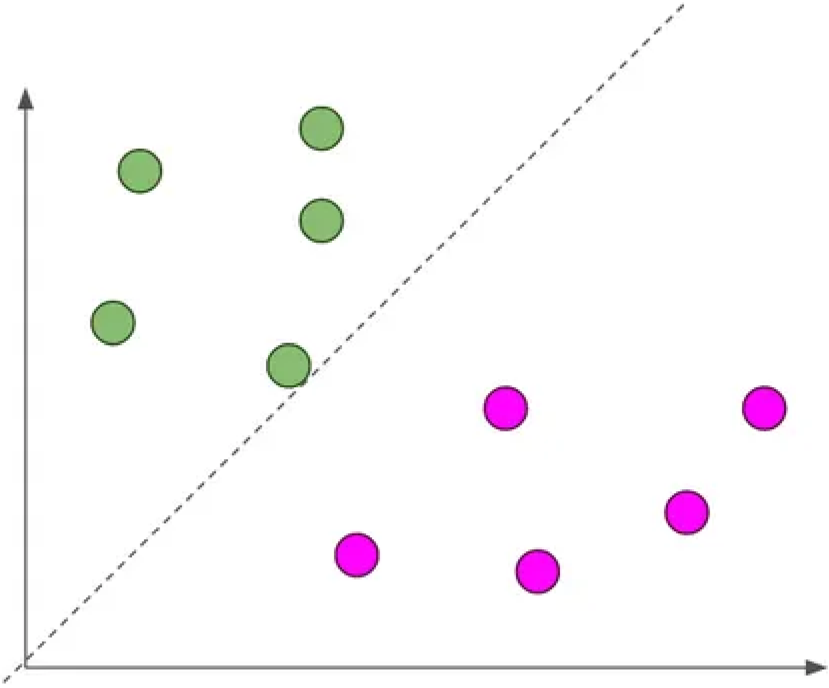
Math behind SVM (Support Vector Machine)

MLMath.io · Follow

9 min read · Feb 10, 2019

SVM is one of the most popular, versatile supervised machine learning algorithm. It is used for both classification and regression task.But in this thread we will talk about classification task. It is usually preferred for medium and small sized dataset.

> The main objective of SVM is to find the optimal hyperplane which linearly separates the data points in two component by maximizing the margin .



dotted line is hyperplane, separating blue and pink classes balls.

Don't panic with the word 'hyperplane' and 'margin' that you have seen above. It is explained in details below.

Points to be covered:

> *1. Basic linear algebra*

Basic Linear Algebra

Vectors

Vectors are mathematical quantity which has both magnitude and direction. A point in the 2D plane can be represented as a vector between origin and the point.

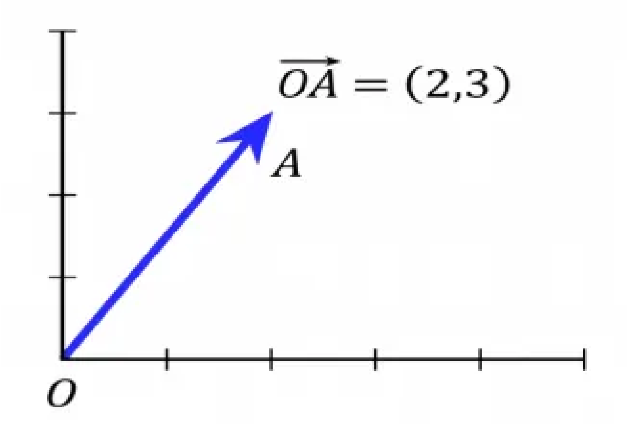


Fig.-1

$\vec{OA}$ is a vector and length between O and A is its magnitude.

Length of Vectors

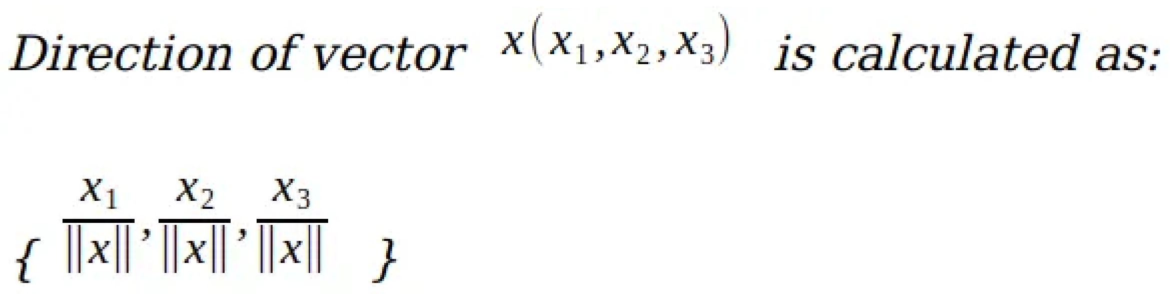
Length of vectors are also called as norms. It tells how far vectors are from the origin.

Length of vector $x(x_1, x_2, x_3)$ *is calculated as :*

$$||x|| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

Direction of vectors

Direction of vector .

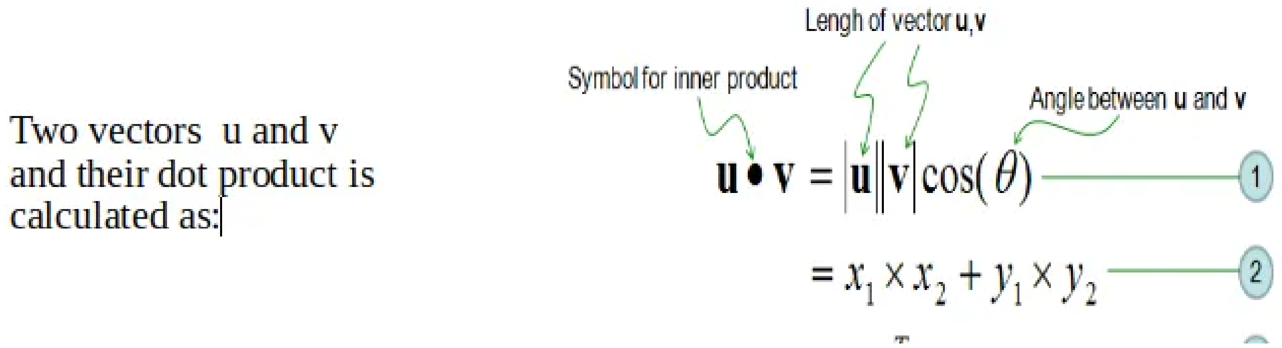


Direction of vector $x(x_1, x_2, x_3)$ is calculated as:

$$\left\{ \frac{x_1}{\|x\|}, \frac{x_2}{\|x\|}, \frac{x_3}{\|x\|} \right\}$$

Dot Product

Dot product between two vectors is a scalar quantity . It tells how to vectors are related.



Two vectors u and v and their dot product is calculated as:

$$\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}||\mathbf{v}|\cos(\theta) \quad (1)$$

$$= x_1 \times x_2 + y_1 \times y_2 \quad (2)$$

Hyper-plane

It is plane that linearly divide the n-dimensional data points in two component. In case of 2D, hyperplane is line, in case of 3D it is plane.It is also called as *n-dimensional line. Fig.3* shows, a blue line(hyperplane) linearly separates the data point in two components.

Fig.3

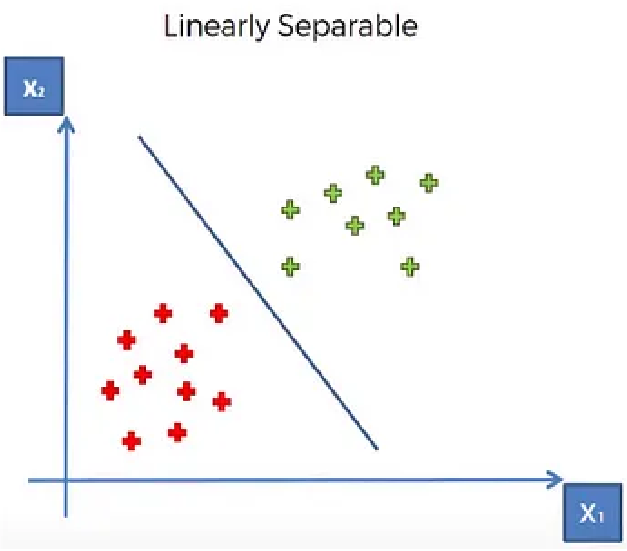
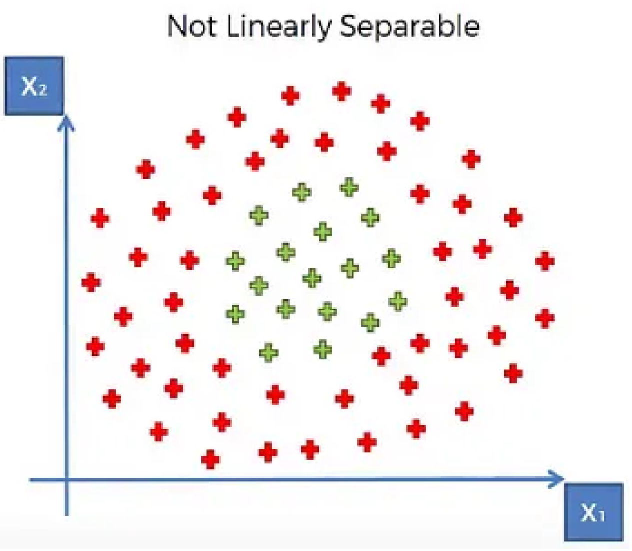


Fig.4



In the *Fig.3, hyperplane* is line divides data point into two classes(red & green), written as

$$y = a * x + b$$

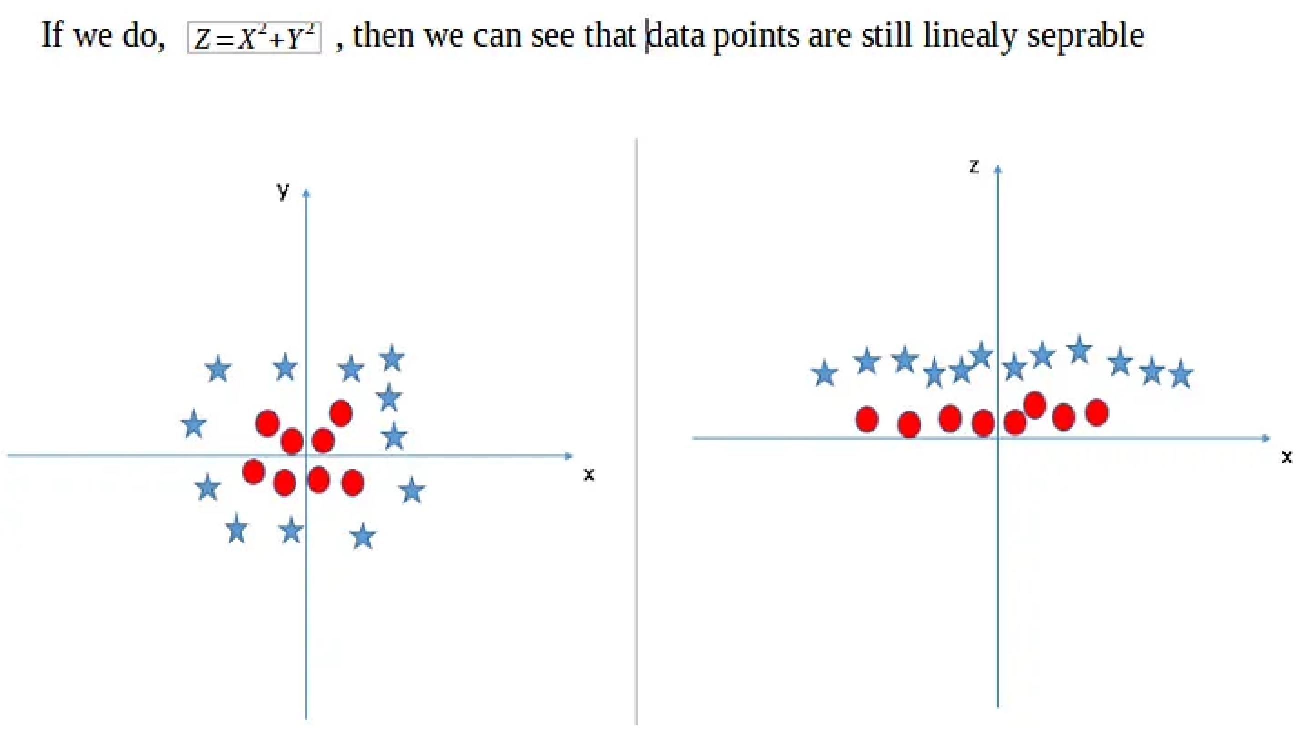
$$a * x + b - y = 0$$

Let vector X=(x,y) and W=(a,-1) then in vector form hyperplane is

$$W . X + b = 0$$

What if data points is not linearly separable ?

Look *Fig.4* how can we separate the data-points linearly? This type of situation comes very often in machine learning world as raw data are always non-linear here.So, *Is it do-able? yes!!.* we will add one extra dimension to the data points to make it separable.



If we do, $Z=X^2+Y^2$, then we can see that data points are still linealy seprable

FIg.5

so , the above process of making non-linearly separable data point to linearly separable data point is also known as Kernel Trick, which we will cover later in details.

Optimal Hyperplane

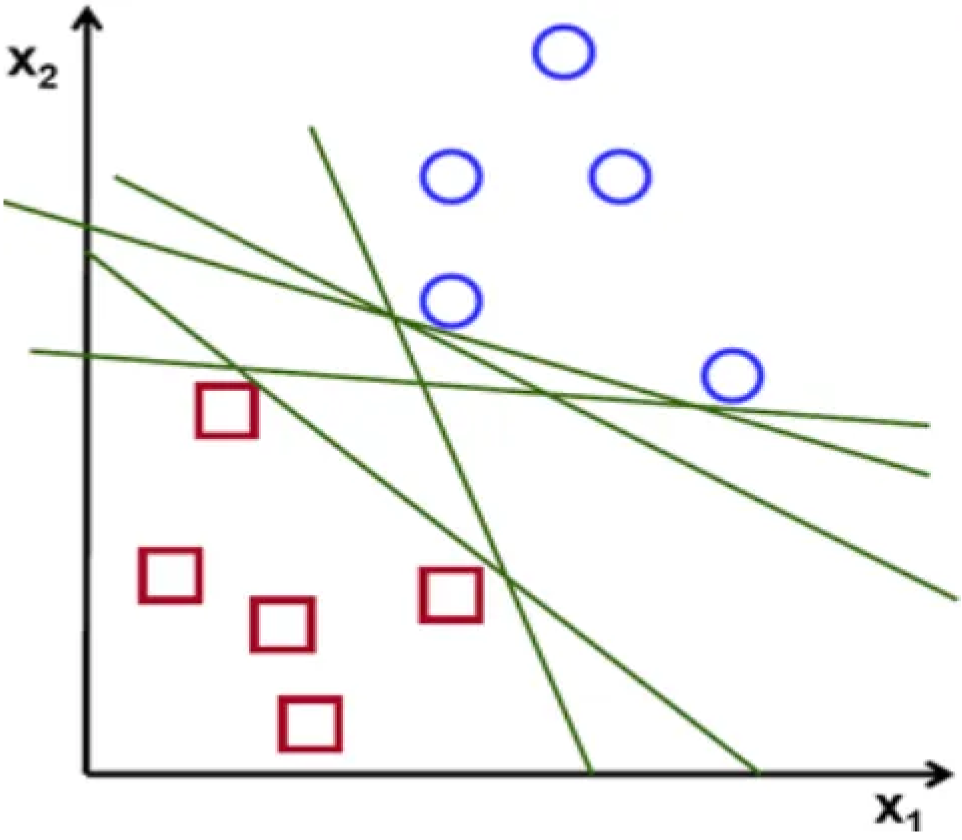


Fig.6

If you look above ***Fig*** there are numbers of hyperplane that can separate the data points in two components. ***So optimal hyperplane is one which divides the data points very well.*** So question is why it is needed to choose optimal hyper plane?

So if you choose sub-optimal hyperplane, no doubt after number of training iteration , training error will decrease but during testing when an unseen instance will come, it will result in high test error. In that case it is must to choose an optimal plane to get good accuracy.

How to choose Optimal Hyperplane ?

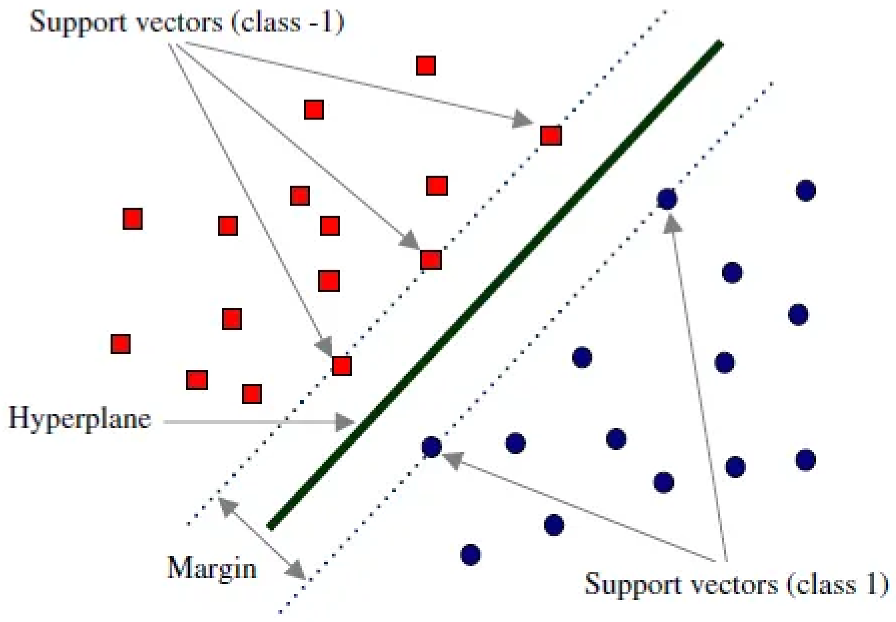


Fig.7

Margin and Support Vectors

Let's assume that solid black line in above Fig.7 is optimal hyperplane and two dotted line is some hyperplane, which is passing through nearest data points to the optimal hyperplane. Then distance between hyperplane and optimal hyperplane is know as margin, and the closest data-points are known as support vectors. Margin is an area which do not contains any data points. There will be some cases when we have data points in margin area but right now we stick to margin as no data points lands.

So, when we are choosing optimal hyperplane we will choose one among set of hyperplane which is highest distance from the closest data points. If optimal hyperplane is very close to data points then margin will be very small and it will generalize well for training data but when an unseen data will come it will fail to generalize well as explained above. So our goal is to maximize the margin so that our classifier is able to generalize well for unseen instances.

So, In SVM our goal is to choose an optimal hyperplane which maximizes the margin.

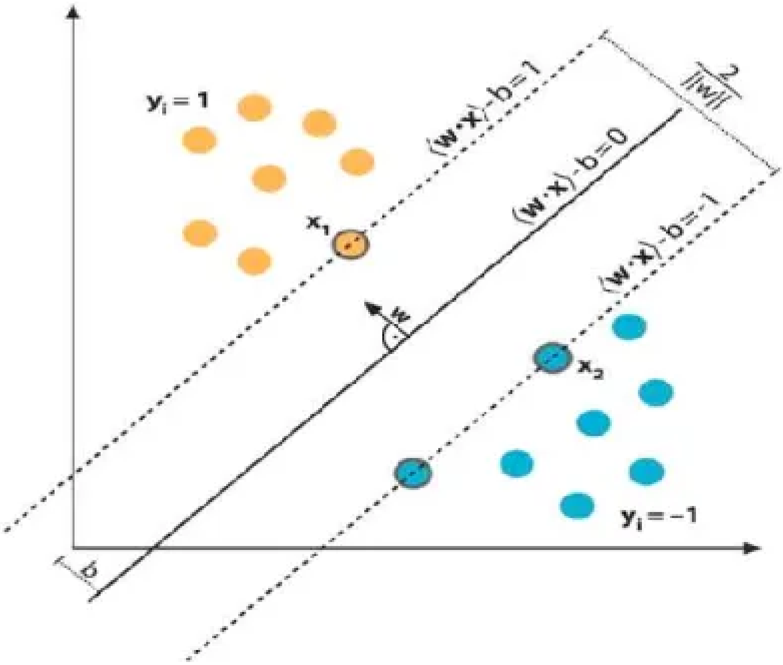
— — — — — — —

Since covering entire concept about SVM in one story will be very confusing. So i will be dividing the tutorial into three parts.

1. Linear separable data points.
2. Linear separable data points II.
3. Non-linear separable data-points.

So, In this story we will assume that data points(training data) are linearly separable.Lets start,

Mathematical Interpretation of Optimal Hyperplane


$y_i = 1$
x_1
$\langle w \cdot x \rangle - b = 1$
$\langle w \cdot x \rangle - b = 0$
$\langle w \cdot x \rangle - b = -1$
$\frac{2}{\|w\|}$
w
x_2
$y_i = -1$
b

we have *l* training examples where each example *x* are of *D* dimension and each have labels of either *y=+1 or y= -1* class, and our examples are linearly separable. Then, our training data is form ,

$$\{\mathbf{x}_i, y_i\} \quad \text{where} \quad i = 1 \dots L, \; y_i \in \{-1, 1\}, \; \mathbf{x} \in \Re^D$$

We consider *D=2* to keep explanation simple and data points are linearly separable, The hyperplane *w.x+b=0* can be described as :

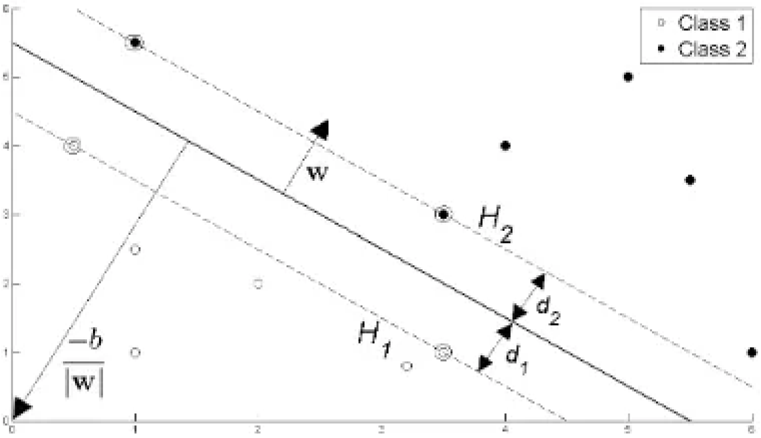
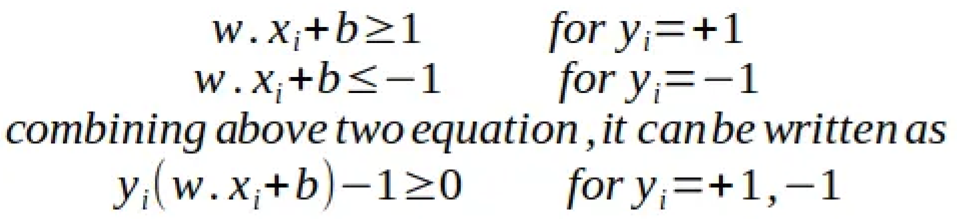


Fig.8

Support vectors examples are closest to optimal hyperplane and the aim of the SVM is to orientate this hyperplane as far as possible from the closest member of the both classes.

From the above Fig , SVM problem can be formulated as,



$$w.x_i+b\geq 1 \qquad for\ y_i=+1$$
$$w.x_i+b\leq -1 \qquad for\ y_i=-1$$
$$combining\ above\ two\ equation\ , it\ can\ be\ written\ as$$
$$y_i(w.x_i+b)-1\geq 0 \qquad for\ y_i=+1,-1$$

From the Fig.8 we have two hyperplane H1 and H2 passing through the support vectors of +1 and -1 class respectively. so

w.x+b=-1 :H1

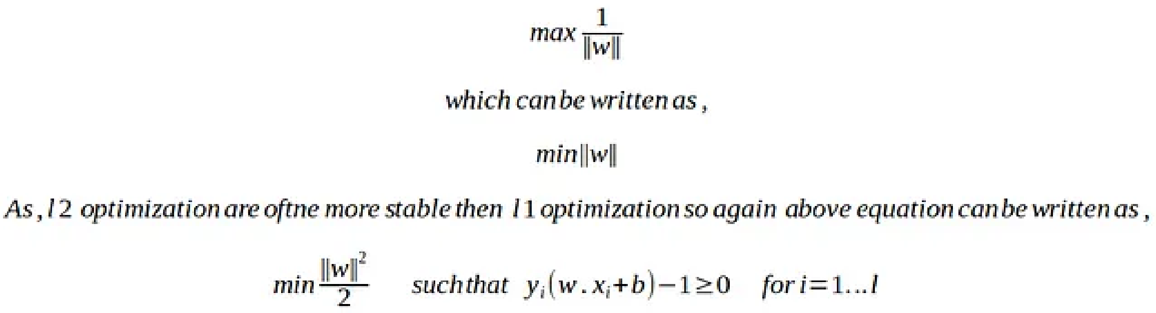
w.x+b=1 :H2

And distance between H1 hyperplane and origin is *(-1-b)/|w|* and distance between H2 hyperplane and origin is *(1–b)/|w|*. So, margin can be given as

M=(1-b)/|w|-(-1-b)/|w|

M=2/|w|

Where *M* is nothing but twice of the margin. So margin can be written as 1/|w|. As, optimal hyperplane maximize the margin, then the SVM objective is boiled down to fact of maximizing the term *1/|w|*,



$$\max \frac{1}{\|w\|}$$

which can be written as ,

$$\min \|w\|$$

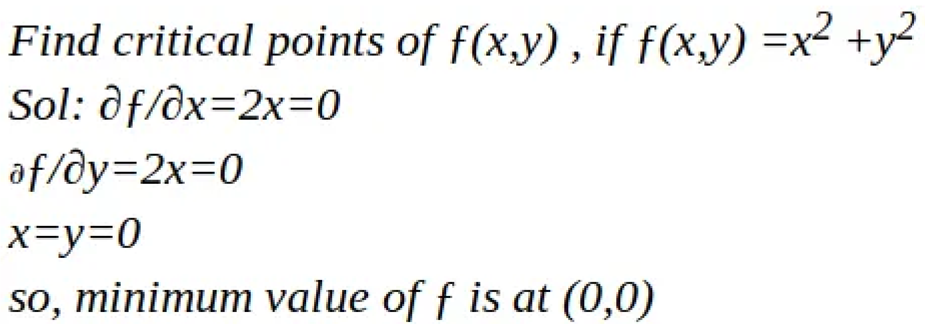
As , l2 optimization are oftne more stable then l1 optimization so again above equation can be written as ,

$$\min \frac{\|w\|^2}{2} \quad such\,that \quad y_i(w.x_i+b)-1\geq 0 \quad for\, i=1\ldots l$$

Basic optimization algorithm terms

Unconstrained optimization

Example will be more intuitive in explaining the concept,



Find critical points of f(x,y) , if f(x,y) =x^2 +y^2

Sol: $\partial f/\partial x=2x=0$

$\partial f/\partial y=2x=0$

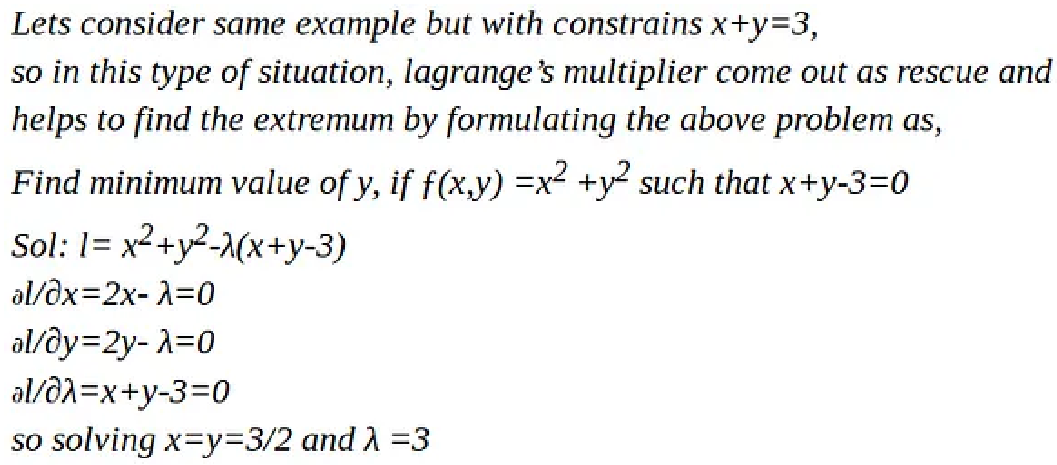
$x=y=0$

so, minimum value of f is at (0,0)

So, it is same that we used to do in higher school in calculus for finding maxima and minima of a function. Only difference is at that time we were calculating for univariate variable, but now we are calculating for multivariate variables.

Constrained optimization

Again it will become clear with an example,



Lets consider same example but with constrains x+y=3,
so in this type of situation, lagrange's multiplier come out as rescue and
helps to find the extremum by formulating the above problem as,

Find minimum value of y, if f(x,y) =x^2 +y^2 such that x+y-3=0

Sol: l= x^2+y^2-λ(x+y-3)

$\partial l/\partial x=2x- \lambda=0$

$\partial l/\partial y=2y- \lambda=0$

$\partial l/\partial \lambda=x+y-3=0$

so solving x=y=3/2 and λ =3

So basically we calculate the maxima and minima of a function by taking into the consideration of given constraints on the variables.

Primal and Dual Concept

Shocked!! what the heck is this now???

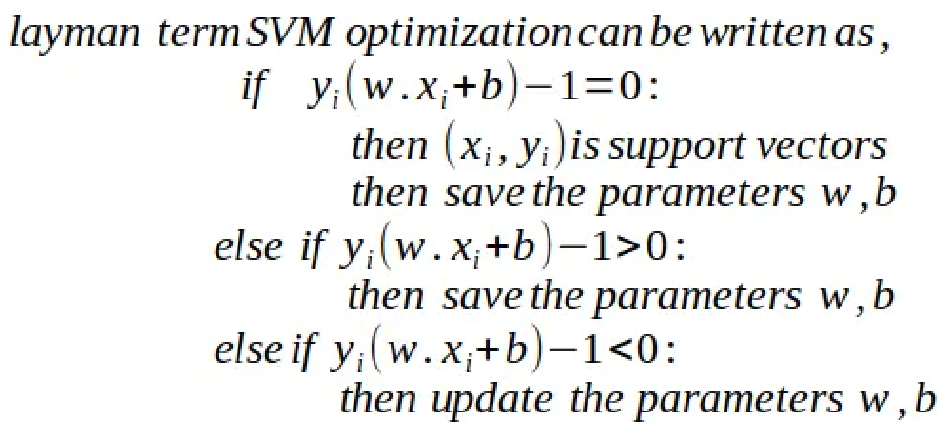
Don't worry, it 's theory only !!

Lets not get deep into these concept. Any optimization problem can be formulated into two way, primal and dual problem. First we use primal formulation for optimization algorithm, but if it does not yield any solution, we go for dual optimization formulation, which is guaranteed to yield solution.

Optimization

This part will be more mathematical, some terms are very high level concept of mathematics, but don't worry i will try to explain each one by one in layman term.

To make you comfortable, Learning algorithms of SVM are explained with pseudo code explain below. This is very abstract concept in SVM optimization. For below code assume x is data point and y is its corresponding labels.



```
layman term SVM optimization can be written as,
        if  $y_i(w.x_i+b)-1=0$:
                then $(x_i, y_i)$ is support vectors
                then save the parameters $w, b$
        else if $y_i(w.x_i+b)-1>0$:
                then save the parameters $w, b$
        else if $y_i(w.x_i+b)-1<0$:
                then update the parameters $w, b$
```

Just above you see learning process of SVM. But here is a catch, how do we update w and b ??

Gradient descent of course!!! — -BIG 'NO'.

why not Gradient descent??

SVM optimization problem is a case of constrained optimization problem, and it is always preferred to use dual optimization algorithm to solve such constrained optimization problem. That's why we don't use gradient descent.

Since it is constrained optimization problem Lagrange multipliers are used to solve it, which is described below, It looks like , will be more mathematical but it is not, its just few steps of finding gradient. We will divide the complete formulation into three parts.

1. In first we will formulate SVM optimization problem Mathematically
2. we will find gradient with respect to learning parameters.
3. we will find the value of parameters which minimizes $||w||$

$$\min \|w\|^2 \quad \text{such that } y_i(w\,.\,x_i+b)-1\geq 0 \quad \text{for } i=1..l$$

$$\text{minimizing } \|w\|^2 \text{ is equivalent as minimizing } \frac{\|w\|^2}{2}$$

$$l=\min \frac{\|w\|}{2} \qquad \text{such that } y_i(w\,.\,x_i+b)-1\geq 0 \quad \text{for } i=1..l$$

PART I — Problem formulation

The above equation is Primal optimization problem.Lagrange method is required to convert constrained optimization problem into unconstrained optimization problem. The goal of above equation to get the optimal value for w and b.

so using lagrange multipliers λ we can write ,

$$l=\frac{\|w\|^2}{2}-\sum_{i=1}^{l}\lambda_i\left(y_i\left(w.x_i+b\right)-1\right) \qquad eq.1$$

$$l=\frac{\|w\|^2}{2}-\sum_{i=1}^{l}\lambda_i\left(y_i\left(w.x_i+b\right)\right)-\sum_{i=1}^{l}\lambda_i \qquad eq.2$$

$$\partial l/\partial w=w-\sum_{i=1}^{l}\lambda_i.y_i.x_i=0$$

$$\partial l/\partial \lambda=\sum_{i=1}^{l}y_i\left(w.x_i+b\right)-1=0$$

$$\partial l/\partial b=\sum_{i=1}^{l}\lambda_i.y_i=0$$

PART II — — — finding the gradient with respect to w ,b and lambda.

May be above equation is looking tricky? but it is not, its just high school math of finding minima with respect to variable.

$$\sum_{i=1}^{l}\lambda_i.y_i=0$$

$$w=\sum_{i=1}^{l}\lambda_i.y_i.x_i$$

PART III — — — — we will get the value of w

As,from the above formulation we only able to find the optimal value of w and that is to dependent on λ, so we need to find the optimal value of λ also. And finding optimal value of b needs both w and λ. So finding the value of λ will be the important for us.

so how do we find the value of λ???

Above formulation is itself a optimization algorithm, but it not helpful to find the optimal value. It is primal optimization problem. As we read above that if Primal optimization doesn’t result in solution, we should use dual optimization formulation, which has guaranteed solution. Also when we move from primal to dual formulation we switch minimizing to maximizing the loss function. Again we will divide the complete formulation into three parts to easier to understand.

1. Problem formulation and substitution value from primal
2. Simplify the loss function equation after substitution
3. final optimization formulation to get the value of λ

$$l_d=\frac{\|w\|^2}{2}-\sum_{i=1}^{l}\lambda_i(y_i(w.x_i+b))+\sum_{i=1}^{l}\lambda_i$$

substituting the value of $w=\sum_{i=1}^{l}\lambda_i.y_i.x_i$ *into above equation,*

$$l_d=\frac{\sum_{i=1}^{l}\|\lambda_i.y_i.x_i\|^2}{2}-\sum_{j=1}^{l}\sum_{i=1}^{l}\lambda_j y_j(\|\lambda_i.y_i.x_i\|.x_j+b)+\sum_{i=1}^{l}\lambda_i$$

PART I — — — — Formulation from primal and Substitution

Above equation is a dual optimization problem. The equation are looking scarier because of substitution of value of *w*.

$$l_d=\frac{\sum_{i=1}^{l}(\lambda_i.y_i.x_i)^T.(\lambda_i.y_i.x_i)}{2}-\sum_{j=1}^{l}\sum_{i=1}^{l}\lambda_j\lambda_i y_j y_i x_i x_j+\sum_{j=1}^{l}\sum_{i=1}^{l}\lambda_j y_j-\sum_{i=1}^{l}\lambda_i$$

since constraint is $\lambda_j\geq 0,\ \sum_{i=1}^{l}\lambda_i.y_i=0$ *so, 3rd term become zero, it can be written as,*

$$l_d=\sum_{i=1}^{l}\lambda_i-\frac{\sum_{i,j=1}^{l}\lambda_j\lambda_i y_j y_i x_i x_j}{2}$$

PART II — — — — — — — — — Simplification

It is simplified equation of above dual optimization problem.

$$\text{where } K(i,j) = y_j y_i \, x_i x_j \, , \quad \text{so in vector form } K = y^T y . x^T x \, , \text{ so}$$

$$\max l_d = \sum_{i=1}^{l} \lambda_i - \frac{\lambda^T \lambda K}{2} \qquad eq.3$$

PART III — — — — — -Final optimization

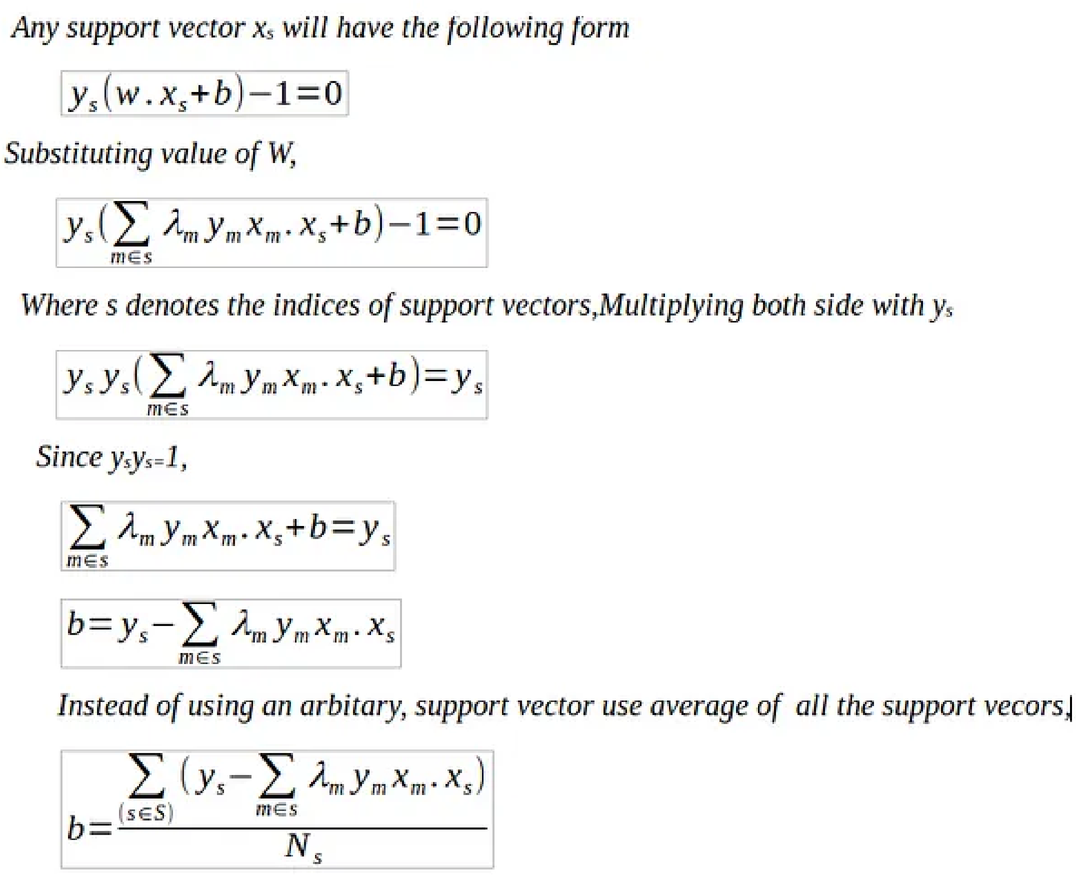
So this is the final optimization problem, to find the maximum value of λ. Here one more term K is there, which is nothing but dot product of input variable *x.* (This K will be very important in future when we will learn about kernel trick and non-linear data points).

Now How do we Solve the above problem??

Above maximization operation can be solved with the SMO (sequential minimization optimization) algorithms . There are also the various library support online for this optimization. Once we get the value of λ we can get w from below equation

$$\boxed{w = \sum_{i=1}^{l} \lambda_i . y_i . x_i}$$

and using value of *w* , *λ* we will calculate *b* as following,



Any support vector x_s will have the following form

$$\boxed{y_s(w.x_s+b)-1=0}$$

Substituting value of W,

$$\boxed{y_s(\sum_{m\in s}\lambda_m y_m x_m.x_s+b)-1=0}$$

Where s denotes the indices of support vectors,Multiplying both side with y_s

$$\boxed{y_s y_s(\sum_{m\in s}\lambda_m y_m x_m.x_s+b)=y_s}$$

Since $y_s y_s=1$,

$$\boxed{\sum_{m\in s}\lambda_m y_m x_m.x_s+b=y_s}$$

$$\boxed{b=y_s-\sum_{m\in s}\lambda_m y_m x_m.x_s}$$

Instead of using an arbitary, support vector use average of all the support vecors,

$$\boxed{b=\frac{\sum_{(s\in S)}(y_s-\sum_{m\in s}\lambda_m y_m x_m.x_s)}{N_s}}$$

As we now have the value for both w and b , then optimal hyperplane that can separates the data points can be written as,

> w.x + b = 0
>
> And a new example x_ can be classified as sign(w.x_ +b)

Thanks for reading, this is the PART1 of SVM, in PART 2 we will discuss about how to deal with a case when data is not fully linearly separable.

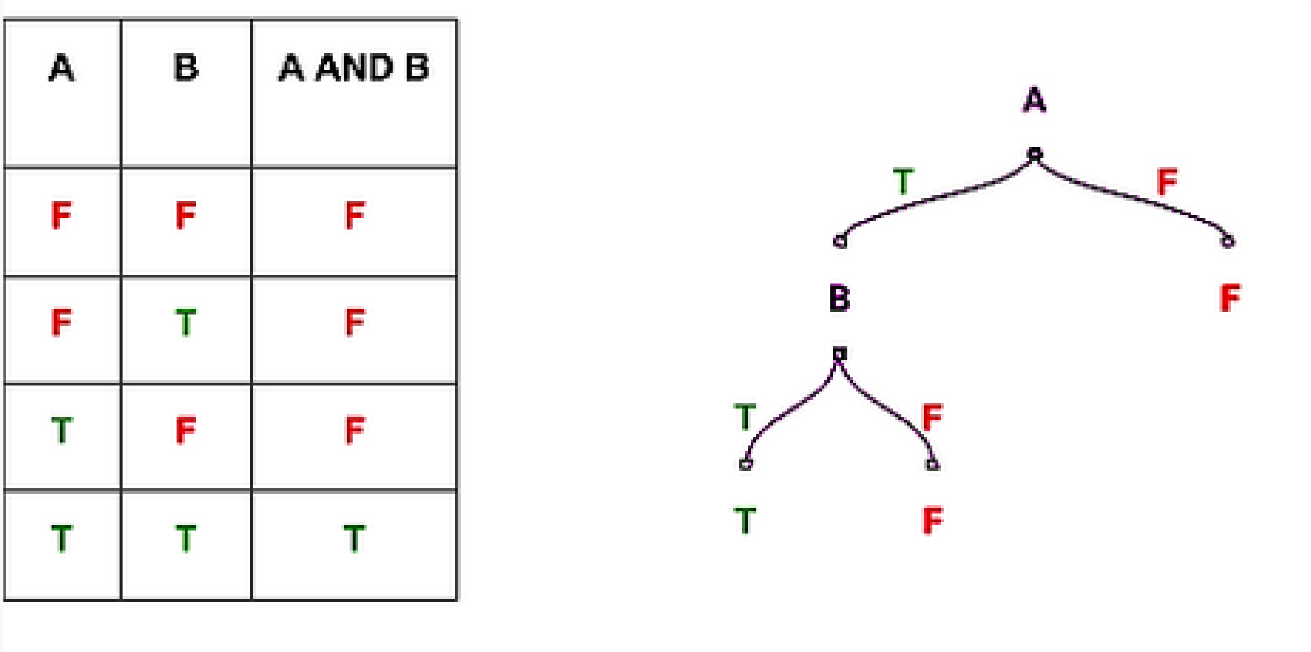
Machine Learning Svm Deep Learning Optimization

Written by MLMath.io

628 Followers

Machine learning | Deep Learning | Reinforcement Learning | Probability

More from MLMath.io



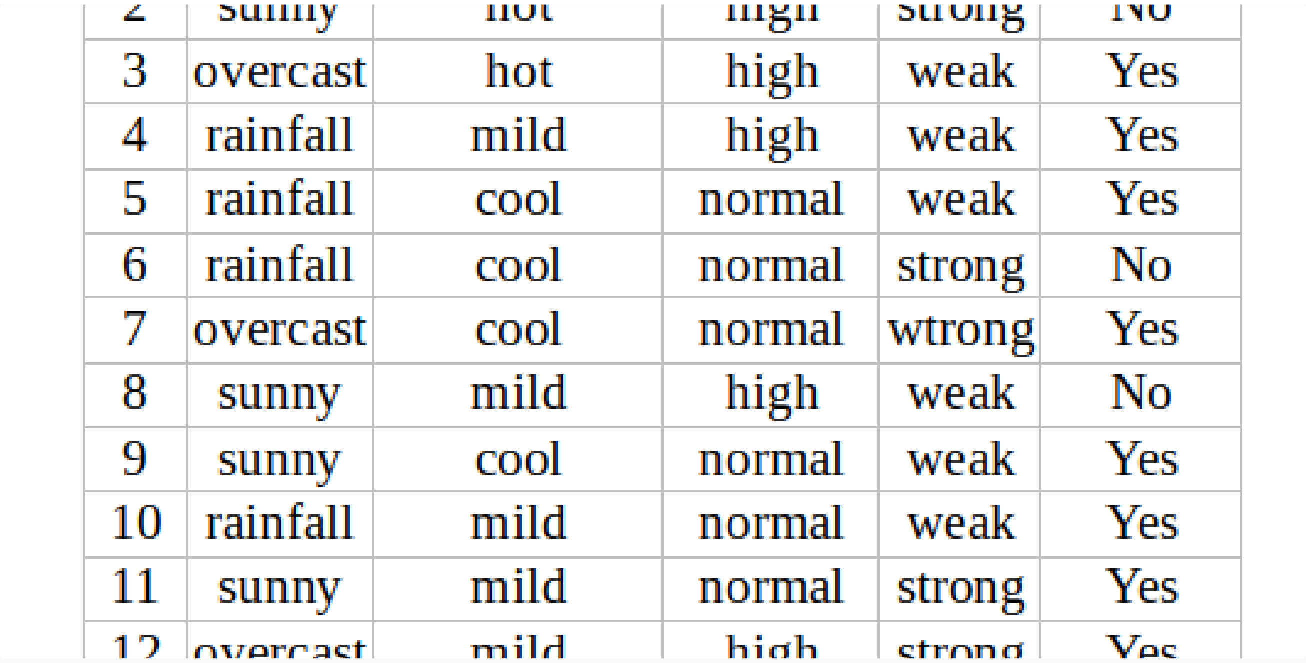
A	B	A AND B
F	F	F
F	T	F
T	F	F
T	T	T

MLMath.io

Math behind Decision Tree Algorithm

Decision tree algorithm is one of the most popular machine learning algorithm. It is a supervised machine learning algorithm, used for...

Feb 19, 2019 679 6



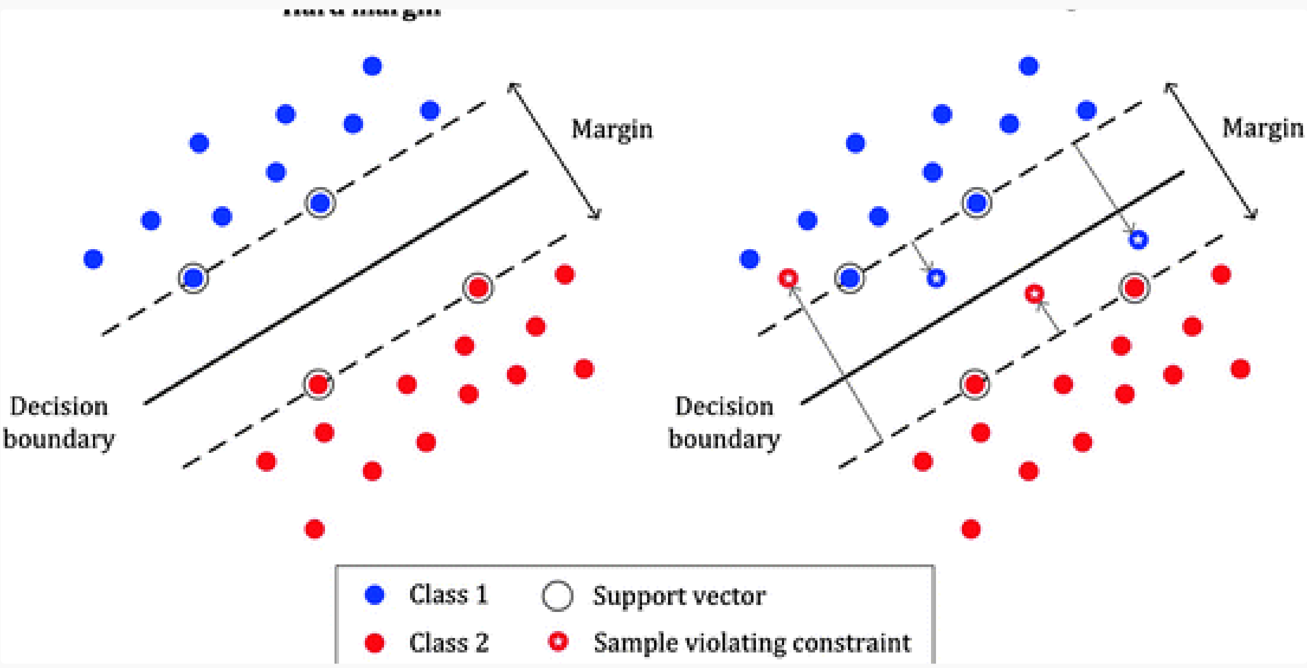
2	sunny	hot	high	strong	No
3	overcast	hot	high	weak	Yes
4	rainfall	mild	high	weak	Yes
5	rainfall	cool	normal	weak	Yes
6	rainfall	cool	normal	strong	No
7	overcast	cool	normal	wtrong	Yes
8	sunny	mild	high	weak	No
9	sunny	cool	normal	weak	Yes
10	rainfall	mild	normal	weak	Yes
11	sunny	mild	normal	strong	Yes
12	overcast	mild	high	strong	Yes

MLMath.io

Decision tree Algorithm (ID3)

This is 2nd part of Decision tree tutorial. In last part we talk about Introduction of decision tree, Impurity measures and CART algorithm...

Feb 20, 2019 195 1

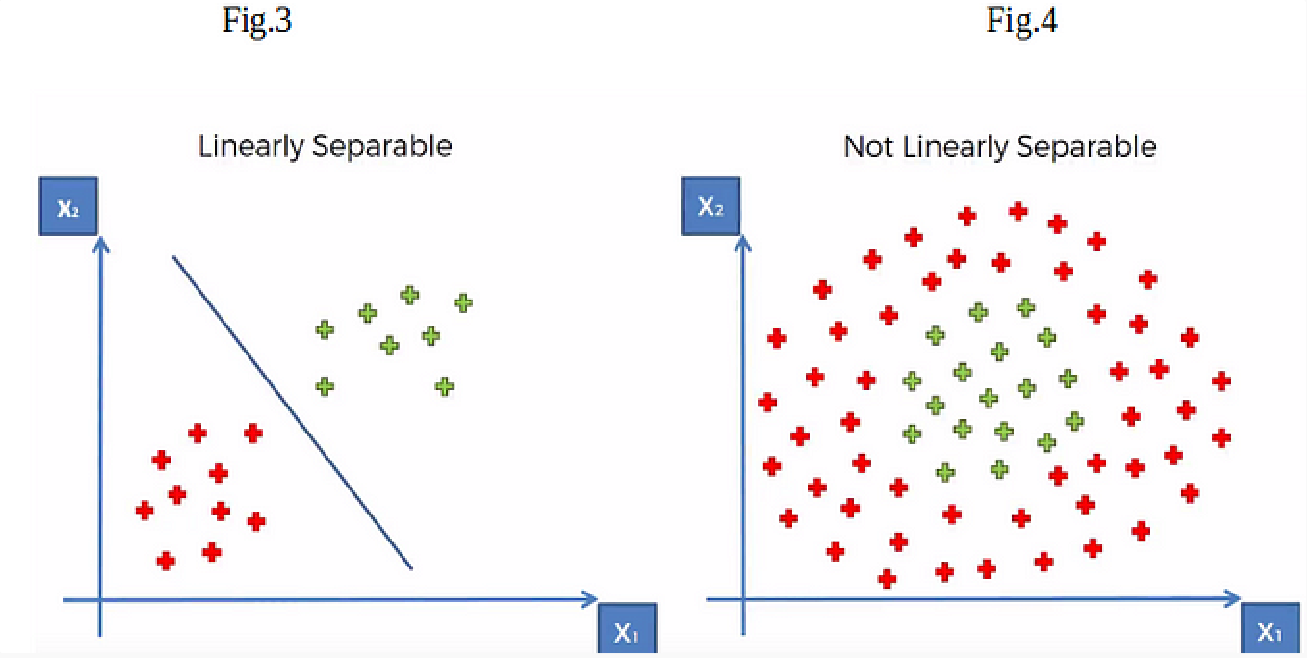


MLMath.io

Math behind SVM(Support Vector Machine)

This is PART II of series of SVM.

Feb 13, 2019 191 4



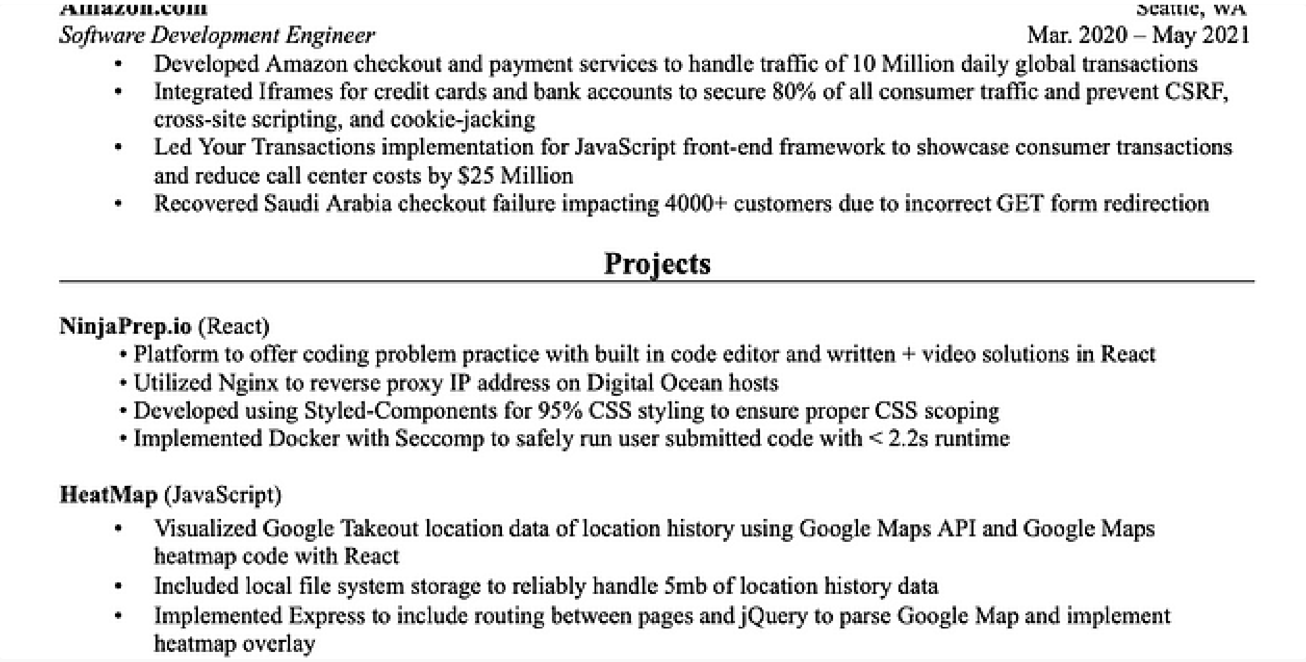
MLMath.io

Math Behind SVM(Kernel Trick)

This is PART III of SVM Series

Feb 16, 2019 130

Recommended from Medium



Amazon.com Seattle, WA

Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

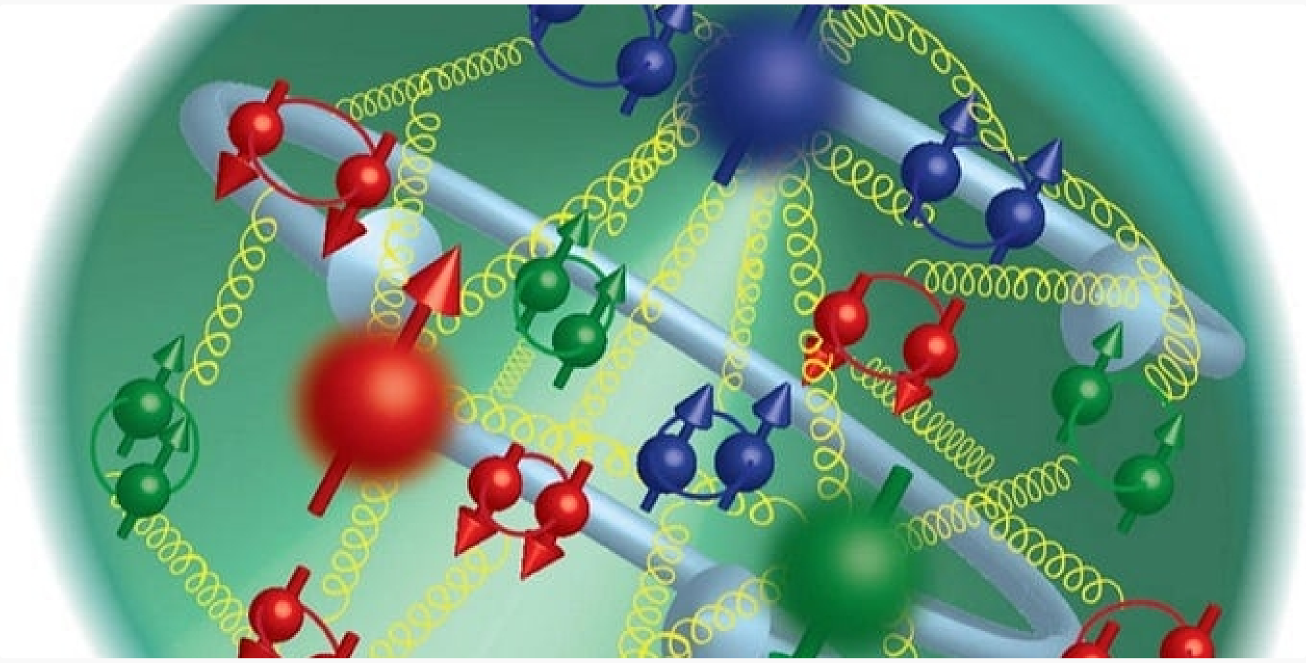
- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

Jun 1 20K 378



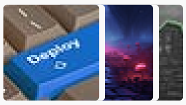
Ethan Siegel in Starts With A Bang!

Ask Ethan: Is the proton stable or unstable?

If we waited long enough, would even protons themselves decay? The far future stability of the Universe depends on it.

Jun 23, 2023 492 3

Lists



Predictive Modeling w/ Python

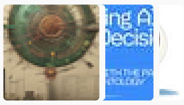
20 stories · 1503 saves

Practical Guides to Machine Learning

10 stories · 1832 saves

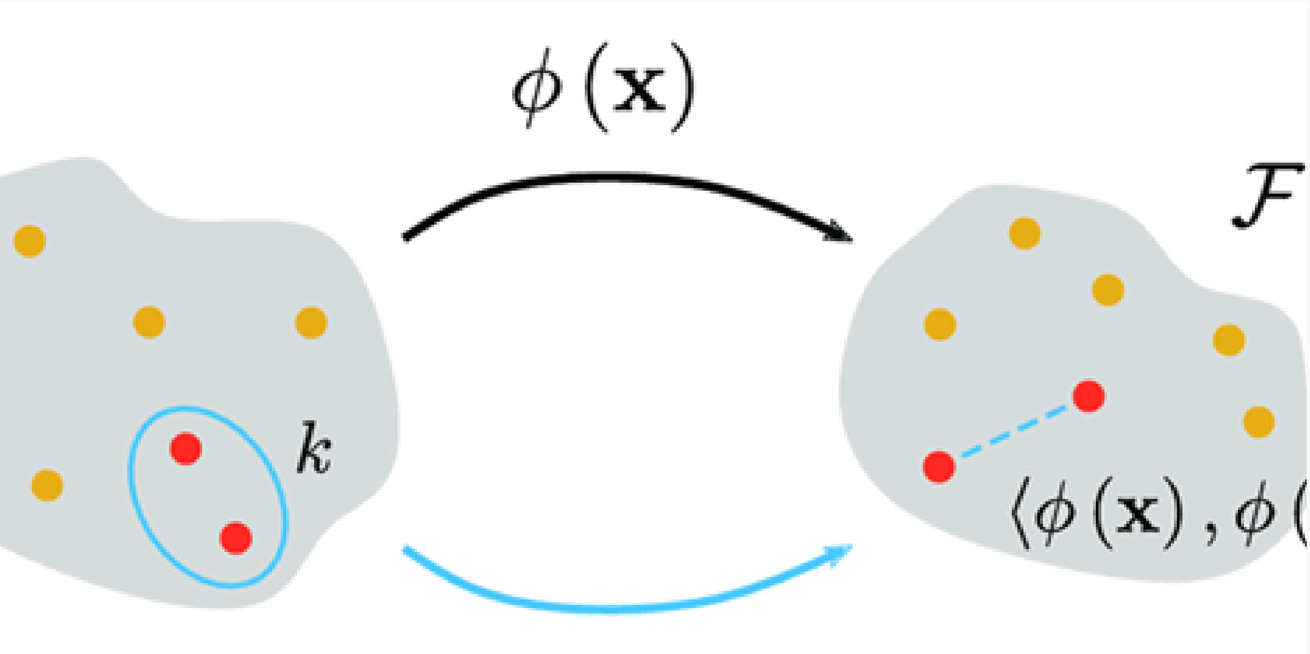
Natural Language Processing

1687 stories · 1258 saves



data science and AI

40 stories · 237 saves

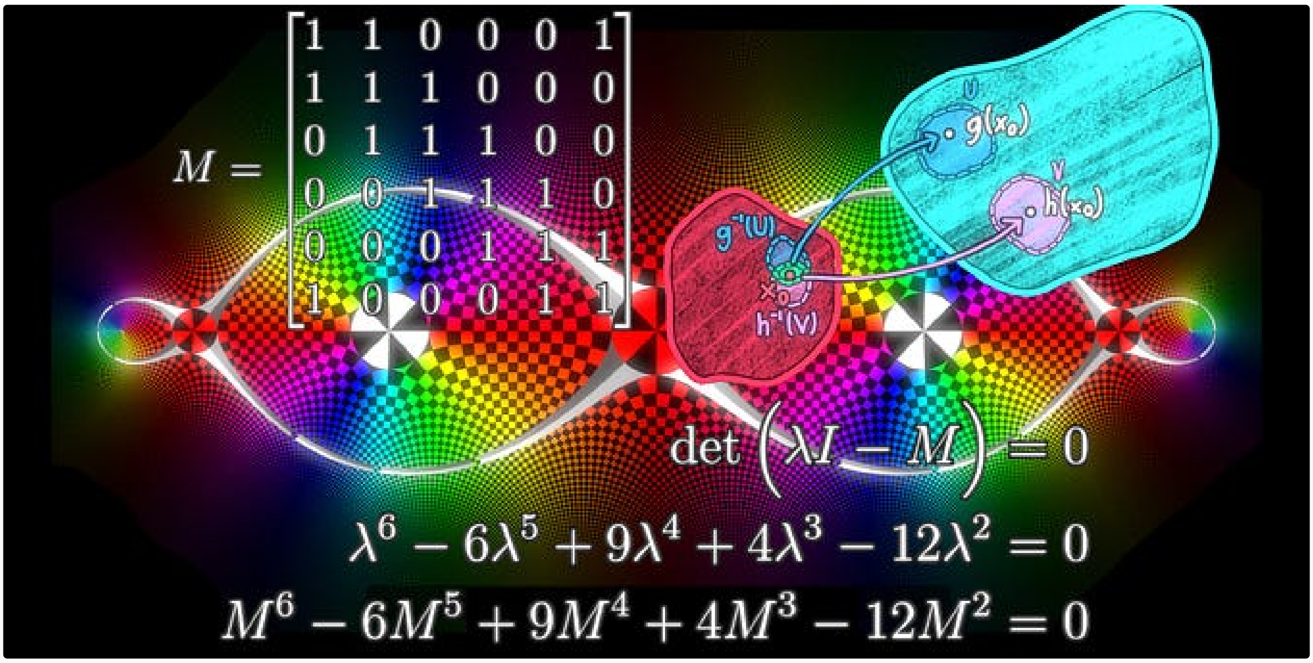


Abhishek Gajawada

UNDERSTANDING THE KERNEL TRICK

INTRODUCTION: -

May 3


$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$
U
$g(x_0)$
V
$h(x_0)$
$g^{-1}(U)$
x_0
$h^{-1}(V)$
$\det(\lambda I - M) = 0$
$\lambda^6 - 6\lambda^5 + 9\lambda^4 + 4\lambda^3 - 12\lambda^2 = 0$
$M^6 - 6M^5 + 9M^4 + 4M^3 - 12M^2 = 0$

Joseph Mellor

The Cayley-Hamilton Theorem, Pt. 1

Every square matrix satisfies its characteristic equation.

✦ Sep 1 👏 123 💬 1

Gregory Roitman

A wonderful coincidence or an expected connection: why π² ≈ g.

Let’s take a brief trip back to our school years and recall some lessons in mathematics and physics. Do you remember what the number π…

Aug 10 154 4

Grant Piper

Why Can’t Robots Click The “I’m Not a Robot” Box On Websites?

Clicking a tiny box tells Google all they need to know about your humanity

May 24 8K 175