Link-Based Text Classification Using Bayesian Networks

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete,
Andrés R. Masegosa, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación,
CITIC-UGR, Universidad de Granada
18071 – Granada, Spain
{lci,jmfluna,jhg,andrew,aeromero}@decsai.ugr.es

Abstract. In this paper we propose a new methodology for link-based document classification based on probabilistic classifiers and Bayesian networks. We also report the results obtained of its application to the XML Document Mining Track of INEX'09.

1 Introduction

This is the third year that researchers from the University of Granada (specifically from the Uncertainty Treatment in Artificial Intelligence research group) participate on the XML Document Mining Track of the INEX workshop. As in previous editions, we restrict our solutions to the application of probabilistic methods to these problems. To be more precise, we are looking to solve the problem of link-based document classification within the field of Bayesian networks [10] (a special case of probabilistic graphical models).

This year, the proposed problem is rather similar to the one considered in the previous edition of the workshop [6]. A training corpus, composed of labeled XML files is provided, and an unlabeled test corpus is left to the participants, in order to be estimated its labeling. Also, a link file is given, which contains specific relations between pairs of documents (either in the training or the test corpus). Thus, the problem can be seen as a graph labeling problem, where each node has textual (XML) content.

The main difference between this INEX track in 2008 and 2009 is the fact that the corpus is composed of multilabeled documents, that is to say, a document can belong to one or more categories. The rest of the rules are esentially the same, although the document collection and the set of categories are also different.

As we did in the past, we can assume that the XML markup (the "internal structure" of the collection) is not very helpful for categorization. In fact, we did not find it very useful for the task in previous editions [4] (by making several transformations from XML to flat text documents). Moreover, the organizers have provided an indexed file of term vectors representing the documents, where XML marks have been removed.

S. Geva, J. Kamps, and A. Trotman (Eds.): INEX 2009, LNCS 6203, pp. 397–406, 2010.

Like our previous participation [5], we will use explicitly the "external structure" of the collection, i.e. the link file (the graph of documents). There, we provided a "graphical proof" that the category of the documents linked by one tends to be similar to the category of the own document. Several experiments in the same direction showed us the same fact for the 2009 corpus, although we do not reproduce them here. Apart from those experiments, the names of the categories (which are explicitly given in the training set), tend to show categories which are probably coming from a hierarchy (for example `Portal:Religion`, `Portal:Christianity` and `Portal:Catholicism`). The two known facts about the relations are summarized here:

- In this linked corpus, due to its nature, a "hyperlink regularity" is supposed to arise (more precisely an encyclopedia regularity, see [15] for more details).
- There are some categories strongly related a priori, because the probable existence of a (unknown) hierarchy.

Although last year we proposed a method that captures some "fixed" relations among categories, given this different problem setting (multilabel) and its higher dimensionality, this year we pretend to learn those relations automatically from data, leading to a more flexible approach.

2 Base Classifiers

Two base classifiers will be used to label the graph nodes based only on their content. They will serve as the baseline, and next will be combined with the Bayesian network learnt from data with our new methodology. We will briefly describe them, in order to make the paper more self-contained.

Both classifiers are probabilistic, i.e. given a document d_j, they compute the probability values $p(c_i|d_j)$ for each category c_i, and assign them as a degree of confidence in that each c_i is an appropriate label for d_j. The advantage of probability is that it is a very well founded approach, and several different probabilistic approaches can be combined together, because they are dealing with the same measures.

Note that here we deal with a multilabel problem by defining a binary classifier for each label, following the "classical" approach to this task [12].

2.1 Multinomial Naive Bayes

The model is the same used by McCallum et al. [9], adapting it to the case of many binary problems. The naive Bayes, in its multinomial version, is a very fast and well performing method. In this model, we firstly assume that the length of the document is independent of the category. We also assume that the term occurrences are independent on each other, given the category (this is the core of the naive Bayes method).

In the multinomial version of this classifier, we see a document d_j as being drawn from a multinomial distribution of words with as many independent trials as the length $|d_j|$ of d_j.

So, given a category c_i, we express the probability[1] $p_i(c_i|d_j)$ as

$$p_i(c_i|d_j) = \frac{p_i(d_j|c_i)\, p_i(c_i)}{p_i(d_j)}. \tag{1}$$

We can rewrite $p_i(d_j)$ using the law of total probability,

$$p_i(d_j) = p_i(d_j|c_i)\, p_i(c_i) + p_i(d_j|\overline{c}_i)\, \big(1 - p_i(c_i)\big). \tag{2}$$

The values $p_i(c_i|d_j)$ can be easily computed in terms of the prior probability $p_i(c_i)$ and the probabilities $p_i(d_j|c_i)$ and $p_i(d_j|\overline{c}_i)$.

Besides, prior probabilities are estimated from document counts:

$$\widehat{p}_i(c_i) = \frac{N_{i,doc}}{N_{doc}} \tag{3}$$

where N_{doc} is the number of documents in the training set and $N_{i,doc}$ is the number of documents in the training set which belong to category c_i.

On the other hand, we can estimate $p_i(d_j|c_i)$ and $p_i(d_j|\overline{c}_i)$ as follows (as a multinomial distribution over the words):

$$p_i(d_j|c_i) = p_i(|d_j|)\, \frac{|d_j|!}{\prod\limits_{t_k \in d_j} n_{jk}!} \prod_{t_k \in d_j} p_i(t_k|c_i)^{n_{jk}},$$

and

$$p_i(d_j|\overline{c}_i) = p_i(|d_j|)\, \frac{|d_j|!}{\prod\limits_{t_k \in d_j} n_{jk}!} \prod_{t_k \in d_j} p_i(t_k|\overline{c}_i)^{n_{jk}},$$

where n_{jk} is the frequency of the term t_k in the document d_j.

Substituting and simplifying in equations 1 and 2 we obtain:

$$p_i(c_i|d_j) = \frac{p_i(c_i) \prod\limits_{t_k \in d_j} p_i(t_k|c_i)^{n_{jk}}}{p_i(c_i) \prod\limits_{t_k \in d_j} p_i(t_k|c_i)^{n_{jk}} \; + \; \big(1 - p_i(c_i)\big) \prod\limits_{t_k \in d_j} p_i(t_k|\overline{c}_i)^{n_{jk}}}.$$

Finally, individual term probabilities $p_i(t_k|c_i)$ and $p_i(t_k|\overline{c}_i)$ are computed by means of the following formulae (using Laplace smoothing):

$$\widehat{p}_i(t_k|c_i) = \frac{N_{ik} + 1}{N_{i\bullet} + M}, \quad \widehat{p}_i(t_k|\overline{c}_i) = \frac{N_{\bullet k} - N_{ik} + 1}{N - N_{i\bullet} + M}, \tag{4}$$

where N_{ik} is the number of times that the term t_k appears in documents of class c_i, $N_{i\bullet}$ is the total number of words in documents of class c_i ($N_{i\bullet} = \sum_{t_k} N_{ik}$), $N_{\bullet k}$ is the number of times that the term t_k appears in the training documents ($N_{\bullet k} = \sum_{c_i} N_{ik}$), N is the total number of words in the training documents, and M is the size of the vocabulary (the number of distinct words in the documents of the training set).

[1] With the notation $p_i(c_i|d_j)$ we are emphasizing that the probability distribution is computed over a binary variable C_i, taking values in $\{c_i, \overline{c_i}\}$. So, we have a different probability distribution over each category.

2.2 Bayesian OR Gate

The Bayesian OR gate classifier was presented in the INEX 2007 workshop by this group [4]. This classifier relies on the assumption that the relationships among the terms and each category follow a so-called *noisy-OR gate* probability distribution. Following the Bayesian networks notation, this model can be graphically represented as a graph having one node for the category C_i (binary variable C_i, ranging in $\{c_i, \overline{c}_i\}$), one node for each term T_k (binary variable T_k, with values in $\{t_k, \overline{t}_k\}$), and arcs going from each term node to the category nodes they appear in (i.e. they form the parent set, $Pa(C_i)$, of the category node C_i).

In the naive Bayes model (a generative one), we are defining $p(d_j|c_i)$, whereas in the Bayesian OR gate (a discriminative model), we are computing directly $p_i(c_i|d_j)$. Instead of using a "general" probability distribution, $p_i(c_i|d_j)$ is modeled by means of a "canonical model" [10], the noisy OR gate, which makes computations and parameter storage feasible tasks.

We can define the probability distribution for this noisy OR gate in the following way:

$$p_i\big(c_i|\, pa(C_i)\big) \;=\; 1 \;-\; \prod_{T_k \in R(pa(C_i))} \big(1 - w(T_k, C_i)\big)$$

$$p_i\big(\overline{c}_i|\, pa(C_i)\big) \;=\; 1 \;-\; p_i\big(c_i|\, pa(C_i)\big),$$

where $R(pa(C_i)) = \{T_k \in Pa(C_i) \,|\, t_k \in pa(C_i)\}$, i.e. $R(pa(C_i))$ is the subset of parents of C_i which are instantiated to its t_k value in the configuration $pa(C_i)$. $w(T_k, C_i)$ is a weight representing the probability that the occurrence of the "cause" T_k alone (T_k being instantiated to t_k and all the other parents T_h instantiated to $\overline{t}_h$) makes the "effect" true (i.e., forces class c_i to occur).

Then, given a certain document d_j, we can compute the posterior probability $p_i(c_i|d_j)$ by instantiating to the value t_k all the terms that appear in the document (i.e. $p_i(t_k|d_j) = 1$), and to the value $\overline{t}_h$ those terms that do not appear in d_j (i.e. $p_i(t_h|d_j) = 0$). The result is [3]:

$$\begin{aligned} p_i(c_i|d_j) &= 1 - \prod_{T_k \in Pa(C_i)} \big(1 - w(T_k, C_i)\, p_i(t_k|d_j)\big) \\ &= 1 - \prod_{T_k \in Pa(C_i) \cap d_j} \big(1 - w(T_k, C_i)\big)\,. \end{aligned}$$

Finally, we have to give a definition for the weights $w(T_k, C_i)$, which is almost the same appearing in [4]:

$$w(T_k, C_i) = \frac{N_{ik}}{nt_i\, N_{\bullet k}} \prod_{h \neq k} \frac{(N_{i\bullet} - N_{ih})N}{(N - N_{\bullet h})N_{i\bullet}}. \tag{5}$$

In this formula, $N_{ik}, N_{\bullet k}, N_{i\bullet}$ and N mean the same than in previous definitions made in the explanation of the multinomial naive Bayes, and nt_i is the number of

different terms occurring in documents of the class c_i. The factor nt_i is introduced here to relax the independence assumption among terms, but some other valid definitions for the weights (which do not use this factor) can be found in [3] and [4].

Finally, in order to make the probabilities independent on the length of the document (thus making the scores of different documents comparable), we introduce the following normalization, which is somewhat similar to the *RCut* thresholding strategy [14], and we return as the final probability $p(c_i|d_j)$:

$$p(c_i|d_j) = \frac{p_i(c_i|d_j)}{\max_{c_k}\{p_k(c_k|d_j)\}}$$

Some experiments [3] have shown that the Bayesian OR gate classifier tends to outperform the multinomial naive Bayes classifier, although the number of parameters needed and the complexity are essentially the same.

3 The Bayesian Network Model

This section describes a new methodology that models a link-based categorization environment using Bayesian networks. We shall build automatically from data a Bayesian network-based model, representing the relationships among the categories of a certain document and the categories present on the related (linked) documents. In this development, we will only use data from incoming links, because we carried several experiments on the corpus, and found them much more informative than outgoing ones. Anyway, information from outgoing links (or even considering undirected links) could also be used in this model.

3.1 Modeling Link Structure between Documents

In this problem, we will consider two binary variables for every category i: one is C_i (with states $\{c_i, \overline{c}_i\}$) which models the probability of a document being (or not) of class c_i, and the variable LC_i (with states $\{lc_i, \overline{lc_i}\}$), which represents if there is a link, or not, from documents of category c_i to the current document[2]. We assume there is a global probability distribution among all these variables, and we will model it with a Bayesian network.

To learn a model from the data, we will use the training documents, each one as an instance whose categories (values for variables C_i) are perfectly known, and the links from other documents. If a document is linked by another training document of category j, we will set $LC_j = lc_j$, setting it to $\overline{lc_j}$ otherwise. Note that a training document could be linked by test documents (whose categories are unknown). In that case, this evidence is ignored, and for the categories j which do not have any document linked to the current document, their variables are set to $\overline{lc_j}$.

[2] As we stated before, we also could represent the existence of outgoing links to a document of category c_i, or both types of interactions.

So, we could learn a Bayesian network from training data (see next section) and, for each test document d_j, we could compute $p(c_i|e_j)$, where e_j represents all the evidence given by the information of documents that link this.

Thus, the question is the following: for a certain document d_j, given $p(c_i|d_j)$ and $p(c_i|e_j)$, how could we combine them in an easy way? We want to compute the posterior probability $p(c_i|d_j, e_j)$, the probability of a category given the terms composing the document and the evidence due to link information.

Using Bayes' rule, and assuming that the content and the link information are independent given the category, we get:

$$\begin{aligned} p(c_i|d_j,e_j) &= \frac{p(d_j,e_j|c_i)\,p(c_i)}{p(d_j,e_j)} = \frac{p(d_j|c_i)\,p(e_j|c_i)\,p(c_i)}{p(d_j,e_j)} \\ &= \frac{p(c_i|d_j)\,p(d_j)\,p(e_j|c_i)\,p(c_i)}{p(c_i)\,p(d_j,e_j)} = \frac{p(c_i|d_j)\,p(d_j)\,p(c_i|e_j)\,p(e_j)}{p(c_i)\,p(d_j,e_j)} \\ &= \left(\frac{p(d_j)\,p(e_j)}{p(d_j,e_j)}\right)\left(\frac{p(c_i|d_j)\,p(c_i|e_j)}{p(c_i)}\right). \end{aligned}$$

The first term of the product is a factor which does not depend on the category. So, we can write the probability as:

$$p(c_i|d_j,e_j) \propto \frac{p(c_i|d_j)\,p(c_i|e_j)}{p(c_i)}$$

As $p(c_i|d_j,e_j) + p(\overline{c}_i|d_j,e_j) = 1$, we can easily compute the value of the normalizing factor, and therefore the final expression of $p(c_i|d_j,e_j)$ is:

$$p(c_i|d_j,e_j) = \frac{p(c_i|d_j)\,p(c_i|e_j)\,/\,p(c_i)}{p(c_i|d_j)\,p(c_i|e_j)\,/\,p(c_i) \;+\; p(\overline{c}_i|d_j)\,p(\overline{c}_i|e_j)\,/\,p(\overline{c}_i)} \tag{6}$$

We must make some final comments about this equation to make it more clear:

- As we said before, the posterior probability $p(c_i|d_j)$ is the one obtained from a binary probabilistic classifier (one of the two presented before, or any other), which is going to be combined with the information obtained from the link evidence.
- The prior probability used here, $p(c_i)$, is the one computed with propagation over the Bayesian network learnt with link information.
- Because the variables C_i are binary, it is clear that $p(\overline{c}_i|e_j) = 1 - p(c_i|e_j)$, $p(\overline{c}_i) = 1 - p(c_i)$ and $p(\overline{c}_i|d_j) = 1 - p(c_i|d_j)$.

3.2 Learning Link Structure

Given the previous variable setting, from the training documents, their labels and the link file, we can obtain a training set for the Bayesian network learning problem, composed of vectors of binary variables C_i and LC_i (one for each training document).

We have used WEKA package [13] to learn a generic Bayesian network (not a classifier) using a hill climbing algorithm (with the classical operators of addition, deletion and reversal of arcs) [1], with the BDeu metric [8]. In order to reduce the search space, we have limited the number of parents of each node to a maximum of 3.

Once the network has been learnt, we have converted it to the Elvira [7] format. Elvira is a software[3] developed by some Spanish researchers which implements many algorithms for Bayesian networks. In this case, we have used it to carry out the inference procedure. This is done as follows:

1. For each test document d_j, we set in the Bayesian network the LC_i variables to either lc_i or $\overline{lc_i}$, depending whether d_j is linked by at least one document of category i, or not, respectively. This is the evidence coming from the links (represented before as e_j).
2. For each category variable, C_i, we compute the posterior probability $p(c_i|e_j)$. This procedure is what is called *evidence propagation*.

Due to the size of the problem (39 categories, which give rise to a network with 78 variables), instead of exact inference, we have used an approximate inference algorithm [2], firstly to compute the prior probabilities of each category in the network, $p(c_i)$, and secondly to compute the probabilities of each category given the link evidence e_j, for each document d_j in the test set, $p(c_i|e_j)$. The algorithm used is called Importance Sampling algorithm, and is faster than other exact approaches.

4 Results

We have tested our proposal using the INEX'09 XML Document mining corpus, which contains 54572 documents, corresponding to a test/train split of 10968 documents in the training corpus (about 20% of the total), and 43604 in the test set.

The performance measures, selected by the track organizers, are Accuracy (ACC), Area under Roc curve (ROC) and F1 measure (PRF), computed over the categories (micro and macro versions), and Mean average precision by document (MAP), computed over the documents.

4.1 Preliminary Results

Four result files were sent to the organization to participate in the Workshop. Two of them, the baselines (that is to say, no link structure was used to label the documents, only their content), were obtained from the two flat-text classifiers commented in Section 2, Naive Bayes (NB) and the OR Gate (OR). The other two were the Bayesian network model (BN) proposed in Section 3 combined with the two baselines (using equation 6), NB+BN and OR+BN.

[3] Available at `http://leo.ugr.es/~elvira`

Table 1. Preliminary results

	MACC	μACC	MROC	μROC	MPRF	μPRF	MAP
NB	0.95142	0.93284	0.80260	0.81992	0.49613	0.52670	0.64097
NB + BN	0.95235	0.93386	0.80209	0.81974	0.50015	0.53029	0.64235
OR	0.75420	0.67806	0.92526	0.92163	0.25310	0.26268	0.72955
OR + BN	0.84768	0.81891	0.92810	0.92739	0.31611	0.36036	0.72508

The results of the models we sent for this track are displayed in Table 1, where M and μ mean the "macro" and "micro" versions of the performance measures, respectively.

In both cases, the Bayesian network version of the classifier outperforms the "flat" version, though the results on the OR gate are surprisingly poor in ACC and PRF. This fact is due to the nature of the classifier, and to the kind of evaluation. As both ACC and PRF require hard categorization, the evaluation procedure needs a criterion to assign categories to the test documents based on the posterior probabilities. The criterion selected by the organizers was to assign the label c_i to a document d_j if $p(c_i|d_j, e_j) > 0.5$.

But for the OR gate classifier is not known, a priori, what is the appropriate threshold τ_i such that c_i is assigned to d_j if $p(c_i|d_j, e_j) > \tau_i$. This is not a major problem to compute, for example, averaged break-even point [12] or ROC measures, where no hard categorization is needed. In this case, as the threshold 0.5 has been adopted, we need to re-adapt the model to this setting in order to perform better.

In the following section we can see how we estimated a set of thresholds (using only training data) and how we scaled the probability values, in order to match the evaluation criteria, dramatically improving the results.

4.2 Scaled Version of the Bayesian OR Gate Results

We have followed this procedure: using only training data, a classifier has been built (both in its flat and BN versions), and evaluated using cross validation (with five folds). In each fold, for each category, we have searched for the probability threshold that gives the highest $F1$ measure per class and, afterwards, all thresholds have been averaged over the set of cross validation folds.

This is what is called in the literature the *Scut* thresholding strategy [14]. Thus, we obtain, for each category, a threshold τ_i between 0 and 1 (different for each of the two models). We should then transform the original results to a scale where each category threshold is mapped to 0.5.

So, the probabilities of the OR gate model are rescaled using a linear continuous function f_i which verifies $f_i(0) = 0$, $f_i(1) = 1$ and $f_i(\tau_i) = 0.5$. The function is:

$$f_i(x) = \begin{cases} \frac{0.5x}{\tau_i} & \text{if } x \in [0, \tau_i] \\ 1 - \frac{0.5}{1-\tau_i}(1-x) & \text{if } x \in (\tau_i, 1] \end{cases}$$

Table 2. Results of the OR gate classifier using thresholds

	MACC	μACC	MROC	μROC	MPRF	μPRF	MAP
OR	0.92932	0.92612	0.92526	0.92163	0.45966	0.50407	0.72955
OR + BN	0.96607	0.95588	0.92810	0.92739	0.51729	0.55116	0.72508

Then, the new probability values are computed, using the old values $p(c_i|d_j, e_j)$, as $\hat{p}(c_i|d_j, e_j) = f_i(p(c_i|d_j, e_j))$. Once again, we would like to recall that these new results are only "scaled" versions of the old ones, with thresholds being computed using only the training set. The new results are displayed in Table 2.

Note that, using the scaling procedure, ROC and MAP values remain equal, whereas PRF and ACC, on the contrary, are considerably improved. However, for the Naive Bayes models the results obtained by the scaled and non-scaled versions were almost the same.

5 Conclusions and Future Works

Given the previous results, we can state the two following conclusions:

- The use of the Bayesian network structure for links can moderately improve a basic "flat-text" classifier.
- Our results are fairly well situated in a middle-high point among all participants in this track.

The first statement is clear, particularly in the case of the OR gate classifier, where some measures, like PRF are improved around 10%. Accuracy is improved 3-4%, while ROC stands more or less equal. Only MAP is slightly decreased (less than 1%). The changes on the naive Bayes classifier are more irrelevant, but they are positive too.

The second statement can be easily proved watching at the official table of results. Our best model (OR + BN) performs in a medium position for ACC, slightly better for PRF, fairly well for MAP (where only 4 models beat us) and very well for ROC measures (the third best performing model in each of the two versions of ROC, among all the participants).

The results could probably be improved with the usage of a better probabilistic base classifier. For example, a logistic regression or some probabilistic version of a SVM classifier (like the one proposed by Platt [11]), which are likely to have better results than our base models (although they can be much more inefficient). We expect to carry out more experiments with different basic classifiers in the future.

Acknowledgments. This work has been jointly supported by the Spanish Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía, Ministerio de Ciencia de Innovación and the research programme Consolider Ingenio 2010, under projects P09-TIC-4526, TIN2008-06566-C04-01 and CSD2007-00018, respectively.

References

1. Buntine, W.L.: A guide to the literature on learning probabilistic networks from data. IEEE Transactions on Knowledge and Data Engineering 8, 195–210 (1996)
2. Cano, A., Moral, S., Salmerón, A.: Algorithms for approximate probability propagation in Bayesian networks. In: Advances in Bayesian Networks, Studies in Fuzziness and Soft Computing, vol. 146, pp. 77–99. Springer, Heidelberg (2004)
3. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E.: OR gate Bayesian networks for text classification: a discriminative alternative approach to multinomial naive Bayes. In: XIV Congreso Español sobre Tecnologías y Lógica Fuzzy, pp. 385–390 (2008)
4. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E.: Probabilistic methods for structured document classification at INEX'07. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 195–206. Springer, Heidelberg (2008)
5. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E.: Probabilistic methods for link-based classification at INEX'08. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 453–459. Springer, Heidelberg (2009)
6. Denoyer, L., Gallinari, P.: Overview of the INEX 2008 XML Mining Track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 401–411. Springer, Heidelberg (2009)
7. Elvira Consortium: Elvira: An environment for probabilistic graphical models. In: First European Workshop on Probabilistic Graphical Models, pp. 222–230 (2002)
8. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning 20, 197–243 (1995)
9. McCallum, A., Nigam, K.: A Comparison of event models for Naive Bayes text classification. In: AAAI/ICML Workshop on Learning for Text Categorization, pp. 137–142. AAAI Press, Menlo Park (1998)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
11. Platt, J.: Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)
12. Sebastiani, F.: Machine Learning in automated text categorization. ACM Computing Surveys 34, 1–47 (2002)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
14. Yang, Y.: A study of thresholding strategies for text categorization. In: 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 137–145 (2001)
15. Yang, Y., Slattery, S.: A study of approaches to hypertext categorization. Journal of Intelligent Information Systems 18, 219–241 (2002)