
Data Mining and Knowledge Discovery, 12, 229–248, 2006

DOI: 10.1007/s10618-005-0015-5

Reinforcing Web-object Categorization Through Interrelationships

GUI-RONG XUE grxue@sjtu.edu.cn
YONG YU yyu@cs.sjtu.edu.cn
Computer Science and Engineering, Shanghai Jiao-Tong University, Shanghai, 200030, P.R.China

DOU SHEN dshen@ust.hk
QIANG YANG qyang@cs.ust.hk
Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong

HUA-JUN ZENG hjzeng@microsoft.com
ZHENG CHEN zhengc@microsoft.com
Microsoft Research Asia, 5F, Sigma Center, 49 Zhichun Road, Beijing, 100080, P.R.China

Received April 6, 2005; Accepted July 27, 2005

Published online: 4 April 2006

Abstract. Existing categorization algorithms deal with homogeneous Web objects, and consider interrelated objects as additional features when taking the interrelationships with other types of objects into account. However, focusing on any single aspect of the inter-object relationship is not sufficient to fully reveal the true categories of Web objects. In this paper, we propose a novel categorization algorithm, called the Iterative Reinforcement Categorization Algorithm (IRC), to exploit the full interrelationship between different types of Web objects on the Web, including Web pages and queries. IRC classifies the interrelated Web objects by iteratively reinforcing the individual classification results of different types of objects via their interrelationship. Experiments on a clickthrough-log dataset from the MSN search engine show that, in terms of the F_1 measure, IRC achieves a 26.4% improvement over a pure content-based classification method. It also achieves a 21% improvement over a query-metadata-based method, as well as a 16.4% improvement on F_1 measure over the well-known *virtual document-based method*. Our experiments show that IRC converges fast enough to be applicable to real world applications.

Keywords: categorization, interrelated Web objects, iterative reinforcement, clickthrough data.

1. Introduction

The advent of the World Wide Web has rejuvenated research interest in text categorization. A huge number of Web pages are available on-line, and categorizing them into meaningful semantic categories is a practical and challenging research problem.

Traditional *content-based* Web-page-categorization approaches use simple representation schemes that are based on word-occurrence statistics. However, such a representation approach is deficient in several aspects. First, non-text objects, such as images and scripts, which are meaningful parts of Web-page content, are unusable by text classifiers since they cannot be represented well in this representation method. Furthermore, even if textual information in the Web pages can be utilized fully, they may still contain too much noisy information which cannot be distinguished by this representation method. Second, the Web pages are created by different authors who may have

no coherent page-construction styles and structures. Thus, it is impossible to include all this information by directly applying the vector space model for text categorization.

Recently, some emerging applications, such as Web mining and collaborative filtering, started to focus on multiple data collections and multiple data types. In such applications, data objects are of different types but highly interrelate with each other. In Figure 1, we illustrate these data objects in the Web environment, where the objects consist of Web pages, users and queries. These three types of objects are interrelated: users issue queries in order to search Web pages, users browse Web pages related to their previously issued queries, and queries reference the Web pages. It is clear from this picture that when we classify the Web users, the pages they browse and the queries that they issue should play an important role. Similarly, when classifying Web pages, users and queries should be considered as well.

The intuitive approach to exploiting the information hidden in the interrelationships is to augment the features by the additional features in the Web objects. This approach takes the interrelated Web objects of an object as its additional features since related objects are likely to have similar properties. For example, in Web-page categorization (Beeferman and Berger, 2000; Wen et al. 2001), the feature vector of a Web page can be constructed by combining its content and the queries that are associated with the page through the clickthrough data. In such cases, the relationships or interactions between Web pages and queries are only considered in the feature representation step. That is, each data object is represented by two sets of features: one is extracted from the data object itself like the content features of Web pages; another is extracted from the objects related to the target object, such as content features in related queries. We denote these methods as *virtual-document* categorization. In our experiment, to exploit the interrelationship in this way can slightly improve the performance over the content-based categorization approach. Thus, there is still headroom for harnessing such interrelationships between heterogeneous objects.

Interrelated heterogeneous objects are likely to have similar topics. Therefore they are also likely to share similar category information. So, after classifying the objects based on their content, is it possible to impose the category information of the interre-

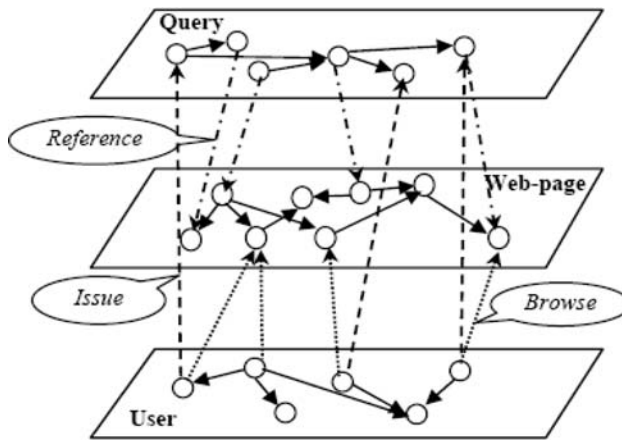

Figure 1. An example of multi-type interrelated data.

lated objects in improving the categorization performance? In this paper, we propose a novel classification algorithm, the Iterative Reinforcement Categorization (IRC), to fully exploit interrelated relationships. In this algorithm, the category information of one object is reinforced by the category information of all its interrelated objects; and the updated category information of the object consequently reinforces the category information of its interrelated objects. That is to say, such an iterative reinforcement process continues until it converges to a conclusive result. The difference with our method from the *virtual-document* method is that we use the category information to reinforce the categorization results instead of the content feature of queries directly. In fact, each Web page is usually interrelated to a few queries while the dimension of the query vector space is very high (up to 46,000). Therefore, the *virtual-document* categorization algorithm may suffer from severe data sparseness; which happens when each query is related only to a few features in a large space of features. This is an issue resolved by our IRC algorithm.

Our IRC algorithm can be directly applied to multiple types of data as well as those from multiple sources, so long as the objects in these data are interrelated. When data come from multiple sources, we can model these linkages where the intra-source links describe the intra-type relationship between the data objects within a single source, and the inter-source links describe the inter-type relationship between different data sources. We show in this paper that our algorithm is general enough to include any number of data sources; however, to focus our attention, we demonstrate how to apply the IRC algorithm to two main Web objects: Web pages and user queries.

The novelty of our work can be seen from several aspects. First, we extend the traditional classification methods to multi-type interrelated data objects. We aim to classify interrelated data objects of different types simultaneously using both their content features and their relationship with other types of objects. Second, we present a reinforcement algorithm to classify interrelated Web data objects by introducing a new way to exploit the interrelationship among the multi-type data objects. In this algorithm, the category information of one type is propagated to reinforce the categorization of another interrelated data object, vice versa, as an iterative process.

We perform comprehensive experiments on the clickthrough dataset from the MSN search engine to evaluate the proposed approach. In these experiments, data are from two different sources: the Web pages and the clickthrough-log data. In two data sources, there exist two types of data objects: queries and Web pages. Our experiments show that, in terms of the F_1 measurement, IRC achieves a 26.4% improvement over a pure content-based classification method, a 21% improvement over a query metadata-based method, and a 16.4% improvement over a virtual document-based method.

The rest of the paper is organized as follows. In Section 2, we review some related work on Web classification and clickthrough data analysis. In Section 3, we explain the IRC algorithm and analyze the properties of the algorithm. Experimental results are reported in Section 4. We give conclusions and discuss possible future work in Section 5.

2. Related work

Considering the problems we discussed, there are two major fields related to our work. One is Web-page categorization and the other is clickthrough data analysis.

2.1. *Web-page categorization*

The first collection of related works is about classification of Web pages which can be divided into two categories: content-based classification techniques and link-analysis-based classification techniques.

The former relies on the textual content of Web pages being classified. The content of Web pages is usually represented by Vector Space Model (VSM). Joachims (1998) proposed a method of using Support Vector Machines (SVMs) to classify Web pages. Dumain and Chen (2000) use text representation to organize search results into an existing hierarchical structure. Not many of these methods achieve a satisfactory performance on Web pages, in some cases due to the noise contained in the Web pages. In fact, these methods cannot handle some Web pages at all if they just contain some non-text information such as images and scripts that are unusable for these methods.

In the link-analysis-based classification techniques, learning algorithms are applied to handle both textual information of the Web pages and the hyperlink relationship among them. Slattery and Craven (2000) explored the hyperlink topology using an extended HITS algorithm. Similarly, Cohn and Hofmann (2001) and Glover et al. (2002) showed that classification performance can be improved by combining link-based and content-based techniques. Chakrabarti et al. (1998) proposed a probabilistic model to utilize both text and linkage information to classify a database of patents and a small Web collection. They showed that directly incorporating words from neighboring pages might reduce the classification performance. However, incorporating category information, such as hierarchical category prefixes, improves performance. Oh et al. (2000) reported similar results on a collection of encyclopedia articles. Getoor et al. (2001) and Lu and Getoor (2003) proposed the PRMs to combine the content feature and its relationship under a probabilistic model. Currently, most of their algorithms just deal with one kind of data object (Web page), but consider many kinds of relationships such as In-link, Out-link, Co-inlink and Co-outlink.

Our work is different from the hyperlink-based classification methods in that we can classify the heterogeneous data objects across different data types; such as Web pages, search queries and users, simultaneously by fully exploiting the interrelated relationships through an iterative process.

Furthermore, our method can be regarded as an extension of multi-database mining (Zhang et al. 2003, 2004). In Zhang et al. 2003, local pattern analysis was used to discover high-performance patterns from a multi-database. However, in our work, we focus on the learning among different types of data sources, which contains different kinds of data objects while they are interrelated with each other. Local patterns (such as category information in our problem) of data objects can be iteratively reinforced by their interrelated data objects.

2.2. *Clickthrough data analysis*

A second group of related work is clickthrough data analysis. Beeferman and Berger (2000) proposed an innovative query clustering method based on clickthrough data. In their work, they treat clickthrough data sets as a bipartite graph and identify the mapping between queries and the associated URLs. Queries with similarly clicked URLs can be clustered together. Our work is different from this work in that they deal

with the query clustering problem while we deal with the categorization problem (for both query and Web pages). Another difference is that we use the interrelationship to infer probability that a query/Web page belongs to a class, while their work only uses the interrelationship as an additional feature for clustering. Jeh and Widom (2002) proposed finding similar objects by utilizing the relationship among different types of data objects, which iteratively calculate the similarities between the objects through their relationships. Wen et al. (2001) described a query clustering method that made use of user logs. Chuang and Chien (2003) proposed a technique for categorizing Web query terms from the clickthrough data into a pre-defined subject taxonomy based on their popular search interests. Wang et al. (2003) put forward a method of using clickthrough data to iteratively reinforce the clusters of queries and Web pages.

To the best of our knowledge, our work on multi-type data objects classification on the Web is one of the first to integrate content information with interrelationships across different data types to improve the performance of classification.

3. Categorization of multi-type interrelated objects

In this section, we first define the problem of classifying multi-type interrelated objects into a general form, followed by our iterative reinforcement categorization algorithm.

3.1. Problem definition

We are given K different types of objects $X_1, X_2, \ldots, X_K$. Each type of data object X_i is described by a set of features F_i. Data objects within the same type are interrelated to each other through intra-type relationships $R_i \subseteq X_i \times X_i$. Data objects from two different types are related through inter-type relationships $R_{ij} \subseteq X_i \times X_j$. ($i \neq j$, $i = 1,2,\ldots,K$, $j = 1,2,\ldots,K$). To distinguish the features extracted from the relationships, F_i is referred to as content feature of data objects.

For a specific object $x \in X_i$, we use $x.F_i$ to represent its content features, and use $x.R_i \subseteq X_i$ and $x.R_{ij} \subseteq X_j$ to denote the objects related to it in X_i and X_j, respectively. For example, considering the Web page, the plain text contained in the Web page is defined as its content feature. The hyperlink between the Web pages is the intra-type relationship while the relationship of "clickthrough" between the Web page and the queries is the inter-type relationship.

An illustration of multiple-type data objects and their relationships is shown in Figure 2. Besides its content feature, R_i is the intra-type relationship of the data object X_i. Considering $X_i, X_1, X_2, \ldots,$ and X_K are the interrelated objects of $X_i, R_{i1}, R_{i2}, \ldots,$ and R_{ik} are the inter-type relationships between X_i and $X_i, X_1, X_2, \ldots, X_K$, respectively. For the data object X_i, there exists the training set X_{it} and testing set X_{is}. The data objects in the training set are labeled while the data objects in the testing set need to be classified.

The problem of classifying multi-type interrelated data objects is to classify each type of data object X_{is} into a set of predefined categories $C = \{c_1, c_2, \ldots, c_k\}$, where k is the number of categories.

Considering that an object $x \in X_i$ has both the content feature and relationships with other objects, we can define the feature vector of x as:

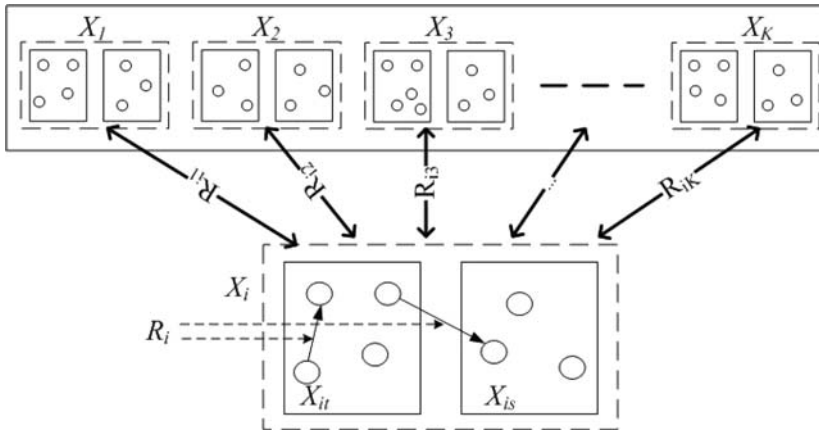

Figure 2. A graph representation for multi-type objects.

$$x = \alpha \cdot x.F + \beta \cdot x.R_{\text{intra}} + \gamma \cdot x.R_{\text{inter}} \tag{1}$$

where α, β, and γ are weights for different features with $\alpha + \beta + \gamma = 1$.

From the above definition, we can see that the different features are combined linearly. By assigning different values to α, β, and γ, we can adjust the weights of different features. For example, if $\alpha = 1$, $\beta = \gamma = 0$, we only consider the content feature and ignore the effects of intra-type features and inter-type features.

By using the additional feature, the problem of classification could be solved using the traditional classification algorithms. That is, after mapping the relationships among data objects as relationship features, each type of data objects could be classified individually. While this approach seems feasible, it would not work well when the number of objects becomes large. Furthermore, due to the sparseness of the relationships among the different types of data objects, this method cannot avoid the curse of the dimension problem.

To address these problems, we propose the IRC algorithm to iteratively exploit the relationship between the heterogeneous data objects.

3.2. Iterative reinforcement categorization algorithm

To fully utilize the relationship among the different types of data objects, we propose a novel iterative reinforcement classification method. The basic idea behind this method is to propagate the categories information computed for one type of object to all related objects by updating their probability distribution of belonging to a certain category. This process is iteratively performed until the classification results for all object types converge.

Our iterative reinforcement algorithm is described in Figure 3.

In the following, we illustrate the process with two object types: Web pages $X = \{x_1, x_2, \ldots, x_m\}$, and queries $Y = \{y_1, y_2, \ldots, y_n\}$, while the clickthrough relationships are taken as the interrelations between them. The training data set and the testing data set of data objects X are defined as X_T and X_S, while the training data set and the testing

Iterative Reinforcement Categorization Algorithm (IRC)

Input: $X_1(X_{1s}, X_{1t})$, $X_2(X_{2s}, X_{2t})$,...., and $X_n(X_{ns}, X_{nt})$.

Output: The categories of each objects in X_{1s}, X_{2s},..., and X_{ns}.

1. For each data type X_i, calculate the probability distribution of the data objects in X_{is} on the k categories based on the content feature;

2. Update the probability distribution of data objects in X_i on the k categories according to:

   (i) the probability distribution of its own content calculated in Step 1;

   (ii) the probability distribution of the interrelated objects in X_1 to X_n on the k categories;

3. Repeat Steps 3 until the probability distributions for each data object reaches a fixed point.

4. Label the object $x \in X_i$ with the category c where $c = \arg\max_{c_i}[P(c_i \mid x)]$;

Figure 3. Iterative reinforcement categorization algorithm.

data set of data objects Y are defined as Y_T and Y_S. The inter-type relationship of object $x(y)$ is defined as $x.R_Y$ ($y.R_X$). By distinguishing the relationship between the training set and testing set, the inter-type relationship of object x to the training set Y_T and Y_S are defined as $x.R_{Y_S}$ and $x.R_{Y_T}$.

Furthermore, the relationship among the different data types of objects is associated with a weight, which reflects the closeness of the relationship. For any two objects $x \in X$ and $y \in Y$, the weight of the relationship between x and y is defined as $x.R_y$.

We can use vectors $P_i = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$ and $Q_i = \{q_{i1}, q_{i2}, \ldots, q_{ik}\}$ to represent the probabilities of the testing data objects $x_i \in X_S$ and the testing data objects $y_i \in Y$ belonging to each category, respectively, where $0 \leq p_{ij}, q_{ij} \leq 1$. Thus, a probability matrix $P_{n\times k}$ ($Q_{m\times k}$) for all n objects in X (m objects in Y) can be constructed, where each entry p_{ij} (q_{ij}) is the probability that object x_i (object y_i) belongs to category j.

To complete the first step shown in Figure 3, we take the SVM classifier as the content feature based classifier. We first train the classifier on the training data set X_T and Y_T by the content features. Then, we classify the objects in X_S and Y_S into categories according to their content features.

3.2.1. Classifying objects based on their interrelated relationship. After initially classifying objects X_S and Y_S according to their contents, we can update the category information of each object in Y_S by considering the relationships with X. That is, we adjust the probability distribution of objects in Y_S belonging to categories according to the category information of their interrelated objects in X. This propagation of probability is based on the assumption that the objects of a certain category are usually related to data objects of the same category.

In general, for any object $y \in Y_S$, its interrelated data objects in our experiment may be from two different sets: the training set X_t and the testing set X_s. Since the categories of objects in the training data set X_t are known and the categories of objects in the testing data set X_s are predicted with probability, the role played by the training data and testing data on the objects should be differentiated. We denote $y.R_X^T(y.R_X^S)$ as the adjacency

matrix between the training data objects in X (testing data objects) and the data objects in Y_S.

In particular, the entry q_{ij} of the probability matrix Q, denoting the probability that object $y_i \in Y_S$ belongs to category j, is computed through the following equation:

$$q_{ij} = \lambda_1 q_{ij} + \lambda_2 \frac{\sum_{x_z \in y_i.R_{X_T}} y_i.R_{x_z} p_{zj}}{\sum_{x_z \in y_i.R_{X_T}} y_i.R_{x_z}} + \lambda_3 \frac{\sum_{x_z \in y_i.R_{X_S}} y_i.R_{x_z} p_{zj}}{\sum_{x_z \in y_i.R_{X_S}} y_i.R_{x_z}} \tag{2}$$

where λ_1, λ_2 and λ_3 are the parameters to be tuned to reflect the relative importance among its content feature, the relationship to the training data X_T and the relationship to the testing data X_S. Meanwhile, $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

We denote the matrix M_T and M_S as the row normalized matrix $R_{Y_S X_T}$ and $R_{Y_S X_S}$. Equation (2) can be re-written with a matrix form:

$$R_S = \lambda_1 R_S + \lambda_2 M_T P_T + \lambda_3 M_S P_S \tag{3}$$

where P_t and P_s are the probability distribution matrices of X_t and X_s on k categories, respectively.

Similarly, just as the objects in X can affect the objects in Y, the classification result of objects in Y can also affect the objects in X. After acquiring the categorization of objects in Y, we can re-classify objects in X_S through the relationship between objects X_S and Y. In this step, we also take the content and the categories of the associated objects into consideration. The element p_{ij} of the matrix P, denoting the probability that object $x_i \in X_S$ belongs to category j, is computed through the following equation:

$$p_{ij} = \lambda'_1 p_{ij} + \lambda'_2 \frac{\sum_{y_z \in x.R_{Y_T}} x_i.R_{y_z} q_{zj}}{\sum_{y_z \in x.R_{Y_T}} x_i.R_{y_z}} + \lambda'_3 \frac{\sum_{y_z \in x.R_{Y_T}} x_i.R_{y_z} q_{zj}}{\sum_{y_z \in x.R_{Y_T}} x_i.R_{y_n}} \tag{4}$$

where λ'_1, λ'_2 and λ'_3 are also the parameters to be tuned to reflect the relative importance among its content feature, the relationship to the training data Y_T and the relationship to the testing data Y_S. Also $\lambda'_1 + \lambda'_2 + \lambda'_3 = 1$.

We denote the matrix N_T and N_S as the row normalized matrix $R_{X_S Y_T}$ and $R_{X_S Y_S}$. Eq. (4) can be re-written in a matrix form:

$$P_S = \lambda'_1 P_S + \lambda'_2 N_T Q_T + \lambda'_3 N_S Q_S \tag{5}$$

3.2.2. Iterative reinforcement categorization (IRC). Using only the above two steps, we still do not fully utilize the interrelationships between two interrelated objects. Therefore we continue to perform an iterative reinforcement on the categorization by exploiting the relationships. Such calculation is an iterative process in which the category information is propagated from one side to the other. Let Q_S^i denote the probability matrix of interrelated objects in Y and P_S^i denote the probability distribution matrix of

objects in X after ith iteration. The algorithm can be re-written in the following matrix:

$$Q_S^{i+1} = \lambda_1 Q_S^0 + \lambda_2 M_T P_T + \lambda_3 M_S P_S^i \tag{6}$$

$$P_S^{i+1} = \lambda_1' P_S^0 + \lambda_2' N_T Q_T + \lambda_3' N_S Q_S^i \tag{7}$$

Based on the Eqs. (6) and (7), we can derive the following equation:

$$Q_S^{i+1} = \lambda_1 Q_S^0 + \lambda_2 M_T P_T + \lambda_1' \lambda_3 M_S P_S^0 + \lambda_2' \lambda_3 M_S N_T Q_T + \lambda_3' \lambda_3 M_S N_S Q_S^i \tag{8}$$

Taken $\lambda_1 Q_S^0 + \lambda_2 M_T P_T + \lambda_1' \lambda_3 M_S P_S^0 + \lambda_2' \lambda_3 M_S N_T Q_T$ as Q, the formula can be written in the following form:

$$Q_S^{i+1} = Q + \omega M_S N_S Q_S^i \tag{9}$$

where ω is equal to $\lambda_3' \lambda_3$.

The equation implies that the probability matrix of objects in Y is affected by the probability calculated on the content of objects in Y and the relationship of the adjacent matrix.

Similar to Eq. (7), the computation of the probability matrix for objects in X is derived as follows:

$$P_S^{i+1} = \lambda_1' P_S^0 + \lambda_2' N_T Q_T + \lambda_3' \lambda_1 N_S P_S^0 + \lambda_2' \lambda_3 N_S M_T P_T + \lambda_3' \lambda_3 N_S M_S P_S^i \tag{10}$$

By denoting $\lambda_1' P_S^0 + \lambda_2' N_T Q_T + \lambda_3' \lambda_1 N_S P_S^0 + \lambda_2' \lambda_3 N_S M_T P_T$ as P, the formula can be written in the following form:

$$P_S^{i+1} = P + \omega N_S M_S P_S^i \tag{11}$$

After several iterations, P_S and Q_S would not change any more. Then, the category of x is taken to be $\arg\max_{c_i}[P(c_i|x)]$, while the category of y is $\arg\max_{c_i}[P(c_i|y)]$.

The above computation is performed iteratively until the change of probability distribution of each object among different categories is trivial. In this paper, we use the Euclidean length of the residual vector $\left\| P_S^{i+1} - P_S^i \right\|$ to measure the change. That is when $\left\| P_S^{i+1} - P_S^i \right\|$ becomes less than a predefined δ, the iteration process stops.

3.3. Convergence of IRC

In this section, we give proof of the convergence property of our proposed algorithm.

Lemma 3.1. *M_S and N_S are nonnegative matrices and sum of each row is equal to 1.*

Proof: It is a direct induction from the definition of M_S and N_S.

Lemma 3.2. *$M_S N_S$ is a Markov matrix and therefore $(M_S N_S)^i$ is also a Markov matrix for any integer i.*

Proof: $M_S N_S$ is a nonnegative square matrix and it is easy to verify that the entries in each row of $M_S N_S$ sum to 1. So it is a Markov matrix. Also from the Markov Chain theory (Grimmett and Stirzaker, 1992), $(M_S N_S)^i$ is also a Markov matrix for any integer i.

Theorem 3.1. *The IRC algorithm converges to a fixed point.*

Proof: Without loss of generality, we only prove that the matrix P_S^i converges to a fixed point.

M_S and N_S are Markov matrices (Lemma 1), and $\omega(0 < \omega < 1)$

From Eq. (11),

$$\begin{aligned} P_S^{i+1} &= P + \omega M_S N_S P_S^i \\ &= P + \omega M_S N_S P + (\omega M_S N_S)^2 P + \cdots + (\omega M_S N_S)^i P + (\omega M_S N_S)^{i+1} \end{aligned}$$

Now we see the convergence as following:

$$\begin{aligned} l \lim_{i\to\infty} ||P_S^{i+1} - P_S^i|| &= \lim_{i\to\infty} ||(\omega M_S N_S)^i (P + \omega M_S N_S - E)|| \\ &= \lim_{i\to\infty} ||\omega^i (M_S N_S)^i (P + \omega M_S N_S - E)|| \end{aligned}$$

Since,

$$\lim_{i\to\infty} \omega^i = 0 \quad (0 < \omega < 1)$$

$(M_S N_S)^i$ is a Markov matrix too (Lemma 2).

So $(M_S N_S)^i$ is nonnegative and sum of each row is equal to 1.

Thus,

$$\lim_{i\to\infty} ||(\omega M_S N_S)^i|| = \lim_{i\to\infty} ||\omega^i (M_S N_S)^i|| = 0$$

Finally,

$$\lim_{i\to\infty} ||P_S^{i+1} - P_S^i|| = 0$$

It is obvious that P_S^i converges to its fixed point eventually.

Analogously, Q_S^i converges to its fixed point eventually.

3.4. Complexity of IRC

We now consider the time and space complexity of IRC. The space required is simply $O(k(m+n))$ to store the results of the probability matrices P_S and Q_S, where m and n denote the number of objects in X and Y, respectively. Let T be the iteration number. In our experiments, IRC usually converges within five iterations. Let d_1 be the average number of the objects in X that is associated with the objects in Y and d_2 be the average number of the objects in Y that is associated with the objects in X. The time for computing the probability matrices of objects in X and Y is $O(Tkmd_1)$ and $O(Tknd_2)$, respectively, where k is the number of categories. Thus, the overall time required is $O(Tkmd_1+Tknd_2)$. Since the relationship between the data objects is very sparse, d_1 and d_2 is very small on average.

4. Experiments

In order to validate IRC, we conducted the experiments on the ODP dataset with MSN clickthrough data. In the data set, there are two kinds of data objects: Web pages and queries. There exist clickthrough relationships between two kinds of data objects.

4.1. Data set

To evaluate the performance of our algorithm, experiments were performed using a set of classified Web pages extracted from the Open Directory Project (ODP) (http://dmoz.org/). ODP contains about 1.2 million Web pages, in which each Web page is classified by human experts into 17 top level categories (*Arts, Business and Economy, Computers and Internet, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports, Adult and World*). Because the Web pages in the regional category are also included in other categories and the Web pages in the category of the world are not written in English, these two categories are removed in our experiments. Accordingly, 15 categories in all are used in the experiments.

Real query clickthrough data is collected from MSN as our experiment data set. The clickthrough data contain about 1.2 million query requests recorded over 12 hours in August 2003. The data we obtained had been already processed into a predefined format; i.e. each query request is associated with one or more clicked Web pages, forming a "query session", which can be defined as follows:

$$\text{Query Session} := \text{query text}[\text{clicked Web page}^*]$$

Some preprocessing steps are applied to queries and Web pages in the raw log. All queries are converted into lower-case, and are stemmed using the Porter algorithm. The stop words are removed too. The query sessions sharing the same query and the same URL are merged into one query session, with the frequencies summed up.

Since we only have 12 hours of clickthrough data on hand, some of Web pages in the ODP data set are not in our query clickthrough log. Hence, in our experiment, we only deal with the common pages which appeared in both the ODP data set and the clickthrough data. Finally, we got 131,788 Web pages in 15 top-level categories, 199,564

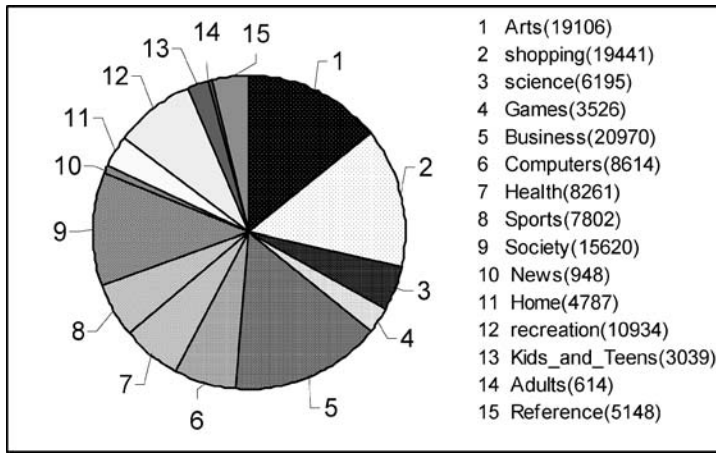

Figure 4. Distribution of Web pages in the 15 categories.

associated queries and 468,696 query sessions. Figure 4 shows the distribution of the number of Web pages in 15 categories.

We test the relevancy of queries to the contents of Web pages from the users' perspective. We randomly select three subsets which contain 600 query sessions in total. Ten volunteer graduate students are invited as our evaluation subjects. They are asked to evaluate whether the queries are relevant to the Web pages according to the content of the pages. The results are shown in Table 1. Figure rom the table it is easy to find that about 81.7% of queries on average are relevant to the contents of the Web pages.

Before the experiment, we conducted another statistical test to see whether query terms can introduce extra information for Web categorization. Several examples of Query session in the raw data are shown in Table 2. From the examples we may come to a conclusion intuitively that the pages linked by the same query are likely to belong to the same category. Statistically 68.4% Web pages fall into the same ODP category when they are clicked by the same query.

4.2. Evaluation criteria

The performance of the proposed methods was evaluated using the conventional precision, recall and F_1 measures. Precision p is defined as the proportion of correctly classified examples in the set of all examples assigned to the target class. Recall r is

Table 1. Relevance between the queries and Web pages.

Subset	Session	Relevant	Ratio
1	300	247	0.82
2	300	262	0.87
3	300	228	0.76
Average			**0.817**

Table 2. A sample of raw MSN clickthrough data.

Query	URL	Category
Software bugs	http://www.cpsr.org/program/y2k	Computer
	http://www.bugnet.com	Computer
Softball hitting	http://www.hitranger.com	Shopping
	http://www.decatursports.com/softball_drills_page.htm	Sports
Playing mantis	http://www.playingmantis.com	Games
	http://www.johnnylightning.com	Recreation
Sniper rifles	http://www.sniperworld.com	Sports
	http://www.snipercentral.com	Sports
	http://www.norcalprecision.com	Sports

defined as the proportion of the correctly classified examples out of all the examples having the target class. F_1 is a combination of precision and recall defined as follows:

$$F_1 = \frac{2pr}{p+r} \tag{12}$$

Furthermore, micro-averaging and macro-averaging (Yang and Pedersen, 1997) were applied to get single performance values over all classification tasks.

4.3. Baseline

As mentioned above, we take two kinds of objects: queries and Web pages as the interrelated data objects in our experiments. Since we do not have the training set and the category information for the queries, we just evaluate the performance of our proposed algorithm on the Web pages.

The *content-based classification method* for Web pages is taken as the baseline. Since we assume that most queries are relevant to the topics of the corresponding Web pages, we can take the interrelated queries as an additional feature for their corresponding pages. Web page d_i is clicked by users through queries $q_1, q_2, \ldots, q_m$ with different frequencies. Thus $M_{i1} \cdot q_1 + M_{i2} \cdot q_2 + \cdots + M_{im} \cdot q_m$ can be taken as additional metadata for Web page d_i where M_{ik} means the frequency that users click on d_i following query q_k. We consider two methods that utilize the query metadata. First, we can use the query metadata directly as additional features of a Web page. We denote this method as *query-metadata* based classification. Second, we can integrate the query metadata and the content of the Web page together and regard them as a *virtual document* of the Web page. To test the relative importance of the query metadata, we try different weights of the metadata and integrate them with the content of the Web pages. For example, if we set "content: metadata" as 1:2, the query metadata is twice as important as the content of the Web page. After removing stop words and feature selection, the dimension of the Vector Space for the content of collection, the query metadata of collection and the virtual document collection are 258,669, 46,002, and 281,259 respectively.

To speed up the classification, a simple feature selection method, known as "Information Gain (*IG*)" (Yang and Pedersen, 1997), is applied in our experiments.

SVM is a promising classification algorithm developed by Cortes and Vapnik (1995) and Joachims (1998). In this paper, we use linear SVM because of its high accuracy when used for text categorization. The SVM^{light} software package is used (http://research.micro-soft.com/~jplatt/smo.html). In all experiments, the trade-off parameter C is set to 1. The widely used one-against-all approach is used for the multi-class case. In the rest of this paper, we only consider the probability of the data objects that belong to the categories, so we utilize the method in Platt (1999) to assign each category a probability.

4.4. Performance

We fixed several parameters for the rest of the experiments: the ratio between the content of Web pages and the query metadata is set as 1: 2 when constructing the virtual documents; λ_1, λ_2, and λ_3 in Eq. (2) are set as 0, 0.6 and 0.4; λ'_1, λ'_2 and λ'_3 in Eq.(4) are set as 0.7, 0, and 0.3; and iteration times equals 5. These parameters are determined based on an experiment conducted on a validation dataset.

We start by analyzing how each source of information such as content, query metadata and virtual document performs without using the iterative algorithm. Table 3 shows the micro-averaged F_1 values for the different classifiers. The highest values for each classifier are shown in bold face. The content-based classifiers, as expected, showed poor results, indicating that the text of the Web pages does not provide sufficient information to reliably classify the Web pages. Since queries are relevant to the topics of Web pages, they can be used to improve the performance.

From Table 3, we also find that IRC, in comparison, achieves a higher performance than the other three methods. Relatively speaking, IRC improved 26.4% over the content method, 21% over the query metadata method, and 16.4% over the virtual document method under the F_1-micro-averageing measure. The reason for the improvement lies in the fact that our algorithm can fully exploit the relationship between the Web pages and the queries.

We conducted a further experiment to show the effect of the clickthrough data by increasing the clickthrough data size as displayed in Figure 5. We randomly selected 10% of the clickthrough data the first time and 20% the second time and so on. We found that the performance of the content-based categorization is quite poor. When clickthrough data is introduced, the performance of the Web-page categorization steadily improves.

Table 3. Performance of the four algorithms.

	MICRO			MACRO		
	Precision	Recall	F1	Precision	Recall	F1
Content	0.561	0.561	0.561	0.642	0.470	0.496
Query metadata	0.586	0.586	0.586	0.523	0.554	0.523
Virtual document	0.609	0.609	0.609	0.575	0.583	0.568
IRC	**0.709**	**0.709**	**0.709**	**0.671**	**0.664**	**0.68**

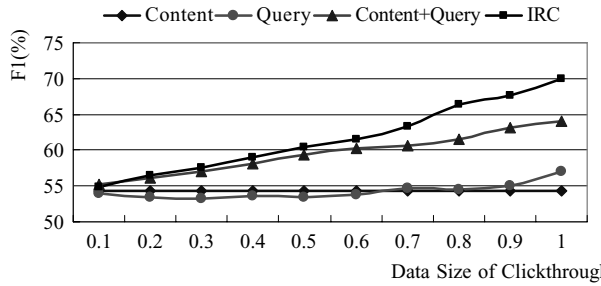

Figure 5. Performance on the different clickthrough data size.

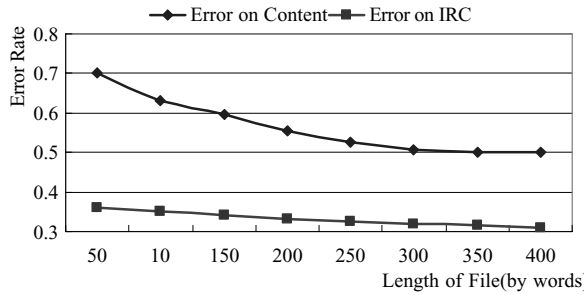

Figure 6. Error rate on different file length.

Since the content-based classification methods rely heavily on the content of Web pages, they may perform badly when the length of Web pages is short. In order to verify such an intuition, two experiments are conducted and the results are shown in Figure 6 and Table 4.

When processing the data, we find many pages that contain too few words to indicate their main topics though they are meaningful with plentiful non-text resources such as pictures and videos. Thus it is hard to identify their labels based only on such words. However, these pages may still be retrieved and clicked on by users when they are relevant to the query given by them. In these cases, the query logs may be especially effective to predict the labels of these pages. In Figure 6, we can see the error rate of the content-based categorization gradually increasing on a large scale with the shorter length of pages while our IRC show a very small change in error rate. We also perform statistics on a testing set with 13173 Web pages using the content-based categorization. The results are shown in Table 4. The average length of correctly classified Web pages is larger than that of wrongly classified Web pages.

Figure 7 shows the performance of the IRC algorithm with the iteration times. The implicit relationships are exploited more and more thoroughly with the increase of the

Table 4. Effect of the length of file on classification.

	Wrongly classified	Correctly classified	Total
Number of files	6060	7113	13173
Average length of files	180.85	375.28	285.84

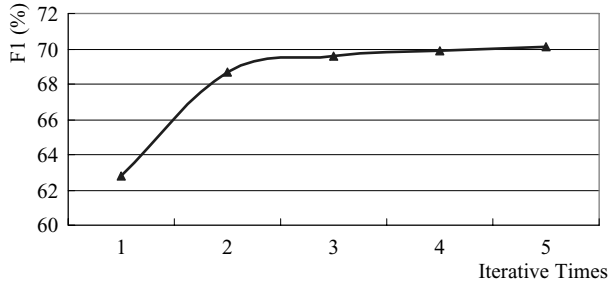

Figure 7. Performance on different iteration counts.

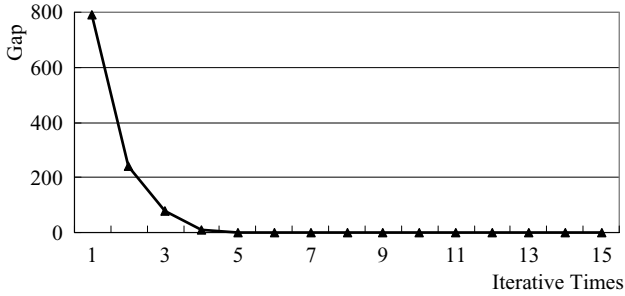

Figure 8. Convergence of the iterative algorithm.

iterations and these relationships contribute to the improvement of the classification performance.

We can also find that the performance of IRC converges quickly as shown in Figure 7. In the following, we give a more detailed experiment to show IRC's convergence property.

The convergence curve of our iterative algorithm is shown in Figure 8. The gap $\left\| P_S^{i+1} - P_S^i \right\|$ denotes the difference between the current iteration and the previous iteration. Figure 8 shows the IRC algorithm converges within five iterations.

We run the IRC algorithm on the Pentium 1.9G PC with 512M of memory. Figure 9 shows the execution time of the algorithm on different data size, where the CPU time is linear with the size of the clickthrough data, which shows that the algorithm scales up well with large data.

4.5. Parameters selection

As mentioned earlier, several parameters are used in the experiments, such as the ratio between the content of Web pages and query metadata when constructing the virtual documents, the ratio of the categories weight between the training data and testing data when classifying queries, the ratio of the categories weight between content features and queries when classifying Web pages and the iteration count of the our iterative algorithm. We conducted the experiments on a validation dataset to determine these parameters.

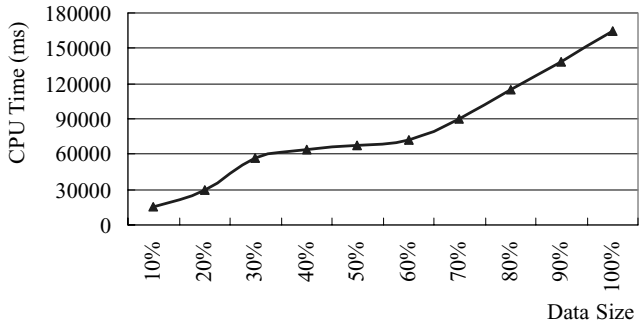

Figure 9. Execution time on different data size.

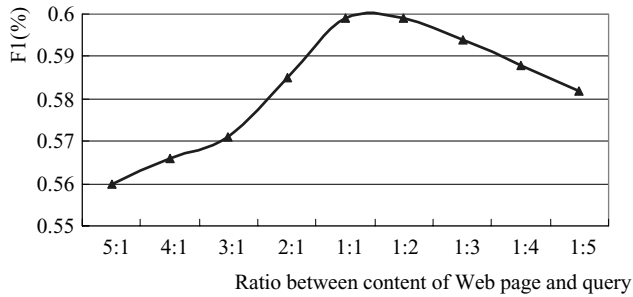

Figure 10. Performance on weight between content and query.

The first experiment shows how to combine the content of the Web page and the query metadata to get the highest performance on the F_1-measure when constructing the virtual document for classification. As shown in Figure 10, we vary the ratio between the content of the Web page and query metadata. We find that classification based on the virtual document can achieve best performance when the ratio of the weight between the content and the query metadata is 1:2.

When classifying the queries, we measure the weight of the category that is set by the training data and the weight of the category that is calculated by the testing data in Eq. (4). Since we do not have the training set of the query, λ_1 is set to 0. We vary the parameter of λ_2 (the weight for the training data) and λ_3 (the weight for the testing data) from 0 to 1. Since $\lambda_2 + \lambda_3 = 1$, we only change the λ_2 in our experiment. The experimental results on F_1-measure are shown in Figure 11. We find that the F_1 value on classifying the Web pages has improved while the weight for the category of the training data has changed from 0 to 0.6. The system achieves the best precision when $\lambda_2 = 0.6$ and $\lambda_3 = 0.4$. If we continue to increase the weight of the training data category, the precision would drop down since the effect of the relationship exploited between queries and testing Web pages is reduced incrementally.

In order to measure the weight of the probability which is calculated by the pure content feature and the weight of the probability which is calculated by queries in Eq. (6), we tune the parameter of λ'_1 (the weight for the content) and λ'_3 (the weight for the query) from 0 to 1 in the SVM model. Here λ'_2 is set to 0 because we do not have the

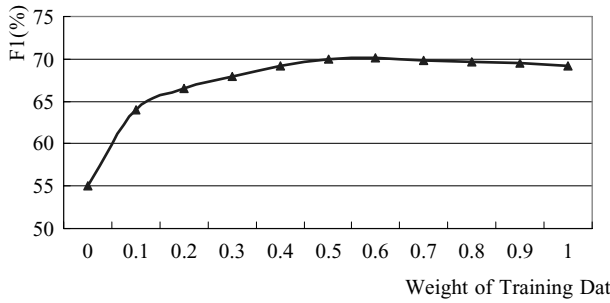

Figure 11. Performance on the weight of the training data.

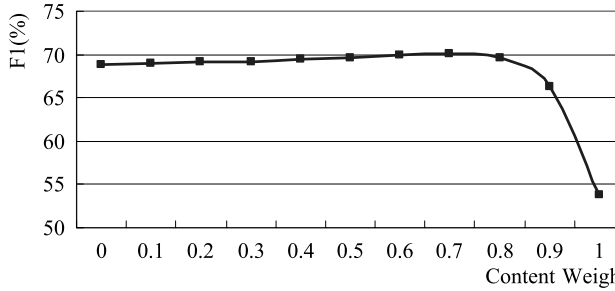

Figure 12. Performance on the content weight.

training set for the query. Since $\lambda_1' + \lambda_3' = 1$, we only change the λ_1' in our experiment. The experimental results on F_1-measure are shown in Figure 12. We found that the F_1 value is improved when introducing content. The system achieves the best precision when $\lambda_1' = 0.7$ and $\lambda_3' = 0.3$. If we continue to introduce more content information, the precision drops as a result of the poor quality in the content classification.

5. Conclusions and future work

In this paper, we proposed an iterative reinforcement classification algorithm to categorize Web pages, which have a relationship to the queries. The proposed algorithm considers both the content feature of data objects and the relationship across the different types of data objects in the iterative classification process. The intermediate classification results of each type are used to update the classification results of their related data objects which may be different types. Such a reinforcement process can exploit the knowledge hidden in the interrelationships effectively. Experiments on the ODP dataset with MSN clickthrough log data show that IRC can significantly improve the Web page classification under the F_1 measure after iteratively exploiting the relationships between the Web pages and queries.

Although IRC can be applied on multiple data sources with multiple interrelationships, in this work, we only consider two types of data objects and the clickthrough relationships during the classification process. In the future, it would be interesting to

apply the algorithm to effectively integrate multi-source data objects and their inter-type relationships to improve the classification performance.

Acknowledgement

Gui-Rong Xue, Hua-Jun Zeng and Zheng Chen are supported by Microsoft Research Asia. Yong Yu is supported by a grant from National Natural Science Foundation of China (NO.60473122). Dou Shen and Qiang Yang are supported by a grant from the Hong Kong Government (RGC central allocation grant CA03/04.EG01).

References

Beeferman, D. and Berger, A. 2000. Agglomerative clustering of a search engine query log. In Proc. of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston: ACM Press, pp. 407–415.

Chakrabarti, S., Dom, B., and Indyk, P. 1998. Enhanced hypertext categorization using hyperlinks. In Proc. of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington: ACM Press, pp. 307–318.

Cortes, C. and Vapnik, V. 1995. Support vector networks. Machine Learning, 20, (3), 273–297, 20:1–25. Springer.

Chuang, S.L. and Chien, L.F. 2003. Enriching Web taxonomies through subject categorization of query terms from search engine logs. Decision Support System, Vol. 35, No. 1, Elsevier Science Publishers, pp. 113–127.

Cui, H., Wen, J.R., Nie, J.Y., and Ma, W.Y. 2003. Query expansion by mining user Logs. IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No. 4, IEEE Computer Society, pp. 829–839.

Cohn, D. and Hofmann, T. 2001. The missing link – A probabilistic model of document content and hypertext connectivity. In Advances in Neural Information Processing Systems 13, Vancouver, Canada: MIT Press, pp. 430–436.

Dumain, S. and Chen, H. 2000. Hierarchical classification of web content. In Proc. of the 23rd annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece: ACM Press, New York: ACM Press pp. 256–263.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. 2001. Learning probabilistic models of relational structure. In Proc. of the 18th International Conference on Machine Learning, Williamstown, MA: Morgan Kaufmann, pp. 170–177.

Getoor, L., Segal, E., Taskar, B., and Koller, D. 2001. Probabilistic models of text and link structure for hypertext classification. In IJCAI Workshop on "Text Learning: Beyond Supervision", Seattle, WA, August.

Glover, E.J., Tsioutsiouliklis Lawrence, K.S., Pennock, D.M., and Flake, G.W. 2002. Using web structure for classifying and describing web pages. In Proc. of the International Conference on the World Wide Web, Hawaii: ACM Press, pp. 562–569.

Grimmett, G. and Stirzaker, D. 1992. Probability and Random Processes, 2nd edition. Oxford, England: Oxford University Press.

Huang, C.K., Chien, L.F., and Oyang, Y.J. 2003. Relevant term suggestion in interactive web search based on contextual information in query session logs. In Journal of the American Society for Information Science and Technology, Vol. 54, No. 7, John Wiley & Sons, Inc., pp. 638–649.

Jeh, G. and Widom, J. 2002. SimRank: A measure of structural-context similarity. In Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada. ACM Press, pp. 538–543.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In Proc. of 10th European Conference on Machine Learning, Chemnitz, Germany: Springer, pp. 137–142.

Lu, Q. and Getoor, L. 2003. Link-based classification. In Proc. of 20th International Conference on Machine Learning, Washington, DC: AAAI Press, pp. 496–503.

Oh, H.J., Myaeng, S.H., and Lee, M.H. 2000. A practical hypertext categorization method using links and incrementally available class information. In Proc. of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece: ACM Press, pp. 264–271.

Platt, J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, pp. 61–74.

Sequential Minimal Optimization, http://research. micro-soft.com/∼jplatt/smo.html.

Slattery, S. and Craven, M. 2000. Discovering test set regularities in relational domains. In Proc. of 17th International Conference on Machine Learning, Stanford, US: Morgan Kaufmann, pp. 895–902.

Wang, J.D., Zeng, H.J., Chen, Z., Lu, H.J., Tao, L., and Ma, W.-Y. 2003. ReCoM: Reinforcement clustering of multi-type interrelated data objects. In Proc. of the ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, CA: ACM Press, pp. 274–281.

Wen, J.R., Nie, J.Y., and Zhang, H.J. 2001. Clustering user queries of a search engine. In Proc. of the Tenth International World Wide Web Conference, Hong Kong: ACM Press, pp. 162–168.

Yang, Y. and Pedersen, J.O. 1997. A comparative study on feature selection in text categorization. In Proc. of the Fourteenth International Conference of Machine Learning, Nashville, Tennessee: Morgan Kaufmann, pp. 412–420.

Zhang, S.C., Wu, X.D., and Zhang, C.Q. 2003. Multi-database mining. IEEE Computational Intelligence Bulletin, Vol. 2 No. 1, IEEE Computer Society, pp. 5–13.

Zhang, S.C., Zhang, C.Q., and Wu, X.D. 2004. Knowledge Discovery in Multiple Databases, Springer.