Using Support Vector Machine for Integrating Catalogs

Ping-Tsun Chang[1] and Ching-Chi Hsu
Department of Computer Science
and Information Engineering,
National Taiwan University,
Taipei 106, Taiwan.
e-mail[1]: r89001@csie.ntu.edu.tw

ABSTRACT

We discuss a problem of integrating classified documents into another master catalog. This problem is useful in electronic commerce and web portals. The key insight of this problem is that many of the data source have their own categorization, and classification accuracy in master categorizations can be improved by using the implicit information in these source categorizations. In this paper, we use support vector machines (SVM), which have been shown to be efficient and effective for classification. In this paper we apply this fine classification method to this problem. Our experienment results show substantial improvement in accuracy of this problem.

Keywords: Support Vector Machines, Electronic Commerce, Categorization, Data Mining, Catalog Integration.

1 INTRODUCTION

In most electronic commerce web sites, catalogs are important information for customers. If you have a well-established web site for electronic commerce, your product data must categorized for user and management. Noticing your success, a major distributor wants to join your electronic commerce systems. This distributor would have a large amount of categorized product data. Your problem is how to effectively and automatically integrate this distributor's catalog into your catalog.

This paper presents a new technique to integrating catalogs automatically by SVM. SVM have presented as very success machine learning and pattern classification technology for text categorization and numerous other domains.

1.1 Problem Definition

First, we define this problem as before in formal. For each document *d*, there is a set of words. We represent *d* by numerical data, which consist of numerous pair of feature and related value [1]. We can regard a document *d* as a product description.

A catalog is a separation of a set of documents into a set of categories. In this problem, we are given two catalogs: (a) Master Catalog *M* and (b)

Source Catalog *N*. The goal is finding the corresponding category in *M* for each document in *N*.

1.2 Related Work

Naïve Bayes Classifier

Bayesian decision theory is a fundamental statistical approach to the problem of text categorization. This approach is based on quantifying the tradeoffs between various classification decisions using probability and the costs that accompany such decisions.

According to Bayes formula (5), we can convert the posteriori probability *P(C|x)* from prior probability *P(C)*. Our training process is in turn to estimate the proper parameters of our model, such as *P(x)* and *P(x|C)*.

$$P(C_i \mid x) = \frac{P(x \mid C_i)P(C_i)}{P(x)} \qquad (5)$$

In a simple case with two categories and only one feature, it's rather instinct that we have better chance to make the right decision if we always choose the category with the larger conditional probability $P(C_i \mid x)$.

In recent research [1], Naïve Bayes classification is well applied in the integration catalogs problem. Using Naïve Bayes classification, we can incorporate the similarity information present in source catalogs easily. The challenge is applying better classification technologies to this problem.

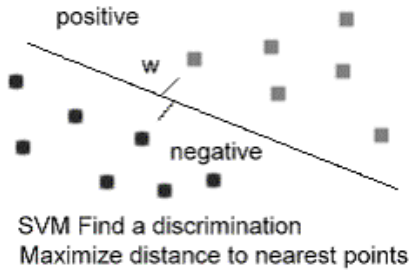
Support vector machine

The support vector machine is a new machine technique used for classifier. Vapnik introduced SVM in his work on structural risk minimization [3]. The general SVM solve the primal problem as (1), which is given training vectors $x_i \in R^n$, i = 1…l, in two classes, and a vector $y \in R^l$ such that $y_i \in \{1,-1\}$ [3].

$$\min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \quad (1)$$

$$y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i, \xi_i \geq 0, i = 1,...,l.$$

In a simplest form, there are binary examples in a two-dimension plane. A linear SVM is a line, two-dimension hyper plane separate s set of positive examples from a set of negative examples with maximum margin as Figure 1. In general case, find a decision function as a hyper plane separating training data and maximize the margin. It is possible that a separable hyper plane in the space does not exist. In such case, we define a parameter *C*, which is the penalty imposed on training examples that fall on the wrong side of the decision boundary. In the linearly separable case, this problem can be expressed as an optimization problem in Eq. 2 [5].



SVM Find a discrimination
Maximize distance to nearest points

Figure 1: A diagram of Linear SVM

$$\text{Minimize } \frac{1}{2}\| \omega \|^2 \text{ subject to}$$

$$y_i(\omega \cdot x_i - b) \geq 1, i = 1,...,l \text{ . (2)}$$

k-Nearest Neighbor

K-nearest neighbor (kNN) classification is an instance-based learning algorithm [13] that has shown to be very effective in text classification. It assumes all instances correspond to points in the *n*-dimensional space $\Re^n$. The similarity (or distance) between instances is defined in terms of the standard Euclidean distance. More precisely, let an arbitrary instance *x* be described by the feature vector

$<a_1(x), a_2(x), \ldots a_n(x)>$

where $a_r(x)$ denotes the value of the *r*th attribute of instance *x*. Then the distance between two instances x_i and x_j is defined to be $d(x_i,x_j)$, where

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2}$$

kNN is a learning method based on similarity. Because the specific characteristics of kNN, it performs well than other learning method in text categorization problem except SVM. In recent research, kNN is still a well-used method in text categorization, because its well performance and easily implementation. The major problem of kNN is insufficient of training data. When training data is not enough, kNN performs incredible worse than the same method with enough training data.

Text Categorization

Text categorization (or classification) is the task of automatically assigning predefined categories to natural language texts. In its broader sense, it can be seen to estimate a membership function, which takes a document d_i as it input and maps this

Input into a set of class c_j. In other words, this membership function tries to determine the binary relationship vector $\{ a_{i1}, a_{i2}, \ldots, a_{im} \}$ for each document d_i according to its own document features (may be extracted from content or other structural information) where *m* is the total number of predefined categories and a_{ij} are values from $\{ 0, 1 \}$. A value of 1 for a_{ij} is interpreted as document d_i belongs to category c_j, while 0 means document d_i doesn't belong to category c_j. The method described above is used common in this problem.

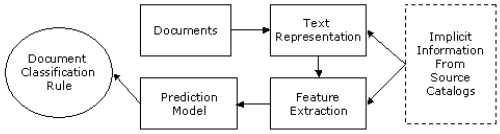


Figure 2: A Overview of Integrating Catalogs

We can solve this problem by the following steps [13]: (1) Text Representation, (2) Feature Extraction, (3) Construct prediction model, (4) Document classification rules. In this problem, we pay more attention to step (2) and (3), then we could apply some research about text classification in recent to this problem as Fig. 2.

2 TEXT CLASSIFICATION USING SUPPORT VECTOR MACHINE

A support vector machine (SVM) algorithm has been shown in previous work to be both fast and effective for text classification problems [6, 7, 10].

In text and language processing domain, the problem are usually based on vector space model. In vector space model, we represent a text document as a numerical vector.

Typically, the document vector is very high dimensional, at least the thousands for large collections. The SVM is for new machine technique alone this line.

2.1 Multi-class Classification

Support vector machines were originally designed for binary classification. There are many on-going researches for extending the original design for multi-class classification. We choose "one-against-one" [11] approach for multi-class SVM. This method was shown that the method is more suitable for practical use than other methods [12]. This method constructs C_2^n classifiers where each one is trained on data from two classes.

Then, we have a decision function:

$$\arg\max_{m=1,\ldots,k}(\sum_{i=1}^{l}(C_i^m A_i - \alpha_i^m)K(x_i, x) + b_m), m = 1,\ldots,k \quad (3)$$

2.2 Using SVM for Text Classification

The document dataset are all text data. Even if we change this dataset to numerical expression by a vector of TF・IDF (Term Frequency * Inverse Document Frequency). TF・IDF is the standard choice for kernel function in text mining or information retrieval community as follow: [4].

$$\phi_i = \frac{TF_i \log(IDF_i)}{\kappa} \quad (4)$$

Where κ is normalization constant ensuring that $\| \phi \|_2 = 1$. The function $K(x,z) = <\phi(x) \cdot \phi(z)>$ is clearly a valid kernel, since it is the inner product in an explicitly constructed feature space.

Using this kernel function, the document is a vector with highly dimension and the dataset is usually unbalanced. The original SVM could not handle this problem. The SVM parameter *C* and *p* must to be determined [2]. *C* is the penalty for examples, which fall on the wrong side. *p* is the decision threshold. We can tune *p* to control classification precision and recall.

Feature Selection

Before compute TF・IDF, we select the terms which could discriminate documents as feature. About feature selection, we focused on Chinese text in this paper. Chinese documents are becoming widely available in the Internet. By the growing up importance of Chinese in the Internet, we discuss the different process approach in feature selection. We have some special problem for Chinese text as follow: (1) For Chinese text (and others like Japanese and Korean), there are no space characters to delimit words. (2) For large amount of Chinese words, we should implement a hashing function for the character index and the posting retrieval speed can be very fast. (3) Not exist a general Chinese dictionary contained enough important words for information retrieval, like proper nouns. We use Bigram-based indexing, which is a completely data-driven technique. It does not have the out-of-vocabulary problem.

3 SVM FOR INTEGRATING CATALOGS

We discuss the basic method in [1] first. Then we illustrate how to apply SVM for this problem.

3.1 On Integrating Catalogs

The goal is using the information from source catalogs to classify the products in source catalogs *S* to master catalogs *M*. Using the implicit information of S, the straightforward idea is that using conditional probability to define this relation. By this way, we discuss the traditional method, which is called Naïve Bayes classifier in this section.

Using Naïve Bayes Classifier

Given a document *d*, the Naïve Bayes classifier estimates the posterior probability of category C_i .

$$P(C_i \mid d) = \frac{P(C_i)P(d \mid C_i)}{P(d)} \quad (5)$$

$P(d)$ is the same for all categories, so that we can ignore this term of Eq. 5. To estimate the term $P(d| C_i)$, we assume that all the words in d are independent of each other. The probability of the document *d* is the product of the multiplication of the probability of all terms *t* in *d*.

$$P(d \mid C_i) = P(t_1 \mid C_i) \cdot P(t_2 \mid C_i),...,\cdot P(t_k \mid C_i) = \prod_{t \in d} P(t \mid C_i) \quad (6)$$

If the document consists of both text and features, we can assume the independence between the text and the features to compute Eq. 7.

$$P(d \mid C_i) = P(d_{text} \mid C_i) \cdot P(d_{attr} \mid C_i) \quad (7)$$

$P(C_i)$ is estimated by Eq. 8.

$P(C_i)$ =(Number of documents in category C_i)/(Total number of documents in the dataset) (8)

To estimate $P(t| C_i)$, we compute the number of documents which occurs term t and in category C_i. Let n(C_i ,*t*) represent the value, and let n(C_i) represent the number of documents in category C_i. Then we can estimate for $P(t| C_i)$ is n(C_i ,*t*)/ n(C_i). For avoiding divided by zero and zero probability, we apply Lidstone's law to the equation. Finally, we can estimate for $P(t| C_i)$ by Eq. 9.

$$P(t \mid C_i) = \frac{n(C_i, t) + \lambda}{n(C_i) + \lambda \mid V \mid}, \lambda \geq 0 \quad (9)$$

Where |V| is the number of vocabulary. (i.e., feature)

According to the equation above, we can integrate catalogs from S to M straightly. First, for each category *C* in *M*, we can evaluate $P(C_i)$ and $P(t| C_i)$ by Eq. 8 and Eq. 9 respectively. Second, for each document *d* in S, we can evaluate $P(C_i|d)$ by Eq. 5 using the result computed in first step. Finally, we assign *d* to the category with the highest value for $P(C_i|d)$.

3.2 Using SVM for Integrating Catalogs

The major problem of Naïve Bayes classification is that (1) the classification is dominated by large category, and (2) the classifier is bias by a few special examples. (3) In practice, Naïve Bayes classifier predictive performance is not as strong as in the training cases. This phenomenon is described as over fitting. Consequently, Naïve Bayes classifier usually has less performance than SVM in text classification community [6]. But in the problem of integrating catalogs, it is easily to use the implicit information of source catalog by conditional probability.

If the information of source catalog is useful for classification, we can assume that a category C_i in S could be mapping to several

category in *M*. To emphasize this information in numerical vector data, we control the weight β of the centroid of all examples in this category of *S*. Increasing β, result in more effect from *S* for classification to *M*. More separate example let SVM finding the hyper plane with maximize margin easily. We illustrate the idea as a new kernel function as Eq. 10. In this equation, we control the distribution of examples to present the information of source catalogs.

$$\phi'(d) = \beta \cdot \phi(d) + (1-\beta)\frac{1}{|C|}\sum_{d \in C \in s}\phi(d) \quad (10)$$

1. For each category C_i in master catalog *S*, determine the weight β_i of the relation between C_i and M
2. Compute the new representation of *d* in *S*.
3. Classification d by SVM trained by all documents in M.

Figure 3: SVM Integrating Catalog Algorithm

This kernel function is a term reweighing scheme. The function emphasizes the important term of the source category. In the view of the vector space, the function aggregates the vector to the centroid. Increasing β indicate that we believe the information of source catalog is useful to classify in master catalog more.

It is possible that the source catalog is orthogonal to master catalog. In other words, the information of source catalog is completing irrelevant for classification in master catalog. For example, if the categorization in source catalog is by region and the categorization in master catalog is by season. We could treat these two catalogs orthogonal. Under this situation, the kernel function will be the same as standard classifier without information of source catalog when β=0.

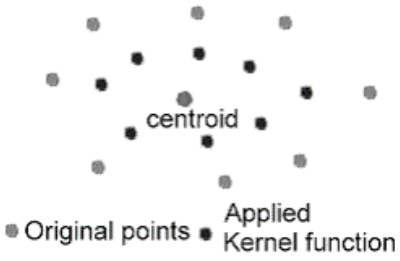


Figure 4: The kernel function for integrating catalogs

4 EXPERIENMENTAL RESULTS

For our experienment, we use a large collection of classified documents from books.com.tw [8] and bookzone.com.tw by Commonwealth Publishing Company [9]. These datasets are both classified book descriptions from their popular electronic commerce web sites.

4.1 Catalogs Description

Dataset	Categories	Docs
Books.com.tw	24	2,374
Bookzone	20	917

Table 1: Dataset Characteristics: Books.com.tw and Commonwealth Publishing Company

These two datasets are Chinese catalogs. In our experienment, we use Bigram-based indexing for feature selection and a hashing function for indexing and retrieval.

In these two catalogs, all documents contain the information of author, publisher, the printed date, printed location, ISBN number, price, content, and a short essay for introduction to the book. The size of each document is about 60~80K bytes with

500~600 Chinese words.

The major difference of these two catalogs is their size. Books.com.tw is a general book stores on Internet, but Bookzone just only sales their own books. Books.com.tw declare that there is 120,000 books in their database. Almost all books in bookzone could be founded at Book.com.tw. So we collect a part of this database. We collect 2,374 documents from Books.com.tw and collect 917 documents from bookzone (Table 1). The book distribution of Bookzone is very unbalanced (2/3 books in category Finance, Science, Psychology, and Literal) and a few books. This is unfavorable for Naïve Bayes classifier.

4.2 Experienment

The purpose of the experienment was to see how much we could increase accuracy by using SVM and such kernel function. For comparison, we also compute using SVM and Naïve Bayesian classifier without any information with source catalog.

Table 2 shows the average accuracy of the different classification strategies: Naïve Bayes classifier and SVM. We can find that even if we just only classify without any information with source catalogs, the performance of SVM is still better than Naïve Bayes classifier.

To measure the performance of integrating catalogs, we compare the accuracy of these classification strategies to our method with the information of source catalogs. If the information of source catalogs is not orthogonal to master catalogs, we can use the information as far as possible. The results show that our method can make a meaningful improvement of classification accuracy.

We tune β as 0.3 to reach a state-of-the-art performance of our method. These two catalogs are similar enough for using the information of source catalog.

In Table 2, we compare the accuracy of traditional learning method. Our experienment also shows that SVM could performs well than Naïve Bayes classifier. We never used implicit classification information. The “improve” column shows the percent improvement from only using training examples to adding the information of classification. Our experiments show that if using this information well, we can have a large improvement of classification accuracy.

	Accuracy (%)			
Dataset	Naïve Bayes	SVM	Our	Improve
Finance& Business	53.45	56.12	64.11	14.24 %
Computers	57.68	57.80	65.50	13.32 %
Science	51.12	57.78	62.00	7.30 %
Literature	37.39	40.13	53.78	34.01 %
Psychology	47.66	54.44	59.96	10.14 %
Average	49.46	53.25	61.07	15.80 %

(a) Train: Books.com.tw, Test: Commonwealth

	Accuracy (%)			
Dataset	Naïve Bayes	SVM	Our	Improve
Finance& Business	42.77	49.74	53.12	6.80 %
Computers	45.60	48.25	55.54	15.11 %
Science	41.14	44.60	47.72	6.99 %
Literature	30.99	37.21	42.21	12.44 %
Psychology	40.05	40.11	43.35	8.08 %
Average	40.11	43.98	43.39	10.08 %

(b) Train: Commonwealth, Test: Books.com.tw

Table 2: Experienment Classification Accuracy

5 CONCLUSIONS

By the results above, we believe that SVM is very useful to the problem of integration catalogs with text documents. Traditionally, SVM is a classification tool. In this paper, we using SVM with a novel kernel function to suit this problem. The experienment here serves as a promising start for the use SVM for this problem. We can also improve the performance by incorporation of another kernel function and proved it, or combining structural information of text document, requires future work.

6 REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, On Integrating Catalogs, *In proceedings of The 10th International World Wide Web Conference*, Hong Kong, 2001.

[2] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001.

[3] Vladimir N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.

[4] Nello Christianini and John Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge University Press, 2000.

[5] Susan Dumais and Hao Chen, Hierarchical Classification of Web Content, *In proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR '00)*, 2000.

[6] Yiming Yang, A re-examination of Text Categorization Methods, *In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR '99)*, 1999.

[7] James Tin-Yau Kwok, Automated Text Classification Using Support Vector Machine, *International Conference on Neural Information Processing(ICNIP '98)*, 1998.

[8] Book.com.tw Company, http://www.book.com.tw

[9] Bookzone by Commonwealth Publishing Company, http://www.bookzone.com.tw

[10] Thorsten Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevent Features. *In European Conference on Machine Learning (ECML '98)*, pages 137-142, Berlin, 1998, Springer.

[11] Knerr, S., Personnaz, L., & Dreyfus G., Single-layer Learning Revisited: a stepwise procedure for building and training a neural netwrk, In J. Fogelman (Ed.), *Neurocomputing: Algorithms, architecture and applications*. Springer-Verlag,1990.

[12] Chih-Wei Hsu and Chih-Jen Lin, A Comparison of Methods for Multi-class Support Vector Machines, 2001.

[13] Sholom M. Weiss, Chidanand Apte, Fred J. Damerau, David E. Johnson, Frank J. Oles, Thilo Goetz, and Thomas Hampp, Maximizing Text-Mining Performance, In IEEE Transactions on Intelligent Systems, (Vol. 14, No. 4), pp. 63-69, 1999.