Date: / /

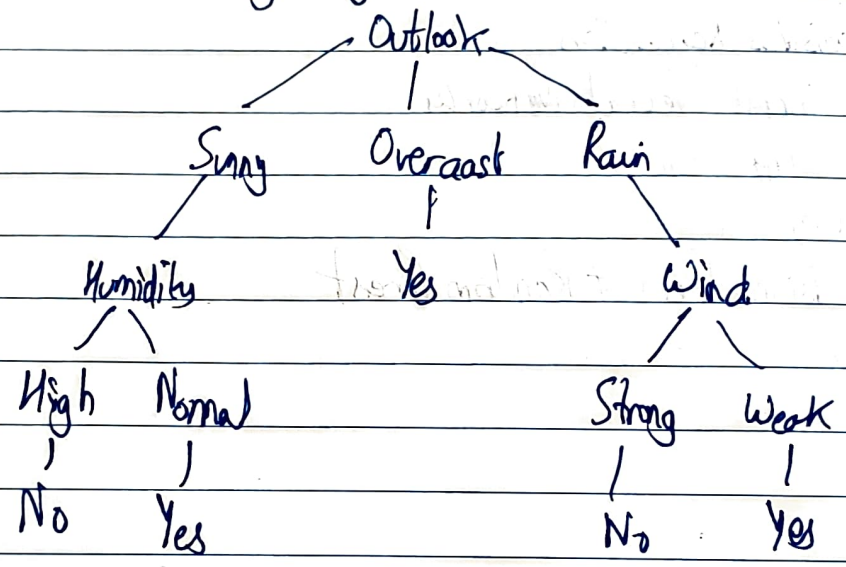
Introduction

☆ ML Algorithms
- Generative
- Discriminant
- Posterior Probability Approach

☆ Supervised Learning

(i) Decision Tree Learning

Task: Learn to predict the value of EnjoySport/Crowded for an arbitrary day



(ii) Fisher's Linear discriminant

(iii) Perceptron Algorithm

(iv) Support Vector Machine - Line with max width to split data.

(v) Artificial Neural Networks
- Networks of processing units (neurons) with connections (synapses) between them
- Large number of neurons and large connectivity
- Parallel Processing and distributed computation/memory

☆ Unsupervised
- Clustering

☆ Probabilistic
- P(c_i/

Probabil
x hai
covid
- We ne

☆ Paramet
- We w
method

Date: / / /

☆ Unsupervised learning
- Clustering Algorithms

☆ Probabilistic ML Approach
- $P(c_i/x) = \frac{P(x/c_i)P(c_i)}{P(x)}$

→ Covid Distribution

$P(x) \rightarrow P(x/c_1)P(c_1) + P(x/c_2)P(c_2)$

$P(c_i/x)$ → Probability of x having covid

$P(x/c_i)$ → Probability a person getting in a covid distribution (Popln. distri.)

- We need immense amount of data

☆ Parametric Methods
- We will learn parameters during training while in non parametric methods we don't have fixed parametric methods.

Date: / / /

Supervised Learning

$(x_1, x_2 \ldots x_{10000})$ — No. of word occurences

$w_1, w_2 \ldots w_{10000}$ — Weight

Threshold

Now $w_1 x_1 + w_2 x_2 \ldots \quad > t$ +ve Points

$\quad \leq t$ -ve Points

$$\hat{w} = \begin{bmatrix} -t \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} \qquad \hat{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}$$

$\hat{w} \cdot \hat{x} > 0$ +ve

$\hat{w} \cdot \hat{x} < 0$ -ve

In perceptron we by default consider bias

Also, in perceptron $k \leq (R/\gamma)^2$

↳ No. of mistakes

$$k(\hat{w}^T \cdot \hat{x}) > 0 \qquad k = 1 \text{ +ve}, \quad k = -1 \text{ -ve}$$

$$-t + w_1 x_1 + w_2 x_2 \ldots w_D x_D > 0 \quad \text{+ve}$$

D+1 Parameters

We need to find w :) One million points, One million eqns

(1) Perceptron Algorithm

$w \leftarrow 0$ while there exists x_i with $x_i l_i \cdot w^T \leq 0$ update

$w \leftarrow w + x_i l_i$ → +ve point means 1 or 0 for -ve

↳ Not Obeying rule — Point

Here x_i are input data and l_i input output

In perceptron we have 1s or -1s

urence

ot d

e Points

Points

$\cdot \vec{x} > 0$ +ve

$\cdot \vec{x} < 0$ −ve

1 +ve

1 −ve

ve

ts y

update

or −ve

t

Date: / / /

If there is a w^* satisfying $(w^* \cdot x_i) y_i \geq 1$ for all i, then the Perceptron Algorithm finds a soln w with $(w \cdot x_i) y_i > 0$ for all i in at most $r^2 / |w^*|^2$ updates where $r = \max |x_i|$

Assume that $(x_1, y_1) \ldots (x_m, y_m)$ is separable, let $B = \min(\|w\| : \forall i \in [m], y_i (w \cdot x_i) \geq 1)$ and let $R = \max_i \|x_i\|$. Then the perceptron stops after at most $(RB)^2$ iterations and when it stops it holds that $\forall i \in M$

$\bar{w} = \frac{w}{\gamma}$ $\quad \gamma = \min\{ y_i (w \cdot x_i) \mid i = 1, 2, \ldots m \}$ $\quad y_i (w \cdot x_i) \geq 0$

$$y_i (\bar{w} \cdot x_i) = y_i \left(\frac{w}{\gamma} \cdot x_i\right) = \frac{1}{\gamma} (y_i (w \cdot x_i)) \geq \frac{1}{\gamma} \gamma = 1$$

Also, in perceptron $k \leq (R/\gamma)^2$

↳ No. of mistakes

$B = \min\{ \|w\| \mid y_i (w_i x_i) > 1 \}$

$R = \max\{ \|x_i\| \mid i = 1, 2, 3 \ldots \}$

- We use $\bar{w}$ for getting $y_i (w \cdot x_i) \geq 1$

$T+1$th iteration

$$\cos\theta = \frac{w^{T+1} w^*}{\|w^{T+1}\| \|w^*\|} = \frac{w^{T+1} w^*}{\|w^{T+1}\| B} \geq \frac{T}{\|w^{T+1}\| B} \geq \frac{T}{\sqrt{T} RB}$$

$$\frac{T}{\sqrt{T} RB} \leq \cos\theta \leq 1$$

$R \to$ max radius of pt.

$T \to$ No. of iterations

$$\sqrt{T} \leq RB$$

$$\boxed{T \leq R^2 B^2}$$

- The problem is you'll never get B
- If the points are not linearly separable, we will never come out of the loop

Date: / / /

(a) $w^{T+1}w^* - w^T w^* = (w^{T+1} - w^T) \cdot w^*$

$= (w^T + y_i(x_i) - w^T) w^*$

$= y_i x_i w^* \geq 1$

$w^{T+1} w^* - w^T w^* \geq 1$

$w^T w^* - w^{T-1} w^* \geq 1$

$w^{T+1} w^* - w w^* \geq T$ Since $w^1 = 0$

$\boxed{w^{T+1} w^* \geq T}$

(b) $\|w^{T+1}\|^2 = \|w^T + y_i x_i\|^2 = \|w^T\|^2 + 2(w^T x_i) y_i + y^2 \|x_i\|^2$

$\leq \|w^T\|^2 + \|y_i\|^2 \|x_i\|^2$

$\leq \|w^T\|^2 + R^2$

$\leq \|w^{T-1}\|^2 + 2R^2$

$\leq TR^2$

$\therefore \|w^{T+1}\| \leq \sqrt{T} R$

Can ask proof.

Kernel Trick

We can use kernel w/o having to deal with a lot of dot products

Positive on top +ve

Negative on -ve

$w \leftarrow 0$

Line

To

$K(x$

As

We

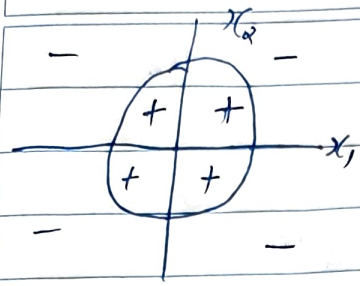
then

Date: / / /

Date / / /

Kernel Trick

We can use kernel w/o having to deal with a lot of dot products



$x_1, x_2 \rightarrow x_1, x_2, x_1^2 + x_2^2$

Here, it gives us a chance to separate them in higher dimensions

$\phi(x_1, x_2) = (\phi_1(x_1, x_2), \phi_2(x_1, x_2), \phi_3(x_1, x_2))$

$\phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x))$

$w \leftarrow 0 \leftarrow w + y_i x_i$

+ve points

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5^2 + 0.5^2 \end{bmatrix} \end{bmatrix}$$

$\hookrightarrow$ Linear Combi of ϕ_1, ϕ_2, ϕ_3

$k(x^1, x^2) = \phi(x^1) \cdot \phi(x^2)$

Positive on top +ve

$w^1 \leftarrow 0$

$w^2 \leftarrow y_1 \phi(x_1)$

$w^3 \leftarrow y_1 \phi(x_1) + y_2 (\phi(x_2))$

Negative on -ve

To figure out the side of the point $\rightarrow$ use dot product

$K(x, y) = \phi(x) \cdot \phi(y)$

$k(x, y) = (1 + x_1 y_1 + x_2 y_2)^2$ $\rightarrow$ x and y are two points

$\rightarrow$ Kernel function

$\varphi(x) = 1, \sqrt{2} x_1, \sqrt{2} x_2, \sqrt{2} x_1 x_2, x_1^2, x_2^2$

As long as $k(x, y)$ is given, there is no need for $\phi(x)$

We define k and check if k satisfies these training data then we use the same eqn.

Q: We have kernel, not transformation matrix to transform into higher dimensions. We are given testing sample, can we tell +ve or -ve.

Date: / /

$w \leftarrow (\ )\phi(x^1) + (\ )\phi(x^2) + \cdots (\ )\phi(x^{1000})$

1000 testing data

x^{test} we have. Now $w \cdot \phi(x^{test}) > 0$ +ve

$w \cdot \phi(x^{test}) < 0$ -ve

This dot product will give

$c_1\phi(x^{test})\phi(x^1) + c_2\phi(x^{test})\phi(x^2) \cdots$

$= c_1 k(x^1, x^{test}) + c_2 k(x^2, x^{test}) \cdots c_{1000} k(x^{1000}, x^{test})$

ϕ n dim vector

- Basically ϕ is a function which transfers a point to higher dimensions and using ϕ we get the plane w.
- Now, say in a question, we are given kernel but not ϕ. Can we find the point is +ve or -ve
- Kernel is a defined thing which is given in terms of initial two parameters only which acts as dot product i.e. used to tell if a point is +ve or -ve.
- DL tries to learn your transformation matrix.
- Now, we can use Kernel and solve.

Kernel is only dependent on initial given conditions and points

Ex: $k(x, x$

$(\phi($

$\phi(x) =$

2

(x^{100})

the
-ve

.. Good $k(x^{100}, x^{100})$

a point to
the plane w.
and) but
r -ve
in terms of
dot product
e.
rix.

Date: / / /

Ex: $k(x,x') = e^{-\frac{\|x-x'\|^2}{2}}$

$$(\phi(x))_n = \frac{1}{\sqrt{n!}} e^{-x^2/2} x^n$$

$$\phi(x) = \phi(x)_0, \phi(x)_1 \ldots$$

$$\hookrightarrow \phi(x)\phi(x') = \sum \left(\frac{1}{\sqrt{n!}} e^{-x^2/2} x^n\right)\left(\frac{1}{\sqrt{n!}} e^{-x'^2/2} (x')^n\right)$$

$$= \sum \frac{1}{n!} e^{-1/2(x^2+x'^2)} (xx')^n$$

$$= e^{-1/2(x^2+x'^2)} \sum \frac{1}{n!} (xx')^n$$

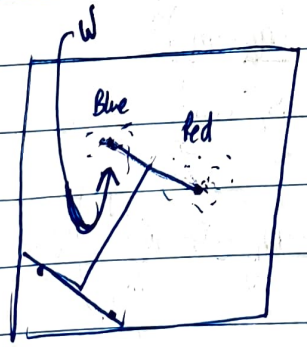
$$= e^{-1/2(x^2+x'^2)} e^{xx'} = e^{-1/2(x^2+x'^2-2xx')}$$

$$= e^{-1/2(x-x')^2}$$

1dimension $\longrightarrow$ n dimensions

due to the kernel you won't need to deal with the space

② Fischer Discriminant



$\bar{x}_{pos} = \frac{1}{N} \sum x_n$

$\bar{x}_{neg} = \frac{1}{N} \sum x_n$

Projected points to mean

$\frac{1}{N} \sum w^T x_n$

$= w^T \sum \frac{1}{N} x_n$

$= w^T \bar{x}_{pos}$

- We choose \ over — and / because \ covers most distance.
- Variance of pos and neg must be as similar as possible.

( ) . → Vector

Date: / /

$\begin{bmatrix} a \\ b \end{bmatrix}_{2\times 1}$ → Maximize

$\|w\| = 1$

We have to maximize the distance b/w the means and try to find the correct line

$$\boxed{w^T \bar{x}_{cpos} - w^T \bar{x}_{neg}}$$

↳ Projected Point

$$\bar{m}_1 = \frac{1}{N_1} \sum x_n \qquad \bar{m}_2 = \frac{1}{N_2} \sum x_n$$

$$m_1 = \frac{1}{N_1} \sum w^T x_n = w^T \left[\frac{1}{N_1} \sum x_n \right] = w^T \bar{m}_1$$

$$m_2 = \frac{1}{N_2} \sum w^T x_n = w^T \left[\frac{1}{N_2} \sum x_n \right] = w^T \bar{m}_2$$

We also minimize in class variance

↓

Horizontal Distance

$$s_1^2 = \frac{1}{N_1} \sum (w^T x_n - m_1)^2 = \frac{1}{N_1} \sum (w^T x_n - w^T \bar{m}_1)^2$$

$$= \frac{1}{N_1} \sum (w^T (x_n - \bar{m}_1))^2$$

$$= \frac{1}{N_1} \sum \left[w^T (x_n - m_1) (w^T (x_n - m_1))^T \right]$$

$$= \frac{1}{N_1} \sum \left[w^T (x_n - \bar{m}_1)(x_n - \bar{m}_1)^T w \right]$$

$x_{n_0} = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}$ → Vector Component

↓ Point

We need to minimize $s_1^2 + s_2^2$ and maximize $(m_2 - m_1)^2$

↳ maximize $\dfrac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$

$$= w^T \left(\frac{1}{N_1} \sum \underbrace{(x_n - \bar{m}_1)}_{2\times 1} \underbrace{(x_n - \bar{m}_1)^T}_{1\times 2} \right) w$$

$$s_2^2 = w^T \left(\frac{1}{N_2} \sum (x_n - \bar{m}_2)(x_n - \bar{m}_2)^T \right) w$$

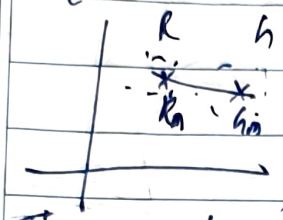
Some Matrix

$AA^T = A^2$

$7^2 = 77^T$

$s_1^2 + s_2^2 = w^T S_w w$ Here $S_w = \sum_{C_1} (x_n - \bar{m}_1)(x_n - \bar{m}_1)^T + \sum_{C_2} (x_n - \bar{m}_2)(x_n - \bar{m}_2)^T$

Now, the point, we need to find the line which seperates them properly, so we observe that if we start taking lines and then we project points on to the line



we take projections onto line and find mean point

Then we try to maximize the distance between these lines and also try to minimize $s_1^2 + s_2^2$ Cause this is variance

$$(m_2 - m_1)^2 = (w^T\bar{m}_2 - w^T\bar{m}_1)^2 = (w^T(\bar{m}_2 - \bar{m}_1))^2$$
$$= w^T(\bar{m}_2 - \bar{m}_1)(w^T(\bar{m}_2 - \bar{m}_1))^T$$
$$= w^T(\bar{m}_2 - \bar{m}_1)(\bar{m}_2 - \bar{m}_1)^T w$$
$$= w^T S_B w$$

If $S_w = 4\times4$ from $x_n - \bar{m}_1$ as 4×1 and $(x_n - \bar{m}_1)^T$ as 1×4, we get $\bar{m}_2 - \bar{m}_1$ as 4×1

$S_w^{-1} = 4\times4$

$w = 4\times1$

$(m_2 - m_1)^2$

maximize $\dfrac{(m_2 - m_1)^2}{s_2^2 + s_1^2} = \dfrac{w^T S_B w}{w^T S_w w}$ → Scalar

Differentiate wrt w

$$2S_B w(w^T S_w w) - 2(S_w w)(w^T S_B w) = 0$$
$$S_B w(w^T S_w w) = S_w w(w^T S_B w)$$
$$S_w w \propto S_B w$$
$$\propto \bar{m}_2 - \bar{m}_1$$
$$\boxed{w \propto S_w^{-1}(\bar{m}_2 - \bar{m}_1)}$$

$S_B w = (\bar{m}_2 - \bar{m}_1)(\bar{m}_2 - \bar{m}_1)^T w$

$\propto (\bar{m}_2 - \bar{m}_1)$

Threshold → Approx mid point

$w^T\bar{m}_1 + w^T\bar{m}_2$

2

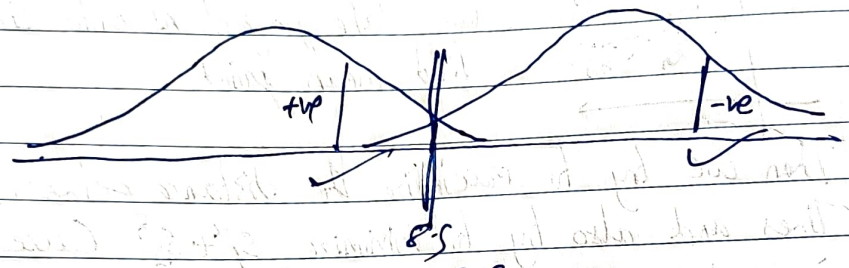
Date: / / /

$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

Note: $\frac{\delta}{\delta w}(w^T S_B w) = 2 S_B w$

S_B → Some Matrix

We generate distribution, check probabilities and find the probability at the point → Probabilistic Approach



$w^T x_{test} < 8.5$ +ve

> 8.5 −ve

So we get $w^T x_{test}$ as the discriminant function

Here, $P(C_1/w^T x_1) = P(w^T x_1 | C_1)\, P(C_1)$

$P(C_2/w^T x_1) = P(w^T x_1 | C_2)\, P(C_2)$

$\therefore P(C_1/w^T x_1) > P(C_2/w^T x_1) \Rightarrow P(w^T x_1 | C_1) P(C_1) > P(w^T x_1 | C_2) P(C_2)$

Take Normal Distribution

$$\frac{1}{\sqrt{2\pi}\,\Sigma^{1/2}} e^{-\frac{1}{2}(x-\mu)^2}$$

For multiclass, we cannot directly use the same method

$S_T = S_W + S_B$

$S_B \rightarrow \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$

$S_W \rightarrow \sum_{1}^{K} S_k = \sum_{n \in C_k} (x - m_k)(x - m_k)^T$

Naive Bayes

We tell a bunch of
16, then we ask the
each number we
probability distribution
Similarly for say
few numbers

Let's say we
initia
even
odd
pow
m

heven − even
$h_2$power − power
heven

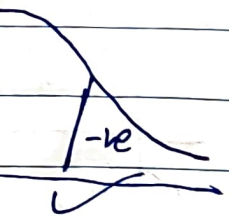
$P(D/h_2$ pow

Smaller Hypo
heven is the

Date: / / /

ilities and

bilistic Approach



ant function

$P(\omega|x_a)|(\omega)P(\omega)$

ibution

e thod

T

T

Naive Bayes

Date: / / /

• Baye's Theorem is to Probability, pythagorean theorem is to algebra

We tell a bunch of people that we have a number, say 16, then we ask the bunch to guess the next number. For each number we make probability and write the probability distribution curve

Similarly for say a series 16 8 2 64, we can predict the next few numbers

Let's say we take 2 8 16 64

	initially	after example
even	□	▭
odd	□	▭
powers of 2	□	▭
mul of 3	□	!
⋮	⋮	⋮

h_{even} – even numbers are generated

h_{power} – powers of 2 are generated

h_{even} → More suspicious → A lot of excessive test cases

$P(D/h_{power}) = \left(\frac{1}{6}\right)^4$ $\quad P(D/h_{even}) = \left(\frac{1}{50}\right)^4$ Likelihood

Smaller Hypothesis is better

h_{even} is the $\boxed{\text{Suspicious Coincidence}}$ and not a valid data.

Date: / /

If there are 2 h(s), if one of them is making all the training data positive, there is more chance of this hypothesis being true.

If we introduce another h → $h_{2^{power}-\{32\}}$ → Smaller set but intuition says that

$h_{2^{power}-\{32\}}$ → Seems more impractical

→ Removed based Intuition

· Subjectivity creeps in

$P\left(\frac{\text{Dataset}}{\text{hypothesis}}\right)$

2 power → 2 4 8 16 32 64

$P\left(\frac{2,8,16,64}{h_{2^{power}}}\right) = \left(\frac{1}{6}\right)^4$

$$P\left(\frac{h}{D}\right) = \frac{\overset{\text{Likelihood}}{P\left(\frac{D}{h}\right)}\ \overset{\text{Prior Probability}}{P(h)}}{\underset{\text{Marginal likelihood}}{P(D)}} = \frac{P\left(\frac{D}{h}\right)P(h)}{\sum P\left(\frac{D}{h'}\right)P(h')}$$

Posterior Probability

$P(h)$ Probability of the Hypothesis

$P(D/h)$ Probability of dataset coming for a particular hypothesis

$P(h/D)$ Probability of a hypothesis given dataset

We are learning from data → Likelihood

$P(\hat{x} \in \text{Concept} \mid \hat{x}, D^{t}$ → Te

$=$

$=$

Weighted avg of posi

average

Naive Bayes (

If we have a data

are Outlook, Temper

Ans $= P\left(\frac{P}{S}\right.$

$P(X \cap Y) = P(X)\,P$ → $P(X,Y)$

$= P(X)\,P($

$P(X,Y,Z) = P$

$= P($

$= P($

if $P((X,Y)|Z)$

Testing Example This means $\hat{x} \in$ Concept.

Date: / /

$$P(\hat{x} \in \text{Concept} \mid \hat{x}, D^t) = P(\hat{y}=1 \mid \hat{x}, D) = \sum P(\hat{y}=1, h \mid \hat{x}, D)$$

$$= \sum P(\hat{y}=1 \mid \hat{x}, D, h)\, P(h/D)$$

$$= \sum P(\hat{y} \mid \hat{x}, h)\, P(h/D)$$

Posterior Probability

Weighted avg of posterior probability is called Bayes' Weighted average

Naive Bayes' Classifier

If we have a data for PlayTennis = Yes/No in cases where parameters are Outlook, Temperature, Humidity, Wind

$$\text{Ans} = P\left(\frac{PT = Y}{S/H/N/W}\right)$$

$$P(X \cap Y) = P(X)\,P(Y) \quad \text{if } X \text{ and } Y \text{ are independent}$$

$P(X,Y)$

$$= P(X)\,P\left(\frac{Y}{X}\right)$$

$$P(X,Y,Z) = P(X)\,P(Y,Z \mid X)$$

$$= P(X)\,P(Y \mid X)\,P(Z \mid Y, X)$$

$$= P(X)\,P(Y)\,P(Z)$$

$$P(Y,Z \mid X) = P(Y \mid X)\,P\left(\frac{Z}{X} \mid X\right)$$

$$= P(Y \mid X)\,P(Z \mid X, Y)$$

if $P((X,Y) \mid Z) = P(X \mid Z)\,P(Y \mid Z)$ if X and Y are independent in Z

Date: / / /

Coming back to tennis' example

$$P\left(\frac{PT = Yes}{S, C, M, W}\right) = P\left(\frac{PT = Y}{h_1 h_2 h_3 h_4}\right)$$

features

$$= \frac{P(h_1 h_2 h_3 h_4 \mid PT = Y)\, P(PT = Y)}{P(h_1 h_2 h_3 h_4)}$$

Difficult to find out

In the world of $PT = Yes$ what is the probability of $h_1 = Cool$ ----

$$P\left(\frac{A}{B}\right) = \frac{P(B/A)P(A)}{P(B)}$$

We check Conditional Independence
- Domain Expertise
- We take testing data and check if they are conditionally independent

This ~~mean~~ has a problem i.e. if data of $P(h_1 h_2 h_3 h_4)$ is not in the given dataset, then $P = 0$.

So, we can do $P(h_1 h_2 h_3 h_4 \mid PT = Y)$

$$= P(h_1 \mid PT = Y)\, P(h_2 \mid PT = Y)\, P(h_3 \mid PT = Y)\, P(h_4 \mid PT = Y)$$

only if they are conditionally independent

i.e. In the world of $PT = Y$ h_1, h_2, h_3, h_4 are independent

Similarly you can find

$$P(h_1 h_2 h_3 h_4) = \sum P(h_1 h_2 h_3 h_4 \mid PT = Y) \cdot P(PT = Y)$$

Use this to solve the

↳ Here we can't write $P(h_1)P(h_2)P(h_3)P(h_4)$ because they are independent CONDITIONALLY

Basic Assumption: Naive Bayes Classifier is the Conditionally Independent.

$$P\left(\frac{PT = Y}{A_1 = S, A_2 = C, A_3 = N, A_4}\right)$$

$$= \frac{}{\sum P(A_1 = S,}$$

Therefore

$V_{NB} = \text{argma}$

Sometimes we can
each other, we

Bayesian Belie

Lighting

$P(S, B \mid G, L, C,$

Date: / / /

$PT=y)$

probability of

$h_1h_2h_3h_4)$ is not in

$h_3 | PT=y)\ P(h_4 | PT=y)$

independent

e independent

$/PT=y)$

$P(h_2)\ P(h_3)\ P(h_4)$

CONDITIONALLY

e Conditionally

→ Hi

Date: P/R/O

$$P\left(\frac{PT=y}{A_1=S, A_2=C, A_3=N, A_4=W}\right) = \frac{P(A_1=S, A_2=C, A_3=N, A_4=W \mid PT=y)\, P(PT=y)}{P(A_1=S, A_2=C, A_3=N, A_4=W)}$$

$$= \frac{}{\sum P(A_1=S, A_2=C, A_3=N, A_4=W, PT=y) + P(\ldots\ldots PT=N)}$$

Therefore

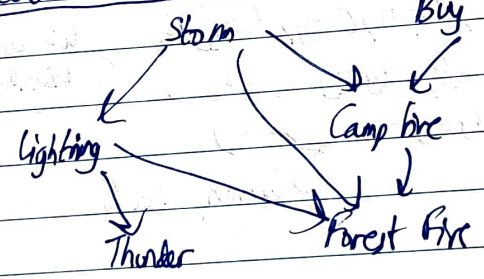
$$V_{NB} = \text{argmax}\ P(v_j)\ \prod_i P(a_i | v_j)$$

v_j = yes or no

Sometimes we cannot assume all variables are independent of each other, we use networks

Bayesian Belief Network

→ In cases where all of them are not independent



Wherever we have Conditionally independent we provide a graph. Here Lightning & CF are independent given Storm

$$P(S, BTG, L, C, T, FF) = P(S)\ P(BTG, L, C, T, FF | S)$$
$$= P(S)\ P(BTG | S)\ P(L, C, T, FF | S, BTG)$$
$$= P(S)\ P(BTG)\ P(L, C, T, FF | S, BTG)$$
$$= P(S)\ P(BTG)\ P(L | S, BTG)\ P(C, T, FF | S, BTG, L)$$
$$\vdots$$

$$= P(S)\ P(BTG)\ P(L|S)\ P(C|S, BTG)\ P(T|L)\ P(FF|S, L, C)$$

$P(X,Y|Z) = P(X|Y,Z)\,P(Y,Z)$

ignore Z and write

Date: / /

How do I generate this graph? We have algorithms

- $P(FF=y \mid S=y, BTG=N, L=y, C=N, T=y)$

$$P\left(\frac{A}{B}\right) = \frac{P(A)P(B|A)}{P(B)} = \frac{P(FF=y, S=y, BTG=N, L=y, C=N, T=y)}{P(S=y, BTG=N, L=y, C=N, T=y)}$$

$P(A,B)$

$P(A \cap B)$

- $P(FF=y \mid Btg=y, T=N) = \dfrac{P(FF=y, Btg=y, T=N)}{P(Btg=y, T=N)}$

$$= \frac{\sum_{S}\sum_{L}\sum_{C} P(S,L,C,FF=y,BTG=y,T=N)}{\sum_{S}\sum_{L}\sum_{C}\sum_{FF} P(BTG=y,T=N)}$$

Can give numbers and ask to solve

We are basically checking all probabilities and finding the answer

- $P(FF=y \mid Btg=y, T=N, C=y, L=y, T=$ → We use Joint Distribution

$$= \frac{P(FF=y, Btg=y \ldots)}{P(Btg=y \ldots)}$$

known directly

$$P\left(\frac{A}{B}\right) = \frac{P(A)P(B|A)}{P(B)} = \frac{P(A,B)}{P(B)}$$

How will we find

Gaussian Distribution

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(x-\mu)^2}$$

$$N(x|\mu,\Sigma) \triangleq \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}}$$

→ Probability Distribution

$\Sigma = U \Lambda U^T$

↳ Orthonormal matrix satisfying $U^T U$ diagonal matrix

Covariance Matrix

$\Sigma^{-1} = U^{-T} \Lambda^{-1} U^{-1} =$

U_i is the ith column

$(x-\mu)^T \Sigma^{-1} (x-$

where $y_i = U_i^T$

have algorithm

c=n, T=y)
N, T=y)
y)

y, Btg=y T=N)
Btg=y, T=N)
BTh=y, T=N)

and finding the

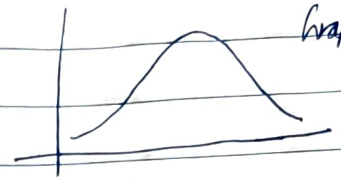
We use Joint Distribution

How will we find Covariance matrix

Date: / / /

Gaussian Distribution

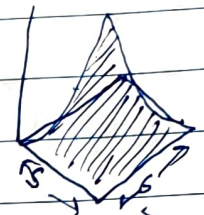
$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(x-\mu)^2}$$



Graph

$$\boxed{N(x|\mu,\Sigma) \triangleq \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]} \approx \text{ie}$$

→ Probability Distribution – Gaussian & Normal



$$X = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$\Sigma = U\Lambda U^T$$

Covariance Matrix

↳ Orthonormal matrix of eigenvectors satisfying $U^TU=1$ and Λ is a diagonal matrix of eigenvalues.

From Linear Algebra ← How?

$$\Sigma^{-1} = U^{-T}\Lambda^{-1}U^{-1} = \boxed{U\Lambda^{-1}U^T = \sum_{i=1} \frac{1}{\lambda_i} u_i u_i^T}$$

NOTE:
$U^TU=1$
$(U^T)^{-1}U^TU = U^{T^{-1}}$
$U=(U^T)^{-1}$

u_i is the ith column of U containing the ith eigenvector

$$(x-\mu)^T\Sigma^{-1}(x-\mu) = (x-\mu)^T\left(\sum_{i=1}^{D} \frac{1}{\lambda_i} u_i u_i^T\right)(x-\mu)$$

$$= \sum_{i=1}^{D} \frac{1}{\lambda_i} \underbrace{(x-\mu)^T u_i}\, u_i^T(x-\mu) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

where $y_i = u_i^T(x-\mu)$ → Recall Ellipse is

$$\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} = 1$$

Date: / /

MLE for a Gaussian. If we have N iid samples $x_i \sim N(\mu, \Sigma)$ then MLE for the parameters,

$$\mu_{mle} = \frac{1}{N}\sum_{1}^{N} x_i \triangleq \bar{x}$$

$$\Sigma_{mle} = \frac{1}{N}\sum_{1}^{N}(x_i-\bar{x})(x_i-\bar{x})^T = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T - \bar{x}\bar{x}^T$$

$$P(C_1|x) = \frac{P(x|C_1)\,P(C_1)}{P(x)} \propto P(x|C_1)\,P(C_1)$$

$C_1 \to$ Covid

$C_2 \to$ No Covid

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}\,\Sigma^{1/2}}\, e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

$$P(C_1|x) > P(C_2|x) \iff P(x|C_1)P(C_1) > P(x|C_2)P(C_2)$$

$$\frac{1}{(2\pi)^{D/2}\,\Sigma_1^{1/2}}\, e^{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}\, P(C_1) > \frac{1}{(2\pi)^{D/2}\,\Sigma_2^{1/2}}\, e^{-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)}\, P(C_2)$$

$$-\frac{D}{2}\log(|\Sigma_1|) - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \log(P(C_1)) > -\frac{D}{2}\log(|\Sigma_2|) - \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) + \log(P(C_2))$$

$$-\frac{D}{2}\log|\Sigma_1| + \log P(C_1) - \frac{D}{2}\log|\Sigma_2| + \log P(C_2) > -\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) + \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)$$

Date: / /

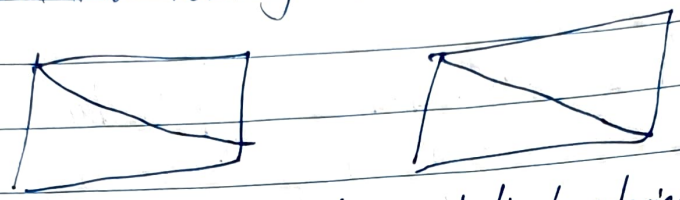
$$(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - (x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) < k$$

but $\Sigma_1 = \Sigma_2$

We get

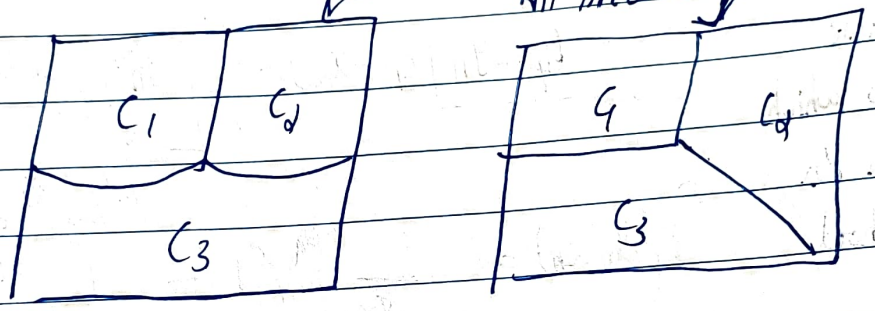
$$-x^T\Sigma\mu_1 - \mu_1^T\Sigma x + \mu_1^T\Sigma\mu_1 - (-x\Sigma\mu_2 - \mu_2^T\Sigma x + \mu_2^T\Sigma\mu_2) < k$$

We get a linear boundary instead of a quadratic boundary



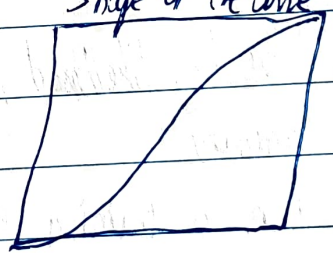
If we combine one linear and two quadratic boundaries

All linear



- We are figuring out ways to find $P(C_1/x)$ without finding $P(x/C_1)$ and see if we can get an answer.
- We assume there is some function for $P(C_1/x)$, should give input between 0 and 1

Shape of the Curve



Sigmoid

- $P(C_1/x) = \dfrac{1}{1+e^{-w_1 x}}$

Function changes based on w

If we have two parameters $y = \dfrac{1}{1+e^{-(w_1x_1 + w_2x_2)}}$ 3D Sigmoid

PRO → DOBBX

$w \to$ Sorta like slope

Date: / /

Let's say we have $(x_1, y_1)\ (x_2, y_2) \ldots (x_n, y_n)$

$$\frac{1}{1+e^{-w^T x_n}} \to 1 - \frac{1}{1+e^{-w^T x_n}} = 1 - \frac{1}{1+\frac{1}{e^{w^T x_n}}}$$

$$= 1 - \frac{e^{w^T x_n}}{1+e^{w^T x_n}}$$

$$P(y_1, y_2 \ldots y_n)_{x_1, x_2 \ldots x_n} = \prod_1^N P(y_n/x_n)$$

$$= \prod_1^N \left(\frac{1}{1+e^{-w^T x_n}}\right)^{y_n} \left(1 - \frac{1}{1+e^{-w^T x_n}}\right)^{1-y_n}$$

~~y Disease~~
~~x features~~

We choose that w which gives us Max Likelihood

$$-\log P(y_1 \ldots y_n / x_1 \ldots x_n) = \sum_1^N \left[y_n \log\left(1+e^{-w^T x_N}\right) + (1-y)\log\left(1+e^{-w^T x_N}\right) \right]$$

$$P(y_n/x_n) = \left(\frac{1}{1+e^{-w^T x_n}}\right)^{y_n} \left(1 - \frac{1}{1+e^{-w^T x_n}}\right)^{1-y_n}$$

0 or 1

Find w which maximizes $P(y_1 y_2 \ldots y_n / x_1 \ldots x_n)$

$\to$ -ve log likelihood function

☆ Summary

- Given n training examples
  - Some +ve
  - Some -ve
- We don't want Generative Approach
- We put function and try to get Parameters

Hes

- We will have a lot o
- We can find Probabi
- If +ve, we multiply
- We need very close
- $(x_1 y_1) \ldots (x_n y_n)$
  - ↳ Output
  - ↳ Input
- We maximize likelih

$x \to$ feature vector

$t \to 0$ or 1 : Could o

Output vector $y_n = y(x$

$P\left(\frac{D}{w}\right) = P($

But $P(t/x)$

$P(t_n/x_n)$

Neg. log lik

$-($

Cross Entropy Function

☆☆ $E(w) = [$

Error Function

Here y is function

$\frac{dE}{dw_1} =$

Date: / /

Hessian: +ve Semidefinite Matrix
Convex function
It will have global minima. / /

- We will have a lot of S shapes for multiple parameters
- We can find Probabilities assuming w
- If +ve, we multiply P (else $1-P$)
- We need very close w for answer
- $(x_1, y_1) \dots (x_n, y_n)$ are independent
  - ↳ Output
  - ↳ Input
- We maximize likelihood function

Note!
Likelihood $= \pi P(x)$
$dy/dx = 0$

$x \to$ feature vector $(x_1, t_1) \dots (x_n, t_n)$
$t \to 0$ or 1 : Covid or Non Covid

Output Vector $y_n = y(x_n) = \dfrac{1}{1+e^{-w^T x_n}}$

$$P\left(\frac{D}{w}\right) = P\left((t_1 t_2 \dots t_n) \,/\, (x_1 \dots x_n)\, w\right) = \prod_{x=1}^{N} P(t_n / x_n, w)$$

↳ We assume everything independent

But $P(t/x) = \dfrac{1}{1+e^{-w^T x}}$

$$P(t_n/x_n) = \underbrace{\left(\frac{1}{1+e^{-w^T x}}\right)^{t_n}}_{+ve\ example} \underbrace{\left(1 - \frac{1}{1+e^{-w^T x}}\right)^{1-t_n}}_{-ve\ example}$$

Neg. log likelihood

$$-\log(P(D/w)) = -\log\left(\prod_1^N \left(\frac{1}{1+e^{-w^T x_n}}\right)^{t_n} \left(1 - \frac{1}{1+e^{-w^T x_n}}\right)^{1-t_n}\right)$$

Cross Entropy Function

$$E(w) = \boxed{-\log P(D/w) = -\sum_1^N \left[t_n \log(y_n) + (1-t_n)\log(1-y_n)\right]}$$

Error function

→ Convex Curve

Here y is a function

$$\frac{dE}{dw_1} = -\sum_1^N \left(t_n \frac{1}{y_n} y_n(1-y_n)(-x_n) + (1-t_n)\frac{1}{1-y_n}(-y_n)(1-y_n)(-x_n)\right)$$

↳ $t_n x_n + t_n y_n x_n - x_n t_n y_n$
$= x_n(y_n - t_n)$

$$= -\sum_1^N \left[(y_n - t_n) x_n\right]$$

Per weight

Date: / /

- If a curve has a convex shape - we get global minima
- Question is how to find this minima

$$w^{k+1} \leftarrow w^{k} - \eta \left(\frac{d\mathcal{E}}{dw} \Big|_{w=w^{k}} \right)$$

In Linear Regression, we have this equation

If η becomes too big → it makes bigger jumps

t_n → targeted value → y_{true}

If $\sigma(z) = \frac{1}{1+e^{-z}} = y_n$

$$\sigma'(z) = (1-\sigma(z))\sigma(z)$$

$$y_n'(z) = (1-y_n)y_n$$

$$\boxed{\frac{dy_n'}{dz} = \frac{(-1)}{(1+e^{-z})^2}(-e^{-z}) = \left(1-\frac{1}{1+e^{-z}}\right)\left(\frac{1}{1+e^{-z}}\right)}$$

We get

$$\frac{d\mathcal{E}}{dw_1} = -\sum_{1}^{N} -(y_n+t_n)x_{n1} = \sum_{1}^{N}(-t_n+y_n)x_{n1}$$

x_n → Vector

$$\nabla\mathcal{E}(w) = \sum_{1}^{N}(t_n - y_n)x_n$$

→ This is for

→ All features

→ Looping through all examples

∴ We get

$$\boxed{w^{k+1} \leftarrow w^{k} - \eta \left[\nabla\mathcal{E}(w) \Big|_{w=w^{k}} \right]}$$

We find w assuming some random w and goon

$$\hat{w} = \max P(D/w)$$

But how to find $P(w/D)$

We computed the likelihood and the respective ω for which we get closest answer

$$P\left(\frac{\omega}{D}\right) = \frac{P(D/\omega)\,P(\omega)}{P(D)}$$

We hand this

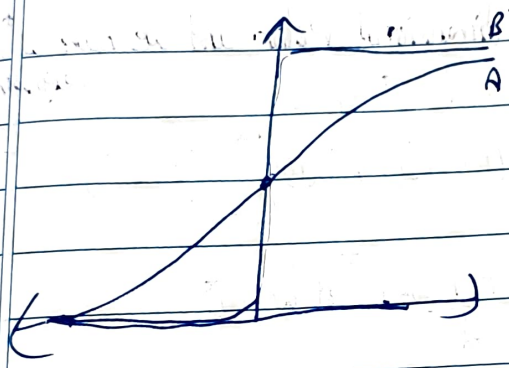
If someone tells based on domain expertise

More insightful

$$\int P(D/\omega)\,P(\omega)\,d\omega$$

$$P(t/x,D) = \int P(t,\omega/x,D)\,d\omega = \int P(t/\omega,x,D)\,P(\omega/x,D)\,d\omega$$

$$= \int P(t/\omega,x)\,P(\omega/D)\,d\omega$$



- A better than B
- B is over fitted and does not give us a range of probabilities

- ω is very high, we get an over fitted curve.
- We need to regularize freedom of ω.
- Decision Boundary $\boxed{\omega_0+\omega_1 x}$

$$\frac{1}{1+e^{-(\omega_0+\omega_1 x_1+\omega_2 x_2 \cdots)}} > 0.5$$

$$2-1 > e^{-(\omega_0+\omega_1 x \cdots \omega_D x)}$$

$$-\log 1 > -(\omega_0+\omega_1 x \cdots \omega_D x)$$

$$-(\omega_1 x+\omega_2 x+\cdots \omega_D x) < (\omega_D+\log 1)$$

Logistic Regression model not linear, but the decision boundary is linear.

L2 Norm: $\omega_1^2+\omega_2^2 < 1$ $\quad$ L_1 Norm $= \left(\sum_{i=1}^{k} |x_i|^n\right)^{1/n}$

Norms are used to curtail boundary in logistic Regression

PRO

Date: / /

get global minima

$\Big|_{w=w^k}$

equation

bigger jumps

$$\boxed{n' = \frac{(-1)}{(1+e^{-2})^2}(-e^{-2}) = \left(1-\frac{1}{1+e^{-2}}\right)\left(\frac{1}{1+e^{-2}}\right)}$$

$(n+\theta_1)x_1$

is for

ll examples

$\boxed{w^k}$

and gorm

We computed the likelihood and the respective $\hat{w}$ for which we get closest answer

Date: / / /

we hard this

$$P\begin{pmatrix} w \\ D \end{pmatrix} = \frac{P(D/w)\,P(w)}{P(D)}$$

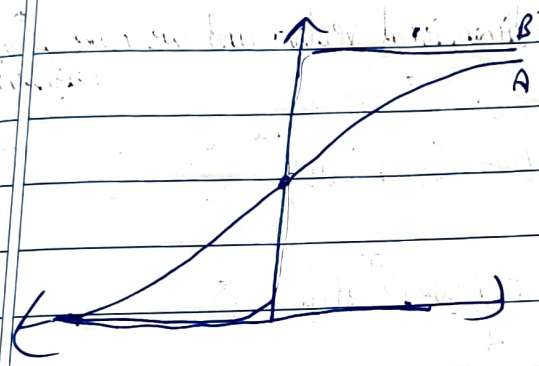
It someone tells based on domain expertise

More insightful

$$\int P(D/w)\,P(w)\,dw$$

$$P(t/x,D) = \int P(t,w/x,D)\,dw = \int P(t/w,x,D)\,P(w/x,D)\,dw$$

$$= \int P(t/w,x)\,P(w/D)\,dw$$



- A better than B
- B is over fitted and does not give us a range of probabilities

- w is very high, we get an over fitted curve.
- We need to regularize freedom of w.
- Decision Boundary $\boxed{w_0+w_1x}$

$$\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2\cdots)}} > 0.5$$

$$2-1 > e^{-(w_0+w_1()\cdots w_D())}$$

$$-\log 1 > -(w_0+w_1()\cdots w_D())$$

$$-(w_1()+w_2()+\cdots w_D()) < (w_0+\log 1)$$

Logistic Regression model not linear, but the decision boundary is linear.

L2 Norm: $w_1^2+w_2^2<1$ $\quad L_1\text{Norm} = \left(\sum_{i=1}^{k}|x_i|^n\right)^{1/n}$

Norms are used to curtail boundary in logistic Regression

PRO

$P(t/x) = \frac{1}{1+e^{-w^T x}}$

Date: / /

$$P(w/D) = \frac{P(D/w)\,P(w)}{P(D)} \Rightarrow P(w/D) \propto P(D/w)\,P(w)$$

$$= \frac{P(D,w)}{P(D)} = \frac{P(D/w)\,P(w)}{P(D)} = e^{-E(w)}$$

$E(w) = -\log P(w/D)$ We take this and get here

$= -\log(P(D/w)\,P(w))$

$$\boxed{E(w) = -[\log P(D/w) + \log P(w)]}$$

$P(w/D) \propto e^{-E(w)}$

In this we have $-\log P(D/w)$

Now, $E(w) = E(w^*) + (w-w^*)\nabla E(w)\big|_{w=w^*}$ + $\frac{1}{2}(w-w^*)^T \nabla^2 E(w)\big|_{w=w^*}(w-w^*)$

w multidimensional vector and we have w^* in neighborhood

$= E(w^*) + \frac{1}{2}(w-w^*)^T H (w-w^*)$

$H = -\frac{d^2(\log P(w/D))}{dw^2}$

$\rightarrow N(w/w^*, H^{-1})$

We assume prior probability $P(w)$ as a normal distribution $N(w/(0, \lambda I))$

Laplace Approximation

$P(w/D) \propto e^{-E(w)}$

$\propto e^{-E(w^*)} e^{-\frac{1}{2}(w-w^*)^T H (w-w^*)}$

$\propto \underbrace{N(w/w^*, H^{-1})}_{\text{Normal distribution}}$ Inverse Hessian

because!

$$N(x/\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}\Sigma^{-1}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

In Laplace Approx. $g(\theta)$

We get $g'(\theta_0) = 0$

$f(\theta_0) = f(\theta_0) e^{(1/2}$

$P_1 = \frac{1}{1+e^{-(w_0+w_1 C)+}}$

$1-P_1 = 1 - \frac{}{1+e^{-(w}}$

Interpretability of M

$\frac{P(\text{Success})}{P(\text{Failure})}$ = Odd

$\frac{P_1}{1-P_1} = e$

$\frac{P_2}{1-P_2} = e$

Now, $\frac{P_2}{1-P_2} \Big/ \frac{P_1}{1-P_1}$

☆ Multivariate M

- We have multi
- Say 5 levers

e is used to take p from (0,1)

$w_1^T x_i \rightarrow S$

We need to find

$\mu_{11} =$

In Laplace Approx. ① Find Second Order Taylor Expansion

$$g(\theta) = \log f(\theta) = g(\theta_0) + g'(\theta_0)(\theta-\theta_0) + \tfrac{1}{2} g''(\theta_0)(\theta-\theta_0)^2$$

We get $g'(\theta_0) = 0$

$$f(\theta_0) = f(\theta_0) e^{\left(\frac{1}{2} g''(\theta_0)(\theta-\theta_0)^2\right)}$$

Date: / / /

$$P_1 = \frac{1}{1+e^{-(w_0 + w_1 () + w_2 () + \cdots w_D())}}$$

$$1 - P_1 = 1 - \frac{1}{1+e^{-(w_0 + () + w_1() \cdots w_D())}}$$

$$C = P(D) = \int_R e^{\left(-\frac{\gamma}{2}(x-\mu)^2\right)} dx = \frac{\sqrt{2\pi}}{\sigma}$$

*Murphy Book

Interpretability of Models

$\frac{P(\text{Success})}{P(\text{Failure})}$ = Odds Ratio

Insight: Random person is x times more prone to success

$$\frac{P_1}{1-P_1} = e^{w_0 + w_1 x + w_2 x_1 \cdots}$$

$$\frac{P_2}{1-P_2} = e^{w_0 + w_1 (x_1 + 1) + w_2 x_2}$$

Now, $\frac{P_2}{1-P_2} \Big/ \frac{P_1}{1-P_1} = e^{w_1}$

☆ Multivariate Model for Logistic Regression (I think?)

- We have multiple features, and also multiple outputs
- Say 5 fevers we can use vectors

fever 1	10000
fever 2	01000
fever 3	00100
	...

e is used to take p from $[0,1]$

$w_1^T x_i \to$ Scalar

We need to find Probabilities for each fever

$$\mu_{11} = \frac{e^{w_1^T x_1}}{\underbrace{\sum_{i=1} e^{w_i^T x_i}}_{\text{SoftMax}}}$$

Malaria, Covid ..., Typhoid

$$w = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1C} \\ w_{21} & w_{22} & \cdots & w_{2C} \\ & & & \\ w_{D1} & w_{D2} & \cdots & w_{DC} \end{bmatrix}$$

w_1

$$\mu_{ic} = \frac{e^{w_c^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}$$

Date: / /

$w \rightarrow$ which makes all training examples accurate

$$P(D/w) = P((x_1,y_1)(x_2,y_2) \cdots (x_n,y_n) / w)$$

$$= P(y_1, y_2 \cdots y_n \mid x_1 x_2 \cdots x_n\, w)$$

$$= \prod_{i=1}^{n} P(y_i / x_i\, w)$$

$$= \prod_{i=1}^{n} \prod_{\alpha=1}^{c} (\mu_{i\alpha})^{y_i \alpha}$$

Need to maximize

$$\max \log \left(\prod \prod (\mu_{i\alpha})^{y_{i\alpha}} \right)$$

$$\min = -\log \left(\prod \prod (\mu_{i\alpha})^{y_{i\alpha}} \right)$$

$$= -\sum_{i=1}^{n} \sum_{\alpha=1}^{c} \log(\mu_{i\alpha}) = -\sum \sum y_{i\alpha} \log \left(\frac{e^{w_\alpha^T x_i}}{\sum e^{w_\alpha^T x_i}} \right)$$

$$\min \left[\sum_{i=1}^{n} \sum_{\alpha=1}^{c} \left(y_\alpha w_\alpha^T x_i - y_i \log \left(\sum e^{-x_i} \right) \right) \right]$$

Cost function

$(x_1, y_1)(x_2, y_2) \cdots (x_n, y_n)$ $\quad y_i = \begin{cases} 1 \\ 0 \end{cases}$

- We make functions for each output

Basics functions

Deep learning ANNs

$$\phi(x) = \begin{bmatrix} \phi(x_1) \\ \vdots \\ \phi(x) \end{bmatrix}$$

Boundary $\rightarrow$ m dim Hyperspace

Decision Boundary: $w_1 \phi_1(x) + w_2 \phi_2(x) \cdots \geq d$

(x_1, y_1) $\quad (x_2, y_2$

$\rightarrow y_1 = (0, 1$

$y_2 = (0$

max μ_{12}

max $(\mu_{11}$ $\quad y_{11}$ $\quad \mu_{12}$

$(\mu_{21}$

$$\max \prod_{i=1}^{n} \prod_{\alpha=1}^{c}$$

$\mu_{10000,1}$

$\mu_{10000,}$

μ_{10000}

Date: / / /

$(x_1, y_1) \quad (x_2, y_2) \qquad\qquad (x_n, y_n)$

$= \downarrow$

$\rightarrow y_1 = (0, 1, 0, 0, 0)$

$y_2 = (0, 0, 1, 0, 0)$

max $\mu_{12} \quad \mu_{2\cdot 3}$

$$\max \begin{pmatrix} y_{11} & y_{12} & y_{15} \\ \mu_{11} & \mu_{12} & \mu_{15} \end{pmatrix}$$

$$\begin{pmatrix} \mu_{21} & \mu_{22} & \mu_{25} \end{pmatrix}$$

$$\max \prod_{i=1}^{n} \prod_{c=1}^{C} \mu_{i\cdot c}^{y_{i\cdot c}}$$

$\underline{\underline{w}}$

x_{10000}

$\mu_{10000,1}$

$\mu_{10000,c}$

$\mu_{10000,5}$

Date: / /

Perceptron

- We have +ve and -ve points, Perceptron algorithm will converge to one hyperplane when it is able to separate all points
- Decision Boundary – That Plane
- Domain expert comes and chooses good features for example we have $f_1 f_2 f_3 \ldots$, we choose $f_2 f_4 f_5$ as good features

- Say the data is not seperable in 20 features but becomes linearly seperable in 10 features
We then apply algorithms on the 10 feature set

$$\phi(f_1 f_2 \ldots f_{20}) = (\phi_1(f_1 f_2 \ldots f_{20}) \; \phi_2(f_1 \ldots f_{20}) \ldots \phi_{10}(f_1 f_2 \ldots f_{20}))$$

ϕ → Basis Function

We get Classifier as

$$w_1\phi_1 + w_2\phi_2 \ldots w_{10}\phi_{10} = \alpha$$

If we want Decision boundary in original dimensions, we expand the basis functions

Linear Discriminant Analysis

We maximize difference between means and minimize $s_1^2 + s_2^2$

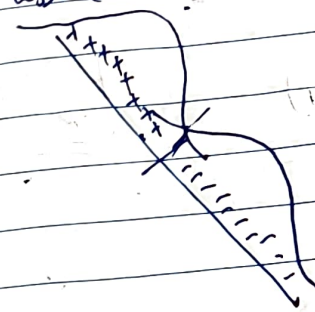
$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

$(m_1 - m_2)^2 \rightarrow w^T S_B w$

$s_1^2 + s_2^2 \rightarrow w^T S_W w$

No guarantee → Training Error = 0

We assume Gaussian Distr



$$P\left(\frac{C_1}{x}\right) > P\left(\frac{C_2}{x}\right)$$

$w^T x_{test} > \alpha$
+ve
Decision

Logistic Regression

$$y(x) = \frac{1}{1+e^{-(w_1 x_1}}$$

$$\frac{1}{1+e^{-w^T x}}$$

1 =

2 =

1

0

Decision Boun

If we don't have
functions, we

reptron algorithm
hen it is able

ood features
we choose

eatures but becomes

set

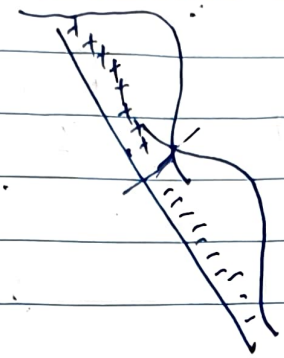
$\phi_2(h_b - h_d)$
$h_e)$

dimensions,

and minimiz

Date: / / /

We assume Gaussian Distribution



Otherway, we use Bayesian Perspective

$$P\left(\frac{C_1}{x}\right) = \frac{P(x/C_1)\,P(C_1)}{P(x)}$$

$$\propto P(x/C_1)\,P(C_1)$$

$$P\left(\frac{C_1}{x}\right) > P\left(\frac{C_2}{x}\right) \iff P\left(\frac{x}{C_1}\right) P(C_1) > P\left(\frac{x}{C_2}\right) P(C_2)$$

The normal distribution curve

$w^T x_{test} > \alpha$	$w^T x_{test} < \alpha$
+ve	−ve

Decision Boundary $= w^T x_{test} = \alpha$

Plane

logistic Regression

$$y(x) = \frac{1}{1+e^{-(w_1 x_1 + w_2 x_2 \cdots)}} = \frac{1}{1+e^{-w^T x}}$$

We use Gradient Descent to find this

$$\frac{1}{1+e^{-w^T x}} > 0.5$$

$$1 > 0.5\,(1+e^{-w^T x})$$

$$2 > 1+e^{-w^T x}$$

$$1 > e^{-w^T x}$$

$$0 > -(w_1 x_1 \cdots)$$

Linear Curve

Decision Boundary

If we don't have basis functions and we want to put/find out these functions, we use ANNs.

Date: / /

$x_1 = [x_{11}, x_{12}, \ldots, x_{1D}]$

$x_{11}^2 + x_{12}^2 - x_{13}$

$\phi_1(x_1)$ $\phi_2(x_1)$ $\phi_M(x_1)$

$$\frac{1}{1+e^{-w^T \cdot \phi(x)}}$$

Activation function: 20 Variables → 10 Variables
↳ linear Combination
↳ Apply σ on this and get Layer 2 feature

Other Gradient Descents
① Batch — 100s
Mini Batch — 10s
Stochastic — 1s

We have D features, Basis functions $\phi_1\ \phi_2 \ldots \phi_m$

LogReg → $\frac{1}{1+e^{-w^T \phi(x_n)}}$ $\phi(x_n) = (\phi_1(x_n), \phi_2(x_n) \ldots \phi_m(x_n))$

Let's say we are not given these functions. We mature D features into m features
We learn these basis function from the data

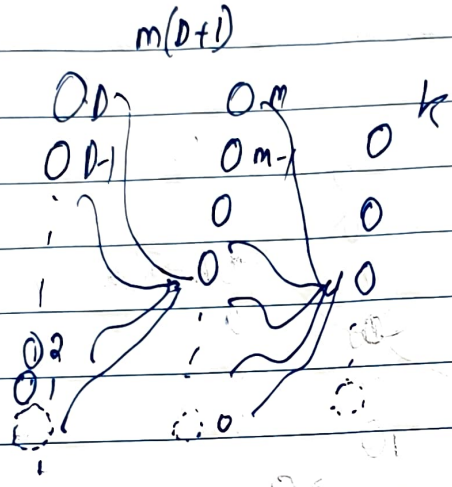
$w^2 \rightarrow$ 2nd layer of w

$$a_j = \sum_{i=1}^{D} w_{ji}^{1} x_i + w_{j0}$$ Linear Combination

$$z_j = \sigma(a_j)$$ Make it non linear

$$h_k = \sum_{1}^{m} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

$$\boxed{y_k = \sigma(h_k)}$$ → For Regression I is used.



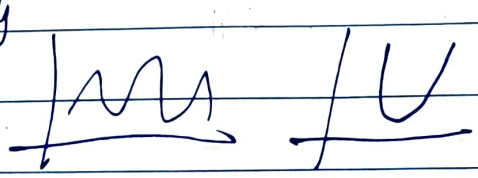
$\therefore$ Total $= m(D+1) + k(m+1)$

In linear regression - activation function - Identity function
↳ finding rent and how much comfort I will have
We apply log-reg on the given functions — ANN
We can use ANN for linreg also and classification also and only the last layer changes

$$h_k = I(a_k) = a_k = \sum w_j^{(2)} z_j + w_{kj}^{(2)}$$
$$= \sum_{j=1}^{m} w_{kj}^{2} \sigma(a_j) + w_{kj}^{2}$$
$$= \sum_{j=1}^{m} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{D} w_{ji} x_i + w_{j0}\right] + w_{kj}^{2}$$

↳ Ends up not being a quadratic

We cannot use gradient Descent Directly ← Loses Convexity ←



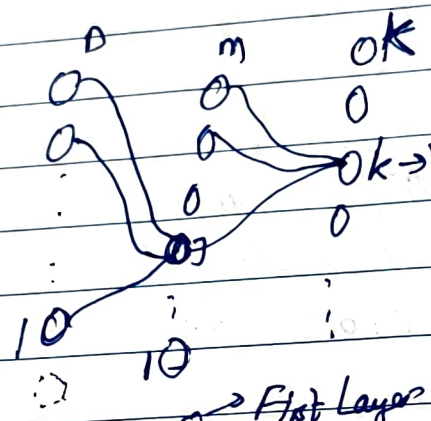
Date: / /

ANNs: Revisit

- If x has say D features, these are not themself enough to predict them.
- We use basis functions and get the ~~answer~~ converted functions

$$y = f(\phi(x))$$
$$= f(\phi_1(x)\,\phi_2(x) \cdots \phi_m(x))$$

- If these functions are not given, can we cook them up using ANNs



No. of Parameters
$(D+1)m + (m+1)k$

$$z_j = \sum_{1}^{D} w_{ji}^{1} x_i + w_{j0}^{1} \quad \text{(1) → First Layer}$$

$$a_j = h(z_j)$$

Activation function

$$z_k = \sum_{1}^{m} w_{kj}^{2} a_j + w_{k0}^{2} \quad \text{(2) → Second Layer}$$

$$a_k = h'(z_k)$$

D Basic Features
m Matured Features } k target variables.

$$a_k = h'\left(\sum_{j=1}^{m} w_{kj}^{2}\, h\left(\sum_{i=1}^{D} w_{ji}^{1} x_i + w_{j0}^{1}\right) + w_{k0}^{2}\right)$$

We convert $\sum_{1}^{D} w_{ji}^{1} x_i + w_{j0}^{1} \rightarrow \sum_{0}^{D} w_{ji}^{1} x_i$

Same for $z_k \rightarrow \sum_{0}^{m} w_{kj}^{2} a_j$

Date: / / /

We have 2 layers and we can have multiple equivalent vectors which give the same prediction

$$h = \tanh$$
$$\tanh(-a) = -\tanh(a)$$

$\rightarrow w_{j0} + w_{j1}x_1 + w_{j2} \cdots w_{jD}x_D \rightarrow \tanh(w_{j0} + w_{j1}x_1 + w_{j2}x_2 \cdots w_{jD}x_D)$

$\rightarrow -w_{j0} - w_{j1}x_1 - w_{j2}x_2 \cdots -w_{jD}x_D$ (Negation) and we apply tanh on this, we get $-\tanh(\sum_0^D w_{ji}x_i)$

- If we add negation in the m-k layer, we get the same answer even though we have a different vector

We have $m2^m$ possibilities of weight vectors for same output
↳ We reorder nodes with their weights

$(x_1, t_1)\ (x_2, t_2) \cdots (x_n, t_n)$

$$t \sim N(y(x, w), \beta^{-1}) \quad \beta = \frac{1}{\sigma^2}$$

$y(x_1, w)\ y(x_2, w) \cdots y(x_n, w)$

$P((x_1, t_1)(x_2, t_2) \cdots (x_n, t_n) \mid w)$

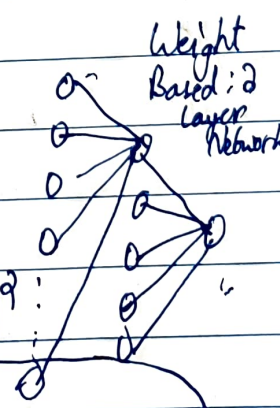
$= P(t_1 t_2 t_3 \cdots t_n \mid x_1 x_2 \cdots x_n, w)$

$= \prod_1^N (P(t_n \mid x_n, w)) = \prod_1^N \frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{-\beta/2 (t_n - y(x_n, w))^2}$

$\max P(t_1 t_2 \cdots t_n \mid x_1 x_2 \cdots x_n\ w)$

$\max \left(\frac{\beta}{2\pi}\right)^{N/2} e^{-\beta/2 \sum (t_n - y(x_n, w))^2}$

$\text{maximize} \rightarrow \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \frac{\beta}{2} \sum_1^N (t_n - y(x_n, w))^2$



minimize sum of squares of errors

Given w, can we find optimal network. What w maximises Probability of training data.

Date: / /

Finding Distribution ⇒ Point Estimation

- We assume follows normal distribution
- We can get y_n using all this
- We use MLE and ~~[illegible]~~ maximize loglikelihood.
- In logreg/reg we used the fact that, they are a concave but here we have multiple pts so we cannot directly use it.
- Using this dw, we find the network
- Point being, Here y is not so same, because if y use tanh, it becomes very hard

Derivative $\lim_{\delta x \to 0} \frac{f(x+\delta x) - f(x)}{\delta x}$ at given x

$$\nabla_w y(x_n, w) = \begin{bmatrix} \frac{\delta}{\delta w_1} y(x_n, w) \\ \frac{\delta}{\delta w_2} y(x_n, w) \\ \vdots \\ \frac{\delta}{\delta w_n} y(x_n, w) \end{bmatrix}$$

But the problem is since $\delta x \to 0$, we'll lose precision and accuracy. → Leading to numerical errors.

w & x_n is known, what is the complexity of knowing

No. of Additions/Operations

$(D+1) + D+1$
↳ $O(D)$

D+1 Multiplications
D Additions
1 Function
we multiply with M,
We get
$O(MD)$

$$\rightarrow ((D+1) + D+1)M + (M+1 + M+1)$$

$$t_n = \begin{bmatrix} t_{n_1} \\ t_{n_2} \\ \vdots \\ t_{n_K} \end{bmatrix} \qquad x_n =$$

↗ Error(n)

$$E_n = \sum_{k=1}^{K} (y_{nk} - t_{nk}$$

$$E = \sum_{n=1}^{N} E_n$$

$$\frac{\delta E}{\delta w_{ji}} = \frac{\delta}{\delta w_{ji}} \left(\sum_{1}^{n} E_n \right.$$

$$\hookrightarrow \frac{\delta E_n}{\delta w_{ji}} = \frac{\delta E_n}{\delta w}$$

$$= \frac{\delta D}{\delta w}$$

$$\boxed{w_{k+1}}$$

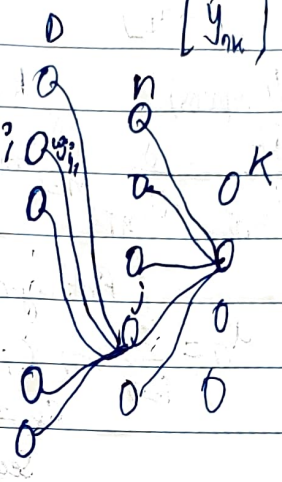
We cannot find as answer

$$\lim_{\delta x \to 0} f$$

Date: / /

$$t_n = \begin{bmatrix} t_{n_1} \\ t_{n_2} \\ \vdots \\ t_{n_K} \end{bmatrix} \qquad x_n = \begin{bmatrix} x_{n_1} \\ x_{n_2} \\ \vdots \\ x_{n_D} \end{bmatrix} \qquad y_n = \begin{bmatrix} y_{n_1} \\ y_{n_2} \\ \vdots \\ y_{n_K} \end{bmatrix}$$

Predicted Outcome of nth training example of kth target attribute



Error(n)

$$E_n = \sum_{k=1}^{K} (y_{nk} - t_{nk})^2$$

$$E = \sum_{n=1}^{N} E_n$$

$$\frac{\delta E}{\delta w_{ji}} = \frac{\delta}{\delta w_{ji}} \left(\sum_{1}^{n} E_n \right) = \sum_{1}^{N} \frac{\delta E_n}{\delta w_{ji}}$$

$$\hookrightarrow \frac{\delta E_n}{\delta w_{ji}} = \frac{\delta E_n}{\delta a_j} \frac{\delta a_j}{\delta w_{ji}}$$

$$= \frac{\delta D}{\delta w} \left(\sum_k \frac{\delta E_n}{\delta a_k} \frac{\delta a_k}{\delta a_j} \right)$$

$$\boxed{w_{k+1} \leftarrow w_k + \eta \frac{\delta E}{\delta w}}$$ Gradient Descent

We cannot find as we did in logreg because it won't give a single answer

$$\lim_{\delta x \to 0} \frac{f(x+\delta x) - f(x)}{\delta x} \rightarrow \frac{f(x+10^{-6}) - f(x)}{10^{-6}}$$

Date: / /

ANN long lecture

$(x_1, t_1)\ (x_2, t_2) \ldots (x_n, t_n)$

$$x_n = \begin{bmatrix} x_{n_1} \\ x_{n_2} \\ \vdots \\ x_{n_D} \end{bmatrix} \qquad t_n = \begin{bmatrix} t_{n_1} \\ t_{n_2} \\ \vdots \\ t_{n_K} \end{bmatrix}$$

(D Inputs) (K Target Attributes)

$$E = \sum^{N} E_n = \sum_{n=1}^{N} \left(\sum_{k=1}^{K} \tfrac{1}{2} (y_{n_k} - t_{n_k})^2 \right)$$

MSE

We assume normal dist. and try to maximize likelihood.

$$\frac{\delta E}{\delta w_{ji}} = \frac{\delta}{\delta w_{ji}} \left(\sum_{n=1}^{N} E_n \right) = \sum_{n=1}^{N} \frac{\delta E_n}{\delta w_{ji}}$$

First layer

$$a_j = \sum_{i=0}^{D} w_{ji} x_i$$

Activation Function (1st layer)

$$z_j = \sigma(a_j) \quad \text{[Matured Feature]}$$

$$a_k = \sum_{j=0}^{N} w_{kj} z_j$$

Second layer

$$z_k = I(a_k) = a_k$$

$I \Rightarrow$ Regression

Relu/σ for Classification

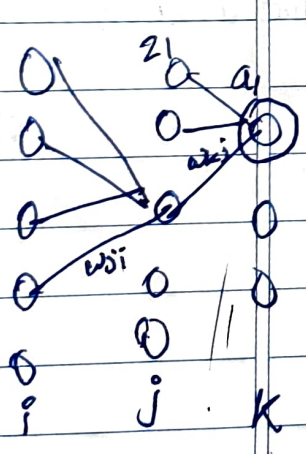
$$(y_k = z_k = a_k)$$

J is hidden layer Number

$$\frac{\delta E_n}{\delta w_{ji}} = \frac{\delta E_n}{\delta a_j} \frac{\delta a_j}{\delta w_{ji}} = \frac{\delta E_n}{\delta a_j} \frac{\delta}{\delta w_{ji}} \left(\sum_{p=0}^{D} w_{jp} x_p \right) = \frac{\delta E_n}{\delta a_j} x_i$$

$$\frac{\delta E_n}{\delta a_j} = \sum_{k=1}^{K} \left(\frac{\delta a_k}{\delta a_j} \right) \left(\frac{\delta E_n}{\delta a_k} \right) = \sum_{k=1}^{K} \frac{\delta \left(\sum_{q=0}^{M} w_{kq} z_q \right)}{\delta a_j} \frac{\delta E_n}{\delta a_k}$$

$$= \sum_{k=1}^{K} \left(\frac{\delta (w_{kj} z_j)}{\delta a_j} \right) \frac{\delta E_n}{\delta a_k} = \sum_{k=1}^{K} \left((w_{kj}) \frac{\delta z_j}{\delta a_j} \right) \frac{\delta E_n}{\delta a_k}$$



$$= \sum_{k=1}^{K} w_{kj} \left[\frac{\delta \sigma(a_j)}{\delta a_j} \right] \left[\frac{\delta E_n}{\delta a_k} \right]$$

$$= \sum_{k=1}^{K} \left(w_{kj} \frac{\delta \sigma(a_j)}{\delta a_j} \right) \left(\frac{\delta E_n}{\delta a_k} \right) = \sum_{k=1}^{K} \left(w_{kj} \sigma'(a_j) \right) \left(\frac{\delta E_n}{\delta a_k} \right)$$

$$= \sigma'(a_j) \left(\sum_{k=1}^{K} w_{kj} (a_k - t_k) \right)$$

Because all other z_j are independent of a_j So they become 0

We need what

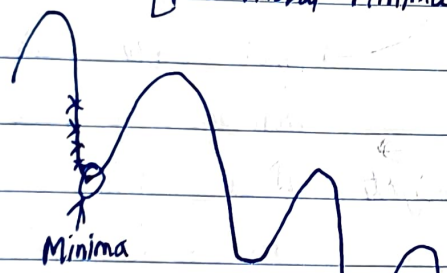
Date / / /

We find error in the last layer → We add up errors for all training examples

Best Weight – less error – Reduce Sum of Square of Errors 'MSE'

y_{nk} → Predicted Value
t_{nk} → Actual Value } nth Training Example for kth target

In LogReg → We have a simple Gradient Descent, we can't do that since we don't even know how many local minima(s) we get.
[Not Global Minima → In ANN]

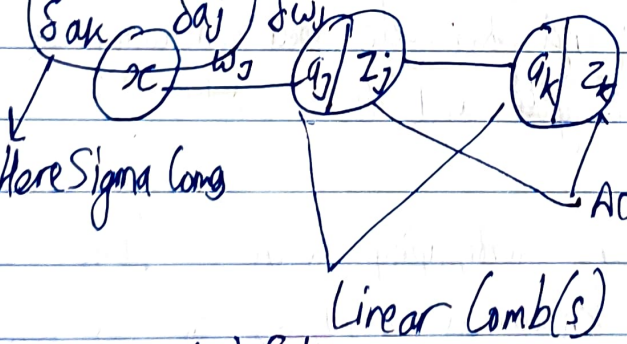


a_j → Linear Combination of $w_{ji} x_i$

We do proof only assuming nth Training Example → We don't mention it

If we go from right to left

$$\frac{\delta \mathcal{E}}{\delta w_j} = \frac{\delta \mathcal{E}}{\delta a_k} \frac{\delta a_k}{\delta a_j} \frac{\delta a_j}{\delta w_j}$$



We have $\sigma'(a_j)\left[\sum_{k=1}^{K} w_{kj}(a_k - t_k)\right]$

↳ Error + w_j's Linear Combination

$$\frac{\delta \mathcal{E}_n}{\delta w_{ji}} = x_i \sigma'(a_j)\left[\sum_{k=1}^{K} w_{kj}(a_k - t_k)\right]$$

↳ Input at Starting Node

Error accumulated at the terminal Way

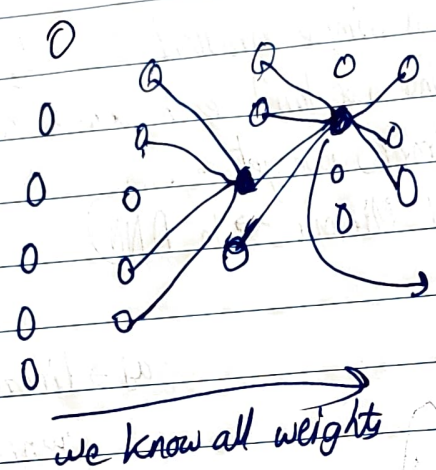
Tich ML

Date: / /

$\left(\frac{\delta \mathcal{E}}{\delta w_{ji}}\right)_{w=w^k}$ we know, so we have

$$w_{ji}^{k+1} \leftarrow w_{ji}^{k} - \eta \left(\frac{\delta \mathcal{E}}{\delta w_{ji}}\right)_{w=w^k}$$

The fact that error is going from right to left
→ Backpropagation

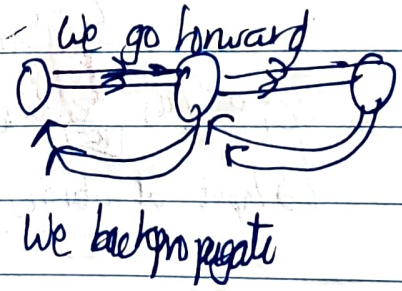


← Error Here we know
We multiply with weights

Input or Error (weights)
Then again we take linear combination and we keep going in a recursive manner.

We forward Propagate the inputs and Backpropagate the error and this cycle goes on, using the matured ~~weights~~ features and the weights

But we are multiplying with σ' (function) but σ' we know as $\underline{\sigma(1-\sigma)}$ so we know all the values



$\frac{\delta \mathcal{E}_n}{\delta w_{kj}} = \frac{\delta \mathcal{E}_n}{\delta a_k} \frac{\delta a}{\delta w}$

$= \frac{\delta \mathcal{E}_n}{\delta a_k} z$

$= \delta \left(\right.$

$= (a_k -$

So in summary

have

$$\left(\frac{\delta \mathcal{E}}{\delta w_{ji}}\right)_{w=wt}$$

m right to left
> Backpropagation

nor Here we know
e multiply with
eight

Error (weights)
ais we take
combination
keep going
ursive manner.
Backpropagate
ing the

nction) but
e know

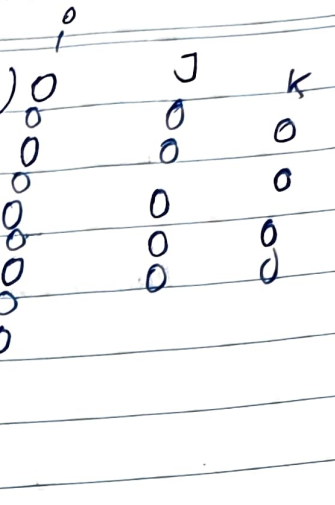
Date / / /

$$\frac{\delta \mathcal{E}_n}{\delta w_{kj}} = \frac{\delta \mathcal{E}_n}{\delta a_k}\frac{\delta a_k}{\delta w_{kj}} = \frac{\delta \mathcal{E}_n}{\delta a_k}\frac{\delta(\mathcal{E} w_{kp} a_p)}{\delta w_{kj}}$$

$$= \frac{\delta \mathcal{E}_n}{\delta a_k} z_j$$

$$= \frac{\delta\left(\frac{1}{2}\mathcal{E}(a_p - t_p)^2\right)}{\delta a_k} z_j$$

$$= (a_k - t_k)\, z_j$$



So in summary $\frac{\delta \mathcal{E}_n}{\delta w_{ji}} = \underbrace{\sigma(a_j)(1-\sigma(a_j))}_{\text{Middle Layer's Differential}}\ \underbrace{x_i}_{\text{Input from first Layer}}\ \underbrace{\left[\sum_{k=1}^{K} w_{kj}(a_k - t_k)\right]}_{\text{Error Accumulated Weights}}$

② ① ③

Date: / /

ML APPLICATIONS

Initially everything paper → IT → Everything digital
↓
AI/ML

We write a lot of test cases and we test the software – To reduce bugs

Should not have bugs

Coding << Testing
30% 50+%

Every requirement – tests

1. Can ML be used for testing and guessing bugs
2. Classify bugs based on Severity
3. Which files have most bugs
4. Code Commenting
   - Code Attributes
   - Change Attributes
   - Program related attributes

☆ ANN – Again FF
(x_1, t_1) (x_2, t_2)
$x_n = (x_{n1}, x_{n2}$
$t_n = (t_{n1}, t_{n2}$ ...

$\prod_{n=1}^{N} \prod_{k=1}^{K} (y_k(x_n,$
and get Answe

As in last time
but when we u
we get classific

$\frac{\delta E_n}{\delta w}$

Date: / / /

ANN - Again FFS

$(x_1, t_1)\ (x_2, t_2) \ldots (x_n, t_n)$

$x_n = (x_{n_1}, x_{n_2} \ldots x_{n_n})$

$t_n = (t_{n_1}, t_{n_2} \ldots t_{n_k})$

$$\prod_{n=1}^{N} \prod_{k=1}^{K} (y_k(x_n, w))^{t_{nk}} (1 - y_n(x_n, w))^{1-t_{nk}}$$

We do max likelihood and get Answer for loss function

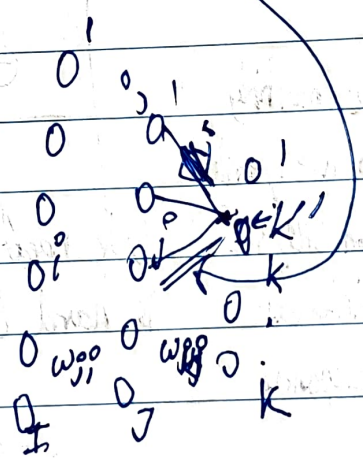
... Classification

As in last time we took regression and used error as Pred - Actual but when we use the things and error function like in logreg we get classification type.

$a_j \rightarrow$ Linear Combination in 1st layer

$$\frac{\delta E_n}{\delta w_{kj}} = \frac{\delta E_k}{\delta a_k} \frac{\delta a_k}{\delta w_{kj}} = \frac{\delta E_k}{\delta a_k} \left(\frac{\partial}{\delta w_{kj}} \left(\sum_1 w_{kj} z_j \right) \right)$$

$$= z_i \frac{\delta}{\delta a_k} \sum_{k=1}^{K} \left[t_{nk} \log (y_k(x_n, w)) + (1 - t_{nk}) \log (1 - y_n(x_n, w)) \right]$$



$$\sum_{j=1}^{J} w_{kj} z_j$$

On this we apply Sigmoid

Mapping

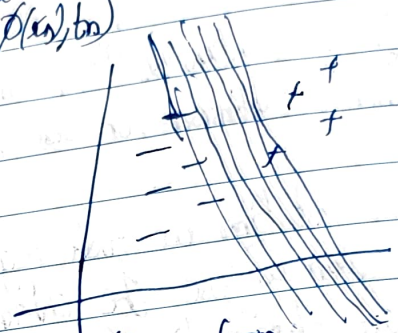
Date: / /

SVM

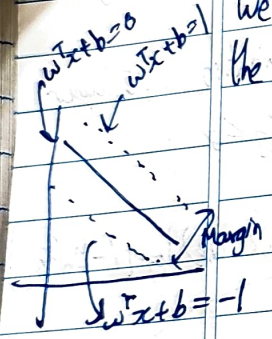
$(x_1, t_1)\ (x_2, t_2) \ldots (x_n, t_n)$ $\quad t_n \in \{-1, 1\}$

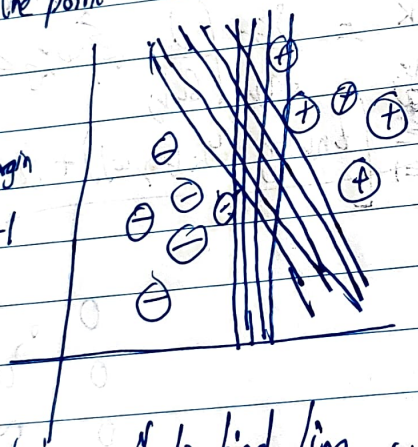
$(\phi(x_1), t_1)\ (\phi(x_2), t_2) \ldots (\phi(x_n), t_n)$

$w^T\phi(x) + b = 0$



We define a margin, we find the distance from the point to the line → this is margin





If we multiply with $n>1$, only on LHS, line shrinks else it expands

We need to find line with max margin and this is hard to compute mathematically

$(w^T\phi(x_n) + b) > 0$ +ve Points

$(w^T\phi(x_n) + b) < 0$ -ve Points

$t_n(w^T\phi(x_n) + b) > 0$ for all points

Date: / /

max slope

margin

$$\max_{w} \min_{n} \frac{[w^T\phi(x_n)+b)\,t_n}{\|w\|}$$

such that $(w^T\phi(x_n)+b)\,t_n > 0$

Say plane $2x+3y+5=0 \rightarrow \frac{2}{10}x+\frac{3}{10}y+\frac{5}{10}=0$

Distance from (1,1) $\frac{\frac{2}{10}+\frac{3}{10}+\frac{5}{10}}{\frac{1}{10}\sqrt{13}} = \frac{2+3+5}{\sqrt{13}}$

Here we are able to make 1

w^* – equivalent Plane of w where numerator comes out to be 1 for the margin point.

If we call $2/10+3/10+5/10$ as $w^T(\phi(x_n)+b$, then we get $\max_{w} \min_{n} \frac{1}{\|w^*\|} \rightarrow$ Distance is maximized

Any other point

Margin Point

$$\frac{(w^T\phi(x_n)+b_n)\,t_n}{\|w\|} \geq \frac{(w^{*T}\phi(x_n)+b_n)\,t_n}{\|w\|}$$

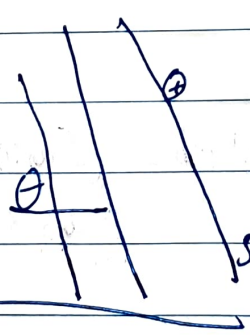
we replace w^T with the w^{*T} such that it because that gives us 1

$$(w^T\phi(x_n)+b)\,t_n \geq (w^{*T}\phi(x_n)+b)\,t_n$$

$$(w^T(\phi(x_n))+b)\,t_n \geq 1$$

Here we take that representation for that line such that distance from margin has min 1

Apparently we take $\pm 1, -1$ on RHS because that would make it easier to move equations left and right



Support Vectors $\rightarrow$ $\boxed{t_n(w^T(\phi(x_n))+b) = 1}$

Date: / /

$$\max \frac{1}{\|w\|^2} \text{ such that } t_n(w^T\phi(x)+b) \geq 1$$

$$\min \frac{1}{2}\|w\|^2 \text{ such that } t_n(w^T\phi(x)+b) \geq 1$$

$\hookrightarrow$ Dual of Previous Expression

When we take lagrange we reverse the operation

$$\text{Lagrange}(a,w,b) = \frac{1}{2}\|w\|^2 - \sum a_n(t_n(w^T\phi(x_n)+b)-1)$$

$$\frac{\delta L}{\delta w} = 0 \rightarrow w - \sum_{n=1}^{N} a_n t_n \phi(x_n) = 0$$

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$\frac{\delta L}{\delta b} = 0 \quad \sum_{n=1}^{N} a_n t_n = 0$$

$$\max \frac{1}{2}\left(\sum_{n=1}^{N} a_n t_n \phi(x_n)\right)^T \left(\sum_{m=1}^{N} a_m t_m \phi(x_m)\right) - \sum a_n \left(t_n \left(\sum a_m t_m \phi^T(x_m)\right)\phi(x_n)\right) - 1$$

$$\max \sum_{n=1}^{N} a_n - b\sum_{n=1}^{N} a_n t_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T\phi(x_m)$$

$$\max \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T\phi(x_m)$$

such that $a_n \geq 0$ and $\sum_{n=1}^{N} a_n t_n = 0$

$$y(x) = w^T\phi(x) + b = \left(\sum a_n t_n \phi(x_n)\right)^T \phi(x) + b$$

$$= \sum_{n=1}^{N} a_n t_n \underbrace{\phi(x_n)^T\phi(x_n)}_{\text{Kernel Function}} + b$$

OK bi

Kernels (again)

The story is like, we try

So that we maximi
equation then we ta
some equations a
max

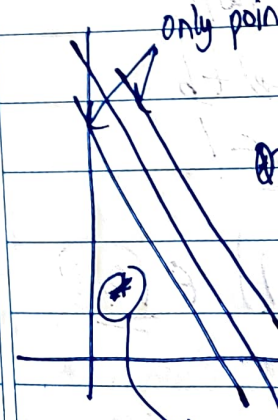
Primal Problem min

max

such that $a_n \geq$

Points on which, the

only poin



Hen

vidyate

Date: / / /

Kernels (again)

NOTE: Don't check questions in exam.

The story is like, we try to find max w for $\frac{1}{\|w\|}$ (min of 1st distances)

so that we maximize margin and so we get the $(w^T\phi(x_n)+b)t_n \geq 1$ equation then we take dual using lagrange's and we get some equations and then since we used lagrange's we take max

Primal Problem $\min \frac{1}{2}\|w\|^2$ such that $t_n(w^T\phi(x_n)+b) \geq 1$

$t_n(w^T\phi(x_n)+b-1 \geq 0$

$$\max \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

such that $a_n \geq 0$ $\sum a_n t_n = 0$ wrt conditions

KKT Conditions

$a_n \geq 0$

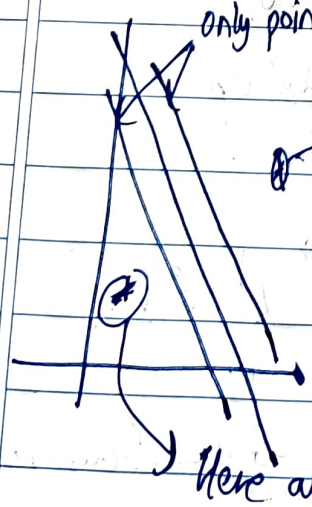
$t_n(w^T\phi(x_n)+b)-1 \geq 0$

$a_n(t_n(w^T\phi(x_n)+b)-1)=0$

Given condition

Points on which the line stays → Support Vectors

only points on the line a_n need not be zero



For this $t_n(w^T\phi(x_n)+b)-1 \geq 0$

So $a_n = 0$

Here also $a_n = 0$

We call sparse Kernel

In the end, Output

$$y = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$$

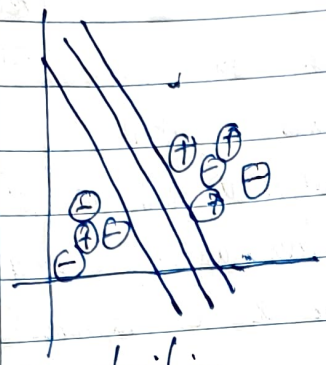
Constants

Kernel function

vidyateja

Date: / /

☆ How SVM works when data is not linearly separable



Max Margin +
Min misclassification

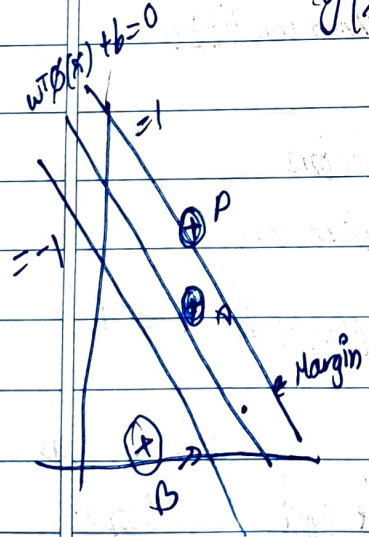
We introduce a penalty based on how bad it is in classifying

Suppose A and B be two the training examples
Define ξ_n as follows

$$\xi_n = \begin{cases} 0 & \text{if data point is on/inside the correct margin boundary} \\ |t - y(x_n)| & \text{for other points} \end{cases}$$

Penalty - how far the point is from the margin.

$$y(x) = w^T\phi(x) + b$$



$$\frac{(w^T\phi(x_p)+b)}{\|w\|} > \frac{(w^T\phi(x_a)+b)}{\|w\|}$$

but P is on $w^T\phi(x_p)+b$

$\therefore w^T\phi(x_a)+b < 1$

$$\xi_A = |1-(\quad)| \in (0,1)$$

On midline - Penalty = 1

$0 < \xi_A < 1 \Rightarrow 0 < w^T\phi(x_a)+b < 1$

$0 < \xi_a < 1$

early separable

rgin ↑
misclassification

penalty
ad it is

example

inside H_0
y

margin.

$+b)$

$(0,1)$

$\phi(x_a)+b<1$

Date: / / /

$y(x_a) \mp 1-\varepsilon_a$ which can be seen as $y(x_a) \geq 1-\varepsilon_a$

Similarly at B, $w^T\phi(x_B)+b \geq 1-\varepsilon_B$

We state that

$$t_n(w^T\phi(x_n)+b) \geq 1-\varepsilon_n \quad \forall \text{ all the training examples}$$

We need to minimize penalization i.e. $\sum_{i=1}^{N} \varepsilon_i$

$$\min \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\varepsilon_n \text{ subject to } t_n(w^T\phi(x_n)+b) \geq 1-\varepsilon_n$$

$\varepsilon_n \geq 0$

Scaling Factor: ↑ → Penalty high → Precise Decision boundary
Else Rough boundary

$$L(w,b,a) = \frac{1}{2}\|w\|^2 + \underbrace{C\sum_{n=1}^{N}\varepsilon_n - \sum_{n=1}^{N} a_n(t_n y(x_n) - 1 + \varepsilon_n) - \sum_{n=1}^{N}\mu_n\varepsilon_n}_{\text{Difference}}$$

$\frac{\delta L}{\delta w} = 0 \qquad w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$

$\frac{\delta L}{\delta b} = 0 \qquad \sum_{n=1}^{N} a_n t_n = 0$

$\frac{\delta L}{\delta \varepsilon_n} = 0 \qquad a_n = C - \mu_n$

KKT Conditions

$a_n \geq 0$

$t_n(y(x_n)) - 1 + \varepsilon_n \geq 0$

$a_n(t_n y(x_n) - 1 + \varepsilon_n) = 0$

$\mu_n \geq 0.$

$\varepsilon_n \geq 0$

$\mu_n \varepsilon_n = 0$

Dual Problem → $\sum_{1}^{N} a_n - \frac{1}{2}\sum_{1}^{N}\sum_{1}^{N} a_n a_m t_n t_m k(x_n, x_m)$

Date: / /

We can find w^* such that

$$w^* = \sum_{1}^{N} a_n^* t_n \phi(x_n)^T$$

↳ Some have $a_n = 0$ if they are correct

Any data point with $a_i = 0$ will not appear in decision boundary

$$\sum_{n \in S} a_n t_n k(x_n, x_n) + b = 0$$

↳ Those points which are misclassified.

$$\text{Margin} = t_n \left(\sum_{n \in S} a_n t_n k(x_n, x_n) + b \right) = 1$$

$$\sum_{m \in S} t_m \left(\sum_{n \in S} a_n t_n k(x_m, x_n) \right) + b \sum_{m \in S} t_m = |S|$$

S → Support Vector Set

Decision

↓

Disjunctions

$-P_1 \log$

P (Tennis

Hi

vn

Max Info

ID3

Algorithm

S

(2+

$\epsilon =$

Date? N / O / FUCK / OFF

Decision Tree

↓

Disjunction of Conjunctions

ANN is not an interpretable model

$-P_1 \log P_1 - P_2 \log P_2 \longrightarrow \sum_{i \text{ all classes}} -P_i \log P_i$

P(Tennis ☺) P(No Tennis ☹)

$0 \leq \underline{\text{Entropy}} \leq \log P$

Highly non uniform — Highly uniform

Max Info gain - highest

ID3 Algorithm

$S = [9+, 5-]$

$E = 0.940$

Outlooks

Sunny $[2+, 3-]$ $E = 0.971$

Overcast $[4+, 0]$ $E = 0.0$

Rain $[3+, 2-]$ $E = 0.971$

$\text{Gain}(S, \text{Outlook}) = -5/14(0.971) - 4/14(0.0) - (5/14)(0.971) + 0.940$

$= 0.247$

↓ Original Entropy

$\text{Gain}(S, \text{Humidity}) = 0.151$ ✓

$\text{Gain}(S, \text{Wind}) = 0.048$

Date: / /

- If 10/15 are noisy examples. With noisy data, tree becomes bigger than expected
- Generalization needs smaller trees

Leads to Overfitting

- How do we counter overfitting in decision tree is an important conversation.
  - ↳ Pruning + take dataset as train + validation

- Rule Post Pruning
  - We check each rule also and then remove if removal leads to higher accuracy

We can build a lot of decision trees and each will have its own tree, and if a testing example comes — we take majority

↑

Random Forest

Probabilistically more efficient

↑

Ensemble of classifiers, but very expensive computationally

tree

Overfitting is

Date / / /

- Positive Classes << Negative Classes - Class imbalance Problem

Ex: Covid, Cancer Patients Data

(a) Oversampling

50 +ve 950 -ve

↳ We repeat these multiple times → 500 samples

(b) Under sampling

50 +ve 950 -ve

↳ We compress → 50 samples

Feature Selection Problem

☆ ID3 Algorithm

- Create root node for tree
- All +ve → +ve
- All -ve → -ve
- Attribute set is empty → Return most common target
- Begin:

  A ← Attribute with highest gain

  Root ← A

  Each value of A → Add branch below Root corresponding to A=v,

  If it goes deeper

  ID3(Attributes - {A})

  Else

  Most common attribute

Date: / /

Important Points

Overfitting
- Low training Error
- High test Error

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

F1 Score

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

- In dual formulation, features only appear as dot products
- Increasing Data reduces variance
- Less training data → Overfit
- Add more layers → bias problem solved

- For small datasets - Naive Bayes > Log reg
  Log reg will overfit
  Larger datasets - Naive Bayes underfits