Very Large Bayesian Networks in Text Classification

Mieczysław A. Kłopotek and Marcin Woch

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. The paper presents results of empirical evaluation of a Bayesian multinet classifier based on a new method of learning very large tree-like Bayesian networks. The study suggests that tree-like Bayesian networks are able to handle a classification task in one hundred thousand variables with sufficient speed and accuracy.

1 Introduction

Automatic classification of natural language texts, especially for very large corpora of documents like WWW search engines, patent databases etc. is on the one hand of increasing importance and on the other hand a challenging task due to: the number of documents (10000–1,000,000 and more), the number of attributes (10000–100000 of terms), and limited understanding of human approach to document classification.

The inherent subjectiveness and partial knowledge may be quite well be treated by a probabilistic bayesian toolset. In this paper we study application of a Bayesian multinet based classifier to text classification task. Similar approaches were reported in the past. However, always a heavily restricted vocabulary was used because Bayesian network learning in high number of variables had so far prohibitive complexity. Recently a new algorithm for Bayesian network learning, ETC algorithm has been proposed [4,5], that has a significantly reduced complexity. The current research explores these results to handle unconstrained or only weakly constrained vocabularies.

In section 2 we recall basics of the ETC algorithm. In section 3 we characterize the multinet classifier used. In section 4 we explain evaluation criteria. In section 5 wepresent the experimental setting, and results concerning proper dependency measure for ETC, accuracy and related characteristics of the classifier as well as the speed of the ETC algorithm. Section 6 contains some concluding remarks.

2 Remarks on ETC

The ETC was described in detail in [4,5]. It constructs a tree-like Bayesian network, but contrary to the Chow/Liu algorithm [2] it does not need to compare all variables with each other so that it saves much calculations of so-called DEP-measure. Recall that Chow/Liu algorithm exploits a DEP defined as

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 397–406, 2003.

$DEP_{chow/Liu}(X,Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$. ETC requires that DEP has the tree-common-sense assumption. Therefore we introduced new DeP's. In simplest case of text categorization we deal only with binary variables. We shall denote variables with capital letters. Assume that all variables, e.g. X, take on two values only, e.g. $dom(X) = \{x, \overline{x}\}$. After [6] define the $DEP_{\rightarrow}()$ measure as follows: $DEP_{\rightarrow}(Y,X) = P(x|y) - P(x|\overline{y})$. Define also $DEP[_{\rightarrow}](Y,X) = (DEP_{\rightarrow}(Y,X))^2$. Out of this measure we can derive symmetric ones in a number of ways: e.g. additively or multiplicatively: $DEP+(X,Y) = DEP[_{\rightarrow}](Y,X) + DEP[_{\rightarrow}](Y,X)$, $DEP*(X,Y) = DEP[_{\rightarrow}](Y,X) \cdot DEP[_{\rightarrow}](Y,X)$. Both $DEP_{+}()$ and $DEP_{*}()$ can be used in the ETC algorithm, as the first fulfills the tree-common-sense assumption under special circumstances and the second fulfills it unconditionally [6].

3 Bayesian Network Based Classifiers

The so-called "naive Bayes" classifier has been used for comparison in this study because ithas been successfully applied in text categorization previously [11]. It may be viewed as a primitive form of a bayesian network (decision node connected to all other variables, which are not connected themselves).

In our approach we use the Bayesian multinets. A multinet-classifier allows for different structures of dependencies between variables for each level of the category variable. So it learns separate Bayesian tree-like network (in our case using the ETC algorithm). For classification purposes we need to identify the a-priori probabilities $P_C(C)$ of the categorical variable C.

Classification with a Bayesian multinet is carried out by identifying the category c maximizing $P(c|x_1,\ldots,x_n)$. This probability is calculated according to the Bayes rule as $P(c|x_1,\ldots,x_n) = \frac{P(x_1,\ldots,x_n|c)P(c)}{P(x_1,\ldots,x_n)}$ As $P(X_1,\ldots,X_n)$ is not category-label dependent, we can simplify the above as: $P(c|x_1,\ldots,x_n) = \eta P(x_1,\ldots,x_n|c)P(c)$ where η is any positive constant.

4 Evaluation Criteria

Let us denote with $C(x)$ the intrinsic category of the instance (document) x, and with $\widehat{C}$ the category proposed by the classifier. Let $CM(i,j)$ denote the number of instances from the category i classified by the classsifier into the category j. Then accuracy is the probability of correct classification, calculated as: $Acc = P(C(x) = \widehat{C}(x)) = \frac{\sum_{i=j} CM(i,j)}{\sum_i \sum_j CM(i,j)}$

Precision, calculated for each category separately, is calculated as the share of correctly classified instances among those classified by the classifier into the given category.

$Precision(C=i) = P(C(x) = i|\widehat{C}(x) = i) = \frac{CM(i,i)}{\sum_j CM(j,i)}$

The Recall (completenress) for a category is the proportion of correctly classified documents of the given category among all the documents truly belonging to that category.

Both general precision and general recall is a (weighed) average over all categories.

Both precision and recall should be optimized. Usually these two goals are contradictory. To balance both, so-called F-measure has been introduced calculated as: $F_\beta(c) = \frac{(1+\beta^2)Precision(c)\cdot Recall(c)}{\beta^2 Precision(c)+Recall(c)}$ where the parameter β defines the balance between both (F_0 means precision, F_{oo} means recall). For the purposes of our evaluation we used F_2, as we assume that for the Internet user the precision is more important than recall. In the results F_2 will be averaged over all categories.

5 Experiments and Results

5.1 Data and Experimental Settings

For testing classification capability we used several sets of pre-classified documents: 20 newsgroups (20 categories) and WebKB4 (4 categories), created within the known CMU World Wide Knowledge Base (Web→KB) project (see http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/index.html), as well as sets created ourselves: Yahoo A set – a collection of 10,899 documents taken from randomly selected 15 Yahoo categories, and a set Yahoo B – a collection of 5,011 documents taken from 21 subcategories of Yahoo's Health: Diseases and Conditions (balanced representation of categories was strived at: in case that the catalogue contained too few documents, the links were followed recursively to obtain more documents), and UseNet discussion groups – 8,094 documents from 12 groups.

The experiments were carried out on a computer with limited capabilities (Celereon CPU 1.70 GHz, 504 MB RAM), therefore the number of categories was restricted to about 20, and the number of documents to about 20,000. The construction time of Bayesian networks varied, but did not exceed 2 min. Hence the construction time for a classifier multinet did not exceed 30-40 min.

All documents were-preprocessed by parsers, removing link information (which often contained directly category information), removing HTML tags, stop-words, and stemming using the Porter algorithm was carried out. Documents were stored as binary vectors.

While testing, dictionaries of varying sizes were used, with 10,20,40,80 etc. terms up to the full dictionary size. As a term selection criterion Information gain was used. A thre-fold hould-out test was carried out for each dictionary size, 30

5.2 Choice of DEP Measure

We estimated also the fitness of ETC to the data by determining the log likelihood: for an artificial set and for the above test data.

Table 1. Comparison of fitness (log likelihood) of Bayesian networks obtainied using various DEP measures. A 25-node artificial network

DEP measure	Average	Max	Min	St. dev.
$DEP_{Chow/Liu}$	**-132530.3**	-129679.5	-136711.9	1916.0
DEP_{+}	-134021.6	-129927.4	-138213.3	2101.4
DEP_{*}	-132732.0	-129932.9	-137532.0	2170.3
Random structure	-137954.9	-134707.7	-138663.8	965.1
Chow/Liu network	-129671.5			

The goal was to check the quality of the structure of a Bayesian network obtained using ETC algiorithm for various DEP functions. It is known that the log likelihood function $ll(x^1, x^2, \ldots, x^N) = \sum_{k=1,\ldots,N} \sum_{i=1,\ldots,n} logP(x_i^k | x_{\pi(X_i)}^k)$ where N is the cardinality of training set and x^k is the k^{th} instance in the sample, is maxmized for the Chow/Liu tree, which is the best tree approximator to a probability distribution. Therefore we used it as a measure of quality of trees generated by ETC.

Table 2. Comparison of fitness (log likelihood) of Bayesian networks obtainied using various DEP measures. A 500-node network for the category talk.religion.misc of the "20 newsgroups"database

DEP measure	Average	Max	Min	St. dev.
$DEP_{Chow/Liu}$	-71436.9	-70073.1	-72371.0	555.0
DEP_{+}	-72394.8	-71457.6	-73388.3	489.9
DEP_{*}	**-71364.6**	-69638.2	-72371.2	548.1
Random structure	-75036.8	-74687.6	-75198.0	108.1
Chow/Liu network	-69553.2			

The results reported in Table 1 were obtained for a set containing 10,000 instances generated using the Hugin Lite program (http://www.hugin.com/) for a tree-like Bayesian network with 25 nodes and randomly generated conditional probability tables. The results reported in Table 2 were obtained using real data for the category talk.religion.misc of the "20 newsgroups"database with a dictionary reduced to 500 terms. In general our tests were concerned with networks of up to 1000 nodes.

For each dataset 50 networks were generated. For each network log likelihood was calculated The tables contain average, minimal and maximal values as well as standard deviations. The results were compared against a network with random structure (conditional probabilities from data). Below the (best possible) result for the Chow/Liu algorithm is presented.

The conclusion from our experiments was that DEP_{*} is the best candidate as for small number of variables $DEP_{Chow/Liu}$ outperformed it only slightly and in other experiments (not reported here) DEP_{*} worked much better. DEP_{+} on the other hand provided usually with worst results.

For this reason ETC with DEP_* was used in experiments reported subsequently.

5.3 Accuracy, Recall, F_2

One important aspect of a classifier is its accuracy. We compared ETC based multi-net classifier accuracy with Naive Bayes accuracy (NB). On the one hand, though NB is not a particularly good one, it scales quite well for tasks with dozenz of thousands of attributes. Onm the other hand we know that whatever is worse than NB is really bad. For comparisons of NB with other classifiers see [10,11]. In Figure 1 we see that ETC classifier performs better than NB for larger vocabularies

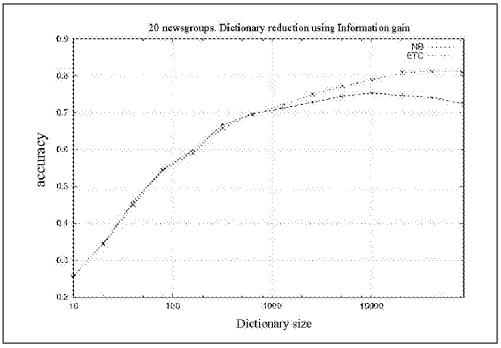

Fig. 1. Impact of dictionary size on the accuracy for ETC and Naive Bayes classifiers (20 newsgroups database)

Standard errors are presented (sometimes so small that invisible). As expected, the ETC multinet without the "naive" assumptions of NB performs much better. Accuracy is comparable only for dictionaries of 100-1000 terms (a dictionary of 160 terms allowed NB to perform as well as ETC, otherwise it performed worse). It is worth nopting, that for databases not depicted here (Yahoo A, Yahoo B, discussion groups) the performance of ETC was even better. This is very important because they were collected fully automatically, without prior removal of irrelevant, "noise" documents. This indicates high practical value of the classifier where we cannot manually check the quality of the training set.

We tested the impact of the number of categories on the performance of the ETC classifier. This test relied on the 20newsgroups database as it contains the largest number of categories. We tested the performance for 2, 4, 8, 16 categories – see Figure 2. As expected, the higher number of categories, the lower accuracy. The differences in accuracy are most strongly visible for small dictionaries, but they narrow to 17–18% for dictionaries with 80,000 values.

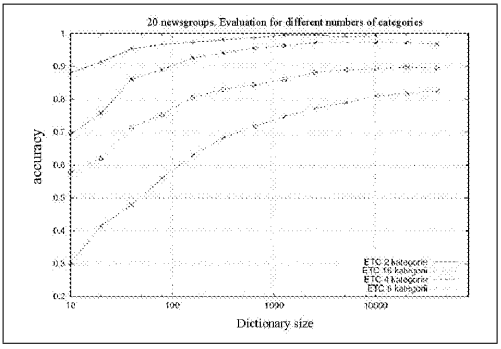

Fig. 2. Impact of the number of categories on accuracy for ETC (20 newsgroups database)

ETC exhibits a bit higher stability than NB. Standard error values are usually slightly lower than those for NB classifier, though the differences are not striking. It turns out that in spite of the possibility of generation of different trees in case of different sequences of variables the quality of the Bayesian networks obtained is similar. An interesting artifact is the worsening of the accuracy after exceeding some optimal dictionary size. The method of dictionary reduction (scoring of terms with Information Gain) prefers terms most stongly differentiating the documents in distinct categories. So the explanation of the phenomenon of decaying classification quality may be as follows: when exceeding some dictionary size terms included are purely random (misspellings, strings not being natural language words, formatting instructions like ASCII Art. Another explanation may be model overfitting, though restriction to tree-like structures of Bayesian networks highly prevents overfitting. Finally, as demonstrated in [7] for Naive Bayes classifier, in case of large dictionaries a model of document representation taking into account frequencies of word occurrence may be more appropriate.

These hypotheses will be subject of future investigations. Figure 3 presents the precision and recall as a function of dictionary size for the 20 newsgroups database. For high values of dictionary size, the recall curve approaches from below the precision curve.

It is an exception for the sets studied, because elsewhere consistently the precision was higher than recall. As stated earlier, we assume that the user is more interested in precision than in recall, therefore we synthesized the precision and recall results in terms of F_2 measure. In Figure 4 we see F_2 value for all the sets. It is apparent that F_2 strongly depends on the dataset under consideration.

Let us split the datasets according to their origin: HTML documents taken from the Internet ("WebKB4", Yahoo! A and B) and discussion groups ("20 newsgroups" and "discussion groups").

We can notice a significantly better classification quality for documents stemming from discussion groups. This can be surely attributed to significantly better

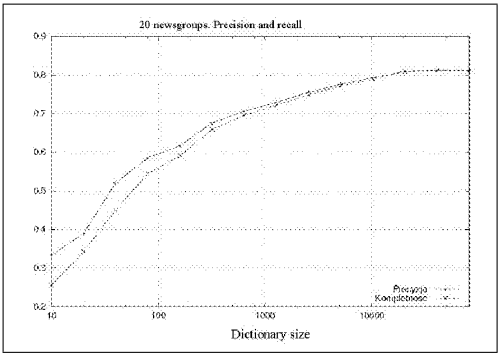

Fig. 3. Impact of dictionary size on precision and recall for ETC (20 newsgroups database)

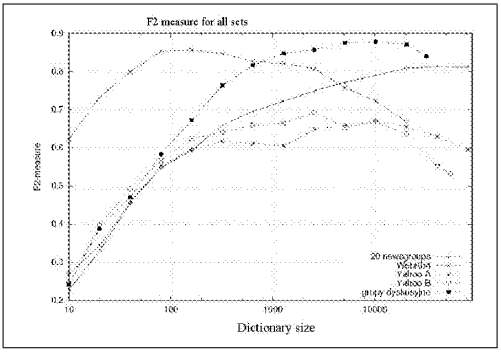

Fig. 4. F_2 measure on dictionary size (all databases)

quality of training data for the classifier: while downloading documents from discussion group we have no chance of misclassifying them. A different situation is with Internet documents. Only in case of documents directly present in respective catalogues we may be assured that they really belong to the category under consideration, whereas the recursively collected additional document may or may not belong to the category of interest (the spider didn't check the relevance of links visited).

The significantly better quality of classification for the discussion group set compared to 20- newsgroups set should not wonder if we recall the results presented in Figure 2. The discussion groups set contains simply less categories, hence the better quality of classification.

We see the same in the internet group. Results for the "WebKB4" with only four categories are significantly better than for Yahoo! A and B. The second of them with more categories and less training data and more close themes of documents should generate worse results.

5.4 Time Complexity

We have also investigated the time complexity of ETC. The theoretical complexity is n log(n) given that the edge tree is ideally balanced. But ideal balancing is in general not possible. It turns out that for Bayesian networks obtained from text documents it happens frequently that one node has many neighbours which leads to heavily unbalanced structured. Luckily, as our investigations demonstrate, the unbalanced structures do not deteriorate significantly the performance of the algorithm. The balancing procedure yields sufficiently "flat" edge trees.

Timing results were byproducts of the previously described accuracy experiments. In Figure 5 we present the comparison of theoretical and averaged real number of calls to DEP function while building an edge tree. Assuming that the tree is balanced ideally in every step, the exact number of DEP calls should be $2 \cdot ((n-1)log_2(n-1) - n + 1)$. Hence the figure contains the curve $p \cdot ((n-1)log_2(n-1) - n + 1)$, where p is a parameter determined for each experiment separately and tells how hard the missing balance deteriorates the performance. Ideal values of p are of course equal to 2.

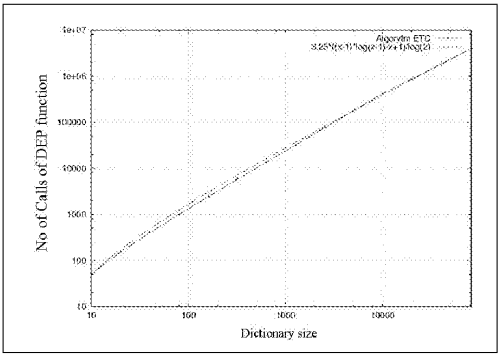

Fig. 5. Real and theoretic number of DEP function calls for ETC on dictionary size (20newsgroups database)

The figure illustrates a good fit of experimental results to the shape of the theoretical curve. Values of the parameter p in general did not exceed 6. For the sets Yahoo A and Yahoo B p-values were 2.62 and 2.40 resp. Which lies close o the ideal value. For the discussion groups p= 2.879. Nieco gorsz1 is also satisfactory. For the smallest set "WebKB4" p=5.18, which is high compared to other sets. This may be related to poor performance of classifier construction for the dictionary with the largest number of terms. For smaller dictionaries the results are closer to expected ones.

Figure 6 presents averaged network construction times for all document sets. It turns out that ETC construction time for domains with up to 1000,000 on average does not exceed 2 min. (PC, processor 1.7 GHz). The shape of the

curves indicates that the network construction time is loglinear in dictionary size. However, the curves fit less the ideal as the program constructing the networks shares time with other processes running on the computer implying distortions.

Worst execution time was obtained for "WebKB4" with large dictionary. This corresoonds to the performance in the number of DEP calls and is related to bad tree balancing.

Characteristic for all the curves is their similar slop (on log scale). This suggests that the complexity measured in the number of DEP calls is in fact multiplied by a constant factor, dependent on the sets. This factor, independent of the dictionary size, represe4nts the computational complexity of the DEP function which in turn depends linearly on the number of documents. Figure 6 shows that DEP computation time for the sets is as follows: WebKB4>>20 newsgroups≥ Yahoo! A>> discussion groups>Yahoo! B.

The execution time is influenced not only by the number of DEP calls, but also by the execution time of DEP itself. A special bipartite representation of document vectors has been applied which is efficvient for sparse data that we have here. The reason for this is clear if we consider the average number of documents per category: for "WebKB4" we3 have 1050, for "20 newsgroups" 1000, for Yahoo! A 726, for discussion groups – 675, and for Yahoo! B 238.

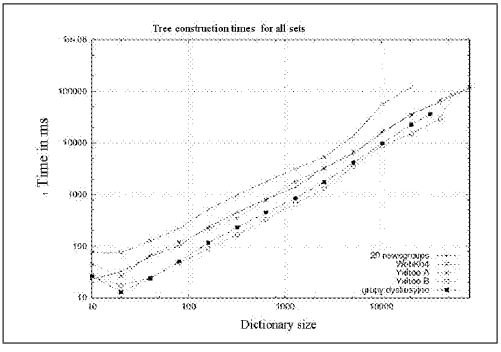

Fig. 6. Real execution time in ms for ETC on dictionary size (all databases)

A separate explanation is necessary for a high average time of net construction for the set This set possesses special characteristics: it has only 4 categories in which documents are very similar(home pages of stuff, of students, course descriptions) which implies a limited vocabulary. Word occurrence vectors are 2-3 times denser than for "20 newsgroups".

To reduce the ETC complexity, the popular words should be removed from the dictionary. But in some cases this may deteriorate the accuracy of the classification (as e.g. for the set "WebKB").

6 Conclusions

The experiments reported in this paper demonstrated the possibility of construction of Bayesian networks with up to 100,000 nodes from data. Execution time measurements indicate good scaling of the ETC algorithm. The obtained average numbers of calls of the DEP function do not deviate significantly from the theoretic estimates for optimistic balanced edge tree. It is wotrth mentioning that the results reported were obtained for Java implementation. If more efficient languages like C were used, we could expect that a 100,000 node network would be created within a time below 30 seconds.

Also the classification accuracy of a multi-net classifier based on ETC appears to be satisfactory, significantly outperforming naive Bayes classifier.

For a number of reasons (see conclusion in [10]) a reliable comparison of our results with those for other methods seems to be difficult. A rough comparison indicates that the quality of classification with ETC is comparable to methods like kNN, SVM or LLSF. See e.g. papers [1,3,8,9] to see results of categorization using alternative methods for the dataset "20 newsgroups"

ETC appears to be a stable algorithm. Both standard errors for classification accuracy and standard deviations in tables 1 and 2 indicate that even with slightly different network structure obtained using different sequences of variables the obtained networks do not diff significantly in quality.

References

1. L. D. Baker, A. K. McCallum: Distributional clustering of words for text classification, Proc. SIGIR-98, ACM Press, New York, US, pp. 96–103, 1998,
2. C.K.Chow, C.N.Liu: Approximating discrete probability distributions with dependence trees, IEEE Trans. on Information Theory, Vol. IT-14, No.3, (1968), 462–467
3. T. Joachims: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, Proc. ICML-97, Morgan Kaufmann, 143–151,1997,
4. M.A.Kłopotek: A New Bayesian Tree Learning Method with Reduced Time and Space Complexity. Fundamenta Informaticae, 49(4)2002, IOS Press, pp. 349–367.
5. M.A.Kłopotek: Minig Bayesian Networks Structure for Large Sets of Variables. In Foundations of Intelligent Systems. LNAI 2366, Springer-Verlag, pp.114–122
6. M.A. Kłopotek On the Distance Hypothesis in Tree-like Bayesian Networks. ICS PAS Report 952, Warszawa, January 2003.
7. A. McCallum, K. Nigam: A comparison of event models for Naive Bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization,1998.
8. A.K. McCallum, K.Nigam: Employing EM in pool-based active learning for text classification, In: Proc.of ICML-98, Morgan Kaufmann, San Francisco, USA, pp. 350–358, 1998,
9. A.K. McCallum, R.Rosenfeld, T.M. Mitchell, A.Y. Ng: Improving text classification by shrinkage in a hierarchy of classes, In: Proc.of ICML-98, , Morgan Kaufmann, San Francisco, USA, pp. 359–367, 1998,
10. Y.Yang: An Evaluation of Statistical Approaches to Text Categorization, J. Information Retrieval, vol, no 1, pp. 69–90, 1999
11. Y. Yang, X. Liu: A re-examination of text categorization methods, in: 22nd Annual International SIGIR, pp. 42–49, Berkley, 1999