



UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2019:2

Past predicts the future with Supervised Learning and James-Stein theory

Sebastian Sjöholm

Examensarbete i matematik, 30 hp
Handledare och examinator: Denis Gaidashev
Januari 2019

A large, faint watermark of the Uppsala University seal is visible in the bottom right corner of the page. The seal features a sun with rays, a cross, and the Latin motto "ALERE FLAMMAM VERITATIS" (to feed the flame of truth).

Department of Mathematics
Uppsala University

Past predicts the future with Supervised Learning and James-Stein theory

Sebastian Sjöholm

Abstract

Different methods has been tested during time, without any success, to fully predict the future price direction of the stock market by using historical data and beat the Efficient Market Hypothesis. In this model we will try and predict the future price direction of different stocks by using a supervised Machine Learning technique called *Support Vector Machine* using a feature called *momentum*. An estimator called *James-Stein* estimator will also be introduce to try and improve the SVM method by adjusting the future prediction of a stock towards its common mean of its own sector. We will see by the result of this paper that the $J - S$ estimator does in fact improve the prediction of the future price direction on some time intervals p , where p is the number of days into the future. The methods are applied to three different portfolios containing 4, 12 and 18 different stocks from the same sector.

Contents

Abstract	1
Support vector machine, (SVM)	3
1 Introduction	3
2 Separating hyperplanes	3
2.1 Optimal Separating Hyperplanes	5
2.2 Optimal Separating Hyperplanes for the non-separable case . .	8
3 Loss function	12
4 James-Stein estimator	14
4.1 James-Stein Theory	14
4.2 Implementation of J-S on stock market	17
5 Model creation and Evaluation Methods	19
5.1 Data	19
5.2 SVM Model	20
5.3 Features	21
5.4 Estimation of model parameters	22
5.5 Model evaluation	24
5.6 Investment strategy	25
5.7 Method	27
6 Results	29
7 Discussion	29
8 Conclusion	35
9 Appendix	36
9.1 A	36
9.2 B	45

Support vector machine, (SVM)

1 Introduction

SVM is a classification method in machine learning whose purpose is to separate the input data into N different classes, $G = [1, 2, \dots, N]$, where each input x_i belongs to one class [6]. In this thesis we will use binary classification which divides the data into two different classes, namely $\{-1, +1\}$. Therefore we will only focus on these two classes here.

We will not introduce the *Kernel Trick* in this thesis. This method takes the original data from the original space into an enlarged space where the data is supposedly separable by a linear hyperplane. This hyperplane then creates a nonlinear separation line when brought back into the original space [6]. Hence, we focus only on linear hyperplanes in the original space as separation of classes. The reason why will become clear later when we introduce the *James – Stein* estimator.

When applying classification methods we consider two different cases of input data, the separable case and the non-separable case. In the separable case one can easily build a linearly separable hyperplane which completely separates the data [6]. For the non-separable input, the data overlap, and hence we must change the method to get the best separating hyperplane. For simplicity, we start by explaining the separable case.

2 Separating hyperplanes

To understand the underlying theory of SVM we will explain the theory of separating hyperplanes for separable data. This theory explains how one can find the best separating hyperplane given some training data $X = [x_1, \dots, x_n]$, where $x_i \in \mathbb{R}^m$ [6]. Remember we only work with binary classification here but the method can also be applied to N different classes [6].

Suppose now that we have a set of n data points $X = [x_1, \dots, x_n]$ and that each input is 2-dimensional. This is because we can visualize it. Now, suppose that for each input x_i we have two possible outcomes, namely $Y \in \{-1, +1\}$, where -1 and 1 is normally dummy variables for some class definitions. For example, in this thesis it will imply if a stock price moves down or up, respectively. We call the output variable Y the response variable. We want to

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

find a hyperplane in the two dimensional plane which separates the data in the best way. Then hopefully, the new input data will be classified according to their right class. But what do we mean by "best" separating hyperplane? According to [6] for two classes this means that we have to find a hyperplane that maximizes the distance to the two closest points of the two classes. The hyperplane will then have same distance to each of these points. To visualize this, we generate 20 uniformly distributed points in the interval $[0,4]$. We then separate the points into $\{-1, +1\}$ randomly by a linear plane. Figure 1 below shows what this looks like.

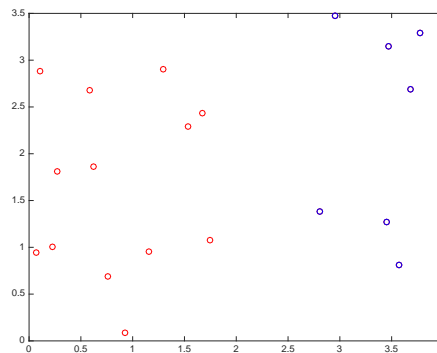


Figure 1: 2 dimensional random points where the different colours represents different classes, i.e. $\{-1, +1\}$.

In figure 1 above we see two different classes, red and blue. We can easily see that the two classes are separable and we need to find a 1 dimensional line which maximizes the distance between the two closest points of the different

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

classes. This distance is called the *margin* and is empty for separable data points. As in [6], we define the separating hyperplane as

$$[x : \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0] \quad (1)$$

and this equation equals 1 or -1 for points on the margin, which we will see later.

There is an algorithm called the *perceptron learning algorithm* which tries to find a separating hyperplane by minimizing the distance of misclassified points to the decision boundary [6]. If a response $y_i = 1$ is misclassified, then $x^T \beta + \beta_0 < 0$, and the opposite for a misclassified response with $y_i = -1$. As in [6] the goal is to minimize

$$D(\beta, \beta_0) = - \sum_{i \in M} y_i (x_i^T \beta + \beta_0) \quad (2)$$

where M indexes the set of misclassified points. Note that the quantity is non-negative and it minimizes the distance to the decision boundary $\beta^T x + \beta_0$ for misclassified points, hence ignoring the points on the right side of its margin. Also when we are dealing with separable data we do not have any misclassified points but with some modification, equation (2) is still useful. We now show how to estimate the optimal separating hyperplane.

2.1 Optimal Separating Hyperplanes

The optimal separating hyperplane maximizes the distance of the closest points from two different classes. According to [6] we can generalize equation (2) and consider the following optimization problem

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} C \\ & \text{subject to } y_i (x_i^T \beta + \beta_0) \geq C, \quad i = 1, \dots, N \end{aligned} \quad (3)$$

where C is the size of the margin. The constraint ensure that all points are at least a signed distance C (the margin) from the decision boundary defined by β and β_0 and we seek the largest such C . We can get rid of the constraint $\|\beta\| = 1$ by writing the condition as

$$\frac{1}{\|\beta\|} y_i (x_i^T \beta + \beta_0) \geq C \quad (4)$$

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

where β_0 is redefined . The condition, equation (4), is equivalent to

$$y_i(x_i^T \beta + \beta_0) \geq C \|\beta\|. \quad (5)$$

Since we need to find any β and β_0 satisfying equation (5), any positively scaled multiple will satisfy it too. Therefore we can set $\|\beta\| = \frac{1}{C}$ and equation (3) is equivalent to

$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1 \quad i = 1, \dots, N. \end{aligned} \quad (6)$$

The thickness of the margin is now defined by $\frac{1}{\|\beta\|}$, note that the problem in (6) minimizes $\|\beta\|$ which is equivalent to maximizing $C = \frac{1}{\|\beta\|}$.

To solve the constrained optimization problem in (6), we will use the method of Lagrange multipliers. The Lagrange function for the problem in (6), with the Lagrange multipliers α_i for the multiple constraints, is given below

$$L_P = \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i^T \beta + \beta_0) - 1]. \quad (7)$$

We then take the derivative w.r.t. β and β_0 and setting them to zero gives us

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i \quad (8)$$

and

$$0 = \sum_{i=1}^n \alpha_i y_i. \quad (9)$$

Now, by substituting equation (8) and (9) into equation (7) we get the Lagrange Dual function

$$\begin{aligned} L_D &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j, \\ & \text{subject to } \alpha_i \geq 0. \end{aligned} \quad (10)$$

The Lagrange dual problem (10) gives us a lower bound for the solution of the primal (minimization) problem in (6). And since its a lower bound it

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

will always be lower or equal to the solution given by the Lagrange Primal function in equation (7). Therefore, the problem given in (10) can instead be seen as a optimization problem w.r.t. β_0 and β . And it is solved by the convex optimization solver, *CVX*, in Matlab, which is explained in section 5.4, table (3).

In addition to the constrained problem in equation (10), the solution, α_i , must satisfy the Karuch-Kuhn-Tucker condition. According to [6], these conditions are

$$\alpha_i[y_i(x_i^T\beta + \beta) - 1] = 0 \tag{11}$$

along with equation (8) and (9).

From the conditions above we see that for $y_i(x_i^T\beta + \beta) = 1$ we have $\alpha_i > 0$, and it means that x_i is on the boundary of the margin for those i 's. When $y_i(x_i^T\beta + \beta) > 1$ x_i is on the "right" side of its margin and thus $\alpha_i = 0$. As we will see later, when $y_i(x_i^T\beta + \beta) < 1$ the points, x_i , is on the wrong side of its margin/decision boundary, but this is for the non-separable case.

The x_i 's for which $\alpha_i > 0$ is called the *Support Vectors* and if we look at equation (8) we see that these points completely define the beta coefficients. Likewise, β_0 is found by solving equation (11) for any (or all) of the support points.

The solution of the optimal separating hyperplane along with its margins are shown in figure 2 below

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

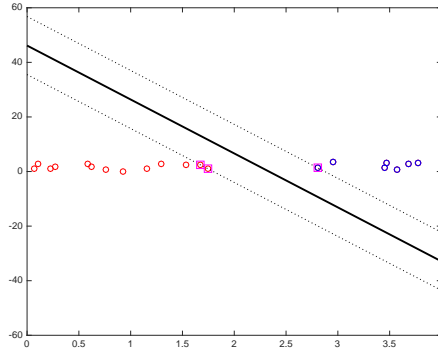


Figure 2: Separating hyperplane(thick line) and two margin boundaries(dashed lines). The support vectors are those in "boxes"

The points in "boxes" are the support vectors and one can observe that CVX used three different support vectors in this case for calculating β and β_0 .

Note : we know that the expression $y_i(x_i^T \beta + \beta)$ tells us the signed distance from the hyperplane. Therefore when $y_i(x_i^T \beta + \beta) = 1$ we know that the point x_i is $C = \frac{1}{\|\beta\|}$ from the hyperplane and $\alpha_i > 0$. And when $y_i(x_i^T \beta + \beta) > 1$ it is on the right side of its margin. And on the wrong side when < 1 .

2.2 Optimal Separating Hyperplanes for the non-separable case

In the last section the data was linearly separable into two classes. We now turn to the non-separable case which means that the data from different classes overlap and cannot be linearly separable. We will see that the method of separating hyperplanes can still be applied to the non-separable case but with some modification and that the decision boundary is still defined by the support vectors.

The training data is generated by 10 2-dimensional standard normally distributed random variables, and 10 2-dimensional normally distributed random variable with mean 1.75 and variance 1. The data is shown in figure 3 below

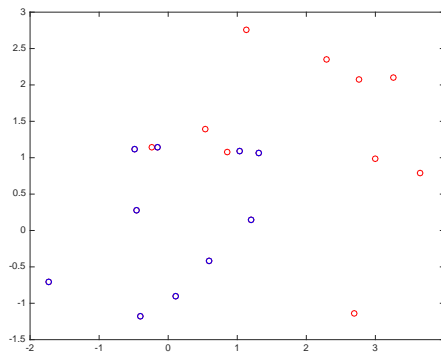


Figure 3: 20 normally distributed 2-dimensional random variables with half with mean 0 and half with mean 1.75

In figure 3 above one can see that the two classes overlap and thus it is impossible to separate by a linear hyperplane.

If we now turn to the minimization problem in equation (6) and modify the condition by introducing the so called *slack variables* we get the following minimization problem, as in [6], for n data points,

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

$$\begin{aligned}
& \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + \gamma \sum_{i=1}^n \xi_i \\
& \text{subject to} \\
& y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \quad i = 1, \dots, N.
\end{aligned} \tag{12}$$

So what is happening here? The value of the slack variable, ξ_i , in the constraint $y_i(x_i^T \beta + \beta_0)$ is the amount by which the predictor $f(x) = x_i^T \beta + \beta_0$ is on the wrong side of its margin. Misclassification occurs whenever $\xi_i > 1$ and $\xi_i = 0$ when x_i is on or on the right side of its margin. Therefore, the minimisation problem in equation (12) is penalised whenever x_i is on the wrong side of its margin. γ is a regularization parameter that controls the trade off between the achieving a low training error and a low testing error, that is, the ability to generalize your classifier to unseen data [6]. If you for example choose a large γ your minimisation problem in equation (12) will try to reduce ξ_i (which then reduces $\gamma \xi_i$) as much as possible (by minimizing $y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i$) leading to an overfit of the training data. Thus, $\gamma = \infty$ represents the separable case. However, by choosing a small γ , we will allow for more misclassification, in fact more the less the γ . Allowing for some misclassification could be good on the training set for generalizing the classifier for unseen data. This means that overfitting the classifier on the training data could lead to a bad testing result.

As in the last section we will explain the solution of the minimization problem in equation (12). According to [6], below is the Lagrange primal function of equation (12),

$$L_P = \frac{1}{2} \|\beta\|^2 + \gamma \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^n \mu_i \xi_i \tag{13}$$

which is maximized w.r.t. β , β_0 and ξ . Setting the respective derivatives to zero yields the following equations

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i, \tag{14}$$

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

$$0 = \sum_{i=1}^n \alpha_i y_i, \quad (15)$$

$$\alpha_i = \gamma - \mu_i \quad (16)$$

and we also have the constraints $\alpha_i, \mu_i, \xi_i > 0 \forall i$. If we substitute these equations into equation (13), as we did in the last section, we get the Lagrange Dual function

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (17)$$

which is maximized subject to $0 \leq \alpha_i \leq \gamma$ and $\sum_{i=1}^n \alpha_i y_i = 0$. In addition to equations (14) to (16) we have the following KKT conditions

$$\alpha_i [y_i (x_i^T \beta + \beta_0) - (1 - \xi_i)] = 0, \quad (18)$$

$$\mu_i \xi_i = 0, \quad (19)$$

$$y_i (x_i^T \beta + \beta_0) - (1 - \xi_i) \leq 0 \quad (20)$$

for $i = 1, \dots, n$. As in the last section, equation (18) tells us that $\alpha_i > 0$ whenever x_i is on or on the wrong side of the margin. These points are again called the support vectors. We can see from equation (14) that β is, again, completely determined by the support vectors. Among the support vectors some will lie on the margin where $\xi_i = 0$ and $x_i^T \beta + \beta_0 = 1$, and some will lie on the wrong side of its margin where $\xi_i > 0$. For the support vectors on the margin we have $0 < \alpha_i < \gamma$ and for those on the wrong side of its margin we have $\alpha_i = \gamma$.

The decision function is still defined as

$$\hat{G}(x) = \text{sign}[x^T \beta + \beta_0]. \quad (21)$$

From the example data presented in figure 3, CVX in Matlab was used with tuning parameter $\gamma = 10$ to find the optimal separating hyperplane with slack variables. The optimal hyperplane and its margins are presented in figure 4 below

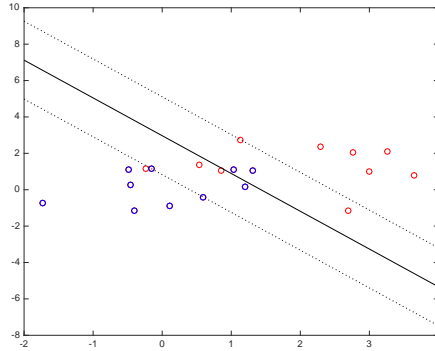


Figure 4: Optimal separating hyperplane(solid line) along with its margin(dashed lines) for the two class case.

In figure 4 above we can see the support vectors on the margin and also the misclassified points. CVX used 3 support vectors in this case and there are 5 misclassified points.

3 Loss function

In the above theory of SVM the solution to the classifier has been formulated as a constrained optimization problem. This can be seen in equation (12). The constraint $y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i$ can be reformulated as

$$\xi_i = \max(0, 1 - y_i f(x_i)) \quad (22)$$

where $f(x) = x^T \beta + \beta_0$ is the decision function. Note that this is valid because in equation (12) we also have the constraint that $\xi_i \geq 0$. According to [6] we can now write the constrained problem in (12) as the

[6]Friedman, J.; Hastie, T; Tibshirani, R. (2009), "The Elements Statistical learning: Data Mining, Inference, and Prediction"

following unconstrained problem

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + \gamma \sum_{i=1}^n \max(0, 1 - y_i f(x_i)) \quad (23)$$

where $\max(0, 1 - y_i f(x_i))$ is the *Hinge* loss function for SVM and γ is a regularization parameter which shrinks the coefficients β towards zero.

If we look at equation (23) and remembering the definition of $y_i f(x_i)$ we notice that the loss function is zero whenever a point x_i lies on or on the right side of its margin, i.e. $y_i f(x_i) \geq 1$. And whenever a point lies on the wrong side of its margin, $y_i f(x_i) < 1$, the loss function will be greater than 0. All this can be seen in figure 5 below (the x-axis is $yf(x)$, and y-axis is loss, L)

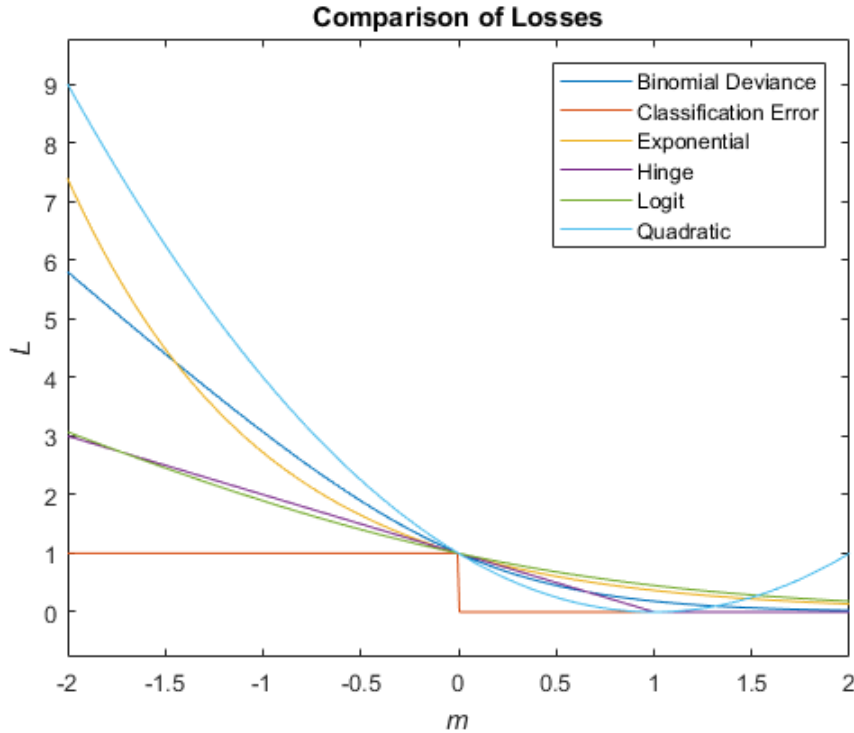


Figure 5: Comparison of different loss functions.

In the above figure the purple line represents the Hinge loss described above and we can see that for $y_i f(x_i) \geq 1$ the loss function becomes zero, i.e. no penalization. Figure 5 also shows the Quadratic loss function and we see that this loss function becomes greater, meaning more penalization, even for points far inside its own margin. This is why this loss function would

be a bad idea for classification problems. In the figure above we can also see the Logistic, Exponential and Binomial loss function. And by observing the figure we notice that all these work better for classification than the Quadratic loss (this because they don't penalize points inside its own margin).

4 James-Stein estimator

In this thesis a James-Stein estimator will be used to shrink the data towards a common mean. Different criteria will be used for which stock to be included in different groups, and then shrunk towards their common mean. In the following sections some general theory of the J-S estimator will be presented followed by the implemented version of J-S estimator in this thesis. Some basic knowledge of Bayesian theory will be preferable in these section.

4.1 James-Stein Theory

The James-Stein(J-S) estimator is a *biased* estimator of the mean of normally distributed random variables. One can show that the J-S estimator dominates the ordinary least squares estimator in terms of mean squared error [5] . There exists an earlier version of the estimator developed by Stein himself in 1956 [1]. But this was later improved by James and Stein in 1961. The theory behind the J-S estimator is explained below.

Suppose we have an observation of the random variable x from the normal distribution, i.e. $x|\mu \sim N(\mu, \sigma^2)$, where σ^2 is known. We wish to estimate the unknown parameter μ . Remember that the Maximum Likelihood estimation (ML) is just $\mu^{ML} = x$. But now, we assume that the parameter of interest μ has some prior distribution $\mu \sim N(M, A)$, and thus we assume that μ is also a random variable. We want to find the posterior distribution of μ given the value of the observed random variables x . According to [7] we have

$$f(x|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right) \propto \exp\left(\frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2}\right) \quad (24)$$

and

-
- [5]James, W.; Stein, C. (1961), "Estimation with quadratic loss"
 [1]Stein, C. (1956) "Inadmissibility of the usual estimator for the mean of a multivariate distribution"
 [7]Efron, B; Hastie, T. (2016), "Computer Age Statistical Inference: Algorithms, Evidence, and Data Science"

$$f(\mu) = \frac{1}{\sqrt{2\pi A}} \exp\left(-\frac{1}{2A}(\mu - M)^2\right) \propto \exp\left(\frac{\mu M}{A} - \frac{\mu^2}{2A}\right). \quad (25)$$

The numerator in Bayes' theorem, $f(H|E) = \frac{f(E|H)f(H)}{f(E)}$, now becomes

$$\begin{aligned} f(x|\mu)f(\mu) &\propto \exp\left(\frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} + \frac{\mu M}{A} - \frac{\mu^2}{2A}\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{A + \sigma^2}{A\sigma^2}\right)(\mu^2 - 2\mu\left(\frac{Ax + \sigma^2 M}{A + \sigma^2}\right))\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{A + \sigma^2}{A\sigma^2}\right)(\mu^2 - 2\mu\left(\frac{Ax + \sigma^2 M}{A + \sigma^2}\right) + \left(\frac{Ax + \sigma^2 M}{A + \sigma^2}\right)^2)\right) \\ &\quad \exp\left(-\frac{1}{2}\left(\frac{A + \sigma^2}{A\sigma^2}\right)\left(\frac{Ax + \sigma^2 M}{A + \sigma^2}\right)^2\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{A + \sigma^2}{A\sigma^2}\right)(\mu^2 - \left(\frac{Ax + \sigma^2 M}{A + \sigma^2}\right))^2\right) \end{aligned} \quad (26)$$

And thus, the posterior distribution is $\pi(\mu|x) \sim N\left(M + \frac{A}{A + \sigma^2}(x - M), \frac{A}{A + \sigma^2}\sigma^2\right)$. The Bayesian estimate of μ is

$$\hat{\mu}^{Bayes} = M + B(x - M) \quad (27)$$

where $B = \frac{A}{A + \sigma^2}$. This also holds if we have a p -dimensional vector x , where each x_i are independently distributed as $x_i|\mu_i \sim N(\mu_i, \sigma^2)$ and $\mu_i \sim N(M, A)$ (note that the μ_i 's differ from each other). Then each independent estimate of $\hat{\mu}_i^{Bayes}$ is

$$\hat{\mu}_i^{Bayes} = M + B(x_i - M), \quad i = 1, \dots, p. \quad (28)$$

Equation (27) and (28) hold if M and A are known. Suppose now that we don't know M and A , we can then estimate them from our observations x_i . We can define each x_i as the sum of two independent normally distributed random variables

$$x_i = \mu_i + \epsilon_i \quad (29)$$

where $\epsilon_i \sim N(0, \sigma^2)$ and the distribution of μ_i is as before. Then by the summation rule of two independently normally distributed random variables we get the following marginal distribution of x_i

$$x_i \sim N(M, A + \sigma^2). \quad (30)$$

Then, from the distribution of x_i , we get the following ML estimation of M (from p independent observations of x_i)

$$\begin{aligned}
L &= \prod_{i=1}^p \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x_i - M)}{2\sigma_i^2}\right) \\
\ln(L) &= -\frac{p}{2} \ln(2\pi) - p \sum_{i=1}^p \ln(\sigma_i) - \sum_{i=1}^p \frac{(x_i - M)^2}{2\sigma_i^2} \\
\frac{\partial \ln L}{\partial M} &\propto \sum_{i=1}^p (x_i - M) = 0 \\
\rightarrow M &= \frac{1}{p} \sum_{i=1}^p x_i = \bar{x}
\end{aligned} \tag{31}$$

where $\hat{M} = \bar{x}$ is an unbiased estimate of M . Moreover, for B we have

$$x_i \sim N(M, A + \sigma^2) \tag{32}$$

then

$$\frac{\sum_{i=1}^p (x_i - M)^2}{A + \sigma^2} \sim \chi^2(p - 1) \tag{33}$$

where $M = \bar{x}$, and $S = \sum_{i=1}^p (x_i - \bar{x})^2$

$$S \sim (A + \sigma^2) \chi^2(p - 1). \tag{34}$$

Now, $\frac{1}{S}$ is inverse χ^2 -distributed with the constant $\frac{1}{A + \sigma^2}$, and we get

$$\begin{aligned}
E\left[\frac{1}{S}\right] &= \frac{1}{A + \sigma^2} \frac{1}{(p - 1) - 2} \\
\rightarrow E\left[\frac{p - 3}{S}\right] &= \frac{1}{A + \sigma^2}
\end{aligned} \tag{35}$$

where $p > 3$. Because we have $B = (1 - \frac{1}{A + \sigma^2} \sigma^2)$, we get the following estimation for B ,

$$\hat{B} = \left(1 - \frac{(p - 3)\sigma^2}{S}\right), \quad S = \sum_{i=1}^p (x_i - \bar{x})^2. \tag{36}$$

Now, the J-S estimation of the mean for the i th random variable is the plug-in version of equation (27),

$$\begin{aligned}
\hat{\mu}_i^{J-S} &= \hat{M} + \hat{B}(x_i - \hat{M}) \\
&= \hat{B}x_i + (1 - \hat{B})\hat{M}
\end{aligned} \tag{37}$$

From the above equation we know from before that $\hat{M} = \bar{x}$ which is the mean value of the observations in our p -dimensional vector x . Thus equation (37) is analogous to

$$\hat{\mu}_i^{J-S} = \hat{B}x_i + (1 - \hat{B})\bar{x}. \quad (38)$$

If we examine the above equation one can interpret it as follows, the J-S estimate of the mean value of our observation $x = (x_1, \dots, x_p)$, is the observed value x_i shrunk towards the overall mean \bar{x} of our observed vector x . One important note is that the J-S estimator only dominates the LS estimator in evaluating the MSE *iff* $p > 3$ [5]. Also we have assumed σ^2 to be known and equal for all x_i which is almost never the case in reality.

4.2 Implementation of J-S on stock market

In the previous section some general theory of the J-S estimator was presented. In this thesis we will apply the J-S estimator in classifying the direction for different stocks using SVM, and this implementation will be explained in this section.

As an example we assume that we have a portfolio of p different stock and that we wish to forecast which direction each individual stock will take in, say, two weeks from today. We also assume that in order to forecast these directions we use the average value of the past 3 months of the price difference for each stock. Thus we have a vector x of p input values. We also have a coefficient vector β with one coefficient for each input value and an intercept β_0 . These two vectors along with its intercept will then generate a response vector which will represent our future predicted direction for each stock two weeks from today.

If we now look at the J-S estimator in equation (38) we see that each input value x_i is shrunk towards the common mean of all stocks price volatilities. This might impose some problems, for example if there exists two stocks which do not share the same variability which is assumed by $x_i|\mu_i \sim N(\mu_i, \sigma^2)$. Then the estimation of \hat{B} in (36) is not valid anymore. Thus there might be some improvements we can do in estimating \hat{B} . If we, for example, instead of choosing the mean value of all stock in our portfolio, we choose stocks which belong to the same stock index. Then it is easier to assume that these sectors have a more similar variability than two stocks from different sectors. This because sectors don't always move in the same

[5]James, W.; Stein, C. (1961), "Estimation with quadratic loss"

direction and also some sectors are more volatile and/or more affected by market fluctuations than others. Also small (or young) companies that are not well-established on the market tend to be more volatile in response to market fluctuations than larger already well established companies. Hence these stocks do not share the same volatility. Further, even though we assume the same volatilities in each sector, we still don't know the right volatility and we would have to estimate these. This creates yet another problem because, in reality, these stocks don't share exactly the same variability and also, we would have to estimate new variances for each stocks new observation where all stocks would have different volatilities. Hence this will not be very efficient.

We will therefore, instead of estimating \hat{B} , evaluate B in $[0, 1]$ using the *Hinge* loss function, $L(Y, f(X))$, on the training data. And the mean values used will be evaluated sectorwise. Then, the B values which minimizes the loss function will be chosen. Hence, for each new observation we might use the following equation to forecast each stocks direction at time $t + \tau$,

$$\hat{Y}_{t+\tau,i} = \text{sign}[\beta(\bar{x}_{t,j} + B(x_{t,i} - \bar{x}_{t,j})) + \beta_0] \quad (39)$$

where $\bar{x}_{t,j}$ is the average of all stocks which belong to sector j at time t , and $x_{t,i}$ is the observed value of stock i (which belongs to sector j) at time t . Also, $\hat{Y}_{t+\tau}$ is the forecasted direction of stock i , τ days into the future.

5 Model creation and Evaluation Methods

In this paper SVM was used to forecast the future price direction of different stocks. The idea is that SVM will be able to make predictions of how the price will move someday in the future given some features. The procedure is explained below.

5.1 Data

The data used in this paper is daily closing prices for stocks retrieved from Nasdaq OMX. The closing prices come from 188 different stocks ranging from 31/12 2004 to 20/3 2017. Because some stocks were not listed until late, these stocks were removed to get as much historical data as possible from the remaining stocks. Also some stocks vanishes from the stock market, e.g. because of failure, hence they where also removed. Thus, after removing these stocks our data consists of 89 different stocks and 3187 daily closing price observations, respectively. In the data we also observe some missing values which we choose to replace by the last observed value for the given stock. The stock data was then divided into 7 groups depending on which type of sector it belongs to, e.g. Healthcare, Telecommunications, etc. These groups, along with the number of stocks, are presented in table 1 below.

Sector	Number of stocks
Consumer Services	13
Technology	22
Consumer Goods	18
Telecommunications	4
Basic Materials	12
Oil and Gas	2
Healthcare	18

Table 1: Number of stocks in different sectors.

As we can see the number of stock in each group differ from 22 stocks in the Technologies sector to only 2 stocks in the Oil and Gas sector which might have some effect on the result when running the algorithm. What also can be seen in this table is that for many sectors we have portfolios of the same, or almost, number of stocks. For this reason I have chosen to analyse Healthcare(18), Basic Materials(12) and Telecommunications(4), where we will see if there is some effect due to the size of the portfolio.

In many previous papers on this subject the data was divided into one group of training set, e.g. 70% of the data, and the rest was used as a test set when the machine learning model had been estimated [2]. In this paper we will take a different approach. The SVM model will be re-estimated for every k th day, hence when the model has been estimated, it will be used as forecast and selection of portfolio strategy for some days until the next k th day arrives and we re-estimate the model once again. By this method the SVM model will, for each time it is re-estimated, have more historical data than before. Hopefully making better predictions. Note that the model can only be estimated for the first time when we have a suitable amount of historical data. The exact amount of data needed to make good SVM model is not known but we assume it needs at least one year of data points.

5.2 SVM Model

The theory of SVM was presented in section 2. When using the SVM model in this paper we will assume that stocks that belong to the same sector share some common variability, and hence stocks from the same sector might help to make better forecast for each stock in that particular sector.

The first method is influenced by the theory of *James – Stein*, presented in section 4, which is an estimation of the mean value of normally distributed random variables. The J-S estimator can be shown to dominate the Maximum Likelihood estimation when $m > 2$ (where m is the number of independent estimations), meaning that it always achieves a lower *MSE* value than the ML method [5]. Therefore, when estimating the SVM model we will *shrink* each value towards the common mean of its sector by a constant B . The forecasted value, $\hat{Y}_{t+1,k}$, for stock k at time $t+1$ with one feature is described by the model below

$$\hat{Y}_{t+1,k} = \text{sign}[(BX_{t,k} - (1 - B)\bar{X}_{t,k}) \cdot W]. \quad (40)$$

In equation (40) above $\hat{Y}_{t+1,k}$ predicts a value in $\{-1,+1\}$ at time $t+1$ for stock k . This means that if the predicted value is $+1$ the stock price is expected to increase and if -1 the price is expected to decrease. The variable $X_{t,k}$ is the feature of stock k at time t and the second variable \bar{X}_{t,k_s} is given by the mean value of all stocks feature values in the same sector as stock k at time t . The parameter W is the coefficient.

The next model also uses all other stocks in its sector to improve its future

[2]Saahil, M. (2015) "Predicting Stock Price Direction using Support Vector Machines"
[5]James, W.; Stein, C. (1961), "Estimation with quadratic loss"

prediction but instead of shrinking each observation towards a common mean, this model will use a weighted value of all stocks features in the same sector when forecasting the future value of each stock. A forecasted value of stock a k at time $t+1$ is given by the model below

$$\hat{Y}_{t+1,k} = \text{sign}[X_{t,k}W]. \quad (41)$$

where $\hat{Y}_{t+1,k}$ is the same as in equation (40). The feature $X_{t,k}$ is now a vector of all stocks feature values in the same sector as stock k at time t . W is thus a coefficient vector of the same size as $X_{t,k}$. Note that equation (41) is the special case in equation (40) when $B = 1$.

5.3 Features

In both SVM models one features was used in order to make a forecast of the future price direction of each stock. According to research done by [3] this feature will be based on the momentum of each stock's daily return from todays value at time t and τ days into the past.

The feature used is based on the momentum for each stock and will be examined with different number of days over which the momentum is calculated. The variable τ which represents the number of days, will be varied from long to short. The calculations are described in table 2 below. In the first model, equation (40), we also introduce the mean momentum of the common sector. The original feature will then be shrunk by a constant $(1-B)$ toward this common sector mean. These are described in equation (40) above and the feature is described in table 2 below.

[3] Jegadeesh; Narasimhan; Titman; Sheridan (1993), "Returns to buying winners and selling losers: implications for stock market efficiency"

Feature name	Description	Formula
Stock Momentum	This is an average of the momentum of the past n momentum observations for a given stock. Each momentum observation represents the daily return for the past n days.	$\sum_{i=t-n+1}^t \log(1 + \Delta C_i) = \log\left(\frac{C_t}{C_{t-n}}\right)$
Sector Momentum	This is the common mean momentum for a sector which is used in the first SVM model. The mean is taken over all stocks momentum observations at time t in sector S . Note that this is done for both n and m .	$\frac{\sum_{k_S} X_{t,k_S}}{q}$

Table 2: In the above formulas ΔC_t is defined as the percental price difference $\frac{C_t - C_{t-1}}{C_{t-1}}$. The number of days n vary as one tries to find the optimal number of days in order to make the best possible forecast. X_{t,k_S} represents the momentum of stock k in sector S at time t .

5.4 Estimation of model parameters

In order to estimate the model parameters in our SVM models we will use *CVX*. This is a Matlab-based modeling system for convex optimization [8]. In order to find the optimal value of the parameters W we can present both models as an optimization problem. The first model, equation (40), can be presented as

[8]Michael C. Grant; Stephen P. Boyd, "CVX: Matlab Software for Disciplined Convex Programming"

$$\begin{aligned}
& \min_W \frac{1}{2} c \|W\|^2 + \sum_{i=1}^n \sum_{j=1}^q \xi_{i,j} \\
& \text{subject to} \\
& \underbrace{Y}_{n \text{ by } q} \cdot \left[\underbrace{(XB + (1 - B)\bar{X}_S)}_{n \text{ by } q} \cdot \underbrace{\text{repmat}(W, [n, 1])}_{n \text{ by } q} \right] \leq \underbrace{1 - \xi}_{n \text{ by } q} \\
& \xi_i \geq 0.
\end{aligned} \tag{42}$$

Now lets say we have a training set of n observations and q different stocks from the same sector S in our portfolio. Then the variable Y is a n -by- q -valued matrix with values in $\{-1,+1\}$, the features vector X is of the same size. Note that in these matrices, each column represents observations from each individual stock. The last variable \bar{X}_S represents the sum over all rows in X and then each row is divided by the number of stocks, q . The parameter W is thus a q -by-1 matrix and (\cdot) is the dot product. Note that `repmat` is a command in Matlab which in our case creates a matrix of n rows and 1 column, but because our vector W has q columns it creates a n -by- q matrix where all rows are identical.

The optimization problem for the second model, equation (41), is presented below

$$\begin{aligned}
& \min_W \frac{1}{2} c \|W\|_F^2 + \sum_{i=1}^n \sum_{j=1}^q \xi_{i,j} \\
& \text{subject to} \\
& \underbrace{Y}_{n \text{ by } q} \left[\underbrace{X}_{n \text{ by } q} \times \underbrace{W}_{q \text{ by } q} \right] \leq \underbrace{1 - \xi}_{n \text{ by } q} \\
& \xi_i \geq 0.
\end{aligned} \tag{43}$$

The variables Y and X in problem (43) is the same as in problem (42) and $\|\cdot\|_F$ is the Forbenius norm. The coefficient matrix is now a q -by- q matrix. Presented below is what this looks like in *CVX* for the first SVM model, equation (40).

```

CVX_begin
variables w(n,1) e(m,n)
minimize (0.5*c*(w'*w) + sum(vec(e))/m);
subject to
YYY.*(XXXX.*repmat(w',[m,1])>= ones(m,n)-e;
e >= 0;
CVX_end

```

Table 3: Algorithm for estimating the parameters in the SVM model.

As can be seen in the algorithm we have to define the value of the constant c which gives us the best future prediction. Also, the feature $XXXX$ includes the constant B (from $XXXX = BX + (1 - B)\bar{X}$), which we also have to find the best (in the sense of best future prediction) value for.

To choose the optimal values for c and B for a given set of data one way is to use cross-validation for finding values for $B \in [0,1]$ and $c \in \mathbb{R}_+$. Although this is a good way to find the optimal value it is very computational intensive and therefore we will use another less computationally intensive approach. This method evaluates the model for $B \in \{0.25, 0.5, 0.75, 1\}$ by maximizing its *Sharpe Ratio* with respect to B . Then c is found in the same way but for values in $\{100, 10, 0.1, 0.01\}$.

5.5 Model evaluation

In finance, the Sharpe Ratio, SR , of an asset is a measure of the excess return on an investment adjusted for its risk. In this thesis we will evaluate the performance of our models according to the Sharpe Ratio defined in [4]

$$SR = \frac{\bar{R}}{\sqrt{\bar{R}^2}} \quad (44)$$

where

$$\bar{R}^2 = \frac{1}{T} \sum_{t=1}^T R_t^2 \quad (45)$$

where R_t is the return on the portfolio at time t given by

$$R_t = \frac{P_t}{P_{t-1}} - 1 \quad (46)$$

[4]Sharpe, William F. (1994), "The Sharpe Ratio"

and \bar{R} is just the mean value of all returns on the portfolio. Note that the mean is taken over all returns, i.e. from $t = 1$ which is our first return up to $t = T$ which is our last return.

Thus, by maximizing the mean value of SR_t for all observations t , we will get the highest return adjusted for the risk of our portfolio.

5.6 Investment strategy

The forecast of the future price direction tells us that the model believes the stock price will go up or down p days into the future. When this forecast is done we will buy stock(long position) if the model believes the stock price will increase or take a short position, i.e. borrow stock from an investor and sell it to another investor, if the price is forecasted to decrease.

Now, lets say we have a portfolio consisting of N different stocks and a total capital of x . Then, each time the model makes a forecast we will have to calculate how many units to invest in each stock.

The return on a given stock k , $R_{t+1,k}$, at time $t + 1$ is given by equation (46)

$$R_{t+1,k} = \frac{P_{t+1,k}}{P_{t,k}} - 1 \quad (47)$$

where $P_{t,k}$ is the price of stock k at time t . We define the expected return, and the variance on a stock k as

$$\begin{aligned} \mu_k &= \mathbb{E}[R_{t+1,k}] \\ \sigma_k^2 &= Var(R_{t+1,k}). \end{aligned} \quad (48)$$

We now make the assumption that all stocks have different expected returns but they all have the same risk-adjusted expected return, μ , i.e.

$$\mu = \mu_k \sigma_k \quad (49)$$

where μ is constant. Note that the constant μ in the above equation is always greater than zero. In our case this is not true due to the fact that some stocks will be expected to decrease where we will take short positions instead. Therefore we introduce the variable $d_k \in \{-1,+1\}$, which takes the value -1 or $+1$ if the stock is expected to decrease or increase, respectively. Now the risk-adjusted return will still be constant $\mu = |\mu_k \sigma_k|$, but with different signs for each stock.

Then the return on a portfolio at time $t + 1$ will be

$$\sum_k w_k R_{t+1,k} \quad (50)$$

where w_k is the number of units of stock k that we have invested. The expected return and total variance on this portfolio is then

$$\begin{aligned} \mathbb{E}[\sum_k w_k R_{t+1,k}] &= \sum_k w_k \mu \sigma_k d_k, \\ \sigma^2 &= \sum_k w_k^2 \sigma_k^2. \end{aligned} \quad (51)$$

Note that in the above equation we make a large assumption by assuming all stocks are uncorrelated. Normally we would have to deal with covariances in the above equation but because of this assumption equation (51) holds. To construct a "good" portfolio we would like to maximize the expected return on our portfolio, constrained by the total amount of risk, V_0 . Thus, from equation (51), we get the following optimization problem

$$\begin{aligned} \max_w \sum_k w_k \mu \sigma_k d_k \\ \text{subject to} \\ \sum_k w_k^2 \sigma_k^2 \leq V_0. \end{aligned} \quad (52)$$

We can rewrite the above problem as a minimization problem instead by just changing the sign of the objective function. Then we get the following Lagrangian function

$$L = - \sum_k w_k \mu \sigma_k d_k - \lambda (V_0 - \sum_k w_k^2 \sigma_k^2). \quad (53)$$

If we now take the derivative with respect to w_k and set to zero, we get

$$w_k = \frac{\mu d_k}{2\lambda \sigma_k} \quad (54)$$

and we see that the amount invested in stock k is inverse proportional to the size of the risk of stock k . Note that when d_k is negative we take a short position, i.e. we will borrow stock and sell to another investor during the time of investment. Thus, all we have to do now is define the constant $C = \frac{\mu}{2\lambda}$. If we look at the total amount of risk in our portfolio in equation (51), we can now define this as

$$\sigma^2 = \sum_k w_k^2 \sigma_k^2 = \sum_k \frac{C^2 d_k^2}{\sigma_k^2} \sigma_k^2 = NC^2. \quad (55)$$

So we have that $\sigma^2 = C^2 N$, where N is the number of stocks in our portfolio. With some algebra we can see that

$$C = \frac{\sigma}{\sqrt{N}} \quad (56)$$

which means that the constant, C , is determined by the total desired risk, σ , of our portfolio. Then the weights are determined by

$$w_k = C \frac{d_k}{\sigma_k}. \quad (57)$$

5.7 Method

The feature in the SVM model was calculated as described in table 2. The momentum feature was calculated by varying the parameter n to try and find if some trends can help us to make more accurate forecasts of the price direction in our portfolio of stocks. The parameter n was varied for $n \in \{180, 240, 300\}$. Then for each n , the mean value for all stock momentum in the same sector were calculated. This was then used for the SVM model in equation (40) to create the new feature according to the formula $BX + (1 - B)\bar{X}$, where B is some constant in $(0, 1]$. The constant $c \in \{100, 10, 0.1, 0.01\}$ was also varied for both models.

A response matrix was also retrieved from our data which shows what the future price direction is p days into the future with values in $\{-1, +1\}$. This was found by just taking the difference of the future price and the price when the forecast was made giving value -1 if the direction is down and +1 if the direction is up. We will try to predict the price direction p days into the future, where $p \in \{5, 20, 60, 90, 120, 240\}$.

Because we choose to estimate the models at day t for the first time and then re-estimate for every k th day, we need some historical data to get a valid estimate of the model the first time. As said before this is not any given number but it should at least be more than a year of data points. The total number of features will thus vary depending on the size of n and how many days into the future we wish to forecast. Note also that the same model is used to forecast future price direction of stocks in our portfolio every day until the model is re-estimated again. This happens every k th day.

Each time the model is evaluated at some time t , our model will choose an investment strategy on taking a long or short position with respect to our

stocks. Because the data set already consist of the real future price difference we can evaluate the performance of our portfolio strategy. The model will forecast some value in $\{-1,+1\}$, which is then taken as the product with the real price difference and the number of units invested. This will give us the real return on our investment by the portfolio strategy. The best SVM model of both versions was chosen by maximizing the modified Sharpe Ration with respect to c and B for different n and k .

6 Results

The result of the best performing estimators are presented below in terms of mean accuracy and the best Sharpe Ratio. Note that p represents the number of days into the future in which the algorithm tries to predict the price direction, τ represents the number of days into the past in which the momentum feature is calculated and c is the penalization constant. The mean accuracy describes the mean performance of the methods future prediction of the price direction. The Sharpe ratio examines the performance of each investment adjusted for its risk. The rest of the results are presented in the tables in the Appendix.

7 Discussion

First of all we should point out that only one feature is used, the momentum. One could search for more features to try and increase the performance of the predictor of the future price direction [2]. Although this is the case, one should also bear in mind that the computations becomes computationally heavy fast before adding too many features to the algorithm.

The result for *Telecommunications*, which contained only 4 stocks, is presented in figure 6 below. The left graph presents the (best)mean accuracy for the algorithm, in terms of how many times it predicted the right pricemovement, for different B when τ varies. The right graph presents the same result, except now p varies. We can easily see that for the *Telecommunications* portfolio, the algorithm predicts the right future price movement more than 50 per cent, most of the time. This can be seen more easily in tables 4 – 5 below.

[2]Saahil, M. (2015) "Predicting Stock Price Direction using Support Vector Machines"

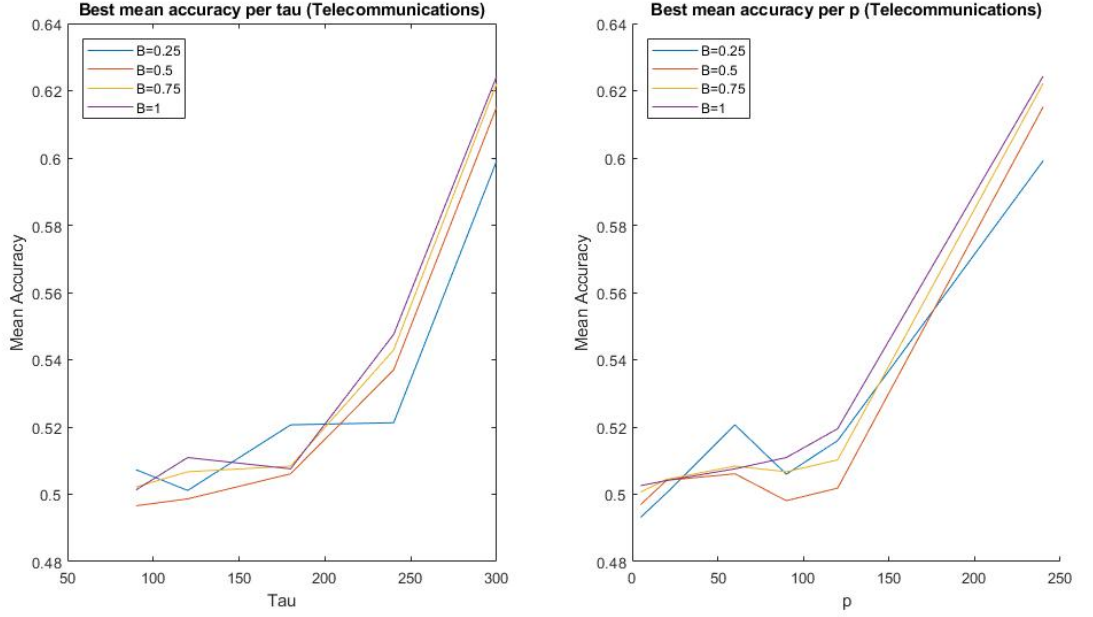


Figure 6: Mean Accuracy Telecommunications

p/B	B=0.25	B=0.5	B=0.75	B=1
p=5	0.4932 _{c=0.01,τ=4}	0.4970 _{c=0.01,τ=5}	0.5007 _{c=0.01,τ=5}	0.5026 _{c=0.01,τ=4}
p=20	0.5003 _{c=0.01,τ=5}	0.5041 _{c=0.01,τ=5}	0.5044 _{c=0.01,τ=5}	0.5041 _{c=0.01,τ=5}
p=60	0.5207 _{c=0.1,τ=3}	0.5061 _{c=100,10,τ=3}	0.5084 _{c=100,10,τ=3}	0.5076 _{c=100,10,τ=3}
p=90	0.5060 _{c=100,10,τ=1}	0.4981 _{c=0.01,τ=5}	0.5067 _{c=100,10,τ=2}	0.5109 _{c=100,10,τ=2}
p=120	0.5160 _{c=0.1,τ=3}	0.5019 _{c=100,10,τ=3}	0.5102 _{c=0.01,τ=5}	0.5195 _{c=0.1,τ=5}
p=240	0.5992 _{c=0.01,τ=5}	0.6152 _{c=0.01,τ=5}	0.6222 _{c=0.01,τ=6}	0.6243 _{c=0.01,τ=5}

Table 4: Telecommunications Mean Accuracy(per p)

τ/B	B=0.25	B=0.5	B=0.75	B=1
τ=90	0.5073 _{c=100,10,P=6}	0.4966 _{c=0.01,P=5}	0.5022 _{c=100,10,P=3}	0.5013 _{c=100,10,P=3}
τ=120	0.5011 _{c=100,10,P=4}	0.4986 _{c=100,10,P=5}	0.5167 _{c=100,10,P=4}	0.5109 _{c=100,10,P=4}
τ=180	0.5207 _{c=0.1,P=3}	0.5061 _{c=100,10,P=3}	0.5084 _{c=100,10,P=3}	0.5076 _{c=100,10,P=3}
τ=240	0.5213 _{c=0.01,P=6}	0.5370 _{c=0.01,P=6}	0.5429 _{c=0.01,P=6}	0.5475 _{c=0.01,P=6}
τ=300	0.5992 _{c=0.01,P=6}	0.6152 _{c=0.01,P=6}	0.6222 _{c=0.01,P=6}	0.6243 _{c=0.01,P=6}

Table 5: Telecommunications Mean Accuracy(per τ)

In these tables we can see the exact best mean accuracy for each B when p (table 4) and τ (table 5) varies. We notice that it is only for $B = 0.25, 0.5$ (table 4-5) when p varies that we get a lower mean accuracy than 50 per cent, and for $B = 0.5$ (table 5) when τ varies, all of them when $p < 100$. This tells us that when the algorithm tries to forecast a price movement close into the future it fails, and that it would perform worse than a simple random walk. This result also suggest that the Efficient Market Hypothesis(EMH) applies here, i.e. market prices should only react to new information, and that in this case past information don't tell us much about the future. We get the best mean accuracy(0.6243) when $B = 1$, $\tau = 240$ and $p = 240$. This tells us that the algorithm performs best with long momentum features predicting over a longer future period. Also notice that the best accuracy is when $B = 1$, which means that the $J - S$ is zero and we do not use any sector mean to improve our predictions. In these 2 tables we also have the penalization constant c . We notice that for the best mean accuracy we have $c = 0.01$ and, on average, the algorithm chose $c = 0.01$.

The next figure(figure 7) below presents the same type of result but for the portfolio *Basic Materials*, including the following 2 tables(table 6-7). This portfolio include 12 different stocks from the same sector. In the right graph in figure 7 we notice that the mean accuracy is very similar for each constant B when p varies. In the right graph however we notice some bigger differences for each B when τ varies. In this graph we notice that the best mean accuracy(76.33 per cent) is achieved when $B = 1$, $\tau = 180$ and $p = 260$. So again, the best performance is when the $J - S$ estimator is excluded from the algorithm, although this is not the case for all combinations. Also, when we try and predict the future price movement for different p , except when $p = 240$, the algorithm without the $J - S$ estimator only outperforms the others when $p = 5$ and when $p = 90$. So we have some improvement implementing the $J - S$ estimator for these time intervals. If we now look at table 6 – 7 we can also notice the algorithm chose the penalization factor to be 0.1 in the majority of times.

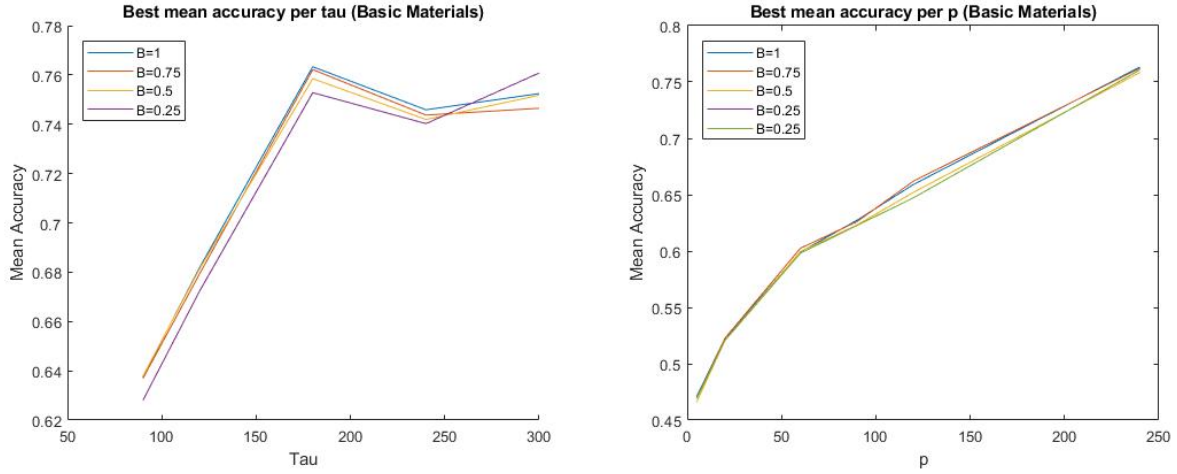


Figure 7: Mean Accuracy Basic Materials

p/B	B=0.25	B=0.5	B=0.75	B=1
p=5	0.4690 _{c=0.01,τ=4}	0.4660 _{c=0.01,τ=5}	0.4690 _{c=0.1,τ=5}	0.4708 _{c=0.1,τ=5}
p=20	0.5206 _{c=0.01,τ=4}	0.5207 _{c=0.1,τ=5}	0.5226 _{c=0.1,τ=4}	0.5224 _{c=0.1,τ=3}
p=60	0.5981 _{c=0.01,τ=3}	0.5996 _{c=0.1,τ=3}	0.6025 _{c=0.1,τ=3}	0.5992 _{c=0.1,τ=3}
p=90	0.6227 _{c=0.01,τ=4}	0.6232 _{c=0.1,τ=4}	0.6260 _{c=0.1,τ=4}	0.6273 _{c=0.1,τ=3}
p=120	0.46473 _{c=0.01,τ=4}	0.6522 _{c=0.1,τ=4}	0.6621 _{c=0.1,τ=4}	0.6591 _{c=0.1,τ=4}
p=240	0.7608 _{c=0.01,τ=5}	0.7585 _{c=0.01,τ=3}	0.7622 _{c=0.1,τ=3}	0.7633 _{c=0.1,τ=3}

Table 6: BasicMaterials Mean Accuracy(per p)

τ/B	B=0.25	B=0.5	B=0.75	B=1
τ=90	0.6280 _{c=0.01,P=6}	0.6378 _{c=0.1,P=6}	0.6369 _{c=0.1,P=6}	0.6374 _{c=0.1,P=6}
τ=120	0.6723 _{c=0.01,P=6}	0.6813 _{c=0.01,P=6}	0.6793 _{c=0.01,P=6}	0.6818 _{c=0.1,P=6}
τ=180	0.7528 _{c=0.01,P=6}	0.7585 _{c=0.01,P=6}	0.7622 _{c=0.1,P=6}	0.7633 _{c=0.1,P=6}
τ=240	0.7403 _{c=0.01,P=6}	0.7419 _{c=0.1,P=6}	0.7437 _{c=0.1,P=6}	0.7459 _{c=0.1,P=6}
τ=300	0.7608 _{c=0.01,P=6}	0.7518 _{c=0.01,P=6}	0.7465 _{c=0.1,P=6}	0.7524 _{c=0.1,P=6}

Table 7: BasicMaterials Mean Accuracy(per τ)

The last portfolio, *Healthcare*, is presented in figure 8 below, along with the following 2 tables(8-9). This portfolio contained 18 different stocks from the same sector and is the largest we examine in this paper. If we look at the right graph in figure 8 below we notice the mean accuracy don't differ much when p is varied, just like in the case for the *Basic Materials* portfolio. This

can also be seen in table 8. If we instead turn to the right graph in figure 8 we notice that the worst mean accuracy is achieved when $B = 0.25$, $\tau = 90$ and $p = 240$. Although this is the worst performance in this comparison we see that it still has an accuracy of 60.05 per cent. The best accuracy, however, is 70.70 per cent and is achieved for $B = 0.70$, $\tau = 300$ and $p = 240$. The penalization factor for the best accuracy was $c = 0.1$ and $c = 0.01$, where $c = 0.01$ was used most of the times. Notice that this was the first time the best mean accuracy was performed by other than $B = 1$, although it was shared. Also notice that for different values for p , $B = 1$ was only used as the best predictor 2 times, which means that the $J - S$ estimator improved the accuracy in the majority of times for different p .

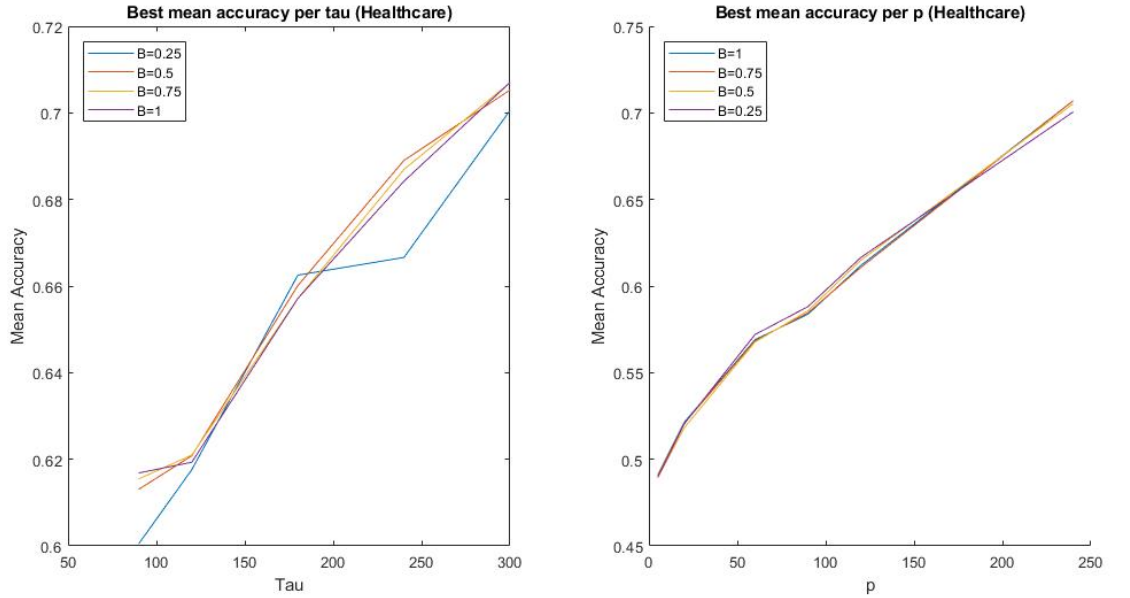


Figure 8: Mean Accuracy Healthcare

p/B	B=0.25	B=0.5	B=0.75	B=1
p=5	0.4894 _{c=100,τ=3}	0.4894 _{c=0.1,τ=4}	0.4905 _{c=100,τ=4}	0.4910 _{c=0.1,τ=4}
p=20	0.5205 _{c=0.01,τ=3}	0.5182 _{c=0.1,τ=4}	0.5209 _{c=0.1,τ=3}	0.5214 _{c=0.1,τ=3}
p=60	0.5720 _{c=0.01,τ=3}	0.5677 _{c=0.1,τ=3}	0.5648 _{c=0.1,τ=3}	0.5690 _{c=0.1,τ=3}
p=90	0.5881 _{c=0.01,τ=3}	0.5858 _{c=0.01,τ=3}	0.5846 _{c=0.1,τ=3}	0.5838 _{c=0.1,τ=3}
p=120	0.6166 _{c=0.01,τ=3}	0.6150 _{c=0.01,τ=3}	0.6109 _{c=0.1,τ=3}	0.6120 _{c=0.1,τ=3}
p=240	0.7006 _{c=0.01,τ=5}	0.7053 _{c=0.01,τ=5}	0.7070 _{c=0.01,τ=5}	0.7070 _{c=0.01,τ=5}

Table 8: Healthcare Mean Accuracy(per p)

τ/B	B=0.25	B=0.5	B=0.75	B=1
$\tau=90$	0.6005 _{$c=0.01, P=6$}	0.6131 _{$c=0.01, P=6$}	0.6155 _{$c=0.01, P=6$}	0.6168 _{$c=0.01, P=6$}
$\tau=120$	0.6176 _{$c=0.01, P=6$}	0.6208 _{$c=0.01, P=6$}	0.6210 _{$c=0.01, P=6$}	0.6193 _{$c=0.01, P=6$}
$\tau=180$	0.6625 _{$c=0.01, P=6$}	0.6601 _{$c=0.01, P=6$}	0.6571 _{$c=0.01, P=6$}	0.6572 _{$c=0.01, P=6$}
$\tau=240$	0.6666 _{$c=0.01, P=6$}	0.6890 _{$c=0.01, P=6$}	0.6869 _{$c=0.01, P=6$}	0.6842 _{$c=0.01, P=6$}
$\tau=300$	0.7006 _{$c=0.01, P=6$}	0.7053 _{$c=0.01, P=6$}	0.7070 _{$c=0.01, P=6$}	0.7070 _{$c=0.01, P=6$}

Table 9: Healthcare Mean Accuracy(per τ)

The Share Ratio was also calculated for each algorithm when B varied. The table below presents the best Sharpe Ratio for different B for each sector. Notice here that the S-R was highest for Healthcare, i.e. a portfolio of 18 different stocks, and the lowest for Basic Materials, 12 stocks. The best S-R was achieved for Healthcare when $B = 1$, $\tau = 180$, $p = 60$ and $c = 0.1$.

Sector	B=0.25	B=0.5	B=0.75	B=1
TC	0.0523 <small>$c=0.01, P=6, \tau=5$</small>	0.0556 <small>$c=0.01, P=6, \tau=5$</small>	0.0598 <small>$c=0.01, P=6, \tau=5$</small>	0.0602 <small>$c=0.01, P=6, \tau=5$</small>
B-M($*10^{-15}$)	4.19 <small>$c=0.01, P=6, \tau=5$</small>	4.19 <small>$c=0.01, P=6, \tau=3$</small>	4.19 <small>$c=0.1, P=6, \tau=3$</small>	4.19 <small>$c=0.1, P=6, \tau=3$</small>
HC	0.1197 <small>$c=0.01, P=3, \tau=3$</small>	0.1195 <small>$c=0.1, P=3, \tau=3$</small>	0.1185 <small>$c=0.1, P=3, \tau=3$</small>	0.1200 <small>$c=0.1, P=3, \tau=3$</small>

Table 10: Best performing Sharpe Ratio (per B)

8 Conclusion

In this paper the purpose was to try to implement the Machine learning technique called Support Vector Machine, SVM, on the stock market to try predict the future direction of a stocks price p days into the future based on a feature, momentum, containing the price movement m days into the past. This paper also tried to increase the performance of the SVM method by introducing the *James Stein* estimator, which uses the mean value of a population to improve the algorithm. The population is thought to share some common properties, thus we used the mean value of the momentum of stocks in the same sector. This was then applied to three different portfolios containing 4,12 and 18 stocks. As discussed the $J - S$ estimator only performed the best mean accuracy for $B = 0.75$, although it was shared with $B = 1$ in this case, for the Healthcare portfolio. Although this is the case we did see the performance increase for different p and τ when the $J - S$ estimator was introduced. As a matter of fact, the $J - S$ estimator improved the performance of the Healthcare portfolio on the majority of times for different values for p . For the Basic Materials portfolio the $J - S$ estimator improved the performance for half of the chosen values for p , and only 2 times for the Telecommunications portfolio, although this portfolio only contained 4 socks. Notice that by looking at the results in this paper, the $J - S$ inspired SVM method performed better against the normal SVM method as the number of stocks in the portfolio increased.

If we again look at the Sharpe Ratio in table 10 in the previous section we see only positive results. In reality the models also returned negative Sharpe Ratios. This normally indicates that a risk-less asset would perform better than the portfolio being analyzed. In our case we would interpret it as bringing us a negative return on our investment, hence for these cases the models fails as a portfolio strategy for these parameters.

9 Appendix

9.1 A

		c=100		c=10		c=0.1		c=0.01	
p	tau	mA	SR (E-15)	mA	SR (E-15)	mA	SR (E-15)	mA	SR (E-15)
1	90	0,46	-	0,46	-	0,47	-	0,46	-
5		0,50	-	0,50	-	0,50	-	0,49	-
20		0,54	-	0,54	-	0,54	-	0,53	-
60		0,54	0,97	0,54	0,00	0,56	2,27	0,55	2,27
120		0,55	1,70	0,55	1,70	0,58	1,98	0,57	2,10
240		0,57	4,19	0,57	4,19	0,61	4,19	0,63	2,46
1	120	0,46	-	0,46	-	0,46	-	0,45	-
5		0,50	-	0,50	-	0,50	-	0,48	-
20		0,54	-	0,54	-	0,53	-	0,51	-
60		0,56	1,62	0,56	1,62	0,56	2,27	0,55	2,27
120		0,56	1,84	0,56	1,84	0,59	1,07	0,59	1,07
240		0,57	4,19	0,57	4,19	0,63	4,19	0,67	4,19
1	180	0,46	-	0,46	-	0,46	-	0,46	-
5		0,52	-	0,52	-	0,52	-	0,50	-
20		0,56	-	0,56	-	0,58	-	0,60	-
60		0,56	2,27	0,56	2,27	0,59	2,27	0,62	2,27
120		0,55	1,98	0,55	1,98	0,60	1,98	0,64	1,78
240		0,59	4,19	0,59	4,19	0,67	4,19	0,75	4,19
1	240	0,46	-	0,46	-	0,46	-	0,47	-
5		0,51	-	0,51	-	0,51	-	0,52	-
20		0,52	-	0,52	-	0,54	-	0,59	-
60		0,53	2,27	0,53	2,27	0,58	2,27	0,62	-2,27
120		0,53	1,98	0,53	1,98	0,61	1,98	0,65	0,00
240		0,58	4,19	0,58	4,19	0,69	4,19	0,74	4,19
1	300	0,46	-	0,46	-	0,46	-	0,47	-
5		0,50	-	0,50	-	0,50	-	0,51	-
20		0,52	-	0,52	-	0,55	-	0,58	-
60		0,54	2,27	0,54	2,27	0,59	2,27	0,61	1,30
120		0,54	1,98	0,54	1,98	0,60	1,98	0,64	1,77
240		0,56	4,19	0,56	4,19	0,67	4,19	0,76	4,19
1	90	0,46	-	0,46	-	0,46	-	0,45	-
5		0,50	-	0,50	-	0,50	-	0,49	-
20		0,53	-	0,53	-	0,53	-	0,53	-
60		0,55	9,71	0,55	0,00	0,56	2,27	0,54	2,27

Continued...

		c=100	c=10	c=0.1	c=0.01		
120		0,56	1,70	0,56	2,06	0,56	2,10
240		0,58	4,19	0,58	4,19	0,63	-1,58
1	120	0,46	-	0,46	-	0,45	-
5		0,50	-	0,50	-	0,49	-
20		0,54	-	0,54	-	0,52	-
60		0,57	1,94	0,57	1,94	0,56	2,27
120		0,58	1,91	0,58	1,91	0,59	1,07
240		0,59	4,19	0,59	4,19	0,67	3,04
1	180	0,46	-	0,46	-	0,46	-
5		0,52	-	0,52	-	0,51	-
20		0,57	-	0,57	-	0,60	-
60		0,58	2,27	0,58	2,27	0,62	2,27
120		0,58	1,98	0,58	1,98	0,64	1,98
240		0,61	4,19	0,61	4,19	0,75	4,19
1	240	0,46	-	0,46	-	0,46	-
5		0,51	-	0,51	-	0,52	-
20		0,53	-	0,53	-	0,59	-
60		0,54	2,27	0,54	2,27	0,62	-2,27
120		0,54	1,98	0,54	1,98	0,65	0,00
240		0,60	4,19	0,60	4,19	0,74	4,19
1	300	0,46	-	0,46	-	0,46	-
5		0,50	-	0,50	-	0,52	-
20		0,53	-	0,53	-	0,58	-
60		0,55	2,27	0,55	2,27	0,61	0,00
120		0,55	1,98	0,55	1,98	0,64	1,56
240		0,58	4,19	0,58	4,19	0,74	4,19
1	90	0,46	-	0,46	-	0,46	-
5		0,50	-	0,50	-	0,49	-
20		0,53	-	0,53	-	0,53	-
60		0,55	1,94	0,55	1,94	0,55	2,27
120		0,57	1,91	0,57	1,91	0,57	2,10
240		0,59	4,19	0,59	4,19	0,64	4,19
1	120	0,46	-	0,46	-	0,45	-
5		0,50	-	0,50	-	0,49	-
20		0,54	-	0,54	-	0,51	-
60		0,57	2,27	0,57	2,27	0,56	2,27
120		0,59	1,98	0,59	1,98	0,59	1,07
240		0,60	4,19	0,60	4,19	0,68	4,19
1	180	0,46	-	0,46	-	0,46	-

Continued...

		c=100		c=10		c=0.1		c=0.01	
5		0,52	-	0,52	-	0,50	-	0,50	-
20		0,58	-	0,58	-	0,60	-	0,59	-
60		0,59	2,27	0,59	2,27	0,63	2,27	0,62	2,27
120		0,59	1,98	0,59	1,98	0,64	1,65	0,64	1,98
240		0,63	4,19	0,63	4,19	0,76	4,19	0,76	4,19
1	240	0,46	-	0,46	-	0,47	-	0,46	-
5		0,51	-	0,51	-	0,52	-	0,50	-
20		0,55	-	0,55	-	0,60	-	0,57	-
60		0,56	2,27	0,56	2,27	0,63	-2,27	0,61	-2,27
120		0,56	1,98	0,56	1,98	0,66	0,00	0,64	0,00
240		0,62	4,19	0,62	4,19	0,74	4,19	0,74	4,19
1	300	0,46	-	0,46	-	0,47	-	0,46	-
5		0,51	-	0,51	-	0,52	-	0,51	-
20		0,54	-	0,54	-	0,59	-	0,58	-
60		0,57	2,27	0,57	2,27	0,61	1,94	0,60	2,27
120		0,58	1,98	0,58	1,98	0,64	1,91	0,64	1,98
240		0,60	4,19	0,60	4,19	0,75	4,19	0,74	4,19
1	90	0,46	-	0,46	-	0,46	-	0,45	-
5		0,50	-	0,50	-	0,49	-	0,49	-
20		0,53	-	0,53	-	0,53	-	0,53	-
60		0,55	2,27	0,55	2,27	0,55	2,27	0,54	2,27
120		0,57	1,98	0,57	1,98	0,57	2,10	0,56	2,10
240		0,60	4,19	0,60	4,19	0,64	4,19	0,63	-2,74
1	120	0,46	-	0,46	-	0,45	-	0,45	-
5		0,50	-	0,50	-	0,49	-	0,49	-
20		0,54	-	0,54	-	0,51	-	0,50	-
60		0,56	2,27	0,56	2,27	0,55	2,27	0,54	2,27
120		0,59	1,98	0,59	1,98	0,59	1,07	0,58	1,07
240		0,60	4,19	0,60	4,19	0,68	4,19	0,68	-1,00
1	180	0,46	-	0,46	-	0,46	-	0,46	-
5		0,52	-	0,52	-	0,50	-	0,50	-
20		0,58	-	0,58	-	0,60	-	0,59	-
60		0,59	2,27	0,59	2,27	0,63	2,27	0,62	2,27
120		0,60	1,98	0,60	1,98	0,64	1,50	0,64	1,98
240		0,65	4,19	0,65	4,19	0,76	4,19	0,76	4,19
1	240	0,46	-	0,46	-	0,47	-	0,46	-
5		0,51	-	0,51	-	0,52	-	0,50	-
20		0,55	-	0,55	-	0,59	-	0,57	-
60		0,57	2,27	0,57	2,27	0,63	-2,27	0,61	-2,27

Continued...

		c=100		c=10		c=0.1		c=0.01	
120		0,58	1,98	0,58	1,98	0,66	0,00	0,63	0,00
240		0,64	4,19	0,64	4,19	0,75	4,19	0,73	4,19
1	300	0,46	-	0,46	-	0,47	-	0,46	-
5		0,51	-	0,51	-	0,52	-	0,51	-
20		0,55	-	0,55	-	0,59	-	0,57	-
60		0,58	2,27	0,58	2,27	0,62	2,27	0,60	2,27
120		0,59	1,98	0,59	1,98	0,64	1,98	0,64	1,98
240		0,63	4,19	0,63	4,19	0,75	4,19	0,74	4,19

Table 11: Result for Basic Materials

		c=100		c=10		c=0.1		c=0.01	
p	tau	mA	SR	mA	SR	mA	SR	mA	SR
5	90	0,48	0,02	0,48	0,02	0,47	-0,02	0,47	-0,06
20		0,49	0,02	0,49	0,02	0,49	0,01	0,49	-0,02
60		0,49	0,02	0,49	0,02	0,50	0,01	0,49	-0,02
90		0,51	0,03	0,51	0,03	0,48	0,01	0,48	-0,01
120		0,46	-0,03	0,46	-0,03	0,47	-0,01	0,49	0,00
240		0,51	0,02	0,51	0,02	0,48	0,01	0,46	-0,02
5	120	0,48	0,04	0,48	0,04	0,49	0,02	0,48	-0,05
20		0,49	0,03	0,49	0,03	0,50	0,03	0,50	0,02
60		0,49	0,02	0,49	0,02	0,49	-0,00	0,48	-0,05
90		0,50	0,01	0,50	0,01	0,49	-0,00	0,50	-0,00
120		0,50	0,03	0,50	0,03	0,50	0,02	0,49	-0,01
240		0,49	0,01	0,49	0,01	0,47	0,00	0,47	-0,01
5	180	0,47	-0,03	0,47	-0,03	0,49	-0,01	0,49	-0,01
20		0,47	-0,02	0,47	-0,02	0,50	0,02	0,49	-0,01
60		0,48	0,01	0,48	0,01	0,52	0,05	0,49	-0,01
90		0,48	-0,03	0,48	-0,03	0,49	0,00	0,47	-0,05
120		0,50	0,01	0,50	0,01	0,52	0,05	0,48	-0,02
240		0,48	-0,02	0,48	-0,02	0,49	-0,00	0,48	-0,01
5	240	0,49	-0,00	0,49	-0,00	0,49	-0,01	0,49	-0,01
20		0,48	-0,00	0,48	-0,00	0,49	0,00	0,50	-0,00
60		0,49	-0,03	0,49	-0,03	0,48	-0,02	0,49	-0,01
90		0,49	-0,01	0,49	-0,01	0,48	0,02	0,48	0,01
120		0,47	-0,02	0,47	-0,02	0,49	0,00	0,49	-0,00
240		0,50	0,01	0,50	0,01	0,52	0,02	0,52	0,01
5	300	0,48	-0,02	0,48	-0,02	0,49	-0,01	0,48	-0,05
20		0,47	-0,02	0,47	-0,02	0,49	-0,00	0,50	-0,01

Continued. . .

		c=100		c=10		c=0.1		c=0.01	
60		0,47	-0,06	0,47	-0,06	0,50	-0,00	0,50	0,01
90		0,46	-0,07	0,46	-0,07	0,47	-0,02	0,50	0,01
120		0,47	-0,03	0,47	-0,03	0,49	-0,01	0,51	0,01
240		0,51	-0,00	0,51	-0,00	0,53	0,02	0,60	0,05
5	90	0,48	-0,00	0,48	-0,00	0,47	-0,05	0,48	-0,06
20		0,49	0,00	0,49	0,00	0,49	-0,01	0,49	-0,01
60		0,50	0,02	0,50	0,02	0,48	-0,02	0,49	-0,02
90		0,49	0,02	0,49	0,02	0,47	-0,01	0,49	-0,01
120		0,48	-0,00	0,48	-0,00	0,49	-0,00	0,50	0,00
240		0,49	0,00	0,49	0,00	0,47	-0,00	0,47	-0,02
5	120	0,49	0,02	0,49	0,02	0,48	-0,03	0,48	-0,04
20		0,50	0,03	0,50	0,03	0,49	0,01	0,49	0,01
60		0,49	0,01	0,49	0,01	0,47	-0,04	0,48	-0,05
90		0,49	0,00	0,49	0,00	0,49	-0,01	0,50	-0,01
120		0,50	0,03	0,50	0,03	0,48	-0,00	0,50	-0,01
240		0,48	-0,00	0,48	-0,00	0,47	-0,00	0,47	-0,02
5	180	0,49	-0,01	0,49	-0,01	0,48	-0,01	0,49	-0,02
20		0,48	-0,02	0,48	-0,02	0,49	0,01	0,49	-0,00
60		0,51	0,03	0,51	0,03	0,49	-0,01	0,50	-0,01
90		0,49	-0,01	0,49	-0,01	0,47	-0,03	0,47	-0,04
120		0,50	0,03	0,50	0,03	0,49	-0,01	0,49	-0,02
240		0,50	-0,01	0,50	-0,01	0,47	-0,01	0,48	-0,01
5	240	0,49	-0,02	0,49	-0,02	0,49	-0,01	0,50	-0,00
20		0,48	-0,01	0,48	-0,01	0,49	-0,02	0,50	0,01
60		0,48	-0,04	0,48	-0,04	0,49	0,00	0,49	-0,01
90		0,48	-0,01	0,48	-0,01	0,48	0,01	0,48	0,01
120		0,48	0,01	0,48	0,01	0,48	-0,00	0,50	0,00
240		0,53	0,02	0,53	0,02	0,53	0,02	0,54	0,02
5	300	0,48	-0,04	0,48	-0,04	0,49	-0,02	0,49	-0,03
20		0,48	-0,03	0,48	-0,03	0,50	-0,01	0,50	-0,00
60		0,49	-0,02	0,49	-0,02	0,50	0,00	0,51	0,02
90		0,47	-0,04	0,47	-0,04	0,48	-0,02	0,50	0,02
120		0,48	-0,00	0,48	-0,00	0,50	0,00	0,49	-0,00
240		0,52	0,01	0,52	0,01	0,55	0,03	0,62	0,06
5	90	0,47	-0,03	0,47	-0,03	0,47	-0,05	0,48	-0,05
20		0,48	-0,02	0,48	-0,02	0,48	-0,02	0,49	-0,01
60		0,50	0,03	0,50	0,03	0,49	-0,02	0,49	-0,02
90		0,48	0,01	0,48	0,01	0,48	-0,00	0,49	-0,01
120		0,49	0,01	0,49	0,01	0,49	0,00	0,50	0,00

Continued...

		c=100		c=10		c=0.1		c=0.01	
240		0,48	-0,01	0,48	-0,01	0,47	-0,01	0,47	-0,02
5	120	0,49	0,01	0,49	0,01	0,48	-0,04	0,48	-0,04
20		0,50	0,02	0,50	0,02	0,49	0,01	0,49	0,01
60		0,50	0,02	0,50	0,02	0,47	-0,05	0,48	-0,04
90		0,51	0,02	0,51	0,02	0,49	-0,01	0,49	-0,02
120		0,49	0,02	0,49	0,02	0,49	-0,01	0,50	-0,01
240		0,47	-0,00	0,47	-0,00	0,47	-0,01	0,47	-0,01
5	180	0,49	0,00	0,49	0,00	0,49	-0,01	0,49	-0,03
20		0,49	-0,00	0,49	-0,00	0,49	-0,01	0,49	0,00
60		0,51	0,03	0,51	0,03	0,49	-0,02	0,49	-0,02
90		0,48	-0,02	0,48	-0,02	0,47	-0,04	0,47	-0,05
120		0,50	0,03	0,50	0,03	0,48	-0,02	0,49	-0,02
240		0,48	-0,01	0,48	-0,01	0,47	-0,01	0,49	-0,01
5	240	0,50	-0,00	0,50	-0,00	0,49	-0,01	0,50	0,02
20		0,49	-0,03	0,49	-0,03	0,49	-0,02	0,50	0,02
60		0,46	-0,06	0,46	-0,06	0,49	-0,00	0,49	-0,02
90		0,47	-0,02	0,47	-0,02	0,49	0,02	0,48	0,01
120		0,48	0,00	0,48	0,00	0,49	0,01	0,51	0,00
240		0,53	0,02	0,53	0,02	0,52	0,01	0,54	0,02
5	300	0,48	-0,04	0,48	-0,04	0,48	-0,04	0,49	-0,03
20		0,49	-0,03	0,49	-0,03	0,50	-0,02	0,50	-0,00
60		0,49	-0,00	0,49	-0,00	0,50	0,01	0,50	0,01
90		0,47	-0,03	0,47	-0,03	0,49	0,00	0,51	0,04
120		0,49	0,01	0,49	0,01	0,51	0,02	0,50	0,01
240		0,53	0,02	0,53	0,02	0,57	0,04	0,62	0,06
5	90	0,48	-0,03	0,48	-0,03	0,47	-0,06	0,48	-0,05
20		0,48	-0,02	0,48	-0,02	0,48	-0,03	0,50	-0,01
60		0,50	0,03	0,50	0,03	0,48	-0,02	0,49	-0,02
90		0,49	0,01	0,49	0,01	0,48	-0,00	0,49	-0,02
120		0,49	0,02	0,49	0,02	0,50	0,01	0,50	-0,00
240		0,48	-0,00	0,48	-0,00	0,47	-0,01	0,47	-0,02
5	120	0,49	-0,01	0,49	-0,01	0,48	-0,05	0,48	-0,04
20		0,50	0,03	0,50	0,03	0,49	0,01	0,50	0,01
60		0,50	0,02	0,50	0,02	0,47	-0,05	0,48	-0,04
90		0,51	0,02	0,51	0,02	0,50	-0,01	0,49	-0,02
120		0,50	0,02	0,50	0,02	0,49	-0,01	0,50	-0,01
240		0,47	-0,00	0,47	-0,00	0,47	-0,01	0,47	-0,02
5	180	0,49	-0,01	0,49	-0,01	0,49	-0,01	0,49	-0,03
20		0,49	-0,01	0,49	-0,01	0,48	-0,03	0,49	0,00

Continued...

		c=100		c=10		c=0.1		c=0.01	
60		0,51	0,03	0,51	0,03	0,49	-0,02	0,49	-0,02
90		0,48	-0,02	0,48	-0,02	0,47	-0,05	0,47	-0,05
120		0,50	0,02	0,50	0,02	0,48	-0,03	0,49	-0,02
240		0,47	-0,01	0,47	-0,01	0,47	-0,02	0,49	-0,00
5	240	0,50	-0,01	0,50	-0,01	0,49	-0,01	0,50	0,03
20		0,48	-0,04	0,48	-0,04	0,50	-0,01	0,50	0,02
60		0,46	-0,05	0,46	-0,05	0,49	-0,00	0,49	-0,02
90		0,46	-0,03	0,46	-0,03	0,49	0,02	0,48	0,01
120		0,48	0,01	0,48	0,01	0,50	0,01	0,51	0,00
240		0,52	0,01	0,52	0,01	0,52	0,01	0,55	0,03
5	300	0,48	-0,05	0,48	-0,05	0,48	-0,04	0,49	-0,02
20		0,49	-0,04	0,49	-0,04	0,49	-0,03	0,50	-0,01
60		0,49	0,01	0,49	0,01	0,50	0,02	0,49	0,00
90		0,46	-0,03	0,46	-0,03	0,50	0,02	0,51	0,04
120		0,50	0,02	0,50	0,02	0,52	0,04	0,50	0,02
240		0,53	0,02	0,53	0,02	0,57	0,04	0,62	0,06

Table 12: Result for Telecommunications

		c=100		c=10		c=0.1		c=0.01	
p	tau	mA	SR	mA	SR	mA	SR	mA	SR
5	90	0,48	0,04	0,48	0,04	0,48	0,04	0,48	0,04
20		0,49	0,03	0,49	0,03	0,50	0,05	0,50	0,06
60		0,51	0,04	0,52	0,04	0,53	0,08	0,54	0,08
90		0,52	0,03	0,52	0,04	0,56	0,08	0,57	0,08
120		0,52	0,03	0,52	0,03	0,58	0,07	0,60	0,08
240		0,52	0,02	0,52	0,02	0,57	0,04	0,60	0,05
5	120	0,48	0,05	0,48	0,05	0,49	0,05	0,49	0,09
20		0,51	0,05	0,51	0,05	0,51	0,05	0,50	0,05
60		0,53	0,05	0,52	0,05	0,52	0,07	0,53	0,06
90		0,52	0,05	0,52	0,05	0,57	0,09	0,59	0,10
120		0,51	0,04	0,51	0,04	0,59	0,08	0,61	0,09
240		0,54	0,03	0,54	0,03	0,59	0,05	0,62	0,05
5	180	0,49	0,05	0,49	0,05	0,49	0,05	0,48	0,06
20		0,51	0,05	0,51	0,05	0,51	0,07	0,52	0,10
60		0,53	0,06	0,53	0,06	0,54	0,08	0,57	0,12
90		0,52	0,04	0,52	0,04	0,56	0,08	0,59	0,10
120		0,51	0,03	0,51	0,03	0,58	0,07	0,62	0,08
240		0,53	0,04	0,53	0,04	0,61	0,06	0,66	0,06

Continued...

		c=100		c=10		c=0.1		c=0.01	
5	240	0,48	0,03	0,48	0,03	0,49	0,04	0,49	0,07
20		0,50	0,04	0,50	0,04	0,51	0,07	0,52	0,10
60		0,52	0,06	0,52	0,06	0,52	0,08	0,53	0,08
90		0,51	0,05	0,51	0,05	0,54	0,08	0,55	0,08
120		0,51	0,05	0,51	0,05	0,56	0,09	0,57	0,07
240		0,50	0,05	0,50	0,05	0,61	0,05	0,67	0,06
5	300	0,49	0,04	0,49	0,04	0,49	0,04	0,48	0,04
20		0,51	0,06	0,51	0,06	0,51	0,07	0,50	0,02
60		0,51	0,06	0,51	0,06	0,54	0,08	0,53	0,09
90		0,51	0,07	0,51	0,07	0,56	0,08	0,56	0,09
120		0,51	0,06	0,51	0,06	0,56	0,07	0,58	0,09
240		0,49	-0,00	0,49	-0,00	0,64	0,06	0,70	0,05
5	90	0,48	0,04	0,48	0,04	0,48	0,04	0,48	0,02
20		0,49	0,03	0,49	0,03	0,50	0,06	0,50	0,04
60		0,52	0,05	0,52	0,05	0,53	0,07	0,53	0,06
90		0,52	0,04	0,52	0,04	0,57	0,08	0,57	0,07
120		0,53	0,04	0,53	0,04	0,59	0,08	0,59	0,08
240		0,53	0,03	0,53	0,03	0,59	0,05	0,61	0,05
5	120	0,48	0,05	0,48	0,05	0,49	0,05	0,48	0,07
20		0,51	0,05	0,51	0,05	0,50	0,05	0,50	0,03
60		0,53	0,06	0,53	0,06	0,53	0,07	0,53	0,05
90		0,52	0,06	0,52	0,06	0,59	0,10	0,58	0,09
120		0,53	0,06	0,53	0,06	0,61	0,09	0,61	0,08
240		0,55	0,04	0,55	0,04	0,60	0,05	0,62	0,05
5	180	0,49	0,05	0,49	0,05	0,48	0,05	0,48	0,05
20		0,51	0,06	0,51	0,06	0,52	0,10	0,52	0,11
60		0,53	0,06	0,53	0,06	0,57	0,12	0,57	0,12
90		0,52	0,05	0,52	0,05	0,58	0,10	0,59	0,11
120		0,51	0,04	0,51	0,04	0,60	0,08	0,62	0,08
240		0,55	0,05	0,55	0,05	0,65	0,06	0,66	0,06
5	240	0,49	0,04	0,49	0,04	0,49	0,06	0,49	0,07
20		0,51	0,06	0,51	0,06	0,52	0,10	0,52	0,07
60		0,52	0,07	0,52	0,07	0,53	0,08	0,54	0,07
90		0,52	0,07	0,52	0,07	0,55	0,08	0,56	0,09
120		0,52	0,08	0,52	0,08	0,57	0,07	0,58	0,07
240		0,53	0,06	0,53	0,06	0,65	0,05	0,69	0,06
5	300	0,48	0,04	0,48	0,04	0,49	0,05	0,48	0,03
20		0,51	0,06	0,51	0,06	0,51	0,05	0,50	0,02
60		0,51	0,06	0,51	0,06	0,54	0,08	0,52	0,06

Continued...

		c=100		c=10		c=0.1		c=0.01	
90		0,51	0,07	0,51	0,07	0,56	0,09	0,55	0,08
120		0,52	0,07	0,52	0,07	0,57	0,07	0,59	0,09
240		0,52	0,04	0,52	0,04	0,68	0,06	0,71	0,05
5	90	0,48	0,04	0,48	0,04	0,48	0,04	0,47	0,01
20		0,50	0,04	0,50	0,04	0,50	0,05	0,50	0,03
60		0,52	0,05	0,52	0,06	0,53	0,07	0,53	0,06
90		0,53	0,06	0,53	0,06	0,57	0,08	0,57	0,07
120		0,54	0,05	0,54	0,05	0,59	0,08	0,59	0,08
240		0,53	0,03	0,54	0,03	0,60	0,05	0,62	0,05
5	120	0,48	0,05	0,48	0,05	0,49	0,08	0,48	0,05
20		0,51	0,05	0,51	0,05	0,50	0,05	0,50	0,03
60		0,52	0,07	0,52	0,07	0,53	0,06	0,53	0,05
90		0,53	0,08	0,53	0,08	0,58	0,10	0,57	0,09
120		0,54	0,07	0,54	0,07	0,60	0,09	0,61	0,08
240		0,55	0,05	0,55	0,05	0,61	0,05	0,62	0,05
5	180	0,49	0,05	0,49	0,05	0,49	0,07	0,48	0,06
20		0,51	0,06	0,51	0,06	0,52	0,10	0,51	0,10
60		0,54	0,07	0,54	0,07	0,57	0,12	0,56	0,11
90		0,53	0,06	0,53	0,06	0,58	0,10	0,58	0,10
120		0,53	0,05	0,53	0,05	0,61	0,08	0,61	0,08
240		0,56	0,05	0,56	0,05	0,65	0,06	0,66	0,06
5	240	0,49	0,05	0,49	0,05	0,49	0,06	0,48	0,05
20		0,51	0,06	0,51	0,06	0,52	0,10	0,51	0,06
60		0,52	0,08	0,52	0,08	0,53	0,07	0,54	0,07
90		0,52	0,08	0,52	0,08	0,56	0,08	0,56	0,09
120		0,53	0,08	0,53	0,08	0,56	0,07	0,58	0,07
240		0,55	0,06	0,55	0,06	0,66	0,06	0,69	0,06
5	300	0,48	0,04	0,48	0,04	0,49	0,05	0,48	0,01
20		0,51	0,06	0,51	0,06	0,51	0,04	0,50	0,01
60		0,51	0,06	0,51	0,06	0,53	0,08	0,52	0,05
90		0,52	0,08	0,52	0,07	0,55	0,08	0,54	0,08
120		0,53	0,08	0,53	0,08	0,57	0,07	0,59	0,08
240		0,53	0,05	0,54	0,05	0,69	0,05	0,71	0,05
5	90	0,48	0,03	0,48	0,03	0,48	0,04	0,47	0,00
20		0,50	0,04	0,50	0,04	0,50	0,04	0,50	0,03
60		0,53	0,07	0,53	0,07	0,53	0,07	0,52	0,05
90		0,54	0,07	0,54	0,07	0,57	0,08	0,57	0,07
120		0,56	0,06	0,56	0,06	0,59	0,08	0,59	0,08
240		0,55	0,04	0,55	0,04	0,60	0,05	0,62	0,06

Continued...

		c=100		c=10		c=0.1		c=0.01	
5	120	0,49	0,05	0,49	0,05	0,49	0,08	0,48	0,05
20		0,51	0,05	0,51	0,05	0,50	0,05	0,50	0,03
60		0,53	0,08	0,53	0,08	0,53	0,05	0,52	0,05
90		0,55	0,09	0,55	0,09	0,58	0,10	0,57	0,08
120		0,56	0,07	0,56	0,07	0,61	0,09	0,61	0,08
240		0,56	0,05	0,56	0,05	0,61	0,05	0,62	0,05
5	180	0,49	0,06	0,49	0,06	0,49	0,06	0,48	0,05
20		0,52	0,08	0,52	0,08	0,52	0,11	0,51	0,10
60		0,54	0,08	0,54	0,08	0,57	0,12	0,56	0,11
90		0,54	0,07	0,54	0,07	0,58	0,10	0,58	0,10
120		0,54	0,06	0,54	0,06	0,61	0,08	0,61	0,08
240		0,56	0,05	0,57	0,05	0,65	0,06	0,66	0,06
5	240	0,49	0,05	0,49	0,05	0,49	0,07	0,48	0,04
20		0,51	0,07	0,51	0,07	0,52	0,09	0,51	0,05
60		0,53	0,09	0,53	0,09	0,53	0,07	0,54	0,07
90		0,53	0,09	0,53	0,09	0,56	0,09	0,57	0,09
120		0,54	0,08	0,54	0,08	0,56	0,07	0,58	0,07
240		0,57	0,06	0,57	0,06	0,67	0,06	0,68	0,06
5	300	0,48	0,04	0,48	0,04	0,49	0,05	0,48	0,01
20		0,51	0,07	0,51	0,07	0,51	0,05	0,50	0,01
60		0,52	0,07	0,52	0,07	0,53	0,07	0,52	0,05
90		0,53	0,08	0,53	0,08	0,54	0,07	0,54	0,08
120		0,54	0,08	0,54	0,08	0,57	0,07	0,59	0,08
240		0,56	0,05	0,56	0,05	0,69	0,05	0,71	0,06

Table 13: Result for Healthcare

9.2 B

Matlab code

Example: Separable data set

```

1 clear all;clc
2
3
4 %create data
5 n = 20; % sample size up to 200000 !
6 rand('seed',2); % fix the randomness
7 Xi = 4*rand(n,2); % build the training set

```

```

8 q = 0; % add useless variables
9 Xi = [Xi 4*rand(n,q)];
10 [n,p] = size(Xi);
11 plot(Xi(:,1),Xi(:,2),'or')
12
13 %make data linearly separable
14 bt = -6; % define the separation line bias
15 wt = [4 ; -1]; % define the separation line vector
16 yi = sign(wt(1) * Xi(:,1) + wt(2) * Xi(:,2) + bt);
17 hold on;
18 plot(Xi(find(yi==1),1),Xi(find(yi==1),2),'ob');
19
20 %draw separating line
21 x1 = 0;
22 y1 = (-bt-(wt(1)*x1))/wt(2);
23 x2 = 4;
24 y2 = (-bt-(wt(1)*x2))/wt(2);
25 plot([x1 x2],[y1 y2],'g','LineWidth',2)
26
27
28
29 %solve minimization problem
30 cvx_begin
31 variables w(p) b
32 minimize( .5*w'*w )
33 subject to
34 yi.*(Xi*w + b) >= 1;
35 cvx_end
36
37
38
39 %Find the indices of the support vectors, and count
    them
40 vec_sup = find(yi.*(Xi*w + b) <= 1+eps^.3);
41 length(vec_sup)
42
43
44 %Draw the separating hyperplane found by the max margin
45 %SVM and the associated margin and support vectors
46
47 x1 = 0; % left bound

```



```

48 y1 = (-b-(w(1)*x1))/w(2); % the separating hyperplane
49 z1 = (1-b-(w(1)*x1))/w(2); % the margin
50 zm1 = (-1-b-(w(1)*x1))/w(2); % the other margin
51 x2 = 4; % right bound
52 y2 = (-b-(w(1)*x2))/w(2); % the separating hyperplane
53 z2 = (1-b-(w(1)*x2))/w(2); % the margin
54 zm2 = (-1-b-(w(1)*x2))/w(2); % the other margin
55 h = plot([x1 x2],[y1 y2], 'k', 'LineWidth', 2);
56 plot([x1 x2],[z1 z2], ':k'); % the margin
57 plot([x1 x2],[zm1 zm2], ':k'); % the other margin
58 plot(Xi(vec_sup,1),Xi(vec_sup,2), 'sm', 'MarkerSize', 10);

```

Example: Non-separable data set

```

1 clear all;
2 clc;
3
4 n = 10; %training set
5 k = 2; %Dimension of input variable
6 mean = 1.75; %swifted r.v. for class -1.
7
8 %Generate non-separable data set
9 randn('seed',k);
10 one = ones(1,k);
11 xi1 = randn(n,k);
12 xi2 = randn(n,k) + ones(n,1)*one*mean;
13 Xi = [xi1 ; xi2];
14 [m p] = size(Xi);
15 Yi = [ones(n,1) ; -ones(n,1)];
16
17 %plot class points
18 plot(Xi(:,1), Xi(:,2), 'or');
19 hold on
20 plot(Xi(find(Yi == 1),1), Xi(find(Yi == 1),2), 'ob');
21
22
23 %Solve min problem with slack variables
24 C = 10; %slack constraint
25
26 cvx_begin
27 variables w(p) b e(m)

```

```

28 dual variables pi px
29 minimize (0.5*w'*w + C*sum(e))
30 subject to
31 pi : Yi.*(Xi*w + b) >= 1 - e;
32 px : e >= 0;
33 cvx_end
34
35
36 %Plot hyperplane
37 xlimit = xlim;
38 xmin = xlimit(1);
39 xmax = xlimit(2);
40 x1 = xmin;
41 x2 = xmax;
42 y1 = (-b-w(1)*x1)/w(2);
43 y2 = (-b-w(1)*x2)/w(2);
44 plot([x1 x2],[y1 y2], 'k')
45
46 %calc and plot margins
47 z1 = (1-b-w(1)*x1)/w(2); %margin
48 z1m = (-1-b-w(1)*x1)/w(2); %other margin
49 z2 = (1-b-w(1)*x2)/w(2); %margin
50 z2m = (-1-b-w(1)*x2)/w(2); %other margin
51 plot([x1 x2], [z1 z2], ':k')
52 plot([x1 x2], [z1m z2m], ':k')
53
54 %find support vectors
55 A = find(round(1000*Yi.*(Xi*w + b)) == 1000);
56 cA = length(A);
57
58 %find the points inside/wrongside of margin
59 B = find(round(1000*Yi.*(Xi*w + b)) < 1000);
60 cb = length(B);

```

Example: k-dimensional case

```

1 clear all;
2 clc
3
4 n = 10; %training set
5 k = 20; %Dimension of input variable

```

```

6 mean = 1.75; %mean value for class -1
7 %Generate non-separable data set
8 randn('seed',k);
9 x11 = randn(n,k);
10 xi2 = randn(n,k) + ones(n,1)*ones(1,k)*mean;
11 Xi = [x11 ; xi2];
12 [m p] = size(Xi);
13
14 Yi = [ones(n,1) ; -ones(n,1)];
15
16
17 %Solve min problem with slack variables
18 C = 1; %slack constraint
19
20 cvx_begin
21 variables w(p) b e(m)
22 dual variables pi px
23 minimize (0.5*w'*w + C*sum(e))
24 subject to
25 pi : Yi.*(Xi*w + b) >= 1 - e;
26 px : e >= 0;
27 cvx_end
28
29 %find support vectors
30 A = find(round(1000*Yi.*(Xi*w + b)) == 1000);
31 cA = length(A);
32
33 %find the points inside/wrongside of margin
34 B = find(round(1000*Yi.*(Xi*w + b)) < 1000);
35 cb = length(B);

```

Code for James-Steine type estimator

In this example we compare the following loss function

$$L = \max(0, y_i * (\beta(x_i - a * \bar{x}) + \beta_0)) \quad (58)$$

for different a with correlation within the 2-dimensional points, i.e. $x_1 = [x_{1,1}, x_{1,2}]$ where $x_{1,1}$ and $x_{1,2}$ are correlated.

```

1 clear all;
2 clc
3

```

```

4 n = 10; %Number of obs for each class, [-1,1]
5
6
7 %Generate data
8 B1 = [0;0.5]; %Mean vector for class 1
9 B2 = [1.5;2]; %Mean vector for class -1
10 A2 = [0.5, 1; 0.5, 2]; %Covariance matrix for class 1
11 A1 = [1, 0.3; 1, 1]; %Covariance matrix for class -1
12 xi1 = randn(2,n); %Normal r.v's
13 xi2 = randn(2,n); %Normal r.v's
14 xi1 = (B1*ones(1,10) + A1*xi1)';
15 xi2 = (B2*ones(1,10) + A2*xi2)';
16
17 %Plot points
18 plot(xi1(:,1),xi1(:,2), 'or');
19 hold on
20 plot(xi2(:,1),xi2(:,2), 'ob');
21
22 xi = [xi1;xi2];
23 yi = [ones(10,1);-ones(10,1)]; %Define classes
24 [m1 n1] = size(xi);
25 xi_ = sum(xi,2)*(1/n1); %calc mean
26 c = 10; %trad-off variable
27 a = 0; %constants in [0,1]
28 a = repmat(a,m1,n1);
29
30 cvx_begin
31 variables w(n1) b e(m1)
32 minimize (0.5*w'*w + c*sum(e));
33 subject to
34 yi.*((xi - a.*repmat(xi_,1,n1))*w + b) >= 1 - e;
35 e >= 0;
36 cvx_end
37
38 Loss = zeros(2,1); %pre-allocate
39 Loss(1,1) = 0.5*w'*w + sum(max(0,yi.*((xi - a.*repmat(
      xi_,1,n1))*w + b)));
40
41 a1 = 0.5;
42 a1 = repmat(a1,m1,n1);
43

```

```

44 cvx_begin
45 variables w1(n1) b1 e1(m1)
46 minimize (0.5*w1'*w1 + c*sum(e1));
47 subject to
48 yi.*((xi - a1.*repmat(xi_ ,1 ,n1))*w1 + b1) >= 1 - e1;
49 e1 >= 0;
50 cvx_end
51
52 Loss(2,1) = 0.5*w1'*w1 + sum(max(0, yi.*((xi - a1.*
53     repmat(xi_ ,1 ,n1))*w...
54     + b1)));
55 Loss %where (1,1) corresponds to a=0 and (2,1) to a>0

```

Matlab code

Main script

```

1 clear all;
2 clc
3
4 %Call scripts
5 data1;
6
7
8 %Define input variables for input1-script
9
10 %Momentum is calculated as the logarithmic difference
11     between the closing
12 %price at time t and time t-tao, tao > 0.
13 tau_ = [90,120,180,240,300];
14 %define the penaltyconstant c
15 c_ = [100,10,0.1,0.01];
16 %Define forecasting days
17 p_ = [5,20,60,90,120,240];
18 %B_ = [0,0.25,0.5,0.75,1];
19 %B = B_(1);
20 %pre-define

```

```

21 Sharperatio_ = zeros(length(p_), length(c_)*length(tau_
    ));
22 meanAccuracy_ = zeros(length(p_), length(c_)*length(
    tau_));
23 meanReturn_ = zeros(length(p_), length(c_)*length(tau_
    ));
24
25
26 %Define input variables for run-script
27
28 %define how many features we need to start
29 %estimating the model
30 begin = 240;
31 %re-estimate the SVM model every kth day
32 k = 120;
33 %Estimate std for price difference
34 d = 60; %number of passed price diff
35 s = 30; %number of days until re-estimation
36
37
38 t = 1;
39 for u = 1:length(c_);
40     c = c_(u);
41     q = 1;
42     for q = 1:length(tau_);
43         tau = tau_(q);
44
45         v = 1;
46         for v = 1:length(p_);
47             p = p_(v);
48             input2;
49             run1;
50             Sharperatio_(v,t) = Sharperatio;
51             meanAccuracy_(v,t) = meanAccuracy;
52             meanReturn_(v,t) = meanReturn;
53         end
54         t = t+1;
55     end
56
57
58 end

```

```

59
60
61 %Divide results into groups depending on tau, i.e. tau
    =[180,240,300,340]
62 %Note that every column in these matrices represents
    the following order of
63 %tau1, [30,60,90,120]. P varies in every row and is the
    same in each column
64 %for a certain row.
65
66 %Sharperatio100 = Sharperatio_(:,1);
67 %Sharperatio10 = Sharperatio_(:,2);
68 %Sharperatio1 = Sharperatio_(:,3);
69 %Sharperatio01 = Sharperatio_(:,4);
70 %Sharperatio001 = Sharperatio_(:,5);
71
72 %meanAccuracy100 = meanAccuracy_(:,1);
73 %meanAccuracy10 = meanAccuracy_(:,2);
74 %meanAccuracy1 = meanAccuracy_(:,3);
75 %meanAccuracy01 = meanAccuracy_(:,4);
76 %meanAccuracy001 = meanAccuracy_(:,5);
77
78
79 %meanReturn100 = meanReturn_(:,1);
80 %meanReturn10 = meanReturn_(:,2);
81 %meanReturn1 = meanReturn_(:,3);
82 %meanReturn01 = meanReturn_(:,4);
83 %meanReturn001 = meanReturn_(:,5);
84
85 save('HealthcareNROMAL')

```

Import data

```

1 clear all;
2 clc
3
4 %Import data from Excel
5 cd /Users/sebastian_sjoholm/Desktop;
6 [Orgdata, ~]=xlsread('OMX_stocks.xlsx',...
7     'Aktier1', 'A5:LK3191');
8 [~, Sektor]=xlsread('OMX_stocks.xlsx',...

```

```

9         'Sektortillhorighet');
10 cd /Users/sebastian_sjoholm/Documents/MATLAB;
11
12 %extract data
13 Sektor = Sektor(2:end,2:end);
14 C = Orgdata(1:end,2:end); %Closing prices , removing
    dates
15
16 %Remove all stock which begins/ends with Nan/zero so we
    no missing obs
17 ind = isnan(C(1,:));
18 C(:,[ind]) = [];
19 Sektor([ind],:) = [];
20 ind1 = find(C(end,:)==0);
21 C(:,[ind1]) = [];
22 Sektor([ind1],:) = [];
23
24
25 %set all missing/zero values equal to closest passed
    registered value
26 ind2 = find(C==0);
27 for i = 1:length(ind2)
28     C(ind2(i)) = C(ind2(i)-1);
29 end
30
31
32 %Find indicator for each group, without ''Nan stocks''
33 [G,GN] = grp2idx(Sektor(:,2)); %GN is sektors and G is
    classification
34
35
36 %Find indicator vector for each group
37 Financials = find(G == 1);
38 ConsumerServices = find(G == 2);
39 Industrials = find(G == 3);
40 Technology = find(G == 4);
41 ConsumerGoods = find(G == 5);
42 Telecommunications = find(G == 6);
43 BasicMaterials = find(G == 7);
44 OilandGas = find(G == 8);
45 Healthcare = find(G == 9);

```



```

46 Utilities = find(G == 10);
47 Unclassified = find(G == 11);
48
49
50 %Choose portfolio , sector
51 C1 = C(:, Healthcare);

```

Create Features

```

1 %Calculate the momentum where tau, tau1 and p is pre-
  defined.
2 momentum = log(C1(tau+1:end-p, :)./C1(1:end-tau-p, :));
3
4
5 %Choose stock to forecast
6 XXX = momentum;
7 % XXX_ = mean(XXX,2);
8 % XXX_ = repmat(XXX_,[1, size(XXX,2)]);
9 % XXX = B*XXX + (1-B)*XXX_ ;
10
11 %Create the response matrix. This matrix is created by
  using the signed
12 %difference at time t and time t+k.The value of k
  indicates
13 %how many days, from today, we wish to forecast the
  stock direction.
14 Y = sign(C1(tau+p+1:end, :) - C1(tau+1:end-p, :));
15 YY = Y;
16
17
18 %Set all zero's to ones in response matrix. This means
  we will record a
19 %zero change in price in k days as upward movement in
  classification.
20 YY(Y==0)=1;
21
22
23 %The real price difference
24 C_diff = (C1(tau+p+1:end, :) - C1(tau+1:end-p, :));

```

Model estimation

```

1 Return = zeros (size (XXX,1)-begin , size (XXX,2) ); %pre-
    allocate
2 dk = zeros (size (XXX,1)-begin , size (XXX,2) );
3
4
5
6
7 %Calculate the constant C by defining the total amount
    of risk V_0
8 V_0 = 1;
9 Z = V_0/sqrt (size (XXX,2) );
10
11
12 for i = begin:length (XXX) -1;
13
14     if mod(i ,k)==0;
15         YYY = YY(1:i ,:);
16         const = sqrt (mean (XXX(1:i ,:).^2));
17         XXXX = bsxfun (@rdivide , XXX(1:i ,:), sqrt (mean(
                XXX(1:i ,:).^2)); %standardize for
18
19
20
21
22
23
24
25         XXXX = XXX(1:i +1 ,:)./ repmat (const , i+1,1); %include
                new day's observation
26
27
28
29
30
31
32
33
34
                %and standardize
                %Calculate the std for each stock price difference
                for [t,t+tao]
                %with the passed d observations as sample size
                if mod(i ,s)==0;
                sigma = std (C_diff (i-d:i ,:));
                end

```

```

35 %Forecast the price direction of each stock
36 dk(i-begin+1,:) = sign(XXXX(i+1,:)*w1);
37
38
39 %weights defines the position we will take and the
    units number of units
40 %we invest in each stock.
41 weights = Z*((dk(i-begin+1,:))./sigma);
42
43
44 %Return is the amount of return of our investment
    dependent on the
45 %units invested in each stock.
46 Return(i-begin+1,:) = C_diff(i+1,:).*weights;
47 end
48
49
50 Return = sum(Return,2);
51
52 %For plotting the Equity curve
53 %n1 = [1:length(Return)];
54 %plot(n1, Equitycurve, '-');
55
56
57 %Calculate a type of ex-post Sharpe ratio. R contains
    the daily realized
58 %returns. Then the Sharp ration is calculated as R/std(
    R).
59 Sharperatio = mean(Return)/mean(Return.^2);
60
61
62 %Calculate the mean prediction accuracy
63 %control C_diff so its the same values as dk
64 meanAccuracy = length(find((dk.*sign(C_diff(begin+1:end
    ,:)))==1))...
65     /numel(dk);
66
67
68 %Calculate the mean,high,low return on investment
    during the total period
69 meanReturn = mean(mean(Return,1)/size(Return,1));

```

```
70 %high = max(Equitycurve);
71 %low = min(Equitycurve);
```

CVX

```
1 function [w1] = estmod(XXXX,YYY,c)
2
3 [m,n] = size(XXXX);
4
5 cvx_begin
6 variables w1(n,n) e(m,n)
7 minimize (0.5*c*((vec(w1)'+vec(w1))) + sum(vec(e))/m);
8 subject to
9 YYY.*(XXXX*w1) >= ones(m,n)-e;
10 e >= 0;
11 cvx_end
12
13 end
```

References

- [1] Stein, C. (1956) "Inadmissibility of the usual estimator for the mean of a multivariate distribution", *Proc. Third Berkeley Symp. Math. Statist.Prob.*, pp. 197-206.
- [2] Saahil, M. (2015), "Predicting Stock Price Direction using Support Vector Machines", *Independent work report spring 2015*.
- [3] Jegadeesh; Narasimhan; Titman; Sheridan (1993), "Returns to buying winners and selling losers: implications for stock market efficiency", *Journal of Finance*, 48:65-91
- [4] Sharpe, William F. (1994) "The Sharpe Ratio", *The Journal of Portfolio Management*, pp. 49-58.
- [5] James, W.; Stein, C. (1961) "Estimation with quadratic loss", *Proc. Fourth Berkeley Symp. Math. Statist.Prob.*, pp. 361-379.
- [6] Friedman, J.; Hastie, T; Tibshirani, R. (2009), *The Elements Statistical learning: Data Mining, Inference, and Prediction*(2nd ed.), Springer.
- [7] Efron, B; Hastie, T. (2016), *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*(Institute of Mathematical Statistics Monographs, pp. 91-107), Cambridge: Cambridge University Press.
- [8] Michael C. Grant; Stephen P. Boyd, *CVX: Matlab Software for Disciplined Convex Programming*,
<http://CVXr.com/CVX/>