



SUPERVISED LEARNING- LINEAR SUPPORT VECTOR MACHINE

UNDERSTANDING SUPPORT VECTOR MACHINE(SVM)

Dev Shrivastava and Shubhanshu Prasad

January 30, 2023

INDEX

1	Introduction	2
2	Regression vs Classification	4
3	Explanation of Linear SVM using Email spam detection	6
4	Mathematical Intuition of Linear-SVM	10
4.1	Dot product in Linear SVM	10
4.2	Deciding the Margin	13
4.3	optimizing the equations to get maximum margin	14
5	Problem with linear SVM	18
6	Inductive Bias Hyperparameter	19
7	Soft Margin SVM	20
8	References:	21

INTRODUCTION

- ▶ SVM are a type of supervised machine learning algorithm, that can be used for classification and regression tasks.
- ▶ An SVM's fundamental purpose is to find a boundary, or hyperplane, that separates various classes of data and has the greatest margin, or distance, between the boundary and the nearest data points in each class.
- ▶ These nearest locations are known as support vectors, and they are essential to the method.

INTRODUCTION

USES AND APPLICATION

- ▶ Fraud detection: SVMs can be used to detect fraudulent activities in credit card transactions, insurance claims, and other areas.
- ▶ A natural Language Processing (NLP): SVMs can be used to classify, predict sentiment in text, identify named entities, and perform other NLP tasks.
- ▶ Gaming: SVMs have been used in computer games to make decisions on the next move, such as predicting the best move for chess.
- ▶ Face detection
- ▶ Handwriting recognition

REGRESSION VS CLASSIFICATION

REGRESSION

Regression algorithms predict a continuous value based on the input variables. It is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome.

Ex. Simple Linear regression, Multiple Linear Regression, Polynomial Regression.

REGRESSION VS CLASSIFICATION

CLASSIFICATION

Classification is a prediction model that uses input variables to approximate a mapping function in order to find discrete output variables, which can be labels or categories. The classification algorithm's mapping function is in charge of predicting the label or category of the provided input variables. A classification algorithm can use both discrete and real-valued variables, but the examples must be classified into one of two or more classes.

Ex. Decision Tree, Random Forest, KNN

EXPLANATION OF LINEAR SVM USING EMAIL SPAM DETECTION

- ▶ Suppose we have our dataset which includes emails and a label suggesting whether its spam or not.
- ▶ Now to process this data a naive approach can be, to use one-hot encoding so that machine can understand the mails in terms of numbers.
- ▶ For that we suppose have an dictionary of all English words and suppose the total number of words are N in that dictionary
- ▶ Then according to one-hot encoding we can make a 2D array of size $[2 \times N]$ so for every word we can have value 0 or 1 suggesting whether our mail has that word or not, and at the end we have the value suggesting whether its spam or not.

a	the	this	Lottery	lucky	win	spam
1	1	1	1	0	0	1

Figure. One-hot encoding

EXPLANATION OF LINEAR SVM USING EMAIL SPAM DETECTION

- Now with this example our feature space is $N+1$ dimensional and imagining the data would be impossible. So for simplicity we are considering only two feature X_1 and X_2 based on which the emails can be represented (shown in graph). The green points represent non-spam emails, the blue one's represent spam emails.

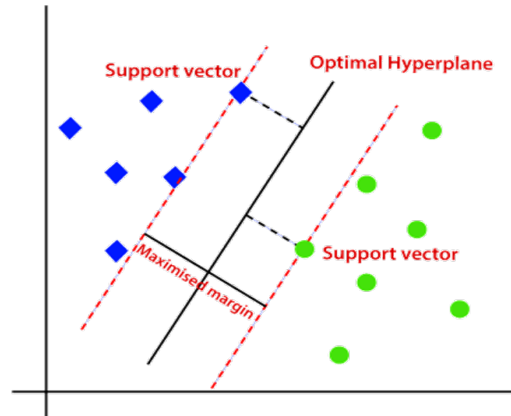


Figure. 2D Model of linear SVM

EXPLANATION OF LINEAR SVM USING EMAIL SPAM DETECTION

- ▶ **Support Vectors:** These are the points that are closest to the hyperplane/decision boundary. A separating line will be defined with the help of these data points.
- ▶ **Hyperplane:** It refers to a subspace whose dimension is one less than the ambient space we are working in.
- ▶ The decision boundary is calculated via support vectors, hence the name.
- ▶ Now seeing the distribution, we can separate this data using a decision boundary/line.
- ▶ The best hyperplane has the maximum distance from both the classes, This is done by finding different hyperplanes which classify the labels in the best way then it will choose the one which is farthest from the data points or the one which has a maximum margin.

EXPLANATION OF LINEAR SVM USING EMAIL SPAM DETECTION

Hence, SVM is basically an optimization problem that tries to find a separating hyperplane with as large a margin as possible. It can be written as a constrained optimization problem as follows:

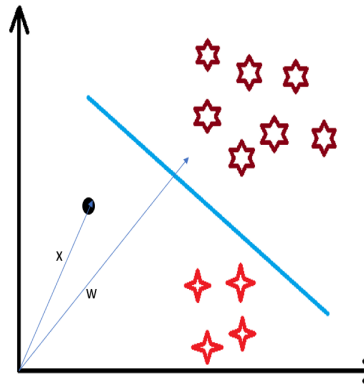
$$\begin{aligned} \min_{(w,b)} \quad & \frac{1}{\gamma(w,b)} \\ \text{subj. to } & y_n (w \cdot x_n + b) \geq 1 \end{aligned}$$

Here, γ denotes the margin which we have to maximize. we have put the condition that the classification of each point to be greater than or equal to 1. However, this can be any other positive constant. The constant just ensures that there is a separation between the two classes.

MATHEMATICAL INTUITION OF LINEAR-SVM

DOT PRODUCT IN LINEAR SVM

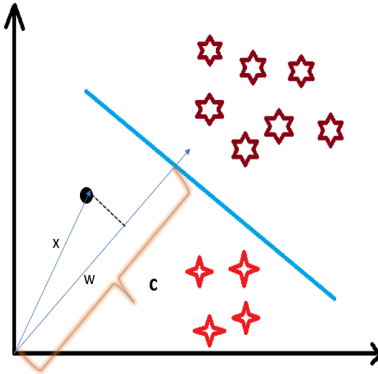
- ▶ Consider a random point X from dataset and we want to know whether it lies on the right side of the plane or the left side of the plane, i.e. whether it is positive or negative data.
- ▶ we assume this point is a vector (X) and then we make a vector (w) which is perpendicular to the hyperplane.



MATHEMATICAL INTUITION OF LINEAR-SVM

DOT PRODUCT IN LINEAR SVM

- ▶ Now we assume distance of vector w from origin to decision boundary is ' c '. Now we take the projection of X vector on w .
- ▶ We already know that projection of any vector or another vector is called dot-product. Hence, we take the dot product of x and w vectors.



MATHEMATICAL INTUITION OF LINEAR-SVM

DOT PRODUCT IN LINEAR SVM

- So Now the decision whether the data point is positive or negative can be taken like this:

$$\vec{X} \cdot \vec{w} = c \quad (\text{the point lies on the decision boundary})$$

$$\vec{X} \cdot \vec{w} > c \quad (\text{positive samples})$$

$$\vec{X} \cdot \vec{w} < c \quad (\text{negative samples})$$

- Hence, now our decision rules looks like:

$$\vec{X} \cdot \vec{w} - c \geq 0$$

putting $-c$ as b , we get

$$\vec{X} \cdot \vec{w} + b \geq 0$$

hence,

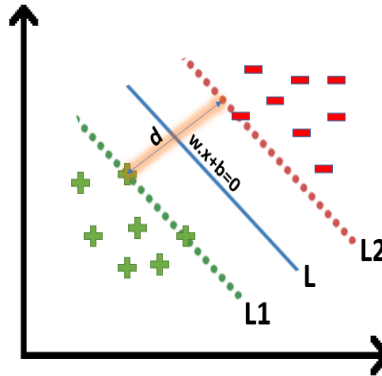
$$y = \begin{cases} +1 & \text{if } \vec{X} \cdot \vec{w} + b \geq 0 \\ -1 & \text{if } \vec{X} \cdot \vec{w} + b < 0 \end{cases}$$

MATHEMATICAL INTUITION OF LINEAR-SVM

DECIDING THE MARGIN

- ▶ Now we need (w, b) such that the margin has a maximum distance. Let's say this distance is ' d '.
- ▶ To calculate ' d ' we need the equation of **L1** and **L2**. For this, we will take few assumptions that the equation of:

L1 is $w \cdot x + b = 1$ and,
for **L2** it is: $w \cdot x + b = -1$



MATHEMATICAL INTUITION OF LINEAR-SVM

OPTIMIZING THE EQUATIONS TO GET MAXIMUM MARGIN

- ▶ Now the constraint is to calculate the distance (d) in such a way that no positive or negative point can cross the margin line. Which can be written mathematically as:

for all the Negative points $\vec{w} \cdot \vec{X} + b \leq -1$

For all the Positive points $\vec{w} \cdot \vec{X} + b \geq 1$

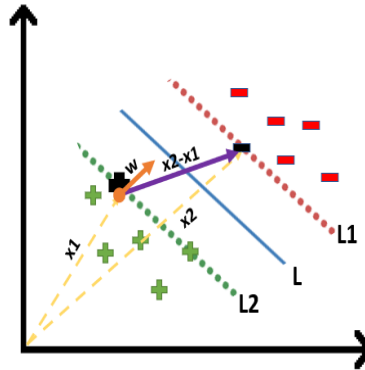
- ▶ Now to reduce the two equation constraints into one equation by assuming that negative classes have $y=-1$ and positive classes have $y=1$.
- ▶ We can say that for every point to be correctly classified this condition should always be true:

$$y_i(\vec{w} \cdot \vec{X} + b) \geq 1$$

MATHEMATICAL INTUITION OF LINEAR-SVM

OPTIMIZING THE EQUATIONS TO GET MAXIMUM MARGIN

- ▶ Now We will take 2 support vectors, 1 from the negative class and 1 from the positive class. The distance between these two vectors x_1 and x_2 will be $(x_2 - x_1)$ vector.
- ▶ The shortest distance between these two points can be found by doing dot product with unit vector (\vec{w}) .



MATHEMATICAL INTUITION OF LINEAR-SVM

OPTIMIZING THE EQUATIONS TO GET MAXIMUM MARGIN

► Therefore we get:

$$\Rightarrow (x_2 - x_1) \cdot \frac{\vec{w}}{\|\vec{w}\|} \Rightarrow \frac{x_2 \cdot \vec{w} - x_1 \cdot \vec{w}}{\|\vec{w}\|}$$

► Since x_2 and x_1 are support vectors and they lie on the hyperplane, hence they will follow y_i^* ($2 \cdot x + b = 1$) so we can write it as:

for positive point $y = 1$

$$\begin{aligned} \Rightarrow 1 \times (\vec{w} \cdot x_1 + b) &= 1 \\ \Rightarrow \vec{w} \cdot x_1 &= 1 - b \quad \text{--- (2)} \end{aligned}$$

Similarly for negative point $y = -1$

$$\begin{aligned} \Rightarrow -1 \times (\vec{w} \cdot x_2 + b) &= 1 \\ \Rightarrow \vec{w} \cdot x_2 &= -b - 1 \quad \text{--- (3)} \end{aligned}$$

MATHEMATICAL INTUITION OF LINEAR-SVM

OPTIMIZING THE EQUATIONS TO GET MAXIMUM MARGIN

- ▶ Putting equations (2) and (3) in equation (1) we get:

$$\Rightarrow \frac{(1 - b) - (-b - 1)}{\|w\|}$$
$$\Rightarrow \frac{1 - b + b + 1}{\|w\|} = \frac{2}{\|w\|} = d$$

- ▶ Hence the equation which we have to maximize is:

$$\operatorname{argmax} (w^*, b^*) \frac{2}{\|w\|} \text{ such that } y_i(\vec{w} \cdot \vec{X} + b) \geq 1$$

PROBLEM WITH LINEAR SVM

- ▶ It works only if the data is linearly separable.
- ▶ Is prone to over-fitting and under-fitting.
- ▶ It is alot sensitive to outliers, Algorithm might not terminate if the data is too noisy as a Linear boundary might not be possible
- ▶ Is not good if feature space is too large as it becomes computationally expensive.

INDUCTIVE BIAS HYPERPARAMETER

Inductive Bias

- ▶ Data is linearly separable

Hyper-parameter

- ▶ **C** : Regularization parameter determines the trade-off between achieving a low training error and a low testing error
- ▶ **Tolerance** : Determines the stopping criteria for optimization
- ▶ **Max. no. of iterations** : Limits the number of iterations in the optimization algorithm

SOFT MARGIN- SVM

- ▶ In real-life applications we don't find any dataset which is linearly separable, what we'll find is either an almost linearly separable dataset or a non-linearly separable dataset.
- ▶ to tackle this problem what we do is modify that equation in such a way that it allows few misclassifications that means it allows few points to be wrongly classified.

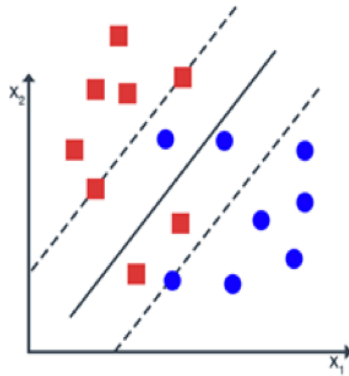


Figure. Soft Margin Linear SVM

REFERENCES:

- ▶ Mishra, Subhankar (2023).“CS460: Machine Learning 2023 Lectures”.
- ▶ Support Vector Machine(SVM): A Complete guide for beginners
- ▶ Wikipedia- Support vector machine
- ▶ MIT Open CourseWare- Learning: Support Vector Machines
- ▶ Chat GPT by open AI