Stock Selection using Support Vector Machines

Alan Fan and Marimuthu Palaniswami

Department of EEE, University of Melbourne, VIC 3010, Australia
{swami@ee.mu.oz.au}

Abstract

We used the Support Vector Machines in a classification approach to 'beat the market'. Given the fundamental accounting and price information of stocks trading on the Australian Stock Exchange, we attempt to use SVM to identify stocks that are likely to outperform the market by having exceptional returns. The equally weighted portfolio formed by the stocks selected by SVM has a total return of 208% over a five years period, significantly outperformed the benchmark of 71%. We have also given a new perspective with a class sensitivity tradeoff, whereby the output of SVM is interpreted as a probability measure and ranked, such that the stocks selected can be fixed to the top 25%.

1 Introduction

Investors are usually faced with an enormous amount of stocks in the market. A crucial part of the their decision process is the selection of stocks to invest in. In a data-mining perspective, the problem of stock-selection aims to identify stocks with potential to outperform the market (i.e. exhibit exceptional returns) in the following year. Given the database of stock prices and indicators, it is a prediction problem that involves discovering useful patterns or relationship in the data, and applying that information to classify stocks.

If the market is efficient such that all stock prices fully reflect all publicly available information, we cannot expect this form of analysis can identify stocks with superior return at all. According to the Efficient Market Hypothesis [1], any useful patterns should have been reflected in the current price, hence making it impossible for us to discriminate between normal and profitable investments in advance. Although the efficient market theory has been found robust in a number of studies [6], there also exist several empirical findings that indicate future stock returns are at least to some extent predictable [2] [3]. Several recent researches also presented encouraging results on stock selection using data mining techniques such as rule induction [7], neural network [8], and combination of classifiers [5].

In this paper we approach the problem of stock selection using Support Vector Machine (SVM) for classification. SVM [11] stems from statistical learning theory as an alternative method to training universal feedfoward networks. It attempts to construct an optimal separating hyperplane in the hidden feature space, using quadratic programming techniques. An important property that made SVM a promising tools is its implementation of Structural Risk Minimization [11] which aims to minimize a bound on the generalization error rather than on the empirical error. We test the usefulness of SVM with Australian stock data, where SVM is trained with accounting information. When SVM was used to select 25% of the stock every year, the equally weighted portfolio formed by SVM has a total return of 207% over a five-year period, outperforming the benchmark return of 71%.

2 Data Collection and Problem Formulation

We examine the financial information of stocks that are trading on the Australian Stock Exchange for the period of 1992-2000. In order to reduce noise level and maintain consistency, only annual reports are considered, and reports with more than 1 missing variables are discarded. The resulting data set consists of 273-537 annual reports each year, with the remaining missing variables calculated by mean imputation. Table 3 shows all the indicators calculated from the financial reports and the price data available. We further group similar financial indicators into one of the following eight categories: Return on Capital, Profitability,

0-7803-7044-9/01/$10.00 ©2001 IEEE

Leverage, Investment, Growth, Short term Liquidity, Return on Investment, Risk. This allows us to use principal component analysis for dimension reduction as we expect indicators from the same group tend to contain similar information. The data is converted to eight-element input vectors before training, with each element representing a single extracted principal component from each group.

For each year we assign a class to every stock to indicate its performance. Since different companies have different report cycles, the stock returns are calculated individually using the price data for the 12 months following the date of the publication. All price data is obtained from the IRESS system and the adjusted stock price is used. The performance of stocks are then ranked, with the top 25% being labeled as exceptional high return stock (Class +1) and the others labeled as unexceptional return (Class -1).

This problem is then formulated as a two-classes pattern recognition task. We represent the financial indicators for the the i^{th} firm as a vector of predictor variables $\mathbf{x}_i = (x_1, x_2, ..., x_n)$ (for our case $n = 8$). And the expected future return of the stock will be a binary dependent variable $y_i = \pm 1$, where $+1$ represents exceptional high return stocks, and -1 as normal stocks. Therefore a training set $(\mathbf{x}, y)$ of l firms will be the following pairs:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_l, y_l)\} \subset \mathbf{R}^N \times \pm 1 \qquad (1)$$

The classifier which attempts to learn from the examples can then be regarded as a set of functions mapping the predictor variables to values of y (± 1), and this ultimately aims to reduce the level of misclassification by adjusting its parameters. If gradient descent methods are used this is achieved by reducing the empirical error. In this paper we employ the Support Vector Machine as our classifier, where SVMs are better formulated to reduce the bound on the actual error instead.

We choose to use three years of data to identify high performance stock for the following year. In particular the first two years of data will be used for training, and the third year for validation, i.e. estimating the classifier's parameters. For example, when we predict the high return stock for 1995, the data from 1992-1993 is used for training, and the data from 1994 is used for validation. Note that the testing data is always strictly out of sample. Therefore with all our available data, we can perform the experiment five times to collect results from 1995 to 1999.

3 Overview of Support Vector Machines

3.1 Linear SVMs

SVMs address the problem that minimization of empirical error (in methods such as gradient descent method of training neural networks) does not guarantee small actual error. Therefore rather than reducing the empirical error (Empirical Risk Minimization), it implements the principle of Structural Risk Minimization [11], which aims to reduce the bound on the misclassification risk. For linearly separable patterns, this is achieved by using optimal separating hyperplane which takes the form of equation (2) with weights $\mathbf{w}$ and bias b:

$$f(\mathbf{x}_i) = sgn((\mathbf{w}'\mathbf{x}_i) + b) = y_i, \qquad i = 1, ..., l \quad (2)$$

For the hyperplane f to be optimal, its margin of separation,

$$\min_{\{\mathbf{x}_i | y_i = +1\}} \frac{\mathbf{w}'\mathbf{x}_i + b}{||\mathbf{w}||} - \max_{\{\mathbf{x}_j | y_j = -1\}} \frac{\mathbf{w}'\mathbf{x}_j + b}{||\mathbf{w}||} = \frac{2}{\sqrt{\mathbf{w}'\mathbf{w}}} \quad (3)$$

must be maximized subjected to equation (2). In fact this can be transformed into a Lagrangian, and there exist a dual problem which is equivalent to maximizing:

$$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i'\mathbf{x}_j)$$
$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (4)$$

For detailed derivation see [9]. The solution for the problem can be found by quadratic programming, in the form of $\alpha = \{\alpha_1, ..., \alpha_l\}$ where each $\alpha_i > 0$ corresponds to a support vector, and then the weights $\mathbf{w}$ and bias b can be calculated. To allow for non-separable patterns, it is equivalent to imposing extra constraint on equation (4) as $\alpha_i \leq C$. Alternatively we can achieve similar results by modifying the kernel matrix with a heavier diagonal:

$$K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda \tag{5}$$

The trade-off parameters C and λ are related. They are usually selected manually to control the trade-off between complexity (hence capacity) and the number of non-separable points (hence training error).

3.2 Non-Linear SVMs

However, SVM takes one more step to map the input vectors to a hidden, high dimensional feature space before the construction of the optimal hyperplane. This can be done with minimum extra computational cost with appropriate mappings. Consider the mapping $K : \mathbf{x}_i \rightarrow \mathbf{z}_i$ with the dot product of the transformation as k:

$$\left(K(\mathbf{x})' K(\mathbf{x}_i)\right) = k(\mathbf{x}, \mathbf{x}_i) \tag{6}$$

Then the quadratic programming problem of minimizing Q of equation (4) can be re-written as

$$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

$$\text{subject to} \qquad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{l} \alpha_i y_i = 0$$

with decision function

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}, \mathbf{x_i}) + b\right) \tag{8}$$

We can see that the solution provided by SVM is a unique solution, in contrast to the possible local minimum for gradient descent techniques like backpropagation.

Notice that the whole problem can still be solved in the same way without knowing the exact mapping K, but using the dot product $k(\mathbf{x}, \mathbf{x}_i)$ instead. With different kernels, the SVM architecture has the form of different classical classifiers. In this paper we used the Radial Basis kernel where $k(\mathbf{x}, \mathbf{y}) = exp(-||\mathbf{x} - \mathbf{y}||^2)$. Once the kernel is chosen, only the upper bound on the alphas (C) is need to be assigned. The parameter selection problem in SVM then becomes a tradeoff between capacity and generalization.

3.3 SVMs for Unbalanced Data

According to our definition of exceptional high return stock (Class +1), our training data is always unbalanced, with the sample size of Class +1 being one third of the other class. Obviously this results in a bias towards the larger class if the SVM is trained normally. Veropoulos et al [12] discussed two approaches to solve this problem for SVM. Firstly different regularization parameters C are assigned to different class. This implies re-writing equation (7) as:

$$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

$$\text{subject to} \qquad \begin{aligned} &0 \leq \alpha_i \leq C_+, \;\; y_i = +1, \\ &0 \leq \alpha_i \leq C_-, \;\; y_i = -1, \\ &\sum_{i=1}^{l} \alpha_i y_i = 0 \end{aligned}$$

In another perspective, one can supply different positive contributions for different classes to the diagonal of the kernel matrix:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_i) &\leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda_+, \qquad & y_i = +1 \\ K(\mathbf{x}_i, \mathbf{x}_i) &\leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda_-, \qquad & y_i = -1 \end{aligned} \tag{10}$$

Both approaches are equivalent to allowing different level of training errors on different classes, hence achieving an effective control over sensitivity.

4 Experiments

For the problem of stock selection, predictive accuracy alone is not a good indication of the classifier performance. To evaluate the usefulness of our approach, we compared the return generated by the selected stock from SVM with a benchmark. The benchmark return (the "market") is determined by an equally weighted portfolio of all the stocks available for classification. Table 1 shows the results for each year of testing data, including the predictive accuracy and average percentage return of the predicted high performance stock. The overall accuracy is calculated by (Class +1 accuracy + Class -1 accuracy)/2. Recall that in order to prevent overtraining, a year of validation data is used to selecting appropriate tradeoff parameters that gives highest overall validation accuracy. The parameter

λ_+ is allowed to be 0.1, 1, and 10, while the ratio λ_+/λ_- ranges from 10 to $\frac{1}{10}$.

From the preliminary results it can be seen that the equally-weighted portfolio formed by the selection of our approach produced excess return over the market benchmark for each of the five years tested. The return is significantly higher for year 1995, 98 and 99, while in other years the selected stocks still slightly ourperformed the benchmark. Over the whole period of 95-99 the accumulated return for the stocks is 122.10%, in contrast to the benchmark overall return of 71.36%. Despite the consistent excess return obtained, the predictive accuracy of this problem are relatively lower than other classification applications. This is expected because even if there exists a relationship between the future stock returns and its accounting information, one would expect it to be a weak relationship.

The results also suggest a tradeoff between the Class +1 and Class -1 accuracy, and the tradeoff determined by the validation exhibits a bias toward Class -1. In fact this is desirable despite Class -1 being the larger class, since the performance of the classifier is based on the stock selected rather than the predictive accuracy. Including a stock without exceptional return in the selection is clearly more costly than excluding a stock with exceptional return. In Figure 1 we have plotted the 1995-1999 accumulated profit, averaged class accuracy against the class accuracy trade-off (the ratio λ_+/λ_-). It illustrates the class accuracy tradeoff and how it is related to the profitability of the classification.

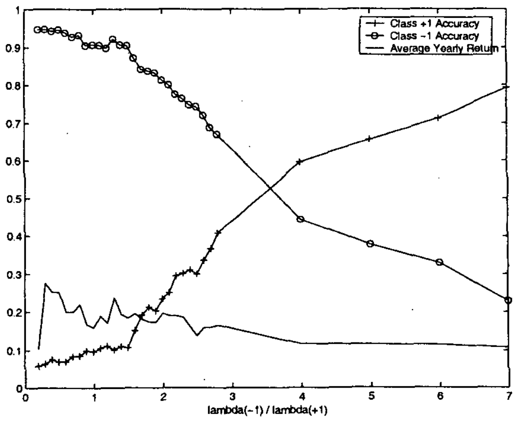

Figure 1: Class Accuracy Trade-off and Average Yearly Return

By examining only the class accuracy for each year, it already shows that the current method of validation failed to obtain an acceptable balance in 1997. That resulted in selecting a large number of stocks which is clearly not sensible. Therefore we propose to extend the task of obtaining the class accuracy tradeoff with a new perspective. Based on the fact that the distance of a test point from classification boundary is related to its probability of misclassification [10], we can utilize the geometrical information of SVM to interpret SVM outputs as probabilities. In [4] we converted SVM output as probabilities and demonstrated that class sensitivity adjustment can be achieved via setting different thresholds to the probability outputs. With that we now propose to rank the SVM output so that we can classify the top 25% stock that give highest Class +1 probability. For consistency the current methodology is kept therefore the class sensitivity adjustment is performed in two stage. Firstly sensitivity is adjusted through the estimation of λ_+/λ_- to consider the unbalanced number of samples in each class. Secondly the class accuracy tradeoff is further adjusted by ranking according to the SVM output to take into account of the different cost of misclassification for this problem.

Table 2 shows the results obtained for this method. Improvements are observed for both predictive accuracy and overall profitability. The accumulated return has been increased from 122.10% to 207.71% over the 5-year period. It is also interesting to see the relationship between the overall profitability against the percentage of stock to be classified as exceptional returns, as illustrated in Figure 2.

5 Conclusions

Support Vector Machine has been shown to be useful to the problem of stock selection. Using stock data on the Australian Stock Exchange, our methodology produces a 208% return over 5 strict out-of-sample years, significantly outperforming the benchmark of 71%. We also pointed out the key to this problem is to achieve the correct balance between the class sensitivity tradeoff. And we demonstrated how the SVM results were improved as the tradeoff is determined by rank-

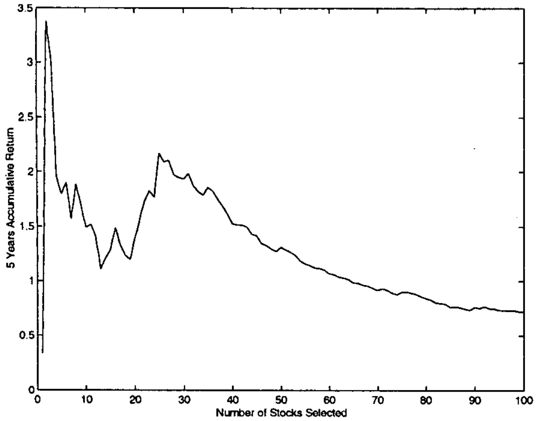

Figure 2: Accumulated Return against Number of Stocks Selected

ing SVM outputs and fixing the proportion of the stocks selected.

References

[1] E. F. Fama. Multiperiod consumption-investment decisions. *American Economic Review*, 60:163–174, 1970.

[2] E. F. Fama and K. R. French. Dividend yields and expected stock returns. *Journal of Financial Economics*, 22:3–26, 1988.

[3] E. F. Fama and K. R. French. The cross-section of expected stock returns. *Journal of Finance*, 47:427–465, 1992.

[4] A. Fan and M. Palaniswami. Corporate loan default predictors with class sensitivity analysis using svms. *Submitted to IEEE Transactions on Neural Networks.*

[5] Albanis G. and R. Batchelor. 21 methodologies to beat the market. In *Proceedings of Computational Finance 2000*, 2000.

[6] R. A. Haugen. *Modern Investment Theory.* Prentice-Hall, Inc., 1997.

[7] G. John, P. Miller, and R. Kerber. Stock selection using recon. In *Neural Networks in Financial Engineering*, pages 303–316, 1996.

[8] A. U. Levin. Stock selection via nonlinear multi-factor models. In *Advances in Neural Information Processing Systems*, 1995.

[9] B. Schlkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proceedings of First International Conference on Knowledge Discovery & Data Mining*, 1995.

[10] J. Shawe-Taylor. Classification accuracy based on observed margin. *Algorithmica*, 22:157–172, 1998.

[11] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., 1995.

[12] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machine. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.

	1995	1996	1997	1998	1999
Total No. of Stock	476	516	537	490	276
No. of Stock Selected	70	69	343	33	56
No. of Class +1 Selected	25	14	90	11	16
No. of Class -1 Selected	45	55	253	22	40
Class +1 Accuracy	21.01%	10.85%	67.16%	9.02%	23.19%
Class -1 Accuracy	87.39%	85.79%	37.22%	94.02%	80.68%
Overall Accuracy	54.20%	48.32%	52.19%	51.52%	51.93%
Return	41.44%	20.73%	-14.99%	28.53%	19.01%
Benchmark Return	29.32%	18.72%	-16.64%	23.03%	8.82%

Table 1: SVM test results (without further sensitivity adjustment)

	1995	1996	1997	1998	1999
Total No. of Stock	476	516	537	490	276
No. of Stock Selected	119	129	134	122	69
No. of Class +1 Selected	46	23	39	41	21
No. of Class -1 Selected	73	106	95	81	48
Class +1 Accuracy	38.66%	17.83%	29.10%	33.60%	30.43%
Class -1 Accuracy	79.55%	72.60%	76.42%	77.99%	76.81%
Overall Accuracy	59.10%	45.22%	52.77%	55.80%	53.62%
Return	58.66%	16.72%	-14.20%	40.23%	38.09%
Benchmark Return	29.32%	18.72%	-16.64%	23.03%	8.82%

Table 2: SVM test results (selecting 25% of the stock)

Return on Capital	**Investment**
Profit Before Tax / Total Assets	PE ratio
Profit Before Tax / Total Capital	Net Tangible Assets per Share
Net Income / Total Capital	Dividend Yield
CashFlow / Total Assets	Earning Yield
CashFlow / Total Capital	Share Holders' Equity / Total Market Value
Profitability	**Growth**
Profit Before Tax / Sales	Sales Growth
Profit After Tax / Sales	Earning Before Tax Growth
Net Income / Sales	Earning After Tax Growth
Cash Flow / Sales	Net Recurring Profit Growth
Profit After Tax /Equity	Operating Profit Growth
Cash Flow / Total Market Value	Shareholders' Fund Growth
Profit After Tax / Cash Flow	Total Assets Growth
Leverage	**Short Term Liquidity**
Debt / Equity	Current Assets / Current Liabilities
Total Liabilities / Total Capital	Current Liabilities / Total Assets
Total Liabilities / Shareholders' Equity	Current Liabilities / Equity
Total Assets / Shareholders' Equity	Long Term Debt / Total Debt
Total Assets / Total Market Value	**Risk**
Return on Investment	Profit Before Tax / Current Liabilities
Return on Assets	Proft After Tax / Current Liabilities
	Cash Flow / Current Liabilities

Table 3: Financial Indicators used