Praveen Kumar Bodigutla
12th December 2009

Abstract
The problem addressed is a sentiment analysis and text classification problem. The idea is to see if the 10K forms filed by companies over the years have any predictive power in forecasting the movement of the stock market. The report presents a new document-labeling scheme based on the performance of a company's stock. Along with presenting various feature selection and dimension reduction techniques, the report discusses the results obtained by applying Supervised classification models – SVM and Naïve Bayes on the selected features. I also use Unsupervised-learning models – Gaussian Mixture Models and K-means, and discuss the motivation behind using them. In the Conclusions section I summarize the results of using various models on the data, and discuss the challenges posed in mining sentiment and classifying the 10K forms. I conclude the report by mentioning the scope of future work. Work presented in this report is also a part of my current quarter's research under Prof. Christopher Manning's supervision.

Introduction
Form 10-K is an annual report required by U.S Securities Exchange Commission (SEC), that gives a summary of company's performance. The motivation for the project comes from the ideas that the [1] timing and lag between the end of fiscal year and the time when 10K forms are filed, leads to under reaction from the investors and [2] the qualitative information, which is harder to process as compared to quantitative information, has greater predictability for returns at longer horizons, suggesting that frictions in information processing generate price drift. Hence there is a value in mining sentiment from the 10K forms, which if mined could provide insights into the performance of a firm's equity in the future.

Assuming that the documents have an impact on the performance of the company and hence the movement of the firm's equity, I try to mine the sentiment of a document and then classify it as belonging to the positive or negative class. Documents belonging to the positive class have a positive impact on the stock and the negative documents have the contrary impact. I used machine-learning techniques to address this classification problem. Since the data is unlabeled and in order to use supervised learning models it is important to have a pre-labeled set of documents. I designed a labeling approach based on the assumption relating the sentiment of document and the movement of firm's stock price.
Along with applying the supervised learning models, I try to explore the inherent structure in the data to further use unsupervised learning models to classify the documents. Typical of the text classification problems, the dimensions of the feature space are much larger than the number of documents. To address this problem I used a variety of dimension reduction techniques, which are the Chi-square feature selection criterion, factor analysis, PCA, stemming and choosing a restricted set of sentiment conveying words. Then on the reduced dimension space of features, I apply the unsupervised and supervised models to see if a successful classification of the documents is obtained.

Data for the project
For the purpose of this project I chose the 10K Forms of the companies constituting the S&P 500 index. I obtained the historical stock prices from Yahoo! Finance and got the stock ticker information from finviz.com. Edgar is the source from which the 10K forms were obtained. Edgars makes these forms along with financial statements available to public. The data is highly unstructured with embedded tables, pictures and is also encoded in HTML, SGML, uuencode formats. The overall size of the forms in compressed format on disk is around 20 GB. A good amount of time was spent on retrieving relevant information from the documents. The sections of the form, from which data is extracted are - Section 1A in which the risk factors including future failures to meet obligations are mentioned, Section3- Legal Proceedings, which talk about an ongoing pending law suits and Section 7 which is the section on Management's Discussion and Analysis of Financial condition & result of operations. This section also gives the quantitative and qualitative disclosures about market risks.
Other than removing tables, HTML tags and other unwanted information, In order to get information on these sections alone I used some rules to extract the content based on the title of the sections, the location of these sections in the document and the size of the content extracted. With these heuristics I could extract information from 95% of the forms. This

ensured that I have as much data as possible, but because the extraction is based on rules assuming some structure to the document, the data extracted does not always exactly represent the content in the sections of interest.

Labeling & Dimensionality Reduction

As mentioned in the previous sections, due to the absence of labeled training set, I assume a cause and effect relationship between the performance of a company's equity and the sentiment mined from 10K document. Hence the labeling scheme is derived from the firm's performance in the stock market. After a 10K document is filed if the stock performs well as compared to the index (measure in terms of excess geometric mean return), for a period of one quarter [2], then that document belongs to the positive class otherwise it belongs to the negative class. The plot for the excess mean return of firm is shown in **Fig.1**, as expected there are not many under/over performers (the documents in the tail). The returns here are continuously compounded.

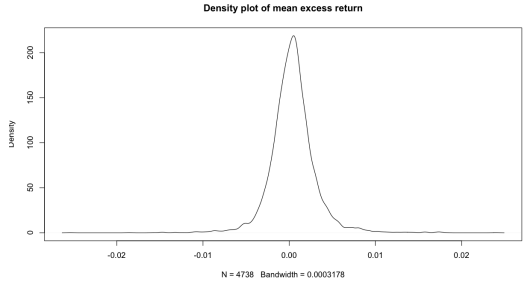

Fig.1 – Excess mean return relative to S&P 500

Along with the excess mean returns, I tried using another labeling scheme based on the mean returns of the company alone, without comparing it with the mean returns of S&P 500 Index (^GSPC). In this labeling scheme, a document is classified as a positive document if the company performs well in one quarter after the document is filed irrespective of how the market performs (log return of the firm > 0). For the training set, I considered forms of firms whose stock price deviated from the mean of the plot by one standard deviation, later to reduce over fitting and to increase the size of the sample, I reduced the range to - half of standard deviation from the mean. The results obtained are mentioned in the coming sections.

Dimensionality Reduction

In the training set the dimensionality of features is very high as compared to the number of documents in training set (n >> m). For a particular training set the feature dimension was around 50,000 and the number of documents less than 1000. This will lead to potentially over fitting when we use supervised learning models. Among various dimensionality reduction techniques [3], I used the Chi-square method, which measures the independence relation between the features (words) and the categories (positive and negative document classes). After applying the Chi-square feature selection technique on the data with stop words removed, the dimensionality of the features reduced by a factor of 100. Some words which the feature selection criterion found out as useful for classification are – *moving, gordon, formulated, coupon, complements, secured, cessation, digitized, permits, allow, withstand, emerged, maturity, achievable, misappropriation.* A sample from the words, which were discarded include – *biology, Geneva, maturing, design, export, defendant, contamination, filtration, personalized.* A glance at the words suggests that the words with some significance towards predicting sentiment are present in feature set.

Feature Selection and Supervised Learning

Once the tokens based on chi-square scores are obtained, the next step is to design features from the tokens. The features I tried using are the word frequency, normalized TFIDF scores of the tokens (words) in the documents, TFIDF scores of the stemmed words, TFIDF scores of the words which convey sentiment [4] and Binary vector of features indicating their presence or absence in the document [5]. As expected the features based on just token frequency performed very poorly. Better predictions were obtained by taking the binary feature vector. As mentioned before the labeling scheme is based on the excess mean returns or the mean returns of the company. The range of dates in the training and test sets show the dates between which the documents were obtained for training and testing respectively. The sentiment and in turn the performance of the company is predicted for a period of one quarter from day its 10K form is filed [2]. For an SVM classifier - C, which is the training error vs. margin trade off and the order of the polynomial kernel were obtained by minimizing the one fold cross validation error. Due to over fitting, the cross validation error was minimum when the slack variable coefficient C was above 50 and the order of polynomial was below 3 in most of the cases. Considering data from all-important sections (1,7 and 3) of the form 10K gives better results than choosing data from one of the three sections. The results obtained after running the SVM classifier [6] on the different features is mentioned in **Tables 1, 2, 3 and 4**.

From the accuracy, precision and recall values

given in the tables, it is clear that choosing binary vector of tokens and labeling using mean returns of the company, gives better results than taking normalized TFIDF of tokens and labeling using excess mean returns. This also implies that the presence or absence of a token in a document is more important than the word frequency. Stemming the tokens doesn't give much improvement in the results.

Train	Test	Accuracy	Precision	Recall
96-00	00-02	0.50	0.48	0.85
96-02	02-04	0.57	0.57	1
96-04	04-06	0.47	0.48	0.98
96-06	06-08	0.74	1	.74
96-08	08-09	0.61	0.63	0.91

Table1:Features:Normalized TFIDF,Labeling:Mean return

Train	Test	Accuracy	Precision	Recall
96-00	00-02	0.56	0.53	0.69
96-02	02-04	0.55	0.55	0.99
96-04	04-06	0.48	0.47	0.60
96-06	06-08	0.53	0.52	0.61
96-08	08-09	0.54	0.55	0.60

Table 2: Features:Normalized TFIDF,Labeling:Excess mean returns

Train	Test	Accuracy	Precision	Recall
96-00	00-02	0.62	0.58	0.71
96-02	02-04	0.55	0.58	0.80
96-04	04-06	0.55	0.53	0.55
96-06	06-08	0.78	1	0.78
96-08	08-09	0.61	0.71	0.60

Table 3: Features: Binary Vector of tokens, Labeling: ,Mean Returns

Train	Test	Accuracy	Precision	Recall
96-00	00-02	0.66	0.62	0.72
96-02	02-04	0.55	0.60	0.64
96-04	04-06	0.54	0.51	0.81
96-06	06-08	0.52	1	0.53
96-08	08-09	0.56	0.64	0.71

Table 4: Features: Binary Vector of stemmed tokens, Labeling: Mean Returns

Another supervised classifier I used is the Naïve Bayes logistic classifier. Being a probabilistic model, Naïve Bayes model gives a good idea on how the documents' tokens are distributed among the two categories. The results obtained by using Naïve Bayes model are mentioned in **Table5**. Due to space constraint I am not adding the results obtained on various features, instead in **Table5** I present the results obtained by training the model on binary valued features an the labeling scheme is based on mean returns.

By running Naïve Bayes classifier with various training sets, the conclusion I derived about the lexicon of the positive and the negative class of documents is that the distribution of tokens across both classes is not very different. For example on a particular test set where all the documents belonged to the positive class the accuracy obtained was close to 100% and the number of positive documents in the training set was marginally higher than the number of negative documents. If I reduced the number of positive documents and made the count lower than that of the negative documents then the accuracy dropped significantly. Poor performance of Naïve Bayes classifier also indicates a possible correlation between the features, which I confirm in later sections. Overall the SVM classifier performs better than the Naïve Bayes classifier.

Train	Test	Accuracy	Precision	Recall
96-02	02-04	0.58	0.61	0.75
96-04	04-06	0.47	0.46	0.56
96-06	06-08	0.66	1	0.66
96-08	08-09	0.58	0.61	0.75

Table5 , Features: Binary vector of tokens, Labeling: Mean returns

Finally as features, I took the words mentioned in Harvard general inquirer [4] and considered as features only those words which conveyed sentiment. This gave very poor results because if we plot the number of positive words in a document against the number of negative words, their ratio is not significant enough to classify the document as conveying positive or negative sentiment **(Fig 2)**

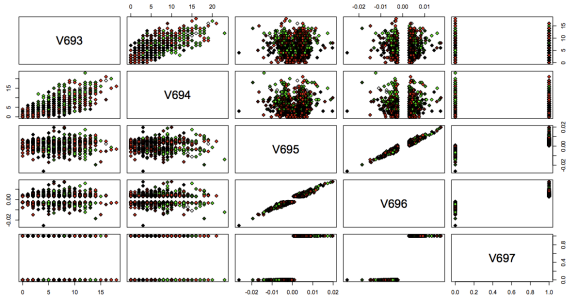

Fig 2. Scatter Plot of positive word count, negative word count, mean returns, excess mean returns and labels against each other in the same order.

On the data set that was obtained after applying the chi-square dimensionality reduction criteria, I tried applying PCA on the documents

obtained between 96-08 to further reduce the feature dimensions and analyze the feature space. 90% of variance in the data was explained by 230 principal components. **Fig 3** shows the scatter plot of the projection of features on the first 3 components. The plot clearly shows that there is no clear separation between the examples belonging to the two classes.

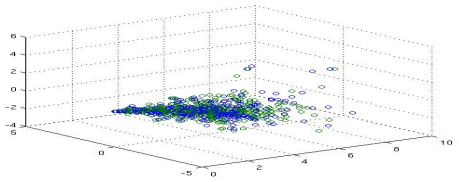

Fig 3. Projection of features onto first 3 Components, labels indicated by different colors.

A linear classifier will not perform well on classifying such data set. The next plot (Fig 4) is the correlation plot of (binary valued) features in the data; this plot indicates that there is some correlation between features. This indicates possibility of natural clusters existing in the data. This also provides motivation for using

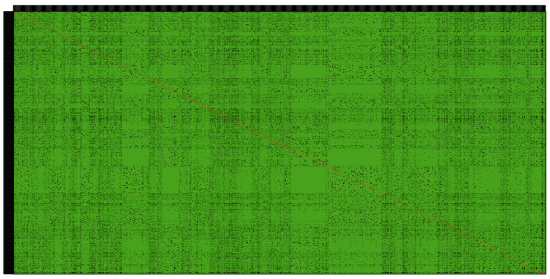

Fig 4: Correlation matrix plot indicating correlation among features (binary valued)

Unsupervised learning models to classify the documents.

Unsupervised Learning – Kmeans and GMM

I applied kmeans to cluster the forms using the minimum sum of square error (SSE) criteria (for choosing the number of clusters). **Fig.5** shows how the SSE dropped as the number of clusters increased. There is no elbow like pattern to choose the ideal number of clusters. Looking at the marginal improvement in SSE (**Fig 6**), I chose the number of clusters to be the point where the marginal improvement in SSE drops below a threshold. **Fig.2** (the colors on the plot) shows the non-linear separation after using kmeans clustering .

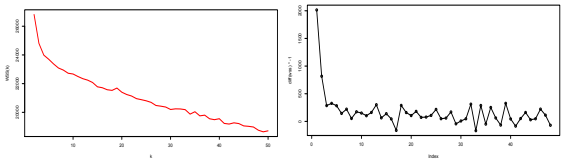

Fig 5 (left) – SSE against number of clusters, Fig 6 (right) - Difference in the SSE against number of clusters

Once the clusters are obtained from the training set, the class for one particular cluster is assigned based on majority vote. A new test document is assigned to the cluster based on its proximity to cluster centroid. Then the label given to the test document is based on the cluster to which it belongs. **Table 6** summarizes the results obtained after using kmeans on the documents from various time intervals. **Fig7** shows the size of kmeans clusters

Training	Test	+ve	-ve	Accuracy
96-00	00-02	143	0	0.53
96-02	02-04	190	28	0.56
96-04	04-06	213	99	0.47
96-06	06-08	180	140	0.57
96-08	08-09	151	20	0.64

Table 6: Number of +ve and –ve labels assigned on test documents and the accuracy obtained by using Kmeans. (Labeling : Mean returns.)

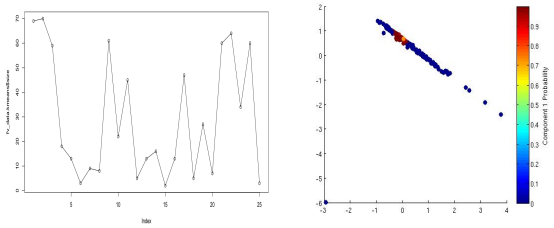

Fig 7. Size of Kmeans clusters (Training 96-08)
Fig 8. Posterior probability of Factors (GMM).

Fig7 also shows that we don't have well separated data between the two classes, as there are no clear dominant clusters where most of the examples lie. The next unsupervised learning model I tried is the Gaussian Mixture Models. The data being sparse binary matrix and the number of examples being almost same as the number of features; the covariance matrix was almost always turning out to be singular. To reduce the dimensions in the feature space, I used factor analysis and got the factors using EM. The results obtained are listed in **Table 7**. From the table it is clear that the data forms dense clusters and GMM fits a

Table 7.Clustering based on GMM, C(train),C(test) indicates number of clusters in training and test set resp.

Training	Test	C(Train)	C(test)	Accuracy
96-00	00-02	2	1	0.47
96-02	02-04	3	1	0.57
96-04	04-06	5	2	0.48
96-06	06-08	12	3	0.50
96-08	08-09	15	5	0.63

Gaussian distribution over a range of values which fall in that dense region, hence it does poorly as compared to all other models tried so far. Fig 8 is the plot of the first two factors obtained from factor analysis using EM algorithm. The color of the plot shows the posterior probability of the components and single color across the data points shows that there was one dominant cluster and the clustering technique by assuming Gaussian distribution yields poor prediction.

Conclusions

* The challenges posed by ill formatted data restrict the amount of relevant content that can be extracted from the documents.
* Chi-square criterion for feature selection is better than using PCA or SVD based methods to reduce dimensionality, as the examples belonging to the two classes are not well separated when projected onto Eigen vectors.
* Considering data from sections 1,7 and 3 together is better than considering data from a particular section alone.
* Between two supervised learning models tried, SVM scores better over Naïve Bayes classifier.
* Binary features better than normalized TFIDF score, presence of a word matters more than its frequency. Using stemmed words does not give significant improvement in the prediction.
* Using only the sentiment conveying words to classify documents is not useful, as the proportion of both positive to negative words in the data does not vary much.
* Prediction seems to improve by considering mean returns for labeling than excess mean returns.
* Using SVM to classify, and Kmeans to cluster the documents gives results, which are better than random guessing.
* Problem of over fitting addressed by taking more documents to train on and by adjusting the degree of the polynomial kernel, and C (Trade off between training error and margin) by minimizing the 1 fold cross validation error.
* Clustering using GMM is not useful as it fits one distribution over the entire dense region and almost always every test document gets classified under the big cluster.

Future Work

* Try better features based on bigrams, Part of Speech of words. As seen in the supervised learning section, there is no huge diversity in the unigrams (words) between documents labeled as positive or negative class.
* As features seem to be correlated, there is a scope to use other classifiers like the Maxent classifiers to classify the documents.
* Though labeling based on mean or excess mean returns provides a good starting point, most of the times a stock might go up or down, completely independent of content in the 10K Form. Alternate labeling techniques based on EAFDR (Earnings announcement Filing date Returns) should be explored [1].
* From the plot for principal components and factors obtained after using Factor Analysis, the data seems to be dense in the center and the points in the periphery clearly belong to one class. Prediction can be improved by using other clustering techniques -Spectral clustering.
* Train with different data sets: Since the vocabulary of formal 10 K documents is not diverse it might be interesting to see if training on documents obtained from different domain, like news and testing on the data from 10K forms or vice versa, gives a better idea of sentiment and helps in classification based on the mined sentiment.
* It's a great value addition to financial analysts if a document is not only classified based on its sentiment but also the sentences, which convey that sentiment are identified. After mining sentiment from the forms this is one area, which I am very eager to explore.
* It's interesting to see if there is any relation between the sentiment mined and the overall movement of the index.
* For this project I took excess mean returns over the Index. Sector based indices probably are good base as compared to S&P500 index.

References and Resources

[1] Haifeng You, Xiao-Jun Zhang, 2009. *Limited Attention and Stock Price Drift Following Earnings Announcements and 10-K Filings*
[2]Engelgerg Joseph, 2008.*Costly Information Processing:Evidence from Earnings Announcement.*
[3] Fabrizio Sebastiani, 2002. *Machine learning in Automated Text Categoraization.*
[4] Positive and negative sentiment words from http://www.wjh.harvard.edu/~inquirer/
[5] Bo Pang and Lillian Lee, Shivakumar Vaithyanathan, 2002, *Thumbs up? Sentiment Classification using Machine Learning Techniques*
[6] Svm light from http://svmlight.joachims.org/