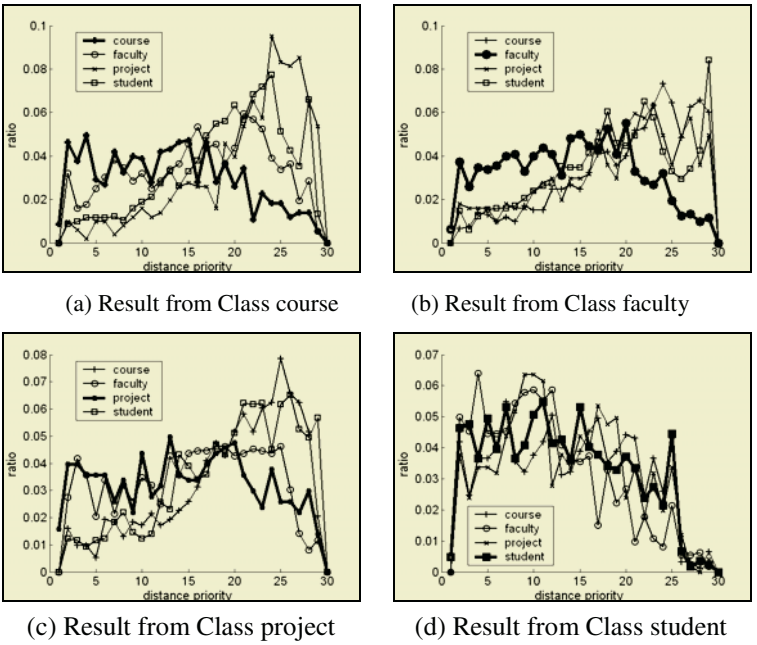
The crawler that we used in this paper was implemented using Java, and the experiments shown here were conducted in Pentium IV with main memory of 1GBytes. The

details of implementation are out of scope in this paper, and you can refer to [6] and [11] for the various implementation issues. For the preprocessing of the HTML documents, we did not execute the stemming, and we filtered the stop words. To make transformation of a page to a vector, we only used TF (Term Frequency) and normalized vectors of unit length.

The experiments were conducted in two steps. First, in Section 3.1, we tested our Modified Proximal Support Vector Machine using the WebKB data set [2]. Then, in Section 3.2, we applied the proposed crawling method on the Web with a specific topic. In order to evaluate the performance of focused crawlers, a harvest rate has often been used [9]. The harvest rate is an average relevance of the fetched pages at any given time. However, a 'true' relevance cannot be obtained as long as millions of Web pages are concerned. Therefore the relevance measure used in the focused crawling is usually used as a relevance measure in the evaluation [9]. In this paper, we used a cosine similarity as a relevance measure in order to compare our focused crawling and a traditional approach using a Naive Bayesian Classifier [9].

3.2 Results from the Modified Proximal SVM

In order to see how our modified Proximal SVM, we experimented over WebKB data set. In this experimental setup, we only used 4 classes, "course", "faculty", "project", and "student". We trained each proximal hyperplane using 10 pages randomly taken out of each class. Then we computed the distances from test pages out of 4 classes to the proximal hyperplane. We quantized the distances as described in Section 2.3 and we set k to 20. We conducted this process for each class.



(a) Result from Class course (b) Result from Class faculty

(c) Result from Class project (d) Result from Class student

Fig. 3. Ratio of true positive pages and false positive pages over the Priority Bins for each class: The bold lines in (a)-(d) indicate a ratio of positive pages to total number pages in each class

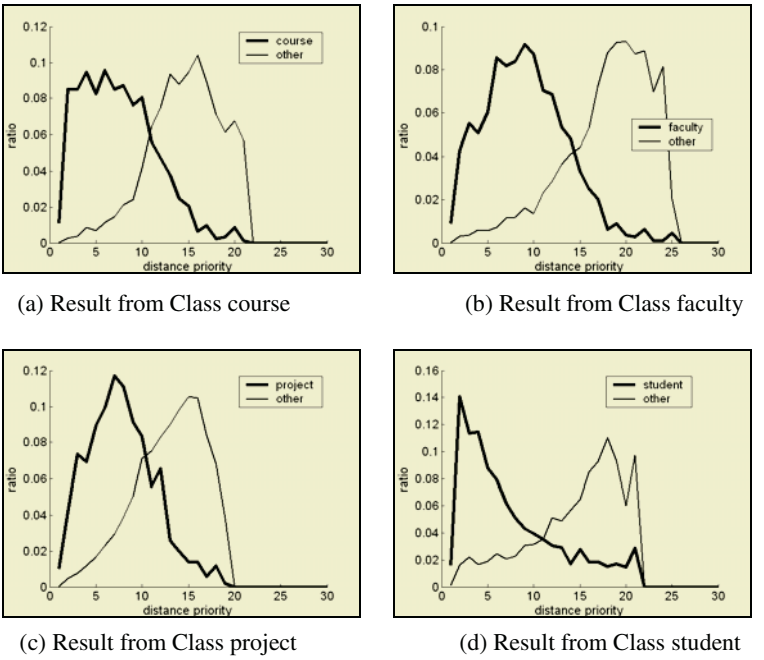
We also tested the modified Proximal SVM with yet another class "others". Class others is a collection of pages that do not belong to a specific class defined in the WebKB project [2].

Figure 4 shows the results against Class others. As we can see in this experiment, all four Proximal Hyperplanes were able to separate themselves well from Class others. These two experiments indicate that the proposed Proximal Hyperplane approach should be effective to the focused crawling where negative examples are not well defined and sometimes only positive examples are available.

3.3 Results from the Proposed Focused Crawling

We tested the proposed focused crawling algorithm over the Web and compared the performance with Naïve Bayesian Classifier based focused crawling [9]. We selected "Computer Science" as a topic of interest and chose 5 positive URLs from the Web. We also chose 5 negative web pages from "Law and Medical" pages for the training of Naïve Bayesian Classifier. Note that the number of samples used in our experiments is much less than that used in others [1-5][9].

We experimented with four different sets of seed URLs. Figure 5 shows the result with http://www.harvard.edu and http://www.eecs.harvard.edu/index/cs/ as Seed URLs. Figure 6 shows the results with http://www.yahoo.com and some URLs in Yahoo! Category "Computer Science" as Seed URLs. In all cases, we confined the crawling range to the same domains as seed URLs.



(a) Result from Class course

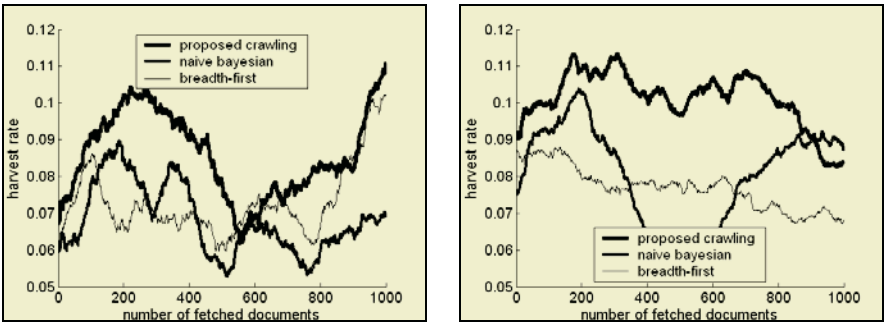
(b) Result from Class faculty

(c) Result from Class project

(d) Result from Class student

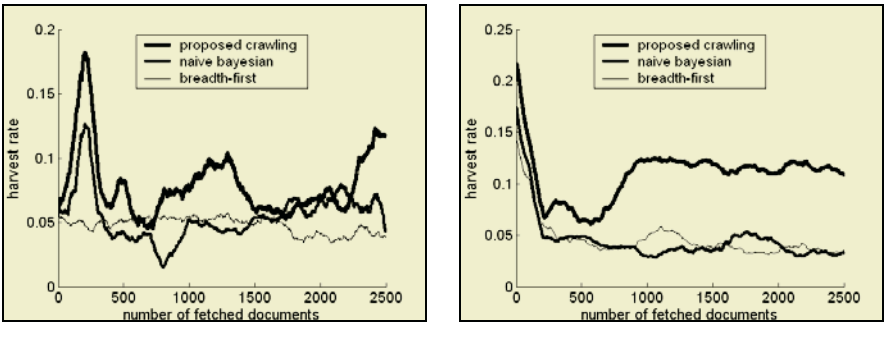
Fig. 4. Ratio of true positive pages and false positive pages over the Priority Bin for each class

As shown in Figure 5 and 6, our proposed method outperformed the Bayesian based approach over a whole crawling duration. It is interesting that the Bayesian approach degenerated into the breadth first crawler as a certain amount of time goes by. The experimental results indicate that the overall performance of the proposed focused crawling was much better than the traditional approach using Naïve Bayesian Classifier. Note also that the harvest ratio from the breadth first crawling goes almost constant across the crawling time as we expected.



(a) (b)

Fig. 5. The results from the proposed crawling, Naive Bayesian based Crawling, and the crawling without focusing starting from (a) http://www.harvard.edu (b) http://www.eecs.harvard.edu/index/cs/



(a) (b)

Fig. 6. The results from the proposed crawling, Naive Bayesian based Crawling, and the crawling without focusing starting from (a) http://www.yahoo.com (b) http://dir.yahoo.com/science/computer_science

4 Conclusions

The proposed crawling method in this paper starts from the idea that a focused crawling for a specific topic can be better formulated as a one-class problem than the two

class classification problem. In order to model a specific topic out of a few positive samples, we modified the Proximal SVMs so that the obtained proximal hyperplane can represent a distribution of positive samples. We have also presented a focused crawling method using the distances to the proximal hyperplanes.

We tested our proposed method over various data sets and also compared the performance with a traditional Bayesian based crawling. The experimental results were very promising and encouraging to do more researches related to this approach. Among them are how to quantize the distances into priority bins properly. It is also worth investigating into the case where some negative samples are available but we are not sure how many different classes they come from.

Acknowledgement. This research was supported by IRC (Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyounggi-Province Regional Research Center designed by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific Web resource discovery. 8th International World Wide Web Conference, Toronto (1999) 1623–1640
2. Aggarwal, C. C., Al-Garawi, F., Yu, P. S.: Intelligent Crawling on the World Wide Web with Arbitrary Predicates. 10th International World Wide Web Conference, Hong Kong (2001) 96–105
3. Rennie, J., McCallum, A. K.: Using Reinforcement Learning to Spider the Web Efficiently. 16th International Conference on Machine Learning (ICML) (1999) 335–343
4. Diligenti, M., Coetzee, F. M., Lawrence, S., Giles, C. L., Gori, M.: Focused Using Context Graphs. 26th International Conference on Very Large Databases (VLDB) (2000) 527–534
5. Cho, J., Garcia-Mlina, H., Page, Lawrence.: Efficient Crawling Through URL Ordering. Computer Networks and ISDN Systems (1998) 161–172
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Proc. 7th Int. World Wide Web Conference, Brisbane, Australia, Computer Networks and ISDN Systems 30 (1998)107–117.
7. Fung, G., Mangasarian, O. L.: Proximal Support Vector Machine Classifiers. KDD2001: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco (2001) 77–86
8. Choi, Y. S., Noh, J. S.: Relevance Feedback for Content-Based Image Retrieval Using Proximal Support Vector Machine. International Conference on Computational Science and Its Applications (ICCSA), Vol. 2. Assisi, Italy (2004) 942–951
9. Charkrabarti, S.: mining the web Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publishers (2003)
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
11. Najork, M., Heydon, A.: High-performance Web crawling. Tech. Rep. Research Report 173, Compaq SRC(2001)