DBWorld e-mail classification using a very small corpus

Michele Filannino
Centre for Doctoral Training
The University of Manchester
filannim@cs.man.ac.uk

Abstract

With the growing of the Web, we have huge amounts of texts that could be analysed. Unfortunately, most of them are not immediately usable for our analysis, especially for supervised classification tasks, because they need pre-annotation. Such procedure is often expensive in terms of money and time. The general aim of this paper is to reproduce a case that happens very often in real life: the availability of a small amount of pre-annotated data to train a classifier as much accurate as possible. I have collected a modest set of e-mails[1] from ACM SIGMOD DBWorld[2] and used it to train different types of classifiers in order to discriminate *announcements of conferences* from *everything else*. In this paper, I evaluate the performance of different learning methods in respect to different feature selection techniques. Although it could seem a trivial task, it gives us the chance of drawing conclusions about the nature of different types of classifiers and the contexts in which their use is recommended in Text Mining.

1 Introduction

Machine learning techniques are widely adopted in a number of different natural language tasks. Research communities have been testing new algorithms using larger and larger datasets. While Banko and Brill [1] have definitively proved the importance of having larger corpora to obtain better performances in NLP problems, I have chosen to investigate what happens when we have just a very small dataset. This scenario is more realistic, in so far as collecting huge amounts of data is time and money consuming. Small datasets are currently of great importance especially in the medical diagnosis field, where collecting labelled data is a very hard task. However, studying small datasets is challenging for several reasons. Firstly, the use of this kind of datasets leads to responses with high variance and second, in some cases, small datasets are built ad-hoc for scientific purposes and are far from representing real phenomena.

I have chosen to build a new dataset because my intention is to emphasize one of the most important criteria [11] for a machine learning algorithm: the interpretability of results. My aim is to evaluate performances of different machine learning algorithms using suitable techniques for a small-sized dataset. The algorithms are: *Support Vector Machine* (SVM), *Support Vector Machine Radial Basis Function* (SVM-RBF) with Gaussian kernel, *Decision Tree* and *Bayesian Network*. I present the performance of each classifier using the same quality measure, i.e. the number of correctly classified e-mails, assuming that both types of misclassification (false positives and false negatives) have the same importance.

1.1 DBWorld datasets

DBWorld mailing list announces conferences, jobs, books, software and grants. Publishing new announcements does not require to provide their category. Some writers use to insert specific acronyms in the title (e.g. CFP, CFW, CFE), although it is not a widely shared practice.

I have manually collected the last 64 e-mails that I received and I have built two different datasets. The first one uses only the subjects, while the second one uses bodies. Both datasets have been represented by a term-document matrix using one of the most common data structure in Text mining: *bag of words* [7, 6]. Every e-mail is represented as a vector containing N binary values, where N is the size of the vocabulary extracted [13] from the entire corpus. The binary value is 1 if the corresponding word belongs to the document, 0 otherwise. Features are unique words extracted from the entire corpus with some constraints: words that have more than 3 characters with a max-

[1] Submitted to UCI Machine Learning Repository
[2] http://www.cs.wisc.edu/dbworld/

Datasets	Subjects	Bodies
Features	242	4702
Samples	64	64
$\#Class_{pos}$	29	29
$\#Class_{neg}$	35	35
Sparsity	97.38%	95.73%

Table 1: Dataset characteristics.

Rank	Subjects (#)	Bodies (#)
1	cfp (16)	research (57)
2	position (14)	http (53)
3	call (12)	www (49)
4	university (7)	information (48)
5	data (7)	applications (47)
6	international (6)	systems (45)
7	web (5)	university (44)
8	systems (5)	computer (43)
9	research (5)	science (42)
10	phd (5)	data (37)
11	network (5)	areas (37)
12	management (5)	web (35)
13	faculty (5)	international (33)
14	special (4)	topics (32)
15	social (4)	technology (32)
16	papers (4)	management (32)
17	mining (4)	include (32)
18	issue (4)	computing (31)
19	conference (4)	limited (30)
20	workshop (3)	interest (30)

Table 2: The 20 most frequent words in both datasets.

imum length of 30 characters. Bag-of-words model produces a large number of features, also in the case in which there are few documents. In both datasets, I have also removed stop words. The dataset of subjects has got 242 features while the second one has got 4702 features. Both have 64 samples. Each dataset contains also a further binary feature that indicates the class of each sample: 1 if the sample is an announcement of conference, 0 otherwise. Table 1 summarizes the characteristics of each dataset. They are different not only in terms of number of features, but also in terms of distribution of words. Table 2 shows the 20 most frequent words and their absolute frequencies.

2 Experiment design

The two datasets previously described have been used with each machine learning algorithm in order to measure their performances. During the experimentation, the datasets have not been edited. The factors of my experimentation are the algorithms.

In the case of a dataset with a small number of rows, it is not possible to perform experiments using resampling methods like *K-fold Cross Validation*, as in my case in which 64 samples are not enough. Each sample contains precious and unique information so we have to use the largest possible amount of data to train the classifiers. In these cases one solution could have been that of using the *Leave-One-Out* validation, i.e. using just a sample to test the classifiers previously trained on the rest of samples. The task is performed a number of times equal to the number of total samples, each time with a different one. The disadvantage of this method consists in an over-estimation of measured performances.

An alternative validation method is *Bootstrapping*, which consists of increasing the number of samples by generating new ones with replacement. With this technique, it is possible to use original and generated samples to train classifiers. The test phase is accomplished by using only those original samples that have not been used during the training task. This technique has two important characteristics that it is necessary to take into account:

- the training set contains on average 63.2% of the original dataset so we are ignoring 36.8% of original data: the error estimation is then pessimistic;
- solving the previous problem requires repeating the measurements of accuracy many times and finally to average them.

Currently, the bootstrapping technique is considered the best way to do validation with very small datasets.

I have applied the bootstrapping on both corpora preserving the uniformity of original data and doubling the original number of instances. After this process, I have obtained two new datasets with 128 instances distributed as in the original ones.

3 Experiments

I have carried out 10 repetitions for each classification task and calculated the average for each classification algorithm and dataset.

3.1 Classifiers

As mentioned before, I have used four different classification algorithms. I provide a brief description of each of them below.

SVM A support vector machine is a classifier that manages to maximize the margin among classes and minimize the error function (*quadratic error*) at the same time. This leads to more reliable predictions and less likelihood of overfitting the data. A SVM uses a line to discriminate among classes in a two-dimensional space (hyper-plane in a N-dimensional space).

Using the default parameter of LIBSVM library [2], I have obtained an accuracy of 97.66% with the dataset of subjects, and the same value with the dataset of bodies.

SVM-RBF: Gaussian kernel Not every dataset is linearly separable. A support vector machine could be extended in order to separate also non linearly separable datasets using a little trick. It is possible to define a function, called *kernel function*, that is aimed at increasing the space dimension. The essence of this trick is that some datasets could become separable by using a linear classifier into a new dimensionality. There are different types of kernel functions [5], but the most widely adopted is the *Gaussian kernel function*. The accuracy of this model depends on the parameters γ, which expresses the width of the Gaussian, and C, which expresses the cost. When I used a SVM-RBF, I have always used $C = 1$ and optimised the parameter γ by choosing the most accurate value among 5 fixed ones.

As a result, I have obtained an accuracy of 97.66% for subjects, and 95.31% for bodies.

Decision tree: C4.5 Instead of analysing data to approximate parameters representing geometric areas in the space, it is possible to start from the solution trying to extract sequences of decisions that represent the subtle model necessary to discriminate among classes. One of the most important decision tree classifiers is *C4.5* [10]. It uses the *Information Gain* as criterion to choose which feature most effectively splits data. I have used the algorithm without constraints on the minimum number of examples in each leaf.

The obtained accuracies are 92.19% for subjects, and 96.88% for bodies.

Bayesian Network: K2 algorithm A Bayesian network [8] structure is a directed acyclic graph in which nodes represent features (words in this case), and arcs between nodes represent probabilistic dependencies. Using the conditional probabilities, the network is gradually augmented. One of the most important algorithms used to build a Bayesian network starting from a matrix is *K2* [3], which is based on four important assumptions: data must be discrete; all features occur independently; there are no missing values and the density function is uniform.

Using default parameters, provided by Weka[3], I obtained an accuracy score of 98.44% for subjects, and a score of 95.31% for bodies.

Datasets:	**Subjects**	**Bodies**
SVM	97.6563%	**97.6563%**
SVM-RBF: Gaussian k.	97.6563%	95.3125%
Decision Tree: C4.5	92.1875%	96.8750%
Bayesian Network: K2	**98.4375%**	95.3125%

Table 3: Accuracy of classifiers in the original datasets.

All the results previously mentioned have been summarised in Figure 3. It is easy to notice that the overall performances, aside from the type of classifier, are very high. In both datasets, the linear SVM appears to be a good choice, although Bayesian network seems to have better performances with the subject dataset. SVM-RBF is never better than the linear one. With bodies, the number of features is high and a map to a higher dimensionality is useless. On the other hand, in the case of subjects, it returns the same number of misclassification as the linear one even with a small number of features.

3.2 Feature selection

Most of the complexity of classifier algorithms usually depends on the amount of data that they are given as input. During the training, it is usually assumed that all the data we provide contain important information. This assumption is often false. The choice of the e-mail domain makes understanding it easier. The bag-of-words model generates one new feature for each unique word (or stem) used in the entire corpus. Some of those words are completely useless in order to discriminate announcements of conferences from everything else. Feature selection techniques [4] are used to drastically reduce the number of features,

[3] http://www.cs.waikato.ac.nz/ml/weka/

Datasets	Subjects	Bodies
Features	229	3621
Sparsity	97.23%	95.02%

Table 4: Datasets characteristics after stemming.

maintaining the best possible accuracy at the same time.

In text mining, the most common feature selection techniques are *stop word removal* and *stemming*. The first one consists of removing some words that are so common that have no discriminative power. This operation has already been done as pre-processing activity. Stemming is the process of reducing words to their stems. By using this process, word such as *compute*, *computer* and *computers* all become *comput*. Effectively, this process compresses the number of features preserving the semantic relations among words. I have stemmed both datasets using Porter's stemmer [9] and obtained two new versions of them (see Table 4). The stemming process has two side effects: the reduction of the vocabulary and the reduction of the sparsity in datasets. These determine a change also in terms of the most frequent stems. Table 5 shows the new 20 most frequent stems. As a result, the stemming process has dropped 13 features from subject dataset and 981 from the other one. Performances have remained the same in terms of accuracy (see Table 6).

Starting from the stemmed datasets, I have applied two other techniques of feature selection: *Mutual Information* and *RELIEF*. Both techniques estimate the importance of each feature in respect to the right predictions, by ordering them from the most informative to the less informative. In both cases, I have trained the four classifiers using the first 5, 10, 50 and 100 most informative features (for each dataset, for each feature selection technique). Accuracy results are shown in Figure 1 while Figure 2 shows their graphical representations.

3.3 Evaluation

Regardless of the classifier, in the subject dataset the use of Mutual Information instead of RELIEF seems to provide slightly better results in terms of classification accuracy. It is important to notice that, although there is no reliable statistical difference between methods (*p-value* = 0.0629), the difference in using 5 or 10 features is statistically significative (p-value = 0.000532). In the body dataset it is possible to identify the same behaviour, as there is no statistical difference between Mutual Information and RELIEF algorithm (p-value = 0.7178), this time even for the first 5 or 10 features. The reason is that the number of samples is too small to decide which feature selection method is the best.

It is important to underline that, in the case of subject dataset, by choosing the first 50 most informative features, I have obtained better accuracies than those obtained by using the entire feature set (stemmed or not). This is because the feature selection has deleted some noisy features, an assumption that is confirmed by the fact that SVM-RBF has performed better with 50 features than 100 features.

Rank	Subjects (#)	Bodies (#)
1	posit (17)	research (60)
2	cfp (16)	comput (55)
3	call (12)	applic (55)
4	univers (7)	http (54)
5	research (7)	system (50)
6	data (7)	inform (50)
7	paper (6)	www (49)
8	network (6)	univers (46)
9	manag (6)	includ (45)
10	intern (6)	scienc (42)
11	web (5)	area (42)
12	system (5)	technolog (41)
13	phd (5)	public (40)
14	faculti (5)	interest (40)
15	workshop (4)	experi (38)
16	special (4)	program (37)
17	social (4)	data (37)
18	propos (4)	web (35)
19	mine (4)	topic (35)
20	issu (4)	submit (35)

Table 5: The top 20 most used stems in both datasets.

Datasets	Subjects	Bodies
SVM	97.6563%	**100.00%**
SVM-RBF: Gaussian k.	97.6563%	95.3125%
Decision Tree: C4.5	91.4063%	93.7500%
Bayesian Network: K2	**98.4375%**	96.875%

Table 6: Accuracy of classifiers in the stemmed datasets.

4 Conclusions

In this paper I have applied machine learning supervised algorithms in the domain of DBWorld e-mails to improve the interpretability of used techniques and their results. The use of this domain has been useful to acquire practical knowledge about the charac-

	5	10	50	100
SVM L.	89.06%	88.28%	99.22%	99.22%
SVM-RBF.	89.06%	89.06%	97.66%	96.88%
C4.5	87.50%	89.06%	93.75%	97.66%
B. Net.	89.06%	90.63%	99.22%	99.22%

(a) Subject dataset with Mutual Information

	5	10	50	100
SVM L.	84.36%	86.72%	95.31%	99.22%
SVM-RBF.	84.36%	86.72%	94.53%	99.22%
C4.5	84.36%	86.72%	89.84%	91.41%
B. Net.	84.36%	86.72%	96.09%	95.31%

(b) Subject dataset with RELIEF

	5	10	50	100
SVM L.	88.28%	92.19%	99.22%	99.22%
SVM-RBF.	91.41%	92.19%	96.88%	96.88%
C4.5	85.94%	90.63%	93.75%	96.88%
B. Net.	91.41%	92.97%	92.19%	92.99%

(c) Body dataset with Mutual Information

	5	10	50	100
SVM L.	89.84%	92.97%	99.22%	99.22%
SVM-RBF.	90.63%	93.75%	96.88%	96.88%
C4.5	86.72%	92.19%	92.97%	96.09%
B. Net.	91.41%	92.97%	92.97%	92.97%

(d) Body dataset with RELIEF

Figure 1: **Feature selection results**: Subject dataset using Mutual Information (a) and RELIEF (b). Body dataset using Mutual Information (c) and RELIEF (d).

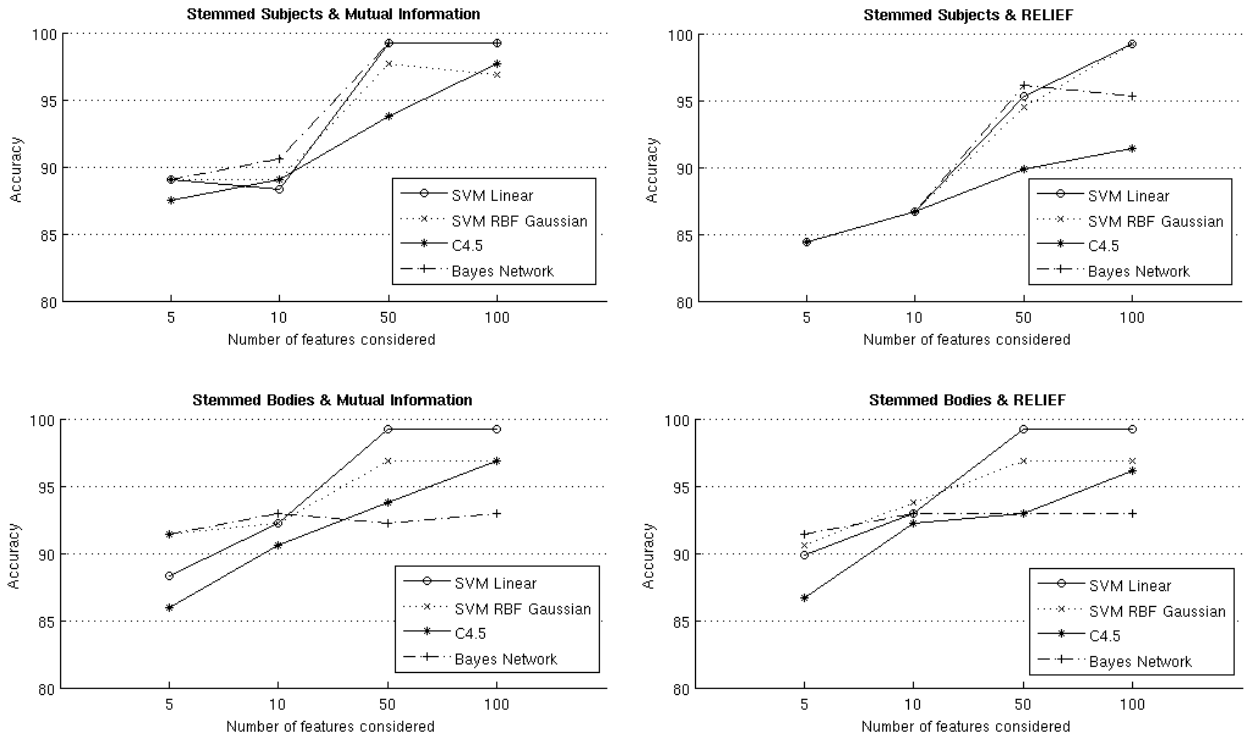

Figure 2: **Feature selection graphical results**: Subject and body datasets, using the most informative features computed with Mutual Information and RELIEF. For the subject dataset, RELIEF seems to provide a worse feature selection. The opposite behaviour is shown in the case of the body dataset.

teristics of some specific feature selection techniques, thus having the possibility of choosing the best one in some common context. I have shown that, regardless of the type of classification algorithm, it is possible to maintain excellent performances even when the number of features taken into account is drastically reduced. This is possible because of the strong sparsity of datasets, a characteristic induced by the use of the bag-of-words representation. The greater the number of features, the higher the probability of reducing them and obtaining a very small error. In the case of the body dataset, by a reduction of 98.94% in the number of features, the classification error has increased of only 0.78%, whereas in the case of subjects, by a reduction of 79.34% in the number of features, the classification error has increased of 1.56%.

For what concerns the number of features, when it is small (5 or 10), performances of SVM-RBF are generally better than those of the linear SVM, whereas for a larger one (50 and more), the results are worse or equal, never better. This behaviour suggests that using a SVM-RBF with a bag-of-words representation might not be a good idea. This representation produces a huge number of features (one for each word/stem in the entire corpus). The addition of others via kernel function has very few chances of introducing new useful information. This result is in line with what had already been proved before [5]. In my experiment I have obtained the best results using a linear SVM. However, its use has not to become an imperative rule, at most a reasonable recommendation [12].

4.1 Future works

Google Mail rules do not make the automatic grab of your own e-mails easy: API changes very frequently and finding a proper library in any programming language seems an hard task. This is one of the reasons for the modest dimension of my dataset, which leads to the lack of statistical significance of my results. Having said that, an interesting proposal for future work might be of repeating the experimentation using a larger number of e-mails, maintaining the same experiment design.

References

[1] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[3] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9:309–347, October 1992.

[4] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, Mar. 2003.

[5] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2000.

[6] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, 1999.

[7] T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. WCB/McGraw-Hill, Boston, MA, 1997.

[8] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, Aug. 1985.

[9] M. F. Porter. An algorithm for suffix stripping. In K. Sparck Jones and P. Willett, editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[10] J. R. Quinlan. C4.5: Programs for machine learning. *Machine Learning*, 16:235–240, 1993. 10.1007/BF00993309.

[11] P. Turney. Types of cost in inductive concept learning. In *Seventeenth International Conference on Machine Learning*, 2000.

[12] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computing*, 8:1341–1390, October 1996.

[13] D. Zeimpekis and E. Gallopoulos. Design of a matlab toolbox for term-document matrix generation. Technical report, Computer Engineering & Informatics Dept., University of Patras, Patras, Greece, 2005.