Binary Classification for Bankruptcy Detection

LIU Man Fung

May 2024

1 Background Introduction

The project's theme is practical solutions for binary classification. Additional aspects of binary classification will be explored, implemented, and compared on a real-world dataset. Julia environment will be used. The objective is to detect bankruptcy and it is achieved by formulating optimization problems and constraining.

2 Model and Theory

2.1 Formulation

The binary classification task aims at learning a linear classifier that distinguishes a feature vector as positive (+1) or negative (-1). The soft-margin classifier is modeled as an optimization problem to minimize the amount of errors.

Let $d \in \mathbb{N}$ be the dimension of the feature vector, the (linear) classifier is described by the tuple $(\boldsymbol{w}, b)$ where $\boldsymbol{w} \in \mathbb{R}^d$ d is the direction parameters and $d \in \mathbb{R}$ is the bias parameter, both specifying the linear model. Given $(\boldsymbol{w}, b)$, a feature vector $\boldsymbol{x} \in \mathbb{R}^d$ is classified into a label $y\{1\}$ such that

$$y = \begin{cases} +1, \text{ if } & \boldsymbol{w}^\top \boldsymbol{x} + b = w_1 x_1 + ... + w_d x_d + b \geq 0, \\ -1, \text{ if } & \boldsymbol{w}^\top \boldsymbol{x} + b = w_1 x_1 + ... + w_d x_d + b < 0 \end{cases} \tag{1}$$

Denote the training dataset with m samples as $\{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^m$, where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ is the ith feature vector with d attributes and $y^{(i)} \in \{\pm 1\}$ is the associated lable. Let R_0 and $\ell_i > 0, i = 1, ..., m$ be a set of positive weights. The following optimization problem designs a soft-margin classifier:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \sum_{i=1}^m \ell_i \max\{0, 1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)\} \quad \text{s.t.} \quad \boldsymbol{w}^\top \boldsymbol{w} \leq R_0 \tag{2}$$

(1) labels the feature vector as positive (+1) or negative (-1).

When the optimal objective value of (2) is zero, it implies that the sum of the terms $\max\{0, 1 - y^{(i)}((x^{(i)})^\top w + b)\}$ is zero, meaning all samples are not mis-classified, i.e., $y^{(i)}((x^{(i)})^\top w + b) \geq 1$.

Any optimal solution to (2) is a classifier $(\boldsymbol{w}^*, b^*)$ that can correctly distinguish the m training samples into the +1 or 1 labels.

Figure 1 shows an example of training dataset with $d = 2$ where the optimal objective value of (2) is not zero. Point F above the classifier, i.e., $y = -x + 10$, is mis-classified as -1, i.e., $y^{(F)} = -1$, $y^{(i)}((x^{(i)})^\top w + b) \ngeq 1$.

As a result, the optimal value of (2) is larger than zero as the term $\max\{0, 1 - y^{(i)}((x^{(i)})^\top w + b)\} > 0$.

The optimization problem is rewritten into:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \sum_{i=1}^m \ell_i \max\{0, 1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)\} \quad \text{s.t.} \quad \boldsymbol{w}^\top \boldsymbol{w} \leq R_0$$

$$\Rightarrow \min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, z \in \mathbb{R}} z \quad \text{s.t.} \quad z \geq \sum_{i=1}^m \ell_i \max\{0, 1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)\}, \quad \boldsymbol{w}^\top \boldsymbol{w} - R_0 \leq 0$$

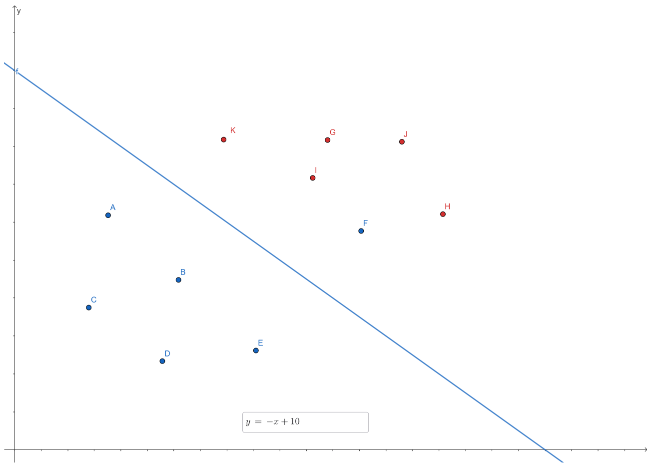

Figure 1: Example Dataset

$$\Rightarrow \min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}, z\in\mathbb{R}, t\in\mathbb{R}} z \quad \text{s.t.} \quad z \geq \sum_{i=1}^{m} \ell_i t_i, \quad t_i \geq (1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)), \quad t_i \geq 0, \quad \boldsymbol{w}^\top \boldsymbol{w} - R_0 \leq 0$$

$$\Downarrow$$

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}, z\in\mathbb{R}, t\in\mathbb{R}} z \quad \text{s.t.} \quad \sum_{i=1}^{m} \ell_i t_i - z \leq 0, \quad (1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)) - t_i \leq 0, \quad -t_i \leq 0, \quad \boldsymbol{w}^\top \boldsymbol{w} - R_0 \leq 0 \tag{3}$$

The Lagrangian function of (3) is defined by:

$$\mathcal{L}(\boldsymbol{w}, b, z, t, \mu_1, \mu_2, \mu_{a_1}, ..., \mu_{a_m}, \mu_{b_1}, ..., \mu_{b_m}) = z + \mu_1(\sum_{i=1}^{m} \ell_i t_i - z) + \mu_2(\boldsymbol{w}^\top \boldsymbol{w} - R_0) + \sum_{i=1}^{m} \mu_{a_i}(1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b) - t_i) - \sum_{i=1}^{m} \mu_{b_i} t_i$$

The Karush-Kuhn-Tucker (KKT) conditions:

1. First Order Necessary Condition (FONC)

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 2\mu_2 \sum_{j=1}^{m} \boldsymbol{w}_j - \mu_{a_i} y^{(i)} \sum_{j=1}^{m} x_j^{(i)} + ... + \mu_{a_m} y^{(i)} \sum_{j=1}^{m} x_j^{(i)} = 0$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\mu_{a_i} y^{(i)} \sum_{j=1}^{m} x_j^{(i)} + ... + \mu_{a_m} y^{(i)} \sum_{j=1}^{m} x_j^{(i)} = 0$$

$$\frac{\partial \mathcal{L}}{\partial z} = 1 - \mu_1 = 0$$

$$\frac{\partial \mathcal{L}}{\partial t} = \mu_1 \sum_{i=1}^{m} \ell_i - \mu_{a_1} - ... - \mu_{a_m} - \mu_{b_1} - \mu_{b_m} = 0$$

2. Complementary Slackness

$$\mu_1(\sum_{i=1}^{m} \ell_i t_i - z) = 0$$

$$\mu_2(1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b) - t_i) = 0$$

$$-\mu_3 t_i = 0$$

$$\mu_4(\boldsymbol{w}^\top \boldsymbol{w} - R_0) = 0$$

3. Primary Constraints

$$\sum_{i=1}^{m} \ell_i t_i - z \leq 0$$

$$(1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)) - t_i \leq 0$$

$$-t_i \leq 0$$

$$\boldsymbol{w}^\top \boldsymbol{w} - R_0 \leq 0$$

4. Dual Constraint

$$\mu_1, \mu_2, \mu_{a_1}, ..., \mu_{a_m}, \mu_{b_1}, ..., \mu_{b_m} \geq 0$$

Solving the KKT conditions, we get $\mu_1 = 1$ in (FONC), it is possible for $\mu_4 > 0$ or $\mu_4 = 0$ from Dual Constraint.

Case 1: $\mu_4 > 0$, $\boldsymbol{w}^\top \boldsymbol{w} - R_0 = 0$ (last constraint of slackness), $\boldsymbol{w}$ may exist more than 1 solutions if $d \geq 2$.

Case 2: $\mu_4 = 0$, $\boldsymbol{w}^\top \boldsymbol{w} \leq R_0$ (last constraint of primary constraint), $\boldsymbol{w}$ may exist more than 1 solutions if $d \geq 1$.

There may exist more than one optimal solution $(\boldsymbol{w}^*, b^*)$ to (2), which is the classifier. Figure 2 shows the example of possible classifiers (blue line) for $d = 2$.

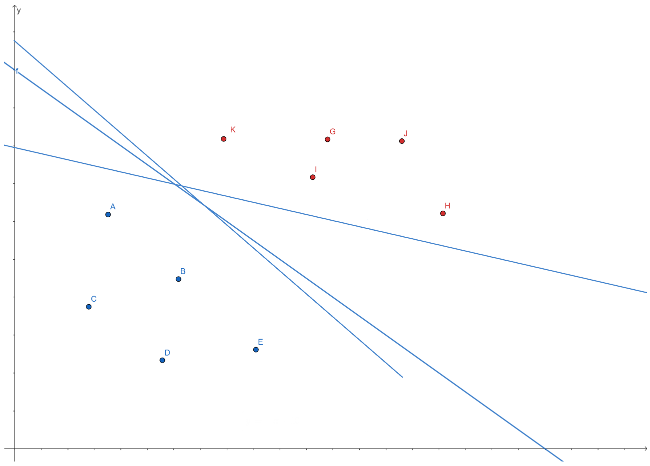

Figure 2: Multiple solutions

2.2 Optimization Formulation

Second-order Cone Programming (SOCP) in the standard form:

$$\min_{x \in \mathbb{R}^n} c^\top x \quad \text{s.t.} \quad \|\boldsymbol{A}_i x + b_i\| \leq c_i^\top x + d_i, \quad i = 1, ..., m.$$

When $\boldsymbol{A}_i = b_i = 0$, $0 \leq c_i^\top x + d_i$, a linear constraint will be a SOCP constraint.
The first 3 constraints are linear and transfer the last constraint into a SOCP constraint.
Second-order Cone Programming (SOCP) of (3):

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, z \in \mathbb{R}, t \in \mathbb{R}} z \quad \text{s.t.} \quad \sum_{i=1}^{m} \ell_i t_i - z \leq 0, \quad (1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)) - t_i \leq 0, \quad -t_i \leq 0, \quad \|\boldsymbol{w}\| \leq \sqrt{R_0} \quad (4)$$

For given $R_0, R_1 > 0$, shaping constraint: $\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w} \leq R_0$ where $\boldsymbol{\Sigma} \in \mathbb{R}^{dxd}$ is a given symmetric, positive definite matrix, and $\boldsymbol{c} \in \mathbb{R}^d$ is a given vector. The directional parameter and bias parameter belongs to an ℓ_1 ball to promote sparsity, i.e., $\sum_{i=1}^{d} |w_i| + |b| \leq R_1$.
Converting 2 new constraints into SOC constraints:

$$\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w} \leq R_0 \Rightarrow \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} \leq R_0 - \boldsymbol{c}^\top \boldsymbol{w}$$

$$\Rightarrow 4\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} \leq 4k \Rightarrow 4\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} - 4k \leq 0, \quad k \leq R_0 - \boldsymbol{c}^\top \boldsymbol{w}$$

$$\Rightarrow 4\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + (1-k)^2 - (1+k)^2 \leq 0 \Rightarrow 4\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + (1-k)^2 \leq (1+k)^2, \quad 0 \leq 1 + k$$

$$\Rightarrow \sqrt{4\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + (1-k)^2} \leq 1+k \Rightarrow \sqrt{4\boldsymbol{w}^\top (\boldsymbol{\Sigma}^{\frac{1}{2}})^\top (\boldsymbol{\Sigma}^{\frac{1}{2}})\boldsymbol{w} + (1-k)^2} \leq 1+k \Rightarrow \sqrt{4((\boldsymbol{\Sigma}^{\frac{1}{2}})\boldsymbol{w})^\top ((\boldsymbol{\Sigma}^{\frac{1}{2}})\boldsymbol{w}) + (1-k)^2} \leq 1+k$$

$$\Rightarrow \left\| \begin{bmatrix} 2(\boldsymbol{\Sigma}^{\frac{1}{2}})w \\ 1-k \end{bmatrix} \right\| \leq 1+k, \quad k \leq R_0 - \boldsymbol{c}^\top \boldsymbol{w}, \quad 0 \leq 1+k$$

$$\sum_{i=1}^{d} |w_i| + |b| \leq R_1$$

$$\Rightarrow \sum_{j=1}^{d} |w_j| + |b| \leq R_1, \quad -w_j \leq v_j \leq w_j, \quad j = 1, ..., d, \quad -b \leq a \leq b$$

$$\Rightarrow \sum_{j=1}^{d} v_j + a \leq R_1, \quad -w_j \leq v_j, \quad v_j \leq w_j, \quad i = j, ..., d, \quad -b \leq a, \quad a \leq b$$

Combining 2 converted SOC constraints into (4):

$$\begin{aligned}
\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}, z\in\mathbb{R}, t\in\mathbb{R}} \quad & z \quad \text{s.t.} \\
& \sum_{i=1}^{m} \ell_i t_i - z \leq 0, \quad (1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)) - t_i \leq 0, \quad -t_i \leq 0, \quad \|\boldsymbol{w}\| \leq \sqrt{R_0} \\
& \| \begin{bmatrix} 2(\boldsymbol{\Sigma}^{\frac{1}{2}})w \\ 1-k \end{bmatrix} \| \leq 1+k, \quad k \leq R_0 - \boldsymbol{c}^\top \boldsymbol{w}, \quad 0 \leq 1+k \\
& \sum_{j=1}^{d} v_j + a \leq R_1, \quad -w_j \leq v_j, \quad v_j \leq w_j, \quad j = 1, ..., d, \quad -b \leq a, \quad a \leq b
\end{aligned} \tag{5}$$

For given $R_0, R_1, S > 0$, shaping constraint: $\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w} \leq R_0$ where $\boldsymbol{\Sigma} \in \mathbb{R}^{dxd}$ is a given symmetric, positive definite matrix, and $\boldsymbol{c} \in \mathbb{R}^d$ is a given vector. Each element in $\boldsymbol{w}$ is bounded such that $-R_1 \leq w_i \leq R_1, \quad i = 1, ..., d$. The number of non-zero elements in the vector $\boldsymbol{w}$ is constrained such that (no. of non-zero elements in the vector $\boldsymbol{w}$) $\leq S$.

$$\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w} \leq R_0, \quad -R_1 \leq w_i \leq R_1, \quad i = 1, ..., d, \quad n_i = \begin{cases} 0, \text{ if } & \boldsymbol{w}_i = 0 \\ 1, \text{ if } & \boldsymbol{w}_i \neq 0 \end{cases}, \quad \sum_{i=1}^{d} n_i \leq S$$

Combining new constraints into (3):

$$\begin{aligned}
\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}, z\in\mathbb{R}, t\in\mathbb{R}} \quad & z \quad \text{s.t.} \\
& \sum_{i=1}^{m} \ell_i t_i - z \leq 0, \quad (1 - y^{(i)}((\boldsymbol{x}^{(i)})^\top \boldsymbol{w} + b)) - t_i \leq 0, \quad -t_i \leq 0, \quad \boldsymbol{w}^\top \boldsymbol{w} - R_0 \leq 0 \\
& \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w} \leq R_0, \quad -R_1 \leq w_i \leq R_1, \quad i = 1, ..., d, \\
& n_i = \begin{cases} 0, \text{ if } & \boldsymbol{w}_i = 0 \\ 1, \text{ if } & \boldsymbol{w}_i \neq 0 \end{cases}, \quad \sum_{i=1}^{d} n_i \leq S
\end{aligned} \tag{6}$$

3 Experiments

3.1 Warmup Exercise

A training dataset of $m = 20$ companies. Each company has 64 attributes (performance indicators). The dataset also contains information of whether the company has bankrupted or not, treated as the *label* $y_i \in \pm 1$. Figure 3 shows different plots with different attributes, 'Bankrupt' in red and 'Not Bankrupt' in blue.

The first and third plot show red and blue points are mixed together, the 2 respective attributes has no relation with 'Bankrupt'.

The second plot also shows red and blue points are mixed together, a straight line could be drawn to fit most data since both attributes involves equity and total assets.

Ranges of Attributes:

total liabilities / total assets: [0.24034, 1.4851]
net profit / total assets: -0.51794, 0.52723]
(equity - share capital) / total assets: [-1.7645, 0.68669]
equity / total assets: [-0.48504, 0.75966]
logarithm of total assets: [2.0061, 5.8511]
(gross profit + depreciation) / sales: [-0.23072, 0.23008]

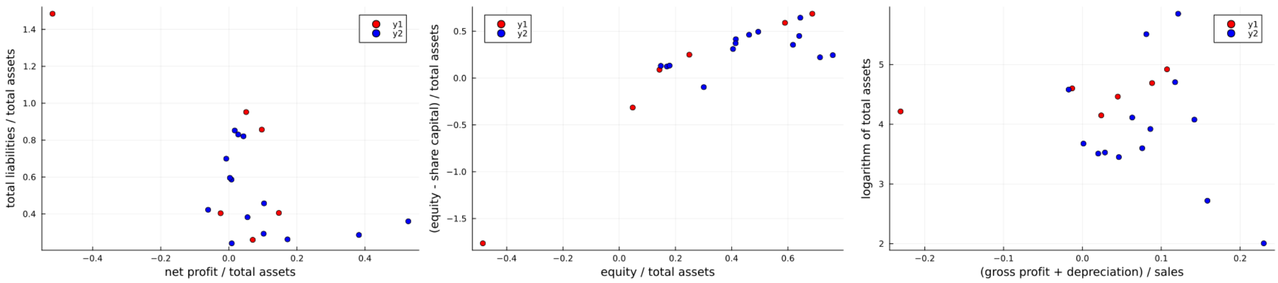

Figure 3: 2-D scatter plots of Bankrupt

3.2 Optimization-based Formulation

Using the solver ECOS in JuMP for SOCP (4), (5) and solver Juniper in JuMP for the MIP (6), marked as SOCP - Task 4a, SOCP - Task 4b and MIP - Task 4c respectively.

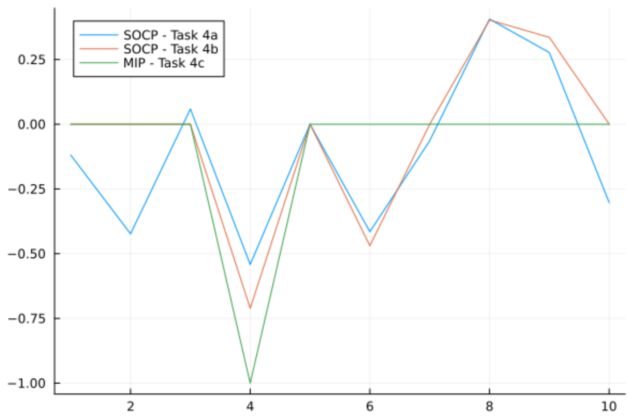

Figure 4: Classifier Solutions

Figure 4 shows the classifier solution of (4), (5) and (6) with default setting **weight** = 1. By observing, number of non-zeros: (4) > (5) > (6), showing the sparsity level: (6) > (5) > (4). It is unreasonable for (6) to determine "Bankrupt" with 1 significant attribute only. (4) and (5) are reasonable as they take much more significant attributes to determine.

3.3 Error Performance

The performance of a classifier can be evaluated by the *error rate* when applied on a certain set of data. It can further be specified into *false alarm rate* and *missed detection rate*. To describe these metrics, note that for a given classifier $(\boldsymbol{w}, b)$, the predicted label is

$$\text{False Alarm (FA) Rate} = \frac{1}{m_-} \sum_{i \in D_-} \mathbb{1}(\hat{y}^{(i)} \neq -1), \quad \text{Missed Detection (MD) Rate} = \frac{1}{m_+} \sum_{i \in D_+} \mathbb{1}(\hat{y}^{(i)} \neq 1) \quad (7)$$

Both error rates are between 0 and 1. Sometimes they are called the Type I and Type II errors.

As our aim is to design a classifier that makes prediction on whether a *future company* that is not found in the training dataset will go bankrupt, it is necessary to evaluate the error rate on a *testing dataset* that is unseen during the training. Denote the testing dataset with m_{test} samples as $\{\boldsymbol{x}_{test}^{(i)}, y_{test}^{(i)}\}_{i=1}^{m_{test}}$, the *testing error rate* for a classifier $(\boldsymbol{w}, b)$ can be estimated using similar formulas as in (7).

The error rate for the classifier solutions for (4), (5), (6) on training dataset ($\text{weight}_{(4),(5),(6)} = 1$):
(4): (FA, MA) = (0, 0.333)
(5): (FA, MA) = (0.0714, 0.333)
(6): (FA, MA) = (0.571, 0.166)

The error rate for the classifier solutions for (4), (5), (6) on testing dataset ($\text{weight}_{(4),(5),(6)} = 1$):
(4): (FA, MA) = (0.46, 0.4)
(5): (FA, MA) = (0.46, 0.4)
(6): (FA, MA) = (0.7, 0.5)

The classifier solution has the lowest error rate in (4) on training dataset and in (5) and (6) on testing dataset.

Adjusting the parameter **weight** ≥ 0 so that it balances between FA and MD rates on the **training dataset**.

The error rate for the classifier solutions for (4), (5), (6) on **training dataset**
($\text{weight}_{(4)} = 1, \quad \text{weight}_{(5)} = 1.35, \quad \text{weight}_{(6)} = 0.565$):
(4): (FA, MA) = (0, 0.333)
(5): (FA, MA) = (0.429, 0.333)
(6): (FA, MA) = (0.0714, 0.5)

The error rate for the classifier solutions for (4), (5), (6) on testing dataset
($\text{weight}_{(4)} = 1, \quad \text{weight}_{(5)} = 1.35, \quad \text{weight}_{(6)} = 0.565$):
(4): (FA, MA) = (0.46, 0.4)
(5): (FA, MA) = (0.72, 0.3)
(6): (FA, MA) = (0.04, 0.7)

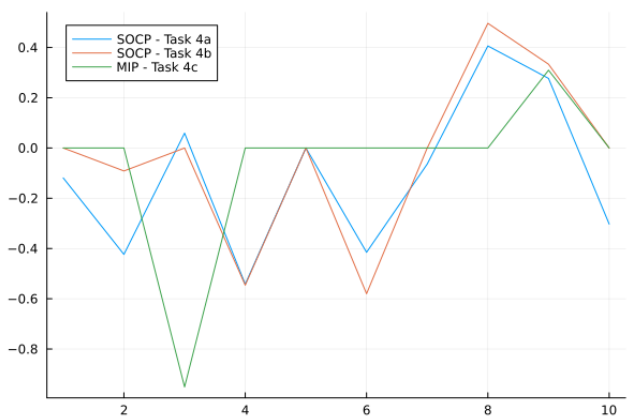

Figure 5: Weighted Classifier Solutions

Figure 5 shows the classifiers solution of (4), (5) and (6) with adjusted weighting. Top-2 most significant features: Attribute 3 (working capital / total assets) and Attribute 9 (sales / total assets) are selected by the optimization from the MIP formulation (6).

Figure 6 shows the scatter plot of the training dataset for the 2 selected features (Attribute 3 and Attribute 9). Only (6) functions well as the selected attributes are not significant for (4) and (5).

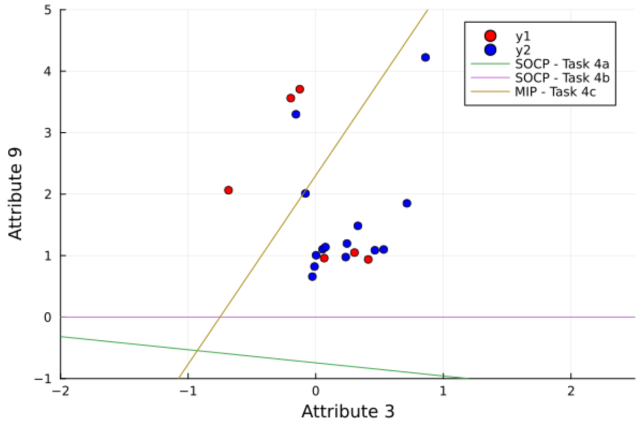

Figure 6: Scatter Plot of the training dataset

4 Customized Solver for Classifier Optimization

To implement the solver to the binary classifier problem, without relying on JuMP and its optimizers such as ECOS, Juniper, etc. A full dataset from m = 8000 samples and all the 64 available attributes to detect bankruptcy are considered.

The optimization problem of the form:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \hat{f}(w,b) \quad s.t \quad (w,b) \in X,$$

$$\text{The logistic loss:} \quad \hat{f}(w,b) = \frac{1}{m}\sum_{i=1}^{m} \ell_i log(1 + exp(-y^{(i)}((x^{(i)})^\top w + b)))$$

$$X = \{w \in \mathbb{R}^d,\ b \in \mathbb{R} :\ |b| + \sum_{i=1}^{d} |w_i| \leq R_1\}$$

Applying the PGD Method to solve the optimizaion problem:

Input: $\boldsymbol{x}^{(0)} \in X$, constant step size $\gamma \leq 0$, max, iteration number K_{max}.
For $k = 0, ..., K_{max}$
$\quad \boldsymbol{x}^{k+1} = Proj_X\{\boldsymbol{x}^{(k)} - \gamma \nabla \hat{f}(\boldsymbol{x}^k)\}$
End For

The projection operator:

Input: $\boldsymbol{x} \in \mathbb{R}^d, R > 0$
Calculate the vector $\boldsymbol{u} = abs.(\boldsymbol{x})$ such that it takes the absolute values of the input $\boldsymbol{x}$.
Sort elements in $\boldsymbol{u}$ with decreasing magnitude, denote the sorted vector as $\boldsymbol{v}, |v_1| \leq ... \leq |v_d|$
For $j = 1, ...d,$
$\quad$ **If** $v_j - \frac{1}{j}(\sum_{r=1}^{j} v_r - R) \leq 0$, **Then** set $j_{sv} = j - 1$ and break for loop
Set $\theta = \frac{1}{j}(\sum_{r=1}^{j_{sv}} v_r - R)$
Return: the vector $\hat{\boldsymbol{x}}$ such that $\hat{x}_i = \text{sign}(x_i)\text{max}\{0, |x_i| - \theta\}$ for $i = 1, ..., d$

To elvaluate the classifier performance:

$$F_1 = \frac{2(1 - P_{MD)}}{2(1 - P_{MD}) + P_{FA} + P_{MD}} \quad \text{(Higher is better)}$$

$$(\#\text{ non-zero elements in } \boldsymbol{w}, b) = \mathbb{1}(|\frac{|b|}{|b| + \sum_{j=1}^{d} |w_j|}| \geq 0.01) + \sum_{i=1}^{d} \mathbb{1}(|\frac{|w_i|}{|b| + \sum_{j=1}^{d} |w_j|}| \geq 0.01) \quad \text{(Lower is better)}$$

By setting weight = 1.0, step size = 0.94, R = 0.5, maximum iterations = 5000, objective value of 1.167 is achieved. Figure 7 shows the plot of objective values.

The classifier performance:
Train $F_1 = 0.680$
Test $F_1 = 0.674$
Number of non-zeros = 2

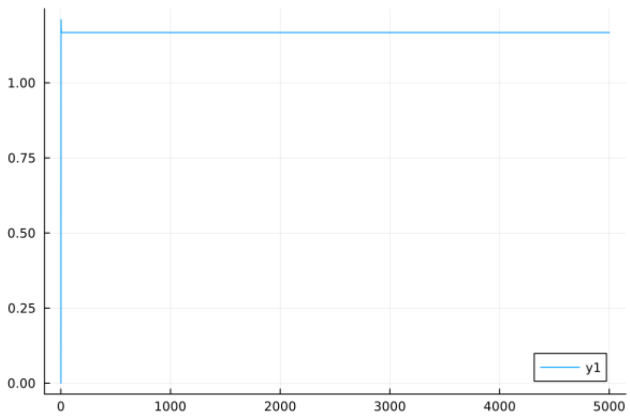

Figure 7: Objective values

5 Conclusions

The project demonstrates the use of SVM and prediction of real-world problems. As demonstrated, optimization is an effective tool to achieve the objective. Due to the limitations of optimization packages such as JuMP, other models will be effective in this case. This project has demonstrated the use of the PGD method, which is one of the feasible algorithm. In real world, there are other alternative algorithms which have its advantages and disadvantages such as the stochastic gradient descent (SGD), are also effective in similar cases.

6 Appendix: Dataset Description

Here is the list of all the 64 features collected in the Bankruptcy dataset:

Attr1 net profit / total assets
Attr2 total liabilities / total assets
Attr3 working capital / total assets
Attr4 current assets / short-term liabilities
Attr5 [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365
Attr6 retained earnings / total assets
Attr7 EBIT / total assets
Attr8 book value of equity / total liabilities
Attr9 sales / total assets
Attr10 equity / total assets
Attr11 (gross profit + extraordinary items + financial expenses) / total assets
Attr12 gross profit / short-term liabilities
Attr13 (gross profit + depreciation) / sales
Attr14 (gross profit + interest) / total assets
Attr15 (total liabilities * 365) / (gross profit + depreciation)
Attr16 (gross profit + depreciation) / total liabilities
Attr17 total assets / total liabilities
Attr18 gross profit / total assets
Attr19 gross profit / sales
Attr20 (inventory * 365) / sales
Attr21 sales (n) / sales (n-1)
Attr22 profit on operating activities / total assets
Attr23 net profit / sales
Attr24 gross profit (in 3 years) / total assets
Attr25 (equity - share capital) / total assets
Attr26 (net profit + depreciation) / total liabilities
Attr27 profit on operating activities / financial expenses
Attr28 working capital / fixed assets
Attr29 logarithm of total assets
FTEC2101/ESTR2520 Project 12
Attr30 (total liabilities - cash) / sales
Attr31 (gross profit + interest) / sales
Attr32 (current liabilities * 365) / cost of products sold
Attr33 operating expenses / short-term liabilities
Attr34 operating expenses / total liabilities
Attr35 profit on sales / total assets
Attr36 total sales / total assets
Attr37 (current assets - inventories) / long-term liabilities
Attr38 constant capital / total assets
Attr39 profit on sales / sales
Attr40 (current assets - inventory - receivables) / short-term liabilities
Attr41 total liabilities / ((profit on operating activities + depreciation) * (12/365))
Attr42 profit on operating activities / sales
Attr43 rotation receivables + inventory turnover in days
Attr44 (receivables * 365) / sales
Attr45 net profit / inventory
Attr46 (current assets - inventory) / short-term liabilities
Attr47 (inventory * 365) / cost of products sold
Attr48 EBITDA (profit on operating activities - depreciation) / total assets
Attr49 EBITDA (profit on operating activities - depreciation) / sales
Attr50 current assets / total liabilities
Attr51 short-term liabilities / total assets
Attr52 (short-term liabilities * 365) / cost of products sold)
Attr53 equity / fixed assets
Attr54 constant capital / fixed assets
Attr55 working capital

Attr56 (sales - cost of products sold) / sales
Attr57 (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
Attr58 total costs /total sales
Attr59 long-term liabilities / equity
Attr60 sales / inventory
Attr61 sales / receivables
Attr62 (short-term liabilities *365) / sales
Attr63 sales / short-term liabilities
Attr64 sales / fixed assets