Or gate Bayesian networks for text classification: A discriminative alternative approach to multinomial naive Bayes

Luis M. de Campos   Juan M. Fernández-Luna   Juan F. Huete   Alfonso E. Romero

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

Abstract

We propose a simple Bayesian network-based text classifier, which may be considered as a discriminative counterpart of the generative multinomial naive Bayes classifier. The method relies on the use of a fixed network topology with the arcs going form term nodes to class nodes, and also on a network parametrization based on noisy or gates. Comparative experiments of the proposed method with naive Bayes and Rocchio algorithms are carried out using three standard document collections.

Keywords: Bayesian network, noisy or gate, multinomial naive Bayes, text classification

1 Introduction: Probabilistic Methods for Text Classification

The classical approach to probabilistic text classification may be stated as follows: We have a class variable C taking values in the set $\{c_1, c_2, \ldots, c_n\}$ and, given a document d_j to be classified (described by a set of attribute variables, which usually are the terms appearing in the document), the posterior probability of each class, $p(c_i|d_j)$, is computed in some way, and the document is assigned to the class having the greatest posterior probability. Learning methods for probabilistic classifiers are often characterized as being generative or discriminative. Generative methods estimate the joint probabilities of all the variables, $p(c_i, d_j)$, and $p(c_i|d_j)$ is computed according to the Bayes formula:

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

The problem in this case is how to estimate the probabilities $p(c_i)$ and $p(d_j|c_i)$. In contrast, discriminative probabilistic classifiers model the posterior probabilities $p(c_i|d_j)$ directly.

The naive Bayes classifier is the simplest generative probabilistic classification model that, despite its strong and often unrealistic assumptions, performs frequently surprisingly well. It assumes that all the attribute variables are conditionally independent on each other given the class variable. In fact, the naive Bayes classifier can be considered as a Bayesian network-based classifier [1], where the network structure is fixed and contains only arcs from the class variable to the attribute variables, as shown in Figure 1.

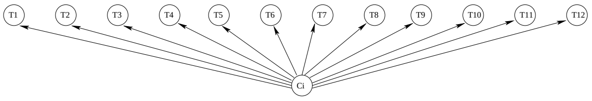

Figure 1: Network structure of the naive Bayes classifier.

In this paper we are going to propose another simple Bayesian network-based classifier, which may be considered as a discriminative counterpart of naive Bayes, in the following senses: (1) it is based on a type of Bayesian network similar to that of naive Bayes, but with the arcs in the network going in the opposite direction; (2) it requires the same set of simple sufficient statistics than naive Bayes, so that the complexity of the training step in both methods is the same, namely linear with the number of attribute variables; the complexity of the classification step is also identical.

The rest of the paper is organized in the following way: in Sections 2 and 3 we describe the two probabilistic text classifiers we are considering, naive Bayes and the proposed new model, respectively. Section 4 is focused on the experimental results. Finally, Section 5 contains the concluding remarks and some proposals for future work.

2 The Multinomial Naive Bayes Classifier

In the context of text classification, there exist two different models called naive Bayes, the multivariate Bernoulli naive Bayes model [2, 4, 8] and the multinomial naive Bayes model [5, 6]. In this paper we shall only consider the multinomial model. In this model a document is an ordered sequence of words or terms drawn from the same vocabulary, and the naive Bayes assumption here means that the occurrences of the terms in a document are conditionally independent given the class, and the *positions* of these terms in the document are also independent given the class[1]. Thus, each document d_j is drawn from a multinomial distribution of words with as many independent trials as the length of d_j. Then,

$$p(d_j|c_i) = p(|d_j|)\frac{|d_j|!}{\prod_{t_k \in d_j} n_{jk}!} \prod_{t_k \in d_j} p(t_k|c_i)^{n_{jk}} , \quad (2)$$

where t_k are the distinct words in d_j, n_{jk} is the number of times the word t_k appears in the document d_j and $|d_j| = \sum_{t_k \in d_j} n_{jk}$ is the number of words in d_j. As $p(|d_j|)\frac{|d_j|!}{\prod_{t_k \in d_j} n_{jk}!}$ does not depend on the class, we can omit it from the computations, so that we only need to calculate

$$p(d_j|c_i) \propto \prod_{t_k \in d_j} p(t_k|c_i)^{n_{jk}} . \quad (3)$$

The estimation of the term probabilities given the class, $\hat{p}(t_k|c_i)$, is usually carried out by means of the Laplace estimation:

$$\hat{p}(t_k|c_i) = \frac{N_{ik} + 1}{N_{i\bullet} + M} , \quad (4)$$

where N_{ik} is the number of times 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, i.e. $N_{i\bullet} = \sum_{t_k} N_{ik}$, and M is the size of the vocabulary (the number of distinct words in the documents of the training set).

The estimation of the prior probabilities of the classes, $\hat{p}(c_i)$, is usually done by maximum likelihood, i.e.:

$$\hat{p}(c_i) = \frac{N_{i,doc}}{N_{doc}} , \quad (5)$$

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 are assigned to class c_i.

The multinomial naive Bayes model can also be used in another way: instead of considering only one class variable C having n values, we can decompose the problem using n binary class variables C_i taking its values in the sets $\{c_i, \overline{c}_i\}$. This is a quite common transformation in text classification [9], especially for multilabel problems, where a document may be associated to several classes. In this case n naive Bayes classifiers are built, each one giving a posterior probability $p_i(c_i|d_j)$ for each document. In the case that each document may be assigned to only one class (single-label problems), the class $c^*(d_j)$ such that $c^*(d_j) = \arg\max_{c_i}\{p_i(c_i|d_j)\}$ is selected. Notice that in this case, as the term $p_i(d_j)$ in the expression $p_i(c_i|d_j) = p_i(d_j|c_i)p_i(c_i)/p_i(d_j)$ is not necessarily the same for all the class values, we need to compute it explicitly through

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

This means that we have also to compute $p_i(d_j|\overline{c}_i)$. This value is estimated using the corresponding counterparts of eqs. (3) and (4), where

$$\hat{p}(t_k|\overline{c}_i) = \frac{N_{\bullet k} - N_{ik} + 1}{N - N_{i\bullet} + M} . \quad (6)$$

$N_{\bullet k}$ is the numbers of times that the term t_k appears in the training documents, i.e. $N_{\bullet k} = \sum_{c_i} N_{ik}$, and N is the total number of words in the training documents.

3 The OR Gate Bayesian Network Classifier

The document classification method that we are going to propose is based on another restricted type of Bayesian network with the following topology: Each term t_k appearing in the training documents (or a subset of these terms in the case of using some method for feature selection) is associated to a binary variable T_k taking its values in the set $\{t_k, \overline{t}_k\}$, which in turn is represented in the network by the corresponding node. There are also n binary variables C_i taking its values in the sets $\{c_i, \overline{c}_i\}$ (as in the previous binary version of the naive Bayes model) and the corresponding class nodes. The network structure is fixed, having an arc going from each term node T_k to the class node C_i if the term t_k appears in training documents which are of class c_i. Let nt_i be the number of different terms appearing in documents of class c_i. In this way we have a network topology with two layers, where the term nodes are the "causes" and the class nodes are the "effects", having a total of $\sum_{i=1}^{n} nt_i$ arcs. An example of this network topology is displayed in Figure 2. It should be noticed that the proposed topology, with arcs going from attribute nodes to class nodes, is the opposite of the one associated to the naive Bayes model.

[1] The length of the documents is also assumed to be independent on the class.

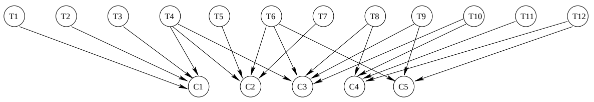

Figure 2: The OR gate classifier.

It should also be noticed that this network topology explicitly requires modeling the "discriminative" conditional probabilities $p(c_i|pa(C_i))$, where $Pa(C_i)$ is the set of parents of node C_i in the network (the set of terms appearing in documents of class c_i) and $pa(C_i)$ is any configuration of the parent set (any assignment of values to the variables in this set). As the number of configurations is exponential with the size of the parent set[2], we use a canonical model to define these probabilities, which reduce the number of required numerical values from exponential to linear size. More precisely, we use a noisy OR Gate model [7].

The conditional probabilities in a noisy OR gate are defined in the following way:

$$\begin{aligned} p(c_i|pa(C_i)) &= 1 - \prod_{T_k \in R(pa(C_i))} (1 - w(T_k, C_i)) \ , \quad (7) \\ p(\overline{c}_i|pa(C_i)) &= 1 - p(c_i|pa(C_i)) \, , \quad (8) \end{aligned}$$

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).

3.1 Classification as Inference

Once the weights $w(T_k, C_i)$ have been estimated, and given a document d_j to be classified, we instantiate in the network each of the variables T_k corresponding to the terms appearing in d_j to the value t_k (i.e. $p(t_k|d_j) = 1$ if $t_k \in d_j$), and all the other variables T_h (those associated to terms that do not appear in d_j) to the value $\overline{t}_h$ (i.e. $p(t_h|d_j) = 0 \; \forall t_h \notin d_j$). Then, we compute for each class node C_i the posterior probabilities $p(c_i|d_j)$. As in the case of the naive Bayes model, we would assign to d_j the class (or classes) having the greatest posterior probability.

The combination of network topology and numerical values represented by OR gates allows us to compute very efficiently and in an exact way the posterior probabilities:

$$\begin{aligned} p(c_i|d_j) &= 1 - \prod_{T_k \in Pa(C_i)} (1 - w(T_k, C_i) \times p(t_k|d_j)) \\ &= 1 - \prod_{T_k \in Pa(C_i) \cap d_j} (1 - w(T_k, C_i)) \ . \quad (9) \end{aligned}$$

In order to take into account the number of times a word t_k occurs in a document d_j, n_{jk}, we can replicate each node T_k n_{jk} times, so that the posterior probabilities then become

$$p(c_i|d_j) = 1 - \prod_{T_k \in Pa(C_i) \cap d_j} (1 - w(T_k, C_i))^{n_{jk}} \ . \quad (10)$$

3.2 Training as Weight Estimation

The estimation of the weights in the OR gates, $w(T_k, C_i)$, can be done in several ways. The simplest one is to compute $w(T_k, C_i)$ as $\hat{p}(c_i|t_k)$, the estimated conditional probability of class c_i given that the term t_k is present. We can do it by maximum likelihood:

$$w(T_k, C_i) = \frac{N_{ik}}{N_{\bullet k}} \, , \quad (11)$$

or using Laplace:

$$w(T_k, C_i) = \frac{N_{ik} + 1}{N_{\bullet k} + 2} \, . \quad (12)$$

Another, more accurate way of estimating $w(T_k, C_i)$ is directly as $\hat{p}(c_i|t_k, \overline{t}_h \, \forall T_h \in Pa(C_i), \, T_h \neq T_k)$. However, this probability cannot be reliably estimated, so that we are going to compute an approximation in the following way:

$$\hat{p}(c_i|t_k, \overline{t}_h \, \forall h \neq k) = p(c_i|t_k) \prod_{h \neq k} \frac{p(c_i|\overline{t}_h)}{p(c_i)} \quad (13)$$

This approximation results from assuming a conditional independence statement similar to that of the naive Bayes classifier, namely

$$p(t_k, \overline{t}_h \, \forall h \neq k | c_i) = p(t_k|c_i) \prod_{h \neq k} p(\overline{t}_h|c_i). \quad (14)$$

In that case

$$\begin{aligned} p(c_i|t_k, \overline{t}_h \, \forall h \neq k) &= \frac{p(t_k, \overline{t}_h \, \forall h \neq k | c_i) p(c_i)}{p(t_k, \overline{t}_h \, \forall h \neq k)} \\ &= \frac{p(t_k|c_i) \left(\prod_{h \neq k} p(\overline{t}_h|c_i) \right) p(c_i)}{p(t_k) \prod_{h \neq k} p(\overline{t}_h \, \forall h \neq k)} \\ &= p(c_i|t_k) \prod_{h \neq k} \frac{p(c_i|\overline{t}_h)}{p(c_i)} \end{aligned}$$

[2] Notice that $|Pa(C_i)| = nt_i$.

The values of $p(c_i|t_k)$ and $p(c_i|\bar{t}_h)/p(c_i)$ in eq. (13) are also estimated using maximum likelihood. Then, the weights $w(T_k, C_i)$ are in this case:

$$w(T_k, C_i) = \frac{N_{ik}}{N_{\bullet k}} \times \prod_{h \neq k} \frac{(N_{i\bullet} - N_{ih})N}{(N - N_{\bullet h})N_{i\bullet}} \qquad (15)$$

Another option is to relax the independence assumption in the following way:

$$p(t_k, \bar{t}_h \,\forall h \neq k|c_i) = \frac{p(t_k|c_i)}{nt_i} \prod_{h \neq k} p(\bar{t}_h|c_i). \qquad (16)$$

We are assuming that the joint probability of the events $[t_k, \bar{t}_h \forall h \neq k]$ is smaller than the pure independence assumption would dictate. The weights $w(T_k, C_i)$ would be in this case

$$w(T_k, C_i) = \frac{N_{ik}}{nt_i N_{\bullet k}} \times \prod_{h \neq k} \frac{(N_{i\bullet} - N_{ih})N}{(N - N_{\bullet h})N_{i\bullet}} \qquad (17)$$

In any case, the set of sufficient statistics necessary to compute the weights are N_{ik} $\forall t_k$, $\forall c_i$, i.e. the number of times each term t_k appears in documents of each class c_i, the same required by multinomial naive Bayes.

4 Experimentation

For the evaluation of the proposed model we have used three document test collections: Reuters-21578, Ohsumed and 20 Newsgroups. Reuters-21578 (ModApte split) contains 12,902 documents (9603 for training and 3299 for testing) and 90 categories (with at least 1 training and 1 test documents). Ohsumed, including 20000 medical abstracts from the MeSH categories (10000 for training and 10000 for testing) of the year 1991, and 23 categories. 20 Newsgroups corpus contains 19997 articles for 20 categories taken from the Usenet newsgroups collection, where only the subject and the body of each message were used. Note that there is no fixed literature split for this collection. All the three collections were preprocessed in the same way using stemming (Porter's algorithm) and stopword removal (SMART's system 571 stopword list). No term selection was carried out.

The evaluation takes into account that the classification process will generate an ordered list of possible categories, in decreasing order of probability[3], instead of a definite assignment of categories to each document. Then, as performance measures, we have firstly selected the typical measures used in multi-label categorization problems (as they are Reuters-21578 and Ohsumed): *breakeven* point[4] (BEP) and the *average 11-point precision*[5] (Av-11). Another measure commonly used is F_1[6]. However F_1 requires a precise assignment of classes to each document, so that we shall use instead F_1 at one (F_1@1) and also F_1 at three and five (F_1@3, F_1@5) document level: the F_1 value obtained by assuming that the system assigns to each document either the first or the first three or five most probable classes. Both breakeven and F_1 values will be computed in micro-average (micr.) and macro-average (macr.). In all the measures, a higher value means a better performance of the model.

We have executed experiments using naive Bayes and the proposed OR gate classifier, although we have also included another non probabilistic classifier in the comparison, namely the well-known Rocchio method [3], used as a perspective. Tables 1, 2 and 3 display the values of the performance measures obtained. We do not display the results of the OR gate classifier using all the different parameter estimation methods commented in Subsection 3.2. Instead, some preliminary experimentation showed that the best performing methods were those based on equation 12 (for estimation based on $\hat{p}(c_i|t_k)$) and equation 17 (for estimation based on $\hat{p}(c_i|t_k, \bar{t}_h \,\forall h \neq k)$).

Several conclusions can be drawn from these experiments: the proposed OR gate model is quite competitive, frequently outperforming Rocchio and naive Bayes. Particularly, this new model seems to perform quite well in terms of macro averages: it gives a more balanced treatment to all the classes, and this is especially evident in those problems where the class distribution is quite unbalanced, as Reuters and Ohsumed. At the same time, the OR gate model (the one based on eq. 17) performs generally well also in terms of micro averages.

5 Concluding Remarks

We have described a new approach for document classification, the so called "OR Gate classifier", with different variants based on several parameter estimation methods. It is based on a Bayesian network representation which is, in some sense, the opposite that the

[3] We are therefore using an instance of the so-called category-ranking classifiers [9].

[4] The point where precision equals recall, by moving a threshold.

[5] The precision values are interpolated at 11 points at which the recall values are 0.0, 0.1,..., 1.0, and then averaged.

[6] The harmonic mean of precision and recall.

	micr.BEP	macr.BEP	Av-11
	Reuters		
NBayes	0.73485	0.26407	0.84501
Rocchio	0.47183	0.42185	0.84501
OR-eq12	0.66649	**0.55917**	0.81736
OR-eq17	**0.76555**	0.54370	**0.89725**
	Ohsumed		
NBayes	**0.58643**	0.49830	**0.76601**
Rocchio	0.42315	0.44791	0.68194
OR-eq12	0.48017	**0.58792**	0.64739
OR-eq17	0.53122	0.56450	0.72925
	20Newsgroups		
NBayes	0.71778	0.73629	**0.88834**
Rocchio	0.60940	0.63875	0.86583
OR-eq12	**0.80732**	**0.81333**	0.87889
OR-eq17	0.77689	0.79779	0.86208

Table 1: Micro and macro breakeven points and average 11-point precision.

	micr.F1@1	micr.F1@3	micr.F1@5
	Reuters		
NBayes	**0.75931**	0.49195	0.35170
Rocchio	0.70047	0.47399	0.34979
OR-eq12	0.67297	0.45352	0.33493
OR-eq17	0.75369	**0.51461**	**0.36825**
	Ohsumed		
NBayes	**0.53553**	**0.53979**	0.42718
Rocchio	0.46064	0.46313	0.40912
OR-eq12	0.41676	0.46329	0.40292
OR-eq17	0.49048	0.50883	**0.42773**
	20Newsgroups		
NBayes	0.80881	**0.48705**	**0.33212**
Rocchio	0.77233	0.48323	0.33153
OR-eq12	**0.81000**	0.47266	0.32614
OR-eq17	0.77858	0.47284	0.32695

Table 2: Micro F_1 values.

one associated to the naive Bayes classifier. The complexity of the training and classification steps for the proposed model is equivalent to that of the naive Bayes too. In fact we can think of the OR gate model as a kind of discriminative version of naive Bayes, which is not a discriminative but a generative method.

According to the results of the experimental comparison carried out using several standard text collections, we found that the new model can compete with naive Bayes, especially in terms of macro averages and in those cases where the class distribution is unbalanced.

We believe that the OR gate model could greatly benefit from a term/feature selection preprocessing, and we plan to test this assertion in the future.

	macr.F1@1	macr.F1@3	macr.F1@5
	Reuters		
NBayes	0.39148	0.28935	0.23116
Rocchio	0.39148	0.28935	0.23116
OR-eq12	0.11263	0.20092	0.26404
OR-eq17	**0.45762**	**0.39169**	**0.30959**
	Ohsumed		
NBayes	0.40627	0.47260	0.40732
Rocchio	**0.45421**	0.50604	0.44945
OR-eq12	0.19980	0.42017	0.45870
OR-eq17	0.43602	**0.53615**	**0.50103**
	20Newsgroups		
NBayes	**0.80985**	0.54983	0.39048
Rocchio	0.77095	0.54086	0.39113
OR-eq12	0.80880	0.58083	0.44502
OR-eq17	0.78682	**0.59099**	**0.44722**

Table 3: Macro F_1 values.

Acknowledgments. This work has been jointly supported by the Spanish Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía, Ministerio de Educación y Ciencia and the research programme Consolider Ingenio 2010, under projects TIC-276, TIN2005-02516 and CSD2007-00018, respectively.

References

[1] S. Acid, L.M. de Campos, J.G. Castellano. Learning Bayesian network classifiers: searching in a space of acyclic partially directed graphs. *Machine Learning* 59(3), 213-235 (2005).

[2] D. Koller, M. Sahami. Hierarchically classifying documents using very few words. In: *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 170–178, (1997).

[3] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pp. 143–151, (1997).

[4] L.S. Larkey, W.B. Croft. Combining classifiers in text categorization. In: *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 289–297 (1996).

[5] D. Lewis, W. Gale. A sequential algorithm for training text classifiers. In: *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12 (1994).

[6] A. McCallum, K, Nigam. A Comparison of event models for Naive Bayes text classification. In: *AAAI/ICML Workshop on Learning for Text Categorization*, pp. 137–142, AAAI Press (1998).

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo (1988).

[8] S.E. Robertson, K. Spärck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science* 27, 129–146 (1976).

[9] F. Sebastiani. Machine Learning in automated text categorization. *ACM Computing Surveys* 34, 1–47 (2002).