Classifying Financial News With Neural Networks

Rafael A. Calvo

Department of Electrical and Information Engineering
The University of Sydney
Bldg J03, Sydney NSW 2006
rafa@sedal.usyd.edu.au

Abstract

One of the biggest challenges facing financial research and trading organizations is how to well exploit unstructured financial information such as textual announcements. The automatic classification of this type of data poses many challenges for learning systems because the feature vector used to represent a document must capture some of the complex semantics of natural language.

In this paper we discuss the use of neural networks in the classification of financial news. The input dimensionality has been reduced using the χ^2 *statistic. Moreover, we have tried several types of averaging showing that this selection significantly affects the micro and macro average performance. The generalization was controlled using a number of cross-validation sets, one for each experiment.*

Keywords Information retrieval, document classification, neural networks

1 Introduction

Financial analysts, traders and government agencies regulating the financial markets must process thousands of daily announcements. These anouncements are sometimes relevant to their particular area of interest, have market-wide impact, or will have an impact in the value of a particular share. Thus, analysts and traders are often searching for the proverbial needle in a haystack, reading through corporate announcements, press releases and even searching for potentially useful “gossip” in discussion forums and chat rooms. Meanwhile, government agencies are trying to find signs of market manipulation, such as an individual or organization trying to change the value of a share by releasing false information. They are also sniffing out insider trading operations, where someone trades public stocks based on information they are not supoused to have. In addition, there is the search for uncertifed advice where someone gives financial guidance without having the certification to do so.

All of the scenarios mentioned above are appropriate for the use of automatic document classification techniques, such as the ones used by a number of authors (see Yang [9], Manning [11] and Croft [4].)

Section 2 of this paper introduces the standard definitions used in the document classification bibliography. In section 3 we discuss the particular application, the data set, and the performance measures. In section 4 we describe the results, and in section 5, the conclusions and comments on future work.

2. Vector models and Classification.

Vector models have been successful in information retrieval (IR) and text classification (TC) (Salton [7], Salton [8]). Vector models are based on the basic assumption that a document can be represented as a vector, dismissing the order of words and other grammatical issues, and that this representation is able to retain enough useful information.

In order to reduce the number of distinct terms in the database, a list of stopwords is removed from the documents. This technique is common in most languages, where some words have low information content. In English articles and prepositions are some of them, and removing 100-300 words, reduces the document length by 30-40%. Another frequent preprocessing is stemming, where words in different tenses and singular/plural are reduced to one term (stem).

Vector models need the vector representation of the documents, and this is done using several weighting schemes. The term a_{ij} of matrix **A** is the weighted value for term *j* of document *i*. Some common weighting schemes are:

Term Frequency (TF) weighting

$a_{ij} = TF_{ij} =$ Times term *j* app appears in document *i*

Inverse Document Frequency (IDF) weighting

$a_{ij} = IDF_{ij} = \log(\frac{N}{O_j}) + 1$, where N is the number of documents in the collection and O_j is the number of docs with term j.

Finally, the products $f(TF_{ij}) \times g(IDF_{ij})$ are also used (Salton [6]). In this work we will describe results using $1 + \ln(TF)$ as the TF factor and the IDF factor defined above.

The dimension of these vector spaces is proportional to the number of terms remaining after stopword removal and stemming. Even for moderate-size text collections this can be tens or hundreds of thousands of terms. This is prohibitively high for some algorithms so dimensionality reduction techniques are needed.

	C present	C absent	
T present	p_{tc}	$p_{t\bar{c}}$	p_t
T absent	$p_{\bar{t}c}$	$p_{\bar{t}\bar{c}}$	$p_{\bar{t}}$
	P_c	$p_{\bar{c}}$	

Table1. Term – Category contingency table

The χ^2 statistic can be used to measure the dependence of class *c* on the occurrence of term *t*. This feature makes it useful for dimensionality reduction. Based on Table 1 we can write:

$$\chi^2 = \frac{(p_{tc} - e_{tc})^2}{e_{tc}} + \frac{(p_{\bar{t}c} - e_{\bar{t}c})^2}{e_{\bar{t}c}} + \frac{(p_{t\bar{c}} - e_{t\bar{c}})^2}{e_{t\bar{c}}} + \frac{(p_{\bar{t}\bar{c}} - e_{\bar{t}\bar{c}})^2}{e_{\bar{t}\bar{c}}}$$

So, the χ^2 has a value of 0 if *t* and *c* are independent.

For each class c_i the χ^2 statistic of term *j*, $\chi^2(t_j, c_i)$ was computed. These values can be combined in several scores, for instance:

$$\chi^2{}_{avg}(t) = \sum_{i=1}^{n} \Pr(c_i)\chi^2(t_j, c_i)$$

where $\Pr(c_i) = \frac{freq(c_i)}{N}$ is the probability of c_i.

$$\chi^2{}_{max}(t) = \max_{i=1}^{n} \chi^2(t_j, c_i)$$

and

$$\chi^2{}_{max \times \Pr}(t) = \max_{i=1}^{n} \Pr(c_i)\chi^2(t_j, c_i)$$

These scores can be used to produce a term ranking table, where highly informative terms are on top of the list, and the last ones will be removed from it. The number of terms to be used will determine the number of inputs to the neural network so the computational costs must be considered. The more terms we keep the higher the probability of retaining non-informative terms that will introduce noise to the learning process.

The underlying assumption in using the χ^2 statistic is that features whose appearence in a document is highly correlated to a class membership will be useful for measuring class membership. The selection of the averaging procedure $\chi^2{}_{avg}(t)$ or $\chi^2{}_{max}(t)$ will affect the performance. If the average includes a $\Pr(c_i)$ weighting the *max* or *avg* will weight classes differently, giving equal weight to every document.

Yang [10] compared different dimensionality reduction (a.k.a. feature selection) techniques, finding that Information Gain (IG) and χ^2 were most effective. They did not specify what averaging of χ^2 they followed, and as we see in the results section this can be an important factor for some applications. These feature selection methods remove non-informative terms according to their statistics. Schutze [8] used mutual information and χ^2 statistic to select features in a neural network architecture. We used the χ^2 statistic with the 3 different averaging schemes defined above. The results obtained are very usefull and show how the selecting the weighting scheme is important and varies the performance in different applications.

Neural Networks have proven useful in many classification tasks (Ripley [5]). Several authors have recently provided results of neural networks applied to document classification. Wiener [1], Ng [3] and Yang [9] have reported on the Reuters-21450 dataset. Ng et al. uses only a 1 layer perceptron, Wiener et al. also tried a 3 layer network, both systems use one network for classifying each class.

Yang et al. approach is similar to ours using one network for all the 90 categories, but did not report the use of cross-validation or other complexity optimization techniques. In their comparison they did not consider that the precision and recall obtained are influenced by the uncertainty of the learning process, so using one single experiment is not enough. An interesting result with neural networks is their high precision performance compared with most other methods.

3. The problem, the data set and the performance measures

Neural Networks and document classification

Neural networks can learn nonlinear mappings from a set of training patterns. The parameters are adjusted to the training data in a number of iterations. We have used a backpropagation algorithm that minimizes quadratic error. The input layer has as many units (neurons) as features retained. The number of hidden units is determined by optimizing the performance on a cross-validation set. There is one output unit for each possible class, with activation values between 0 and 1. If one (or more) of the output units has activation greater than 0.5, the classifier will state that the document belongs to that (those) categories. Hertz [2] and Ripley [5] give an introduction to neural networks that goes out of the scope of this paper.

Yang [9] has performed an evaluation of different statistical approaches applied to document classification, including k-nearest neighbors (kNN), linear least square fit (LLSF), support vector machines (SVM), Naive Bayes (NB) and neural networks (NNet). Calvo [12] has compared this results with neural networks in English and Spanish corpus.

The Reuters document collection

Plenty financial news are available on the web, but it is hard to find in standard benchmark sets for document classification, where each method can be tested and its performance compared reliably with other methods. The Reuters sets are a notable exception. Although different versions are available, many researchers use it for benchmarking. We will use the ApteMod version of Reuters-21450. The ApteMod set has 7769 documents for training and 3019 for testing, 24240 unique terms after stemming and stop word removal and an average of 1.3 categories per document, with a total of 90 categories that occur in both sets.

The data set is composed of news form the Reuters cable produced in 1987. Each document has a document ID, one or more classes that it belongs to (related industry or type of the news), a title and the main text. An example is shown below.

```
<Document>
  <DID>9</DID>
  <CLASS>earn</CLASS>
  <TITLE>CHAMPION PRODUCTS <CP>
APPROVES STOCK SPLIT </TITLE>
<TEXT>Champion Products Inc said its
board of directors approved a two-
for-one stock split of its common
shares for shareholders of record as
of April 1, 1987. The company also
said its board voted to recommend to
shareholders at the annual meeting
April 23 an increase in the
authorized capital stock from five
mln to 25 mln shares. </TEXT>
</DOCUMENT>
```

Figure 1: Example of a document from the Reuters corpus

The news have a title and a content section, and we have used both indistinctly. From all the distinct terms in the training set we tested different thresholds for the χ^2 statistic so only the top 500 & 1000 terms were retained.

Performance measures

Table 2 describes the possible outcomes of a binary classifier. The system YES/NO results refer to the classifier output and the real YES/NO refers to what the correct output is. The perfect classifier would have a value of 1 for a_j and d_j, and 0 for b_j and c_j.
Using table 2 we define 3 performance measures common in the document classification literature.

$$recall = r = \frac{a}{a+c}$$ if a+c >0, else =1

where $\frac{a}{a+c}$ is the number of classes found over total classes correct.

$$precision = p = \frac{a}{a+b}$$ if a+b>0, else=1.

where $\frac{a}{a+b}$ is the number of classes found and correct over total classes found.

The tradeoff between recall and precision is controlled by setting the classifiers parameters. Both values should be provided to describe the performance. Another common performance measure is the F-measure

$$F_\beta(r,p) = \frac{(\beta^2+1)pr}{\beta^2 p + r}$$

the most commonly used F-measure in document classification is F_1:

$$F_1(r,p) = \frac{2pr}{p+r}$$

When dealing with multiple classes there are two possible ways of averaging these measures, namely, **macro-average** and **micro-average**. In the macro-averaging, one contingency table as Table 2 per class is used, the performance measures are computed on each of them and then averaged. In micro-averaging only one contingency table is used, an average of all the classes is computed for each cell and the performance measures are obtained therein. The macro-average weights equally all the classes, regardless of how many documents belong to it. The micro-average weights equally all the documents, thus favoring the performance on common classes.

Different classifiers will perform different in common and rare categories. Learning algorithms are trained more often on more populated classes thus risking local overfitting.

	Correct	
Assigned	YES	NO
YES NO	a_j c_j	b_j d

Table 2. Contingency table for class *j*

Results

We summarize here the steps followed in the preparation of the experiments, and the results therein.

1. **Preparing the data**: A list of 571 stopwords was removed from the document collection. A standard stemming algorithm was also used.

2. **Dimensionality reduction**: The three χ^2 defined above were compared. Since $\chi^2{}_{avg}(t)$ and $\chi^2{}_{avg\times\Pr(t)}$ include the probability $\Pr(c_i)$ the top features will be accounting for the common classes, this means that all documents are effectively weighted the same, this results in a higher micro average. $\chi^2{}_{max}(t)$ does not have this term so the maximization is independent on the probability of the class. This kind of feature selection should produce better macro-averages. Table 3 shows the maF_1 and miF_1. For a comprehensive comparison we trained 20 different networks, starting with different random weights and cross-validation sets. The 3 averaging schemes produce micro and macro averages different with statistical significance (α=0.05). This result should be exploited when deploying a particular application. If we need the best performance possible on classes with very few examples $\chi^2{}_{max}(t)$ should be used. For example, when we need a set of classes with similar number of documents but our tagged corpus has more documents in some classes than other.

	$\chi^2{}_{max}(t)$	$\chi^2{}_{max}\times\Pr(t)$	$\chi^2{}_{avg}(t)$
Macro F_1	.269	0.244	.265
Micro F_1	.805	.833	.835

Table 3. χ^2 performance on 20 runs of nets with 500 input and 50 hidden unit (test set)

The results in tables 4 and 4 were obtained using $\chi^2{}_{max\times\Pr(t)}$.

1. **Vectorization and weighting**: The resulting documents were represented as vectors, using TFIDF weighting. Other weighting schemes were tried and this was found optimum. The Smart text processing package (Salton [6]) was used for this stage. All the experiments described use a cosine normalization.
2. **Network architecture**: The selected terms were used as input features to the network. We tried networks with 500 and 1000 input features. The network has 90 output units, one for each class. Tables 4 and 5 display the of MiF_1 and MaF_1 over for networks with 50, 100 and 150 hidden units.
3. **Sampling**: We tried the sampling scheme described above. The optimum number for the sampling threshold and for the number of hidden units are determined seeking a minimum error on the average for the cross-validation sets. Probably due to the size of the database and that we have only 1.3 categories per document this sampling was not very efficient. For tables 4 and 5 we did not use any sampling.
4. **Training**: We generated 5 cross-validation sets with a number of random documents each. These documents were set aside and the network was trained on the remaining ones. As a rule of thumb is common to use 5-10% of the training set, we tried using 500 and 1000 documents, the smaller cross-validation set resulted in better performance. The learning rate η was reduced or the training stopped ($\eta \le \frac{\eta_0}{4}$) when the error found a minimum on the cross-validation set. Reducing η during learning is a commonly used technique, similar to ``cooling down'' in simulated annealing. In table 5 we show the Micro and Macro average F_1 performance for networks with 500 and 1000 input features. In both cases the best results are obtained for 100 hidden units. In table 4 the results obtained by other authors and us is displayed. We show in this table the best results for 500, and 1000 inputs.

	MiR	MiP	MiF_1	MaF_1
SVM	.914	.812	.860	.513
KNN	.834	.881	.857	.524
NB	.769	.825	.796	.389
NN 500inp	.788	.911	.845	.282
NN 1000inp	.800	.908	.853	.326

Table 4. Performance of different classifiers

Conclusions

In this paper we have applied neural networks to the classification of financial news. The Reuter newswire corpus is one of the most popular benchmark sets in the text classification literature. We compare the performance of the classifier with the results of other authors that used SVM, kNN and NB classifiers.

	50 hidden	100 hidden	150 hidden
Macro F_1- 500 inputs	.263	.288	.281
Micro F_1- 500 inputs	.841	.845	.845
Macro F_1- 500 inputs	.295	.321	.306
Micro F_1- 500 inputs	.849	.853	.851

Table 5. Performance of the network with 500 and 1000 input features. The average for 5 different run is shown.

Neural Networks perform well compared to other methods. Together with Support Vector Machines and k-nearest neighbors they produce the best overall models for this data set. These techniques should make possible automatic classification systems with a performance comparable to human classification.

The micro averaged precision is the best for all the methods compared. The F_1 performance, in particular maF_1 is not as good as good as with SVM or kNN, this is due to a poor recall level. The tradeoff between high recall or high precision must be considered for each application.

The dimension of the vector models for the documents is very high, this makes the classification problem untractable until the dimension is reduced. The χ^2 statistic is successfully used to reduce the input dimension. We have compared three averaging schemes for the χ^2 described above. It is shown that some averaging schemes improve maF_1 and other miF_1. Therefore the kind of averaging should be considered for each given application. We have shown that the 3 schemes perform statistically different (α=0.05.)

Neural networks performance has a degree of randomness, inherent to the learning process and by distinct initial conditions, so it is not posible to determine the method performance by a single experiment. Several trainings are needed, preferably with different cross-validation sets. We have used 5 to 20 independent experiments to determine each value.

Future work includes trying these methods on corporate anouncements from the Australian Stock Exchange (ASX).

Acknowledgments

The author acknowledges partial support by the Language Technology Institute of Carnegie Mellon University, and Instituto de Fisica at Universidad Nacional de Rosario.

References

[1] E. Wiener, J.P., & Weigend., A. (1995). A neural network approach to topic spotting. *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval* (SDAIR'95)

[2] Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation.* Redwood, CA: Addison-Wesley.

[3] H.T. Ng, W. G., & Low, K. (1997). Feature selection, perceptron learning, and usability case study for text categorization. ACM SIGIR *Conference on Research and Development in Information Retrieval* (SIGIR'97)} (pp.\/ 67--73).

[4] Croft, W.B. Editor. *Advances in Information Retrieval.* Kluwer Academic 2000.

[5] Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge Univeristy Press.

[6] Salton, G. (1989). *Automatic text processing: The transformation, analysis and retrieval of information by computer*. Reading, MA: Addison-Wesley.

[7] Salton, G. (1991). Developments in automatic text retrieval. *Science*, **253**, 974--979.

[8] Schutze, H., Hull, D., & J.O., P. (1995). A comparison of classifiers and document representations for the routing problem. Int. ACM SIGIR *Conference on Research and Development in Information Retrieval* (pp.22--34).

[9] Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. ACM SIGIR *Conference on Research and Development in Information Retrieval* (SIGIR'99)}

[10] Yang, Y., & Pedersen, J. (1997).Feature selection in statistical learning of text categorization.*The Fourteenth International Conference on Machine Learning* (pp.\/ 412--420).

[11] Manning C and H. Schutze. *Foundations of Statitical Natural Language Processing.* MIT Press 1999.

[12] Calvo R. A. and H.A. Ceccatto (2000). *Intelligent document classification* in Intelligent data Analysis, 4(5)