SUPPORT VECTOR MACHINES-ML METHOD

DATA SCIENCE
INSTITUTE

Contents

DATA SCIENCE INSTITUTE

Introduction to Support Vector Machines

- Support Vector Machines (SVM's) are a relatively new learning method generally used for classification problem.
- Although the first paper dates way back to early 1960's it is only in 1992-1995 that this powerful method was universally adopted as a mainstream machine learning paradigm

The basic idea is to find a hyper plane which separates the d-dimensional data perfectly into its classes.

What is a Hyper Plane

In two dimensions, a hyper plane is defined by the equation:

$$W_1 X_1 + W_2 X_2 + b = 0$$

This is nothing but **equation of line.**

The above equation can be easily extended to the p-dimensional setting:

$$W_1 X_1 + W_2 X_2 + \ldots + W_p X_p + b = 0$$

In short,

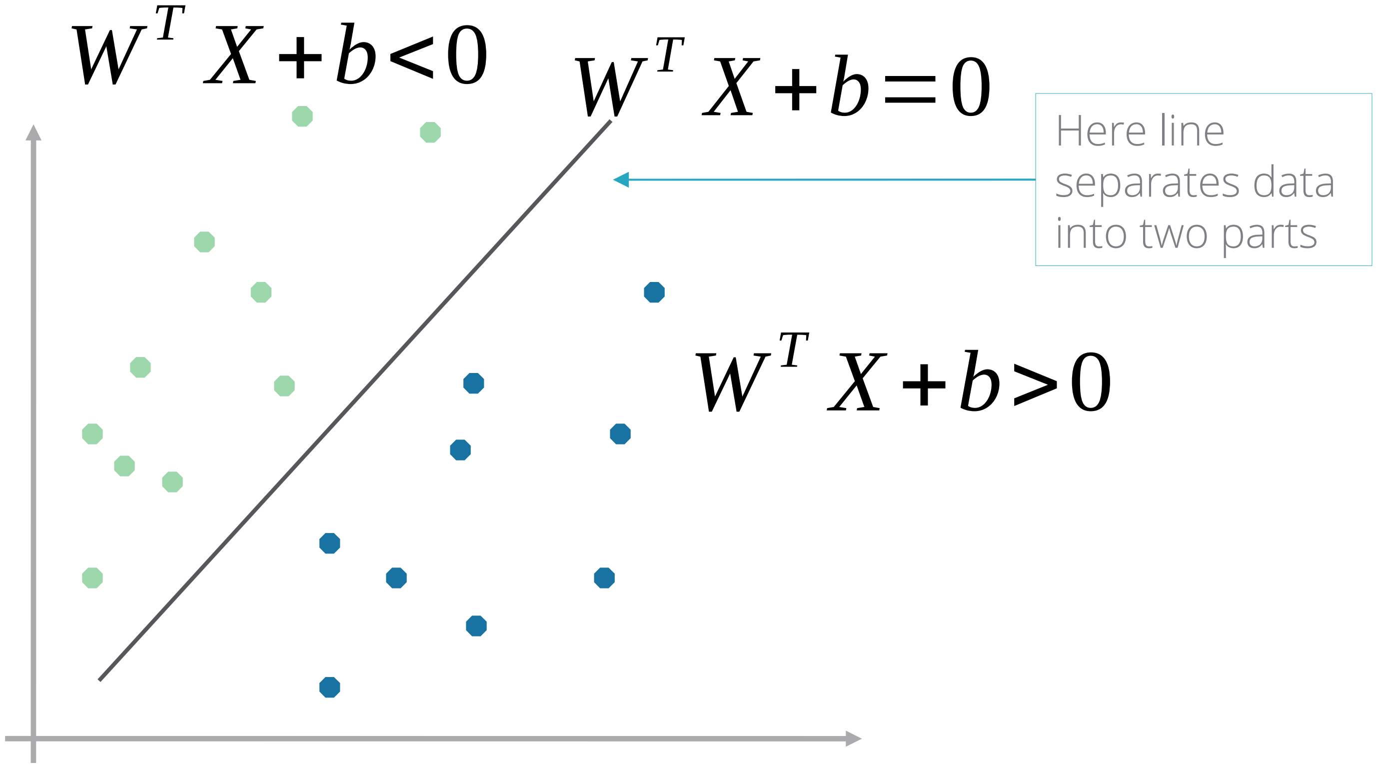
$$\boldsymbol{W^T X + b = 0}$$

In p > 3 dimensions, it can be hard to visualize a hyper planes.

Separating a Hyper Plane

- Binary classification can be viewed as the task of separating classes in feature space:

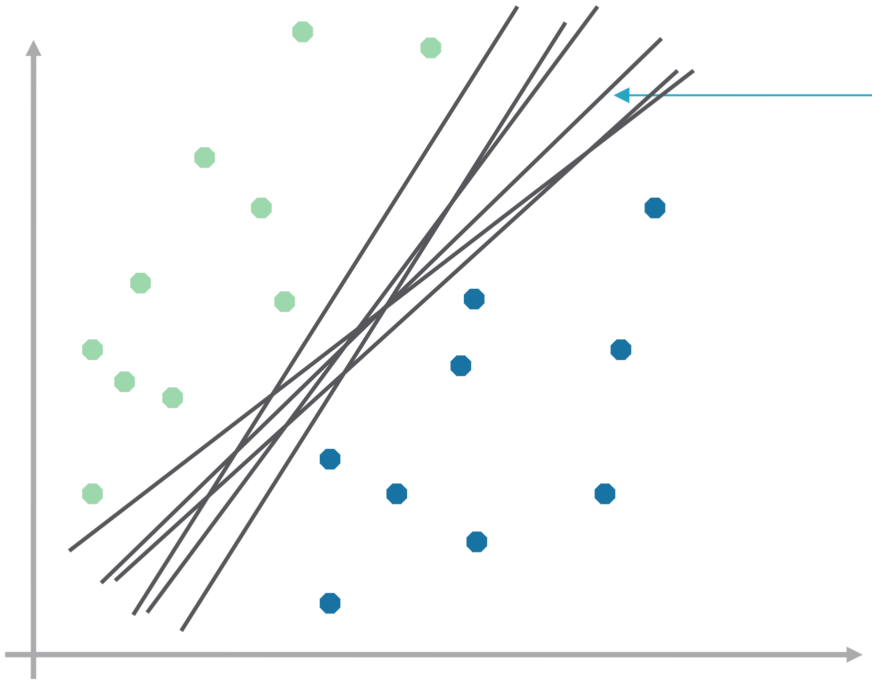
Fig. 01: Binary Classification



Linear Separators

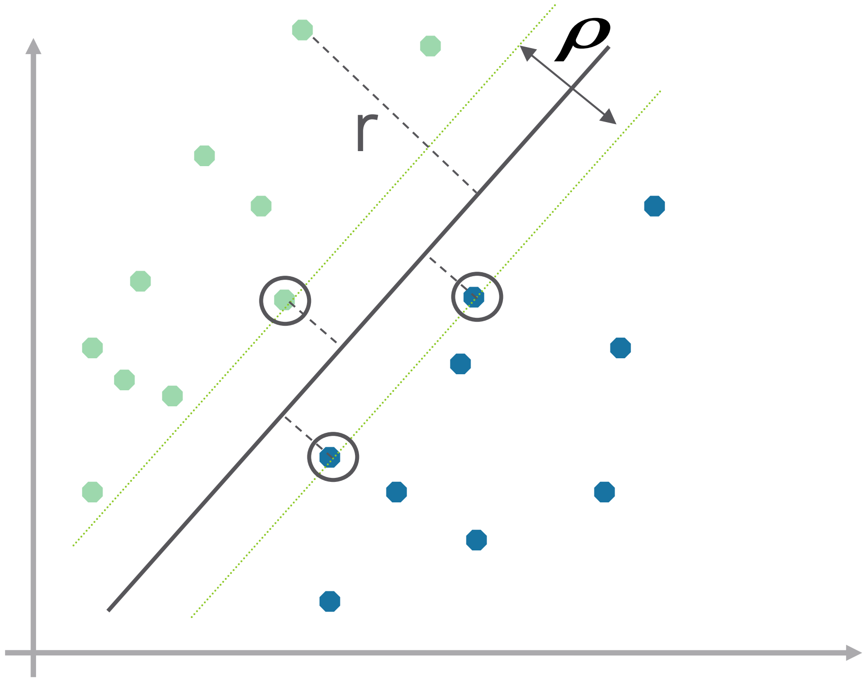
The objective in SVM is to find optimum separator

Fig. 02: Linear Separators



Which of the linear separators is optimal?

Classification Margin



- Distance from case x_i to the separator is

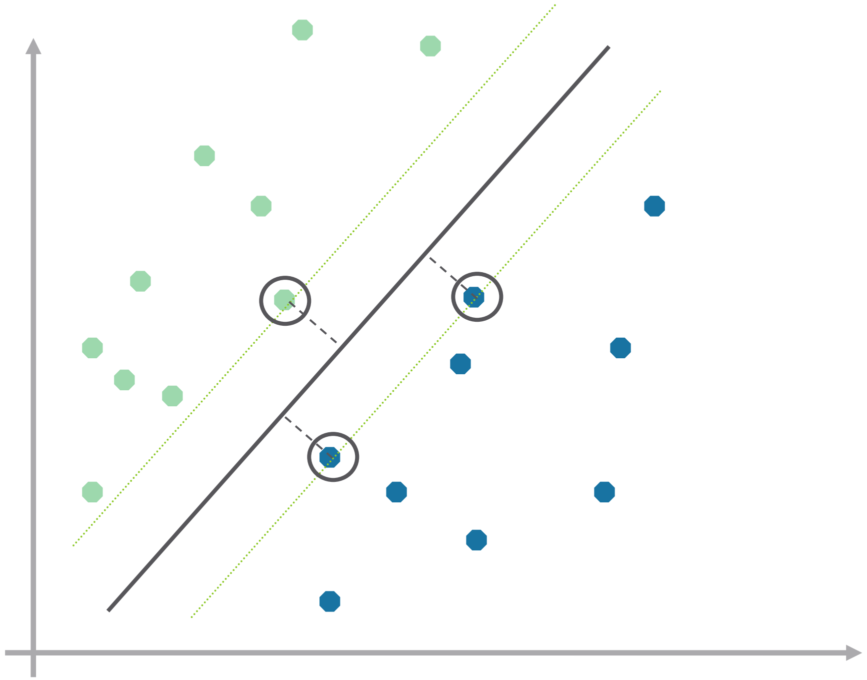
$$r = \frac{w^T x_i + b}{\| w \|}$$

Here is length of a vector given by sqrt(sum(W^2))

- Cases closest to the hyper plane are Support Vectors
- Margin ρ of the separator is the distance between support vectors

DATA SCIENCE INSTITUTE

Maximum Margin Classification



- The objective is now to maximize the margin ρ of the separator
- The focus is on ‘Support Vectors’
- Other cases are not considered in the algorithm

Mathematical Approach to Linear SVM

Let training set be separated by a hyper plane with margin . Then for each training observation

$$\leftrightarrow y_i\left(w^T x_i + b\right) \geq \rho/2$$

For every support vector the above inequality is an equality

After rescaling and by in the equality, we obtain that distance between each and the hyper plane is

$$r = \frac{y_i\left(w¿¿T x_s + b\right)}{\| w \|} = \frac{1}{\| w \|} ¿$$

Margin can be expressed through (

$$\rho = 2r = \frac{2}{\| w \|}$$

DATA SCIENCE INSTITUTE

Mathematical Approach to Linear SVM

Quadratic Optimisation problem is:

Find and such that

is maximised

and

which can be reformulated as:

Find and such that

is minimised

and

Case Study – Predicting Loan Defaulters

Background

- The bank possesses demographic and transactional data of its loan customers. If the bank has a robust model to predict defaulters it can undertake better resource allocation.

Objective

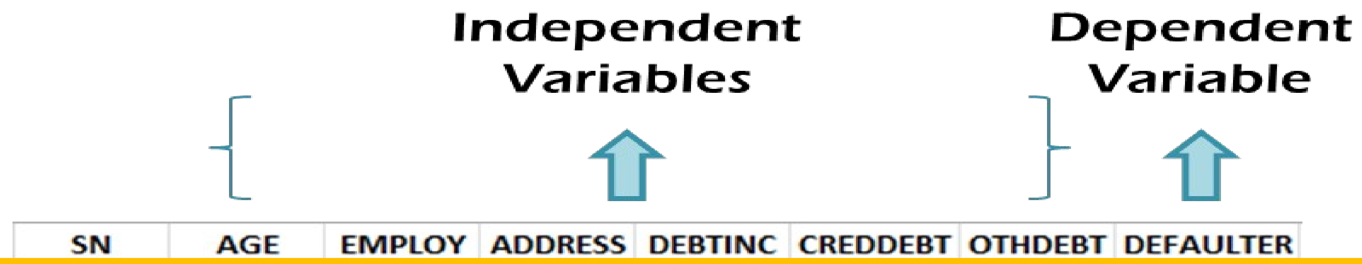
- To predict whether the customer applying for the loan will be a defaulter

Available Information

- Sample size is 700
- Age group, Years at current address, Years at current employer, Debt to Income Ratio, Credit Card Debts, Other Debts are the independent variables
- Defaulter (=1 if defaulter, 0 otherwise) is the dependent variable

Data Snapshot

BANK LOAN



Column	Description	Type	Measurement	Possible Values
SN	Serial Number	-	-	-
AGE	Age Groups	Integer	1(<28 years),2(28-40 years),3(>40 years)	3
EMPLOY	Number of years customer working at current employer	Integer	-	Positive value
ADDRESS	Number of years customer staying at current address	Integer	-	Positive value
DEBTINC	Debt to Income Ratio	Continuous	-	Positive value
CREDDEBT	Credit to Debit Ratio	Continuous	-	Positive value
OTHDEBT	Other Debt	Continuou	-	Positive value

DATA SCIENCE INSTITUTE

SVM in R

```
# Importing and Readying the Data
bankloan<-read.csv("BANK LOAN.csv",header=T)

bankloan$AGE<-as.factor(bankloan$AGE)

str(bankloan)
```

as.factor() changes age from an integer to a factor variable.

str() is used to check if the conversion to factor has taken place and if all other variable formats are appropriate, before moving to SVM modeling.

```
# Output
'data.frame':   700 obs. of  8 variables:
 $ SN      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ AGE     : Factor w/ 3 levels "1","2","3": 3 1 2 3 1 3 2 3 1 2 ...
 $ EMPLOY  : int  17 10 15 15 2 5 20 12 3 0 ...
 $ ADDRESS : int  12 6 14 14 0 5 9 11 4 13 ...
 $ DEBTINC : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
 $ CREDDEBT : num  11.36 1.36 0.86 2.66 1.79 ...
 $ OTHDEBT  : num  5.01 4 2.17 0.82 3.06 ...
 $ DEFAULTER: int  1 0 0 0 1 0 0 0 1 0 ...
```

SVM in R

```
# SVM Using Package "e1071"
install.packages("e1071")
library(e1071)

model<-svm(formula=DEFAULTER~AGE+EMPLOY+ADDRESS+
           DEBTINC+CREDDEBT+OTHDEBT,data=bankloan,
           type="C",probability=TRUE,kernel="linear")

Model
coef(model)
```

- **svm()** trains a support vector machine.
- **formula=** gives the model to be fit.
- **data=** specifies the data object.
- **type=** specifies whether SVM is used for classification or regression or novelty detection. Default for **type=** is "C".
- **probability=** logical for indicating whether model should allow for probability predictions.
- **kernel=** specifies the kernel used in training and predicting. Here, we have kept kernel as linear.

SVM in R

Output

```
> model

Call:
svm(formula = DEFAULTER ~ AGE + EMPLOY + ADDRESS + DEBTINC + CREDDEBT + OTHDEBT,
 data = bankloan, type = "C", probability = TRUE,
    kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  312
```

DATA SCIENCE INSTITUTE

Predictions Based on SVM

```
# Predictions
pred1<-predict(model,bankloan,probability=TRUE)
```

- **predict()** returns predicted probabilities based on the model results and historical data.
- First argument is the **svm()** model object while the second argument is original dataset.
- **probability=TRUE** returns raw probabilities. This argument is valid only when **type="probability“** is specified in **svm()**.
- ***pred1<-predict(model,bankloan,decision.values =TRUE) gives scores***

```
pred2<-attr(pred1,"probabil
```

- **attr()**, from base R, is used get or set specific attributes of an object. Here, we want to get the predicted probabilities obtained by the **svm()** model.
- First argument is the name of the object whose attributes we want to extract.
- Second argument is the character string specifying which attribute is to be accessed. Check **pred1** to know the exact name, which is

CIENCE
STITUTE

ROC Curve and Area Under ROC Curve

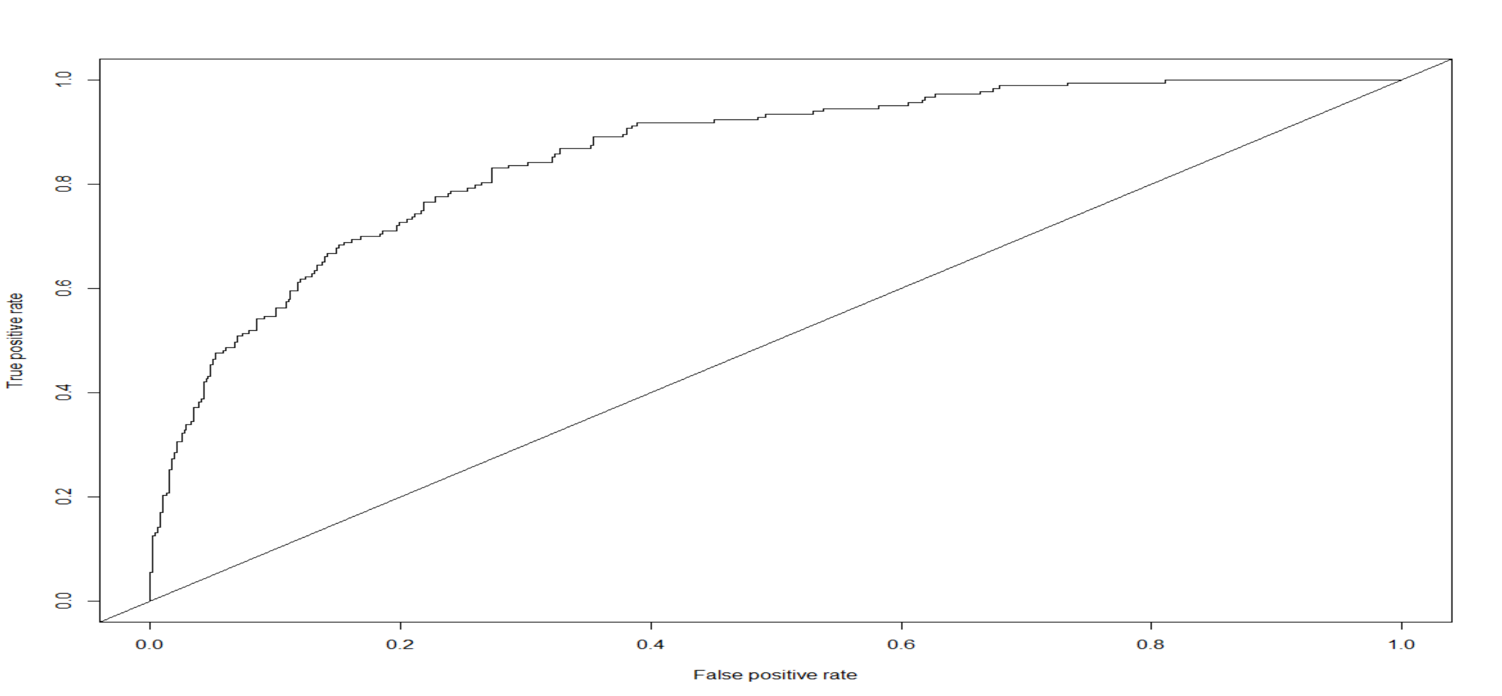
```
#ROC Curve

install.packages("ROCR")
library(ROCR)

pred<-prediction(pred2,bankloan$DEFAULTER)

perf<-performance(pred,"tpr","fpr")

plot(perf)

abline(0,1)
```

- **prediction()** creates object of class prediction, required for ROC curve. **performance()** calculates predictor evaluations.
- Using **measure="tpr", measure="fpr"** we can plot an ROC Curve.
- **abline()** adds a straight line to the plot.

DATA SCIENCE INSTITUTE

ROC Curve and Area Under ROC Curve

\# Output



\# Area Under ROC Curve

```
auc<-performance(pred,"auc")
auc@y.values
[[1]]
[1] 0.855577
```

“auc” in performance() calculates Area Under ROC Curve.

DATA SCIENCE INSTITUTE

Quick Recap

Support Vector Machines

- SVMs find a hyper plane which separates the d-dimensional data perfectly into its classes

SVM in R

- Package **`"e1071"`** has **`svm()`** that trains a support vector machine
- The function takes arguments to specify whether **`svm()`** is to be used for classification or regression; if probabilities are to be returned and which kernel to use for training and predicting

DATA SCIENCE INSTITUTE

THANK YOU!!