ELSEVIER

Expert Systems with Applications 31 (2006) 427–435

Expert Systems with Applications

www.elsevier.com/locate/eswa

Web page classification based on a support vector machine using a weighted vote schema

Rung-Ching Chen *, Chung-Hsun Hsieh

Department of Information Management, Chaoyang University of Technology, 168 Gifeng E.Rd., Wufeng, Taichung, Taiwan, ROC

Abstract

Traditional information retrieval method use keywords occurring in documents to determine the class of the documents, but usually retrieves unrelated web pages. In order to effectively classify web pages solving the synonymous keyword problem, we propose a web page classification based on support vector machine using a weighted vote schema for various features. The system uses both latent semantic analysis and web page feature selection training and recognition by the SVM model. Latent semantic analysis is used to find the semantic relations between keywords, and between documents. The latent semantic analysis method projects terms and a document into a vector space to find latent information in the document. At the same time, we also extract text features from web page content. Through text features, web pages are classified into a suitable category. These two features are sent to the SVM for training and testing respectively. Based on the output of the SVM, a voting schema is used to determine the category of the web page. Experimental results indicate our method is more effective than traditional methods.

Keywords: Latent semantic analysis; Support vector machine; Web page classification; Feature extraction

1. Introduction

According to Google index, the number of web pages now exceeds 8 billion John (2000), and is increasing by 1.5 million per day. The global population of internet users is also growing rapidly. For example, accord to the statistics, to the end of January of 2005, the population of internet users in Taiwan rose by 892,0000, while the total number of broadband internet users has reached 3.17 million.

Users can find the web pages they want in the enormous database of web pages represented by the Internet, and many search engines are available to users, including the popular Yahoo! Kimo Yahoo, (website) Pchome, and Google (website). Typical search engines work through keyword inputs. However, pages retrieved in this manner usually include invalid links and irrelevant web pages. A good web page classification method is thus an urgent need in facilitating user searches.

There are many classification methods for web pages. A decision tree Apte et al. (1998) is a general data classification method. Its two major advantages are (1) it is faster; and, (2) the classification result can be transformed into an IF-THEN relation that the user can easily understand. Common decision tree methods include ID3 Mitchell (1997) and C4.5 Quinlan (1993). The disadvantage is that when categories are more numerous, it makes mistakes more easily. Mccallum and Nigam (1998) transform the frequency of keywords to condition probabilities in which Baysian probability is used to calculate the probability value between every document and category. Under this system, the category with highest probability is the one the document belongs to. The advantage is that the correlation between two documents can be represented by a probability. However, the processing load is higher. A support vector machine, named SVM, is a supervised method Cortes and Vapnik (1995), Gunn (1998) and Joachims (1998) that uses a portion of the data to train the system and then forms a learning model that can predict the category of documents. k-NN method is often used in text document classification Tan (2005). Woog and Lee (2003) use a k-nearest neighbor (k-NN) approach to calculate the likelihood of a category and relevant web page. In order to improve performance, they add a feature selection, HTML tags, and a new similarity measure and evaluation. Selamat and Omatu (2004) use a training sample to do the stemming and remove stop words, then the feature vector dimensions for a portion are reduced, while another portion is used for each category extraction of the keyword and to assign the weight value. The two types of feature vectors are then combined and inputted to

* Corresponding author. Tel.: +886 4 23323000x4463; fax: +886 4 23742337.

E-mail address: crching@mail.cyut.edu.tw (R.-C. Chen).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.09.079

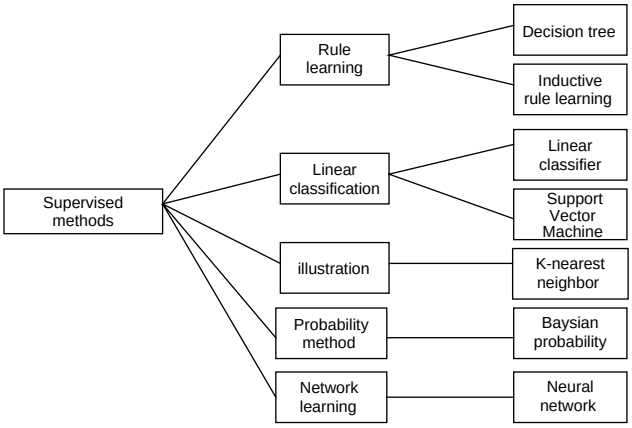

Fig. 1. The supervised classification methods.

a neural network for training. The system can then classify the web pages into the desired categorization . Unfortunately, a long time is required for training, and the convergence speed is low.

Yang and Liu's experiment (1999) shows that if fewer items of training data are used, the nearest neighbor method and SVM will be more effective than a neural network and Baysian probability. But if training data has a uniform distribution and the number items of training data exceeds 300, then there are no significant differences in the above four methods. Sebastian (2002) indicated there are no method suits all data types, but on the whole, a support vector machine is one of the best classification algorithms available at present. In addition to the above methods, rule learning and linear classifiers are also used. Fig. 1 shows the taxonomy of supervision of the classification methods.

This paper proposes a web page classification method, which uses a support vector machine combining latent semantic analysis and web page feature selection. We name it WVSVM (weighted voting support vector machine). First, latent semantic indexing is used to obtain the semantic relation between selected documents, then a selection of web page features is analyzed to obtain the features of the web page. Use these two types of features to input to SVM for training and testing, respectively. Then a voting schema is used to determine the category of the web page.

The remainder of the paper is organized as follows. We describe SVM in Section 2. Section 3 is the system overview. The classification method that uses WVSVM is described in section 4. The experimental results are given in Section 5. We make conclusions in Section 6.

2. Support vector machine

The primary idea of support vector machine (SVM) is using a high dimension space to find a hyper plane to do binary division, where the achieved error rate is minimum. An SVM can handle the problem of linear inseparability.

An SVM uses a portion of the data to train the system and finds several support vectors that represent training data. These support vectors will be formed into a model by the SVM,

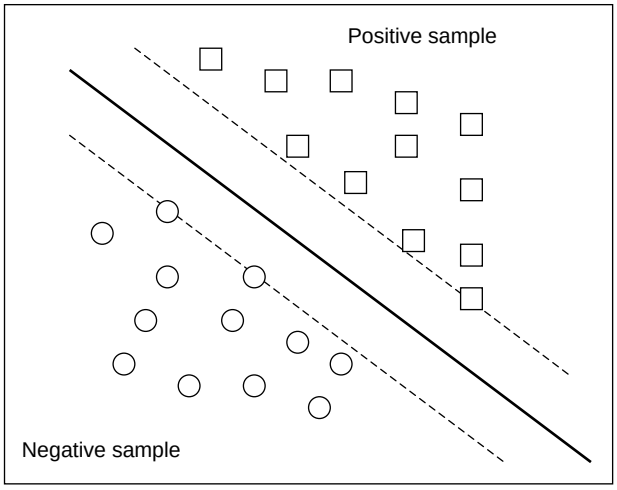

Fig. 2. The hyper plane of SVM.

representing a category. According this model, the SVM will classify a given unknown document by the following classification decision formula

$$(x_i, y_i), \ldots, (x_n, y_n), x \in R^m, y \in \{+1, -1\}.$$

Where $(x_i,y_i),\ldots,(x_n,y_n)$ are training samples, n is the number of samples, m is the input dimension, and y belongs to the category of $+1$ or -1, respectively.

In a linear problem, a hyper plane is divided into two categories. Fig. 2 shows a high dimension space divided into two categories by a hyper plane. The hyper plane formula is: $(w \cdot x) + b = 0$.

The classification formula is:

$$(w \bullet x_i) + b > 0 \quad \text{if } y_i = +1 \qquad (w \bullet x_i) + b < 0 \quad \text{if } y_i = -1$$

However, for many problems it is not easy to find a hyper plane to classify the data. The SVM has several kernel functions that users can apply to solve different problems. Selecting the appropriate kernel function can solve the problem of linear inseparability.

Another important capability of the SVM is that it can deal with linear inseparable problems. Internal product operations will affect the classification function. A suitable inner product function $K(x_i \cdot x_j)$ can solve certain linear inseparable problems without increasing the complexity of the calculation. Table 1 lists four kernel functions that are often used. The different kernel functions are suited to different problem types.

3. System overview

Generally, when looking at a broad web page category such as news, users are only interested in certain topics within that category, for example, business, entertainment, or sports. Our approach explores the category of sports news web pages and classifies sports news items.

We proposed a web page classification method called WVSVM (weighted voting support vector machine), which uses a latent semantic analysis (LSA) and Web page feature selection (WPFS) to extract semantic and text features. The

Table 1
Four kernel functions

Kernel	Kernel function	Parameter
Dot	$k(x,y)=x\cdot y$	None
Polynomial	$k(x,y)=(x\cdot y+1)^d$	d (degree) <integer>
Neural	$k(x,y)=\tanh(ax\cdot y+b)$	a,b <float>
Anova	$k(x,y)=\left(\sum_i \exp(-\gamma(x_i-y_i))\right)^d$	γ (gamma), d (degree) <integer>

framework of the workflow is shown on Fig. 3, described as follows:

(1) Preprocessing: preprocessing includes removal of HTML tags and Chinese word segmentation. HTML tags are removed but the text is retained, to prevent interference. Then, the text is compared with a Chinese lexicon to extract Chinese keywords for word segmentation.
(2) Latent semantic analysis: after preprocessing, the system constructs a term-document matrix X. SVD is applied to decomposing the matrix X and the original data vectors are reduced to a small number of features. The latent semantic relationships between keywords and documents are thus obtained.
(3) Web page feature selection: after segmentation of the Chinese word, the system extracts the web page text features. Such features include the number of keywords in a term database, the number of words in a document, the ratio between the number of keywords and the number of words, and the average interval between each term.
(4) Classification: we use semantic features and text features to train the SVM. The two SVM category models are used to predict the category of the web pages.
(5) Voting policy: after the two SVM models classify the web pages, the two classification results will be used to vote on which category the web page should be placed in.

4. Weighted voting support vector machine

4.1. Preprocessing

Before extracting the web page features, the web pages must be preprocessed. The preprocessing includes removal of HTML tags and segmentation of the Chinese words.

4.1.1. Removal of HTML tags

Most web pages are written in HTML at present. HTML uses open/closed tags to indicate web page commands, represented by '<' and '/>', respectively. Since content is not marked by these tags, we remove the HTML tags to reduce the burden of analysis.

4.1.2. Constructing the lexicon

English sentences have spaces between the words, but Chinese sentences do not. Therefore, computers find it difficult to analyze exactly how many Chinese words comprise a term. Different methods of extracting Chinese terms lead to different semantic meanings. We used the Chinese word segmentation

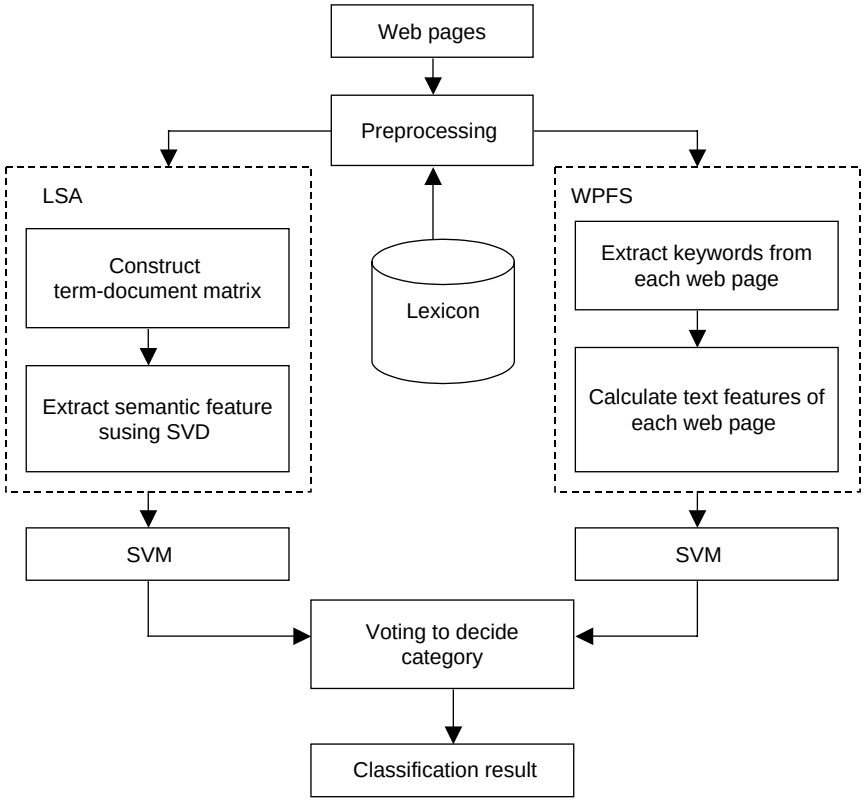

Fig. 3. The WVSVM framework.

Table 2
A term-document matrix

Document	Term			
	1	2	n	n
1	3	2	…	5
2	4	2	…	3
3	6	2	…	3
4	1	6	…	7
5	1	1	…	1
6	3	7	…	3

program named CKIP system (1999), developed by the Academia Sinica, for that purpose.

CKIP divides the inputted Chinese text into several terms and marks word classes such as noun, verb, or adjective. Hence, we used the word class to select terms. In general, in Chinese, certain phrases mean different things in different categories. Some Chinese sentences also contain English proper nouns or acronyms, some of which also represent categories, whose meaning is familiar to the readership, just as an English legal text might contain Latin or an English fashion text might contain French. Therefore, we selected:

Noun (Na): 球員, 本壘, 果嶺….
Verb (V): 揮桿, 灌籃, 發球….
English proper noun (FW): PGA, NBA, CPBL….

After stop words are removed, we selected terms to represent different categories. We then calculated their term frequency. When a term's frequency is greater than a given threshold, this term will be retained and become a keyword in our lexicon.

4.2. Features extraction

This paper uses two types of features extraction, latent semantic analysis and web page features selection, for web page classification using SVM. The latent semantic analysis extracts common semantic relations between keywords and a document. The web page feature selection extracts the text features from a given web page for inputting to the SVM for training and classification.

4.2.1. Latent semantic analysis

Traditional information retrieval methods use a document and keyword relation to show the results of a document query. Normally, different users will use different keywords to search the same topics. One weakness of this methodology is that relevant documents, which nevertheless do not contain the keyword, will be missed under this system.

Latent semantic analysis applies a vector space concept. All keywords and documents form a two-dimension term-document matrix, singular value decomposition is used to decompose the term-document matrix to obtain the semantic features.

In math, the value analysis is often done with a matrix. In general, using a matrix operation generates solutions. However, if the matrix is a nonsingular matrix, it has a unique solution. But if a matrix is a singular matrix, the value of the determinant is zero. In order to solve linear least squares and singular matrices, the SVD (singular value decomposition) uses the eigenvalue and eigenvector to reduce the dimensions of the original data, filtering irrelevant information. The original matrix has a high dimension. An SVD can reduce the original high term-document matrix dimensions to a low term-document matrix.

Assume a term-document matrix X, which is a $t\times d$ matrix, where t is the number of keywords and d the number of documents (Table 2). Each element $X[t,d]$ is the number of occurrences of keyword t in document d. For example, if the position of X [1,1] is 3, $Term_1$ occurs three times in document Doc_1.

The X of SVD is defined as $X=USV^{\mathrm{T}}$. $S=\mathrm{diag}(\sigma_1\ldots\sigma_n)$, where the elements of S are all singular values of X. Let $n=\min\{t,d\}$, and the singular value is represented by $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_n\geq 0$. U and V are $d\times d$, $t\times t$ matrices, respectively. After processing by the SVD, $X=USV^{\mathrm{T}}$ simplifies to $X_k=U_kS_kV_k^{\mathrm{T}}$, as shown in Fig. 4. The dimensions of U_k, S_k, V_k^{T} are reduced to $d\times k$, $k\times k$, and $k\times t$. The common element k is less than the original vector space. S_k retains k large singular value in term-document. U_k is a document vector, V_k is a term vector. For the training sample, after the Chinese words have been segmented, we construct a term-document matrix for each category. For term-document matrix X_i of each category, we use the SVD to decompose X_i, obtaining three matrixes U, S, V. Because we want to find the common semantic relation between different documents, we only process document vector. For the singular value matrix, the top k singular value is selected. The top k singular value is most important for this data set, as it contains the latent semantic relationship. We add these latent semantic relations into each document vector for the same semantic document. Therefore, we operate $(U_k\times S_k)$

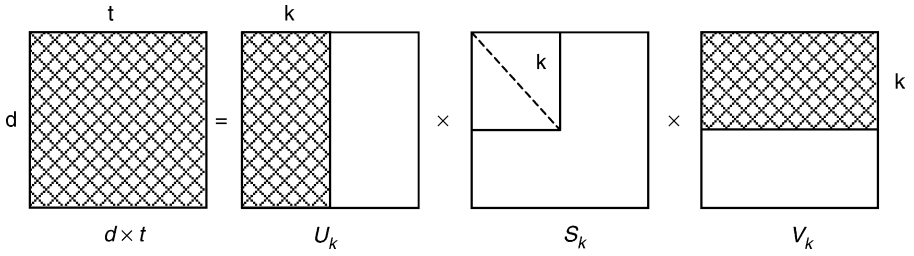

Fig. 4. Using SVD decomposition $d\times t$ matrix.

Table 3
The semantic feature vector

Document	Semantic value				
	1	2	3	.....	*K*
1	−8.30137	0.910919	0.673545	.....	1.787467
2	−5.69583	−1.19023	−0.47056	.....	1.302077
3	−4.84201	−2.27981	−1.87301	.....	0.624073
4	−5.20006	−1.36386	−0.64771	.....	0.833638
5	−7.30432	1.706085	−0.41151	.....	−0.99389

to obtain the semantic feature vector of each document (Table 3).

4.2.2. Web page features selection

Web page features can be used to judge the category of a given web page. In this paper, we extracted four different features of the text, including: (1) The frequency of keywords in a document. A higher keyword frequency value represents a higher probability of belonging to the category containing the keyword. (2) The total number of words displayed in a document. (3) The ratio of the number of keywords to the total number of words in a document. If the ratio is high, then the web page content probably belongs to the category containing those keywords. (4) The average interval between terms. The four features are listed as follows.

$$\text{Total frequency} : \sum_{k=1}^{n} \text{term}_k \tag{1}$$

$$\text{Total of words} : W \tag{2}$$

$$\text{Ratio} : \frac{\sum_{k}^{n} \text{term}_k}{W} \tag{3}$$

$$\text{Average interval} : \frac{I_k}{\sum_{k}^{n} \text{term}_k} \tag{4}$$

where term_k is a number representing the frequency of term_k. I_k is the interval between k-th terms.

If the interval is shorter, the distance between keywords is shorter, and the probability that the web page belongs to this category is higher. These four features are used by the system to determine the category of the web pages (Table 4). These text features are directly extracted from web page content, which clearly determine the category of the page.

Table 4
Text features in one of categories

Document ID	Features			
	Total frequency	Total of words	Ratio	Average interval
1	3	921	0.325733	192
2	11	981	1.121305	79
3	5	700	0.714286	126
4	2	420	0.47619	225
5	5	748	0.668449	95

Table 5
Web page features: values and categorization

Document	Features				Category
1	10	1179	0.848	121	1
2	12	980	1.224	70	1
3	15	899	1.668	57	1
4	17	938	1.812	49	1
5	4	1179	0.339	309	-1
6	4	1221	0.327	318	-1
7	2	1217	0.164	789	-1

4.3. Classification test

After extracting semantic features and text features to make a training sample for each category, these features are then inputted into two different SVMs, one for each category. Two SVM category models are obtained in this way. Test samples were then inputted into the SVM for classification.

The SVM model's learning process is a supervised one. In the training process, a '1' is marked for each web page that belongs to a given category and '−1' for each web page that does not belong to this category (Table 5). After training, the SVM model itself will determine whether sample web pages belong to a given category (Table 6). If the output value is negative, the web page belongs to the '−1' category. If the value is positive, then the web page is belongs to the '1' category.

4.4. Voting and categorization

Voting is a very simple and instinct classification approach to categorization. The voting approach classifies a document into certain category by taking into account a majority of the classifiers. In our experiment, semantic features and text features were used to train two respective SVM models. The two models in turn yielded two types of classification results. Based on the results of the two models, we determined the classification strength of the feature in question, and successfully classified web pages. The weighting voting process workflow is shown in Fig. 5.

Because judgments may not be consistent, we adopted a weighting voting schema to determine the category of web pages. The binary characteristic function T_γ^m is defined as follows:

$$T_\gamma^m(x_i) = \begin{cases} 1, & F_\gamma(x_i) \in \Lambda_m \\ -1, & F_\gamma(x_i) \notin \Lambda_m \end{cases}$$

Table 6
The output of SVM model

Document	Category	Result	Correctly or not
1	−1	−0.445	Y
2	−1	−0.5523	Y
3	−1	−0.2653	Y
4	1	0.2065	Y
5	1	−0.2809	N
6	1	−0.0765	N

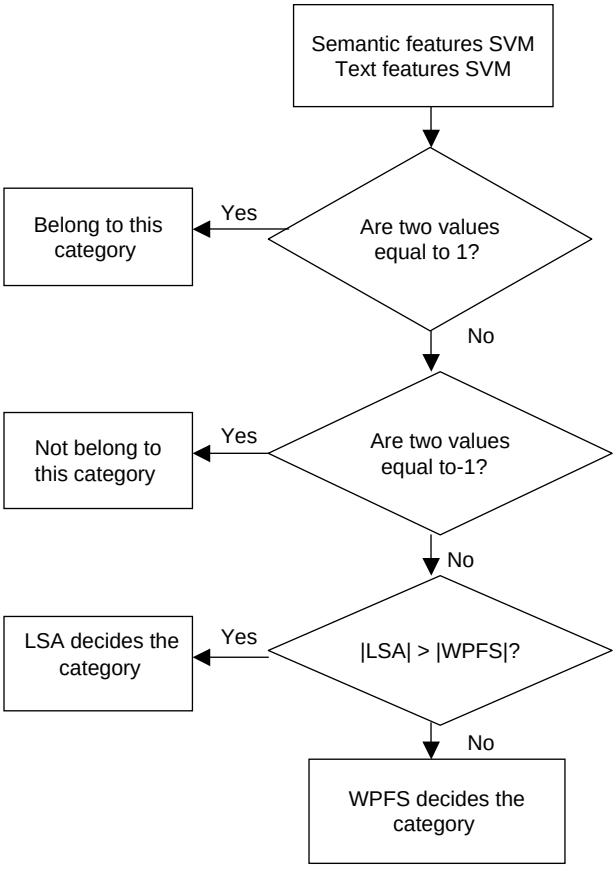

Fig. 5. The workflow of voting process.

which F_γ is semantic features or text features, x_i is document, index γ is model of SVM, Λ_m is type of categories.

When both these two features to deem 1 or -1, the web page is belong to this category or not belong to this category. It is expression as follows:

$$E^m(x_i) = \begin{cases} \Lambda_m, & \bigcap_{y=1}^{2} T_\gamma^m(x_i) = 1 \\ \overline{\Lambda_m}, & \bigcap_{y=1}^{2} T_\gamma^m(x_i) = -1 \end{cases}$$

When one feature is deemed 1 and another one is deemed -1, we vote on which category the web page belong to, according the weight of features. The voting schema is depicted:

IF $|T_1^m(x_i)| > |T_2^m(x_i)|$
  THEN $T_1^m(x_i)$ decision the classification
ELSE
  $T_2^m(x_i)$ decision the classification

Table 7 is the output of two features using SVM. For example, in the document 4 in Table 7, the value of LSA $|1.01013|$ is higher the value of WPFS $|-0.423037|$, so this web page belongs to the '+1' category.

Table 7
Deciding whether a web page belongs to a category

Document	LSA	WPFS
1	−0.927559	−0.852245
2	−0.82182	−0.82322
3	−1.29774	−0.983878
4	1.01013	−0.423037
5	−0.341236	−0.966568
6	−1.04097	−1.03357
7	−2.36632	−0.922749
8	−1.19163	−0.952946
9	−1.04793	−1.03955
10	0.595351	0.903189
11	−0.233816	−1.02759

Table 8
Four situations of classification result

	The system classified category X	The system does not classify category X
Belongs to category X	A	B
Not belongs to category X	C	D

A, The number of pages classified to Category X and belonging to Category X; B, The number of pages not classified to Category X, but belonging to Category X; C, The number of pages classified to Category X but not belonging to Category X; D, The number of pages not classified to Category X and not belonging to Category X. The formula of precision, recall and *F*-value is listed as follows. Precision (P) = A/(A + B). Recall(R) = A/(A + C). F-value = 2PR/(P + R)

Table 9
Data set

Category no.	Category name	Number of web pages
1	Basketball	400
2	Baseball	346
3	Golf	300
4	Tennis	200
5	Volleyball	150
6	Soccer	95
7	Billiards	54
8	Football	96
9	F1 race	83
Total		1724

Table 10
The *F*-value of kernel functions

Category no.	Category name	Anova *F*-value (%)	Polynomial *F*-value (%)	Dot *F*-value (%)	Neural *F*-value (%)
1	Basketball	99	98	98	98
2	Baseball	80	80	72	72
3	Golf	99	99	99	99
4	Tennis	93	92	89	88
5	Volleyball	99	94	95	96
6	Soccer	100	96	84	84
7	Billiards	100	90	100	100
8	Football	54	53	55	54
9	F1 race	94	65	98	99
Average		91	85	87	87

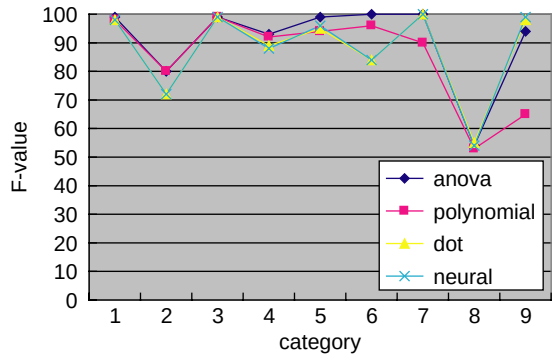

Fig. 6. The comparison of the four kernel functions.

4.5. Performance evaluation

In our experiment, each category uses two SVM models to do the training and testing. Therefore, in the performance evaluation we used the *F*-value to evaluate the classification performance of each category. The *F*-value integrates two norms: precision and recall. The classification result of each category has four possible outcomes (Table 8). We used the *F*-value to evaluate our performance.

5. Experiment

In this section, we designed an experiment to test the performance of the WVSVM. We also investigated LSA-SVM models and BPN (back propagation network) to compare their classification performances. The experiments are described below.

5.1. Experiment environment

Our experiment uses a Pentium-4 2.4 G MHz computer with 256 MB RAM. Java language was implemented on a Windows 2000 Professional operating system. The lexicon is stored in an MS Access database.

Table 11
Precision of WVSVM, LSA- and BPN methods

Category no.	Category name	WVSVM precision (%)	LSA-SVM precision (%)	BPN precision (%)
1	Basketball	100	100	87
2	Baseball	84	67	76
3	Golf	100	99	75
4	Tennis	95	90	83
5	Volleyball	100	98	67
6	Soccer	90	100	68
7	Billiards	100	100	57
8	Football	47	43	73
9	F1 race	93	89	82
Average		90	87	74

Table 12
Recall of WVSVM, LSA- and BPN methods

Category no.	Category name	WVSVM recall (%)	LSA-SVM recall (%)	BPN recall (%)
1	Basketball	97	99	91
2	Baseball	100	100	94
3	Golf	100	100	94
4	Tennis	95	95	85
5	Volleyball	100	100	56
6	Soccer	100	100	72
7	Billiards	100	100	66
8	Football	78	72	75
9	F1 race	100	100	69
Average		97	96	78

5.2. Data set

We use sports news to test system performance. The sports news was downloaded from the udndata website udndata, a popular sports news site with data going back many years. The popularity of various sports news varies, and less popular sports are reported at lower frequencies. By contrast, news of popular sports is reported very frequently. Therefore, we selected sports items that were reported more frequently in the news. This included basketball, baseball, golf, tennis, volleyball, soccer, billiards, football, and Formula 1 Racing. Table 9 shows the data set. For the training set, we randomly selected a part of data from each category, leaving the remainder for the test set. The ratio of training set to test set is 2:1 approximately.

5.3. Experiment design

First, in order to obtain the highest classification performance, we tested anova kernel, polynomial kernel, dot kernel and neural kernel in the SVM, and compared the classification performance of these four kernel functions. The best effective kernel function was then selected. We classified the Chinese sports news web pages using SVM with the best kernel function. We also tested LSA-SVM and BPN classification performance using the same data set. After the classification experiment, the *F*-value was used to evaluate the classification performance.

Table 13
The *F*-value of WVSVM, LSA-and BPN methods

Category no.	Category name	WVSVM *F*-value (%)	LSA-SVM *F*-value (%)	BPN *F*-value (%)
1	Basketball	98	99	89
2	Baseball	92	80	76
3	Golf	100	99	77
4	Tennis	95	93	83
5	Volleyball	100	99	60
6	Soccer	95	100	70
7	Billiards	100	100	56
8	Football	59	54	73
9	F1 race	96	94	73
Average		93	91	73

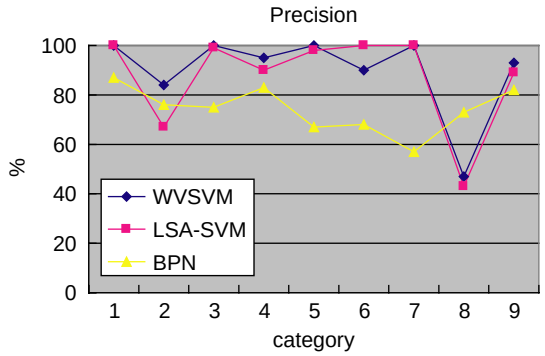

Fig. 7. The comparison of the precision of the three methods.

5.4. Simulate result

Table 10 shows that the average of *F*-values for anova kernel, polynomial kernel, dot kernel and neural kernel in SVM are 91, 85, 87, and 87%, respectively. The performance of the polynomial kernel is clearly below that of the other kernels, while the dot and neural kernel are nearly identical. The anova kernel has the best result. Fig. 6 shows that for the same category of news, an anova kernel function will yield better results. As a result, an anova kernel was selected for the next step.

In this experiment, an LSA-SVM and back propagation network (BPN) are given the same data set, respectively, to compare them to a WVSVM. The LSA-SVM sends the features extracted from the LSA operation to the SVM using anova kernel function to train and classify. For the neural network, back propagation network was adopted. We merged the features of the LSA and the WPFS to input to the BPN for training and classification. Precision, recall and F-value are used to measure the performance of these three methods.

The precision of the WVSVM, LSA-SVM, and BPN is shown in Table 11, the recall of the WVSVM, LSA-SVM and BPN is shown in Table 12 and the *F*-value of the WVSVM, LSA-SVM and BPN is shown in Table 13.

The average precision for WVSVM, LSA-SVM and BPN are 90, 87, 74, respectively (Table 11, Fig. 7). With the exception of category number 8 (football category), the precision of each category for WVSVM and LSA-SVM is higher than that of the BPN. This indicates that the two SVM-based methods yield better precision than the BPN.

The recall results for the WVSVM, LSA-SVM and the BPN are 97, 96, and 78%, respectively. The recall of WVSVM and

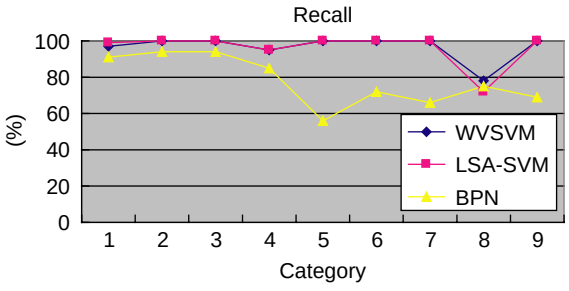

Fig. 8. The comparison of the recall of the three methods.

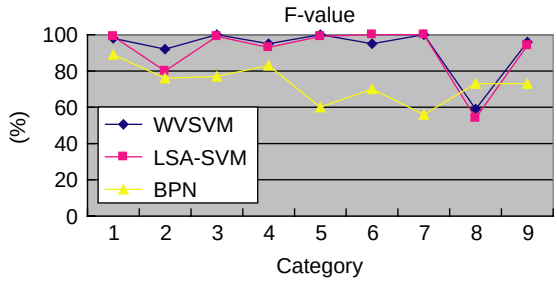

Fig. 9. The comparison of the *F*-value of the three methods.

LSA-SVM are nearly the same and both are higher than the BPN. The WVSVM and LSA-SVM can find related web pages mapping to recall, and classify them into the correct category mapping to precision, with a high degree of reliability (Fig. 8).

In the LSA-SVM method, only the features of the LSA are used, but the BPN method not only uses the features of the LSA, but also uses the features of the WPFS. The simulated result shows that the average *F*-values of the LSA-SVM and BPN are 91 and 73%, respectively. This indicates that the SVM yields a better classification result than the BPN. The average *F*-values of the LSA-SVM and WVSVM are 91 and 93%, respectively. This shows that the WVSVM method, which adds the features of the WPFS, is able to increase classification accuracy. The Billiards and Formula 1 race categories, numbers 7 and 9, contain a smaller number of web pages (54 and 83). The *F*-values of the BPN for these categories are 56 and 73%, respectively, while for the WVSVM they are 100 and 96%, respectively. This indicates that the WVSVM is able to effectively process categories with fewer documents. Thus, WVSVM can correctly categorize unknown web pages with lower number training samples. Fig. 9 shows that among the three methods, the WVSVM has the highest classification result. In category number 8 (football), the precision, recall and F-value are the lowest of these nine categories, probably because this category is relatively more closely related to another category, soccer. After segmentation of the Chinese words, a number of keywords are shared between the two categories, such as '射門', '四分衛', and so forth. These keywords will interfere with the extraction of web page features, making categorization more difficult.

6. Conclusion and future work

In this paper, we have proposed a web page classification method using an SVM based on a weighted voting schema. The feature vectors are extracted from both the LSA and WPFS methods. The LSA can extract common semantic relations between terms and documents. The LSA then classifies semantically related web pages, offering users more complete information. The WPFS extracts four text features from the web page content. The category of a web page can be correctly determined by the WPFS. We also compared the SVM performance using four different kernel functions performance. The experimental results show that the anova kernel function yields the best result of these four kernel functions.

The LSA-SVM, BPN and WVSVM were then compared. The experiment demonstrated that the WVSVM yields better accuracy even with a small data set. When the smaller category has less training data, the WVSVM is still able categorize web pages with acceptable accuracy.

In future research, we will incorporate domain ontology to assist in domain web page classification. The resulting system will be able to find the related semantic web pages from other domains.

References

Apte, C., Damerau, F., & Weiss, S. M. (1998). *Text mining with decision trees and decision rule. Proceeding of the automated learning and discovery conference*. Carnegie-Mellon University.

CKIP (1999) *Chinese words segmentation program*, Central Academia Sinica, Taiwan, http://ckip.iis.sinica.edu.tw/CKIP/tool

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Google, http://www.google.com

Gunn, S. R. (1998). *Support vector machines for classification and regression. ISIS technical report*. Image speech and intelligent systems group of University of Southampton.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European conference on machine learning ECML-98* (pp.137–142).

John, M. P. (2000). Practical issues for automated categorization of web pages. In *proceeding of ECDL2000 workshop on the semantic web*.

Kwon, O. W., & Lee, J. H. (2003). Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing and Management*, *39*, 25–44.

Mccallum, A., & Nigam, K. (1998). *A comparison of event models for Naive Bayes text classification. AAAI-98 workshop on learning for text categorization*.

Mitchell, T. M. (1997). *Machine learning*. Boston, MA: McGraw-Hill.

Pchome, http://www.pchome.com.tw

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann Publishers.

Sebastian, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*, 1–47.

Selamat, A., & Omatu, S. (2004). Web page feature selection and classification using neural networks. *Information Sciences*, *158*, 69–88.

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications* , 1–5.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd international ACM SIGIR conference on research and development in information retrieval* (pp. 42–49).

udndata, http://udndata.com

Yahoo, http://tw.yahoo.com