# Traditional Machine Learning models.

## Dataset

Set of features or attributes → $f_1$ $f_2$ $f_3$ .. $f_n$

Set of instances →
$i_{11}$ $i_{12}$ $i_{13}$ $i_{1n}$
$i_{21}$ $i_{22}$ $i_{23}$ $i_{2n}$
⋮

for Supervised ML tasks we require a set of Labeled data
Given X predict Y.
Given X predict P(y)
ie P(Y/X)

### 1) Regression

Suppose you have a property at a location. L having size S, having rooms R, having amenities A ..... Based on these, determine the rent price. P.

### 2) Classification

For a tumor, given the size S of the tumor, location L of the tumor, comorbidities C of patient, age A, family history F, determine whether the tumor is malignant or benign

### 3) Clustering

The features are available but not the ~~labels~~ output or Labels. Used to learn useful properties of the structure of data

hypothesis $Y = f(X)$

## Linear Regression

$D = X =$

| (Locality) | (Size) | (Rooms) | (Amen) |
|---|---|---|---|
| $x_{11}$ | $x_{12}$ | $x_{13}$ .. | $x_{1k}$ |
| $x_{21}$ | $x_{22}$ | $x_{23}$ .. | $x_{2k}$ |
| ⋮ | | | |
| $x_{n1}$ | $x_{n2}$ | $x_{n3}$ | $x_{nk}$ |

Set of Labels (Price)

$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

Input $X \in \mathbb{R}^{n \times k}$
outputs $Y \in \mathbb{R}^n$

Objective: Given an input vector $\vec{x} \in \mathbb{R}^k$ predict a scalar value $y \in \mathbb{R}$

Assume that the output is a linear function of the input features (Hypothesis)

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \cdots \beta_k x_{ik}$$

Let $\hat{y}_i$ be the predicted value for the $i^{th}$ instance (ie for a given $\vec{x}$)

$\hat{y} = \theta^T \vec{x}$ where $\theta$ is the parameter vector $= [\beta_0, \beta_1 \cdots \beta_k]^T$

Q is also $\theta \in \mathbb{R}^{k+1}$ is the weight parameters

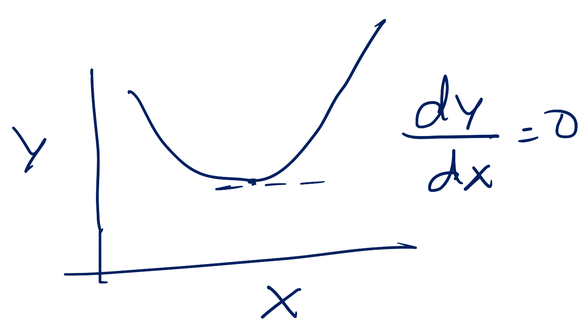
from the data set learn $\beta_0, \beta_1 \dots \beta_k$ such that the Error

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \text{ is minimized}$$

→ (Mean Square Error (MSE))

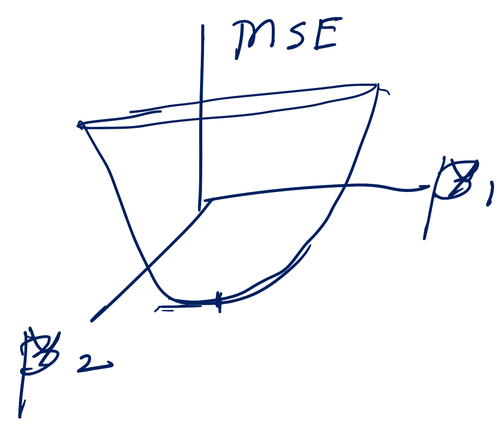
Divide the data into 2 parts, training set & test set
(usually 80-20 or 75-25)

$$\text{minimize}\left(MSE = \frac{1}{n}\|\hat{y} - y\|_2^2\right) \quad \cdots (1)$$




$$\nabla_\theta MSE_{train} = 0$$

$$\nabla MSE = \begin{bmatrix} \frac{\partial MSE}{\partial \beta_0} \\ \frac{\partial MSE}{\partial \beta_1} \\ \vdots \end{bmatrix} = 0 \quad \text{(for minima)}$$




$$\hat{y} = X\theta$$

Replace $\hat{y}$ by $X\theta$ in Eqn (1)

$$MSE = \frac{1}{n}\|X\theta - y\|_2^2$$

$\nabla_\theta MSE = 0$ (Derivation not done)

$$\theta = \frac{X^T y}{X^T X} = (X^T X)^{-1} X^T y$$

for python Scikit Learn provides all traditional ML package.

Eg. Approximating betweenness by Linear Regression.

Given a graph, ~~with degree of the nodes, extract~~ with the adjacency matrix, derive features of nodes like degree of the nodes, average degree of neighbor, Local clustering coeff and then use the same to predict the betweenness.

## Classification

$X \in \mathbb{R}^k$ $y \in \{0,1\}$ or $y \in \{C_1, C_2 .. C_k\}$

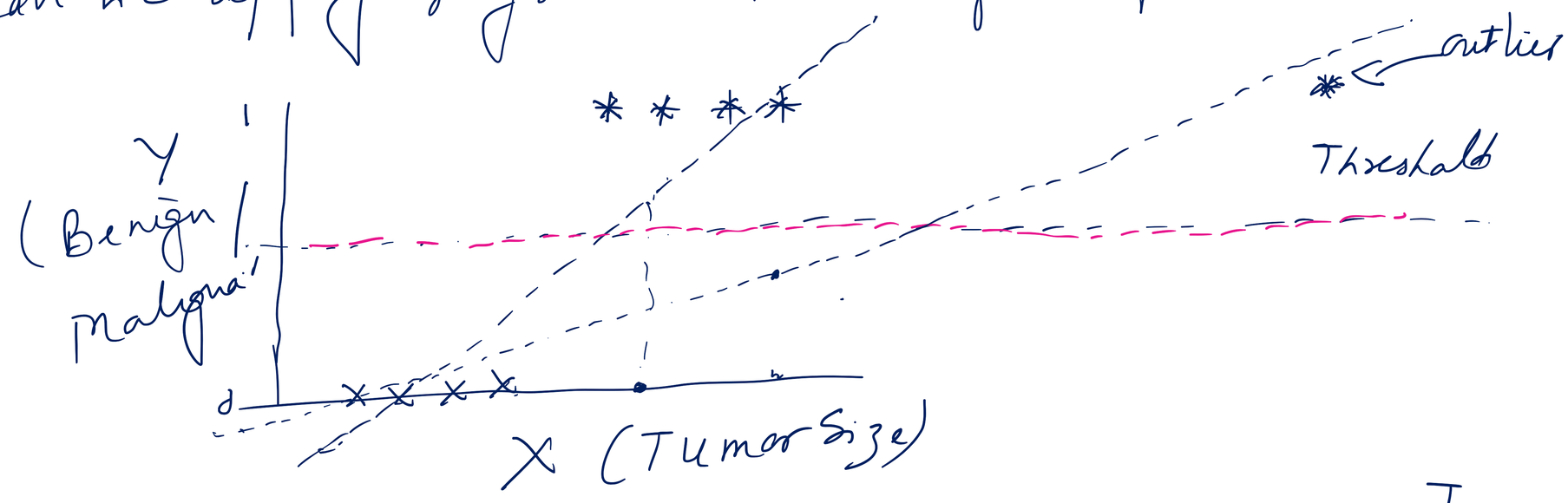
- Tumor :- Malignant / Benign
- Online Transaction :- Licit / Illicit
- Emails :- Spam / Non spam
- Graphs — Link exists / not Exists

when $y \in \{0,1\}$ it is called Binary classification

$y \in \{C_1, C_2, C_3 .. C_k\}$ it is called Multiclass classification.

for Regression we have $\hat{y} = f(x)$

Can we apply regression techniques for classification




If we use the same Linear Regression principle $h_\theta(x) = \theta^T x$

We can use a threshold (say values of $h_\theta(x) > 0.5$ then (+)ve else (−)ve class)

## Problems

- outliers can create errors and makes wrong learning
- Another problem is $h_\theta(x)$ may be much greater than 1 or lesser than 0.

- Fix — We want $0 \leq h_\theta(x) \leq 1$

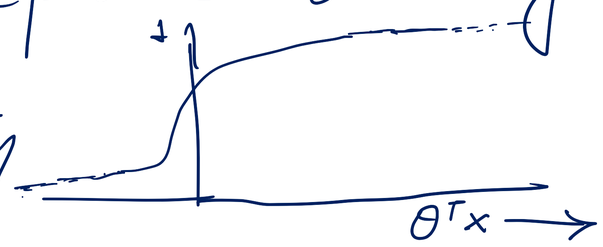
↳ Goal of Logistic Regression model.

Logistic Regression Model — for linear regression we have $h_\theta(x) = \theta^T x$

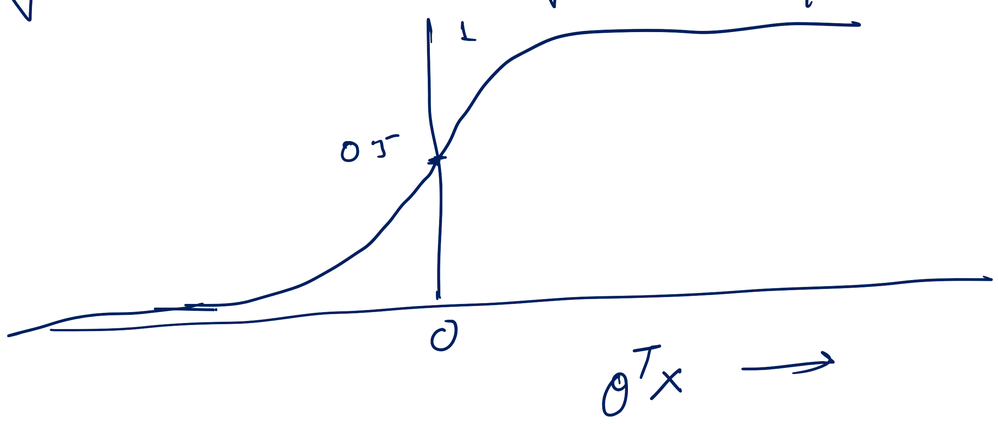
$\theta^T x$ can be much larger than 1 or much lesser than 0.

For (Log Reg) we replace $\theta^T x$ by another function $g(\theta^T x)$

$g(\theta^T x)$ will have a property



So lr $g(\theta^T x)$ be a sigmoid function given as $\frac{1}{1+e^{-\theta^T x}}$




Interpretation of the hypothesis output

$h_\theta(x)$ = Estimated prob that $y=1$ for input $x$.

So If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumor size} \end{bmatrix}$

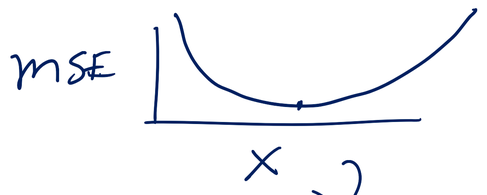
Then $h_\theta(x) = 0.7$ indicates that there 70% chance of the tumor being malignant.

So $h_\theta(x) = P(y=1 \mid x; \theta)$

↳ Prob that $y=1$ given input $x$, parameterized by $\theta$.

For linear regression to derive $\theta$ we minimized the Mean Square Error (MSE)

i.e $\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$ → when $\hat{y}_i = \theta^T x$



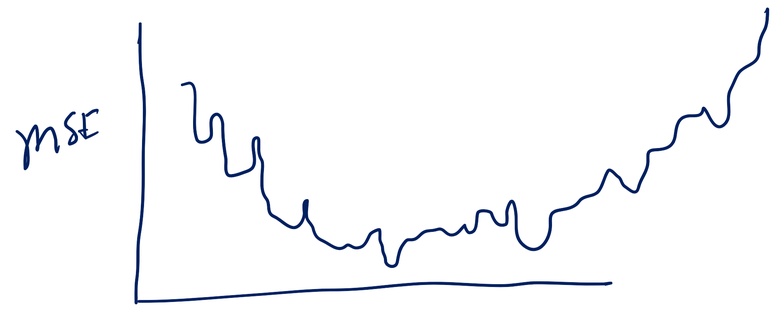

for training set $\{(x_1, y_1)(x_2, y_2) \ldots (x_m, y_m)\}$

However when $\hat{y}_i = h_\theta(x_i) = \frac{1}{1+e^{-\theta^T x_i}}$ is exponential

the MSE

is not Convex

i.e will have lot of Local minima




So instead the following loss function is used

Use Cross Entropy loss

~~Cost~~ $\text{Loss}(h_\theta(x), y) = -\text{Log}(h_\theta(x))$ when $y=1$

$= -\text{Log}(1-h_\theta(x))$ when $y=0$

Cross Entropy loss is given as

$\mathcal{L} = -y\,\text{Log}(h_\theta(x)) - (1-y)\,\text{Log}(1-h_\theta(x))$

for $(x_1, y_1)(x_2, y_2) \ldots$ $\mathcal{L}_{tot} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_i$

# Decision boundary of Logistic Regression

When is $\hat{y} = 1$, when $h_\theta(x) \geq 0.5$ or $\theta^T x > 0$

" " $y = 0$ " $h_\theta(x) \leq 0.5$ or $\theta^T x \leq 0$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_k \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_k \end{bmatrix}$$

$$\theta^T X = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots \theta_k x_k > 0$$

~~This is a line, so the~~

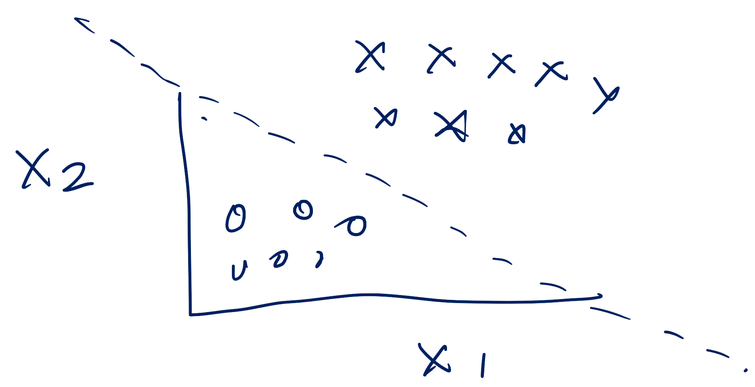
This is a hyperplane, so the decision boundary of logistic regression is a hyperplane

Eg. $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

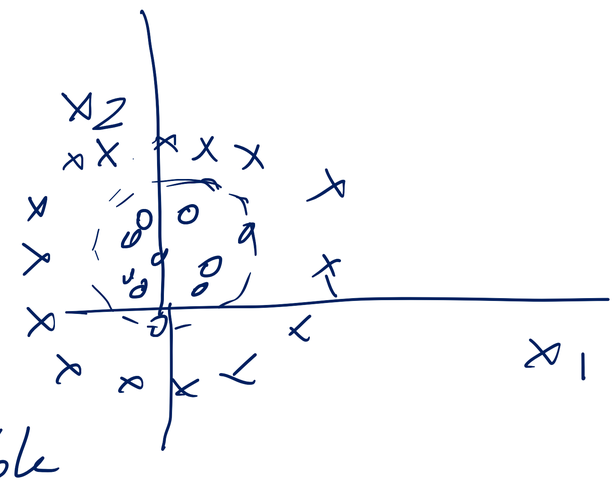
& $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

Then $h_\theta(x) \geq 0.5$ when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0$

& $x_1 + x_2 \geq 3$




Suppose data is




This is not Linearly separable

For a non-linear separation we have to use different function.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \cdots)}}$$

if we use $\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

$h_\theta(x) \geq 0.5$ when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \geq 0$

or $x_1^2 + x_2^2 \geq 1$

## Support Vector machines

In Logistic Regression we had: $h_\theta(x) = \hat{y} = \frac{1}{1+e^{-\theta^T x}}$

objective when y is true output $y=1$ then $\hat{y}=1$

i.e. $h_\theta(x) = 1$

i.e. $\theta^T x \gg 0$

when $y=0$ the $\hat{y}=0$

i.e $h_\theta(x) = 0$

$\theta^T x \ll 0$

Cost

Loss func $\mathcal{L} = -\left(y \log \frac{1}{1+e^{-\theta^T x}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$

Generally for any loss func we also add a regularization term func $f(\theta)$

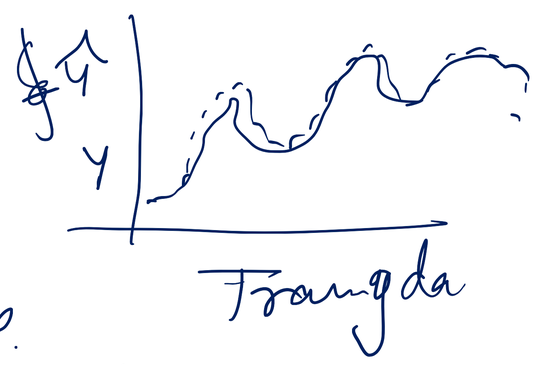
L2 regularization the $f(\theta) = \frac{1}{2}\sum_{i=1}^{k} \theta_i^2$ k is the no. of parameter

L1 regularization the $f(\theta) = \sum_{i=1}^{k} |\theta_i|$

Regularized Loss func $= \mathcal{L} + \frac{1}{2}\sum_{i=1}^{k} \theta_i^2$

Regularization prevents the m/c learning model from overfitting.

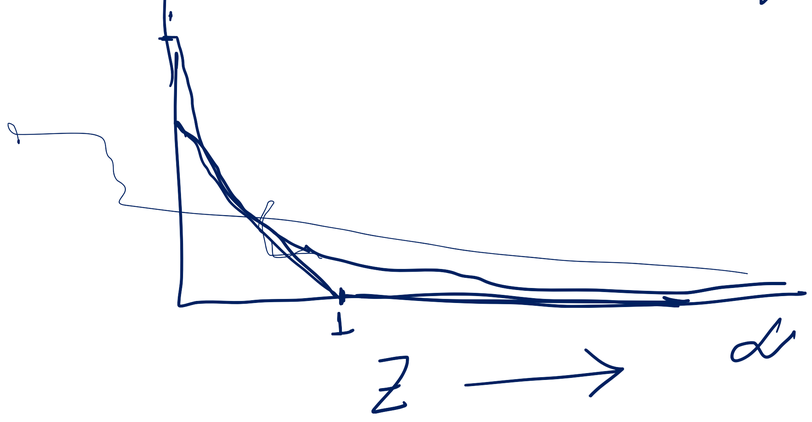
overfitting



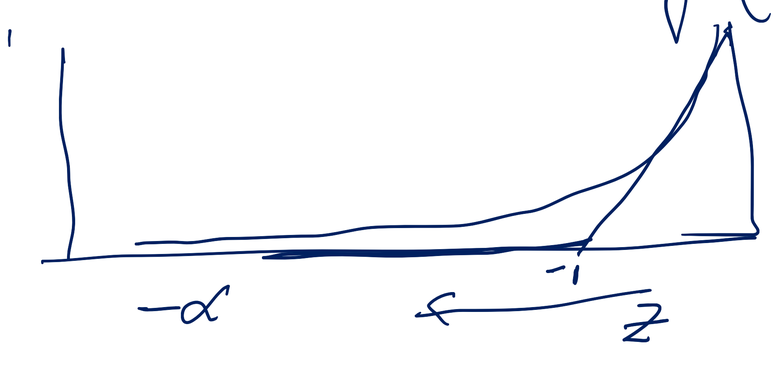


For Logistic Regression

if $y=1$ then in the loss func only $-\left(y \log \frac{1}{1+e^{-\theta^T x}}\right)$ is active




When $y=0$ the the loss func only $= -\log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$ is active



For SVM we modify the cost funcn such that

as $y=1$ we have $\text{Cost}_1(z) = 0$ as $z \geq 1$

as $y=0$ we shall have $\text{Cost}_0(z) = 0$ as $z \leq -1$

New Loss funcn is $\frac{1}{m} \sum_{i=1}^{m} \left[ \left[ y_i \, \text{Cost}_1(z_i) + (1-y_i) \text{Cost}_0(z_i) \right] + \frac{1}{2} \sum_{j=1}^{K} \theta_j^2 \right]$

We may also use a ~~weighted~~ loss funcn that weights the regularization and the main loss funcn

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} \underbrace{\left[ y_i \, \text{Cost}_1(z_i) + (1-y_i) \text{Cost}_0(z_i) \right]}_{A} + \frac{\lambda}{2m} \underbrace{\sum_{j=1}^{m} \theta_j^2}_{B} \quad -(4)$$

So the optimization funcn can be written as

$$\min_{\theta} A + \lambda B$$

or $\min_{\theta} CA + B$ where C is a constant that controls weight of A & B.

So what happens when C is kept very high

Then ~~minimal~~ minimization of the loss funcn tends to make $A \approx 0$ and hence chooses $\theta$ accordingly

So when $y_i = 1$ for $\text{Cost}_1(z_i)$ i.e $\text{Cost}_1(\theta^T x) = 0$

when ~~$\theta^T x \gg 1$~~ $\theta^T x \geq 1$

& $y_i = 0$ for $\text{Cost}_0(z)$ i.e $\text{Cost}_0(\theta^T x) = 0$

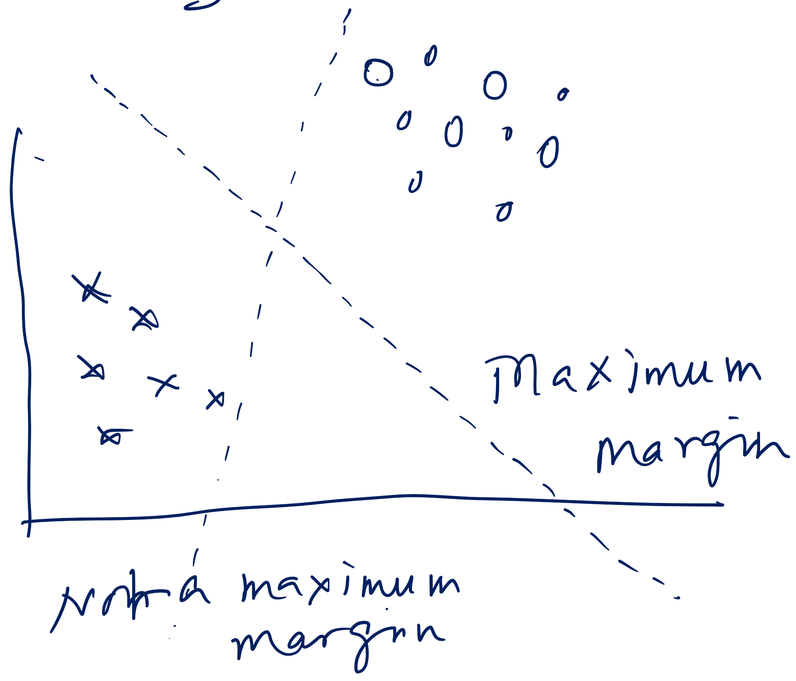
$\theta^T x \leq -1$

So the ~~minimal~~ loss funcn in Eqn 4 can be written as

$$\text{Minimize } \frac{1}{2} \sum_{j=1}^{m} \theta_j^2 \leftarrow \|\theta\|^2 \quad -(5)$$

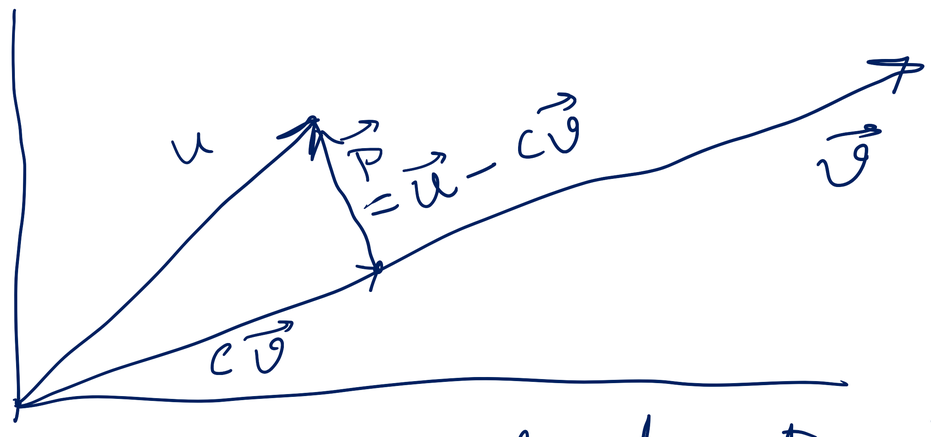
$$\text{s.t.} \quad \theta^T x \geq 1 \quad \text{when } y = 1$$

$$\& \quad \theta^T x \leq -1 \quad \text{when } y = 0$$



# The maths behind the large margin

For 2 vectors $\vec{u}$ & $\vec{v}$ $u^T v$ can be represented as $\vec{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_k \end{bmatrix}$




$c\vec{v} + \vec{p} = \vec{u}$

or $\vec{p} = (\vec{u} - c\vec{v})$

$\vec{p}$ is perpendicular to $c\vec{v}$

$V^T v$

So $\vec{p} \cdot c\vec{v} = 0$

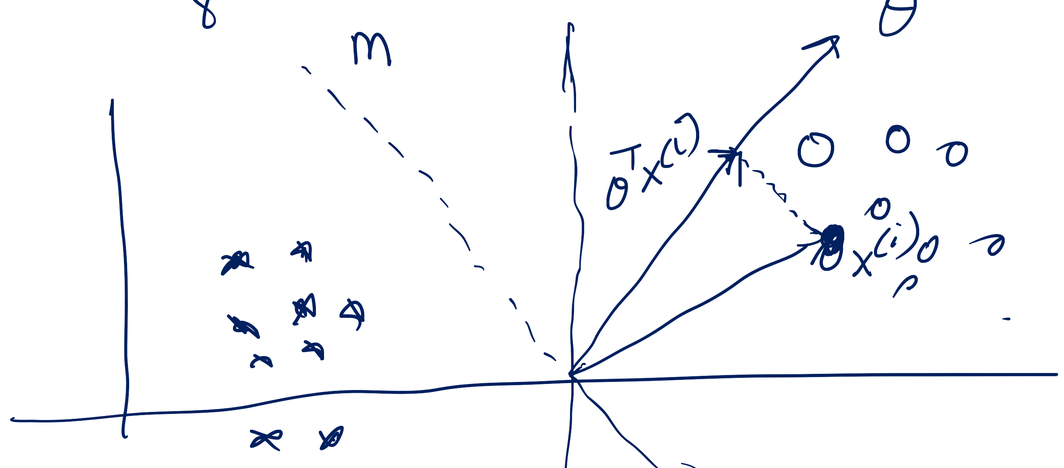
$(\vec{u} - c\vec{v}) \cdot c\vec{v} = 0$

or $c(\vec{u} \cdot \vec{v}) = c^2 \|\vec{v}\|^2$

or $\vec{u}^T \vec{v} = c\|\vec{v}\|^2$

So $\vec{u}^T \vec{v}$ given the magnitude or the length of the projection of $\vec{u}$ on vector $\vec{v}$

So




So $\theta^T x^{(i)}$ gives the length of the projection of $\vec{x}^{(i)}$ on $\vec{\theta}$

If we consider $\theta$ as normal to a margin m then if the margin is very close to $x^{(i)}$ then the projection length reduces and hence to have $c\|\theta^2\|^2 > 1$

$\|\theta\|^2$ must increase which is a conflict as we minimize Eqn 5.

$\theta^T x^{(i)} = c\|\theta\|^2$

* For non-linear decision boundary.

Predict $y = 1$ if

Say $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 \geq 0$

So hypothesis is now $h_\theta(x) = 1$ if $\theta_0 + \theta_1 x_1 + \cdots \geq 0$

$= 0$ otherwise.

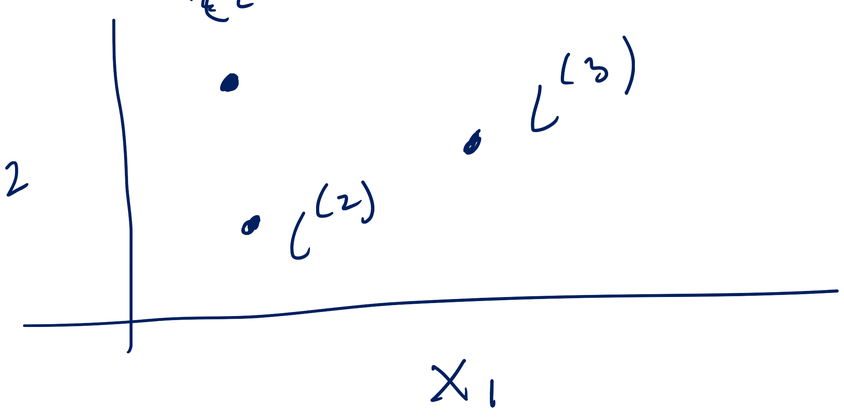
Other approach is create new features by combining feature ie

Let ~~$x_4 = x_4$~~ $f_1 = x_1$, $f_2 = x_2$ $f_3 = x_1 x_2$, $f_4 = x_1^2$

$f_5 = x_2^2$

and apply the previous concept.

So how to create new features




Choose few land mark points and find the distance of the points to the land marks.

Given $x$: $f_1 = \text{similarity}(x, l^{(1)})$

One similarity functio can be

$$\exp\left(-\frac{\|x - l^{(1)}\|_2^2}{2\sigma^2}\right)$$

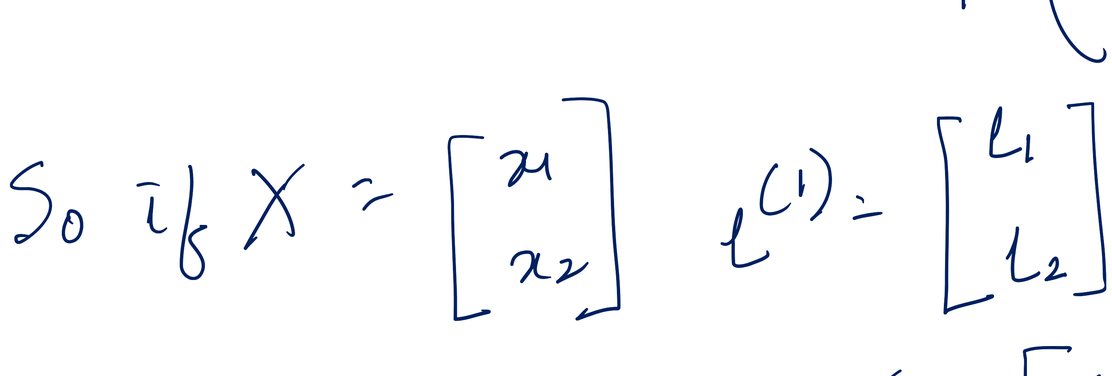
Similarly $f_2 = \text{Similarity}(x, l^{(2)})$

$$\exp\left(-\frac{\|x - l^{(2)}\|_2^2}{2\sigma^2}\right)$$

So if $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $l^{(1)} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$

Th $f_1 = \exp\left(-\frac{[(x_1 - l_1)^2 + (x_2 - l_2)^2]}{2\sigma^2}\right)$

What is exactly does?




If $\theta_0 = -0.5$

$\theta_1 = 1$

$\theta_2 = 1$

$\theta_3 = 0$

Then $h_\theta(f) = 1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 > 0$

or $f_1 + f_2 > 0$

What should be the land marks?

In practice, every point is considered as a land mark