Books →

1) Machine Learning – Tom Mitchell

2) Pattern Classification – Duda, Hart and Stork.

* 3) Introduction to Machine Learning – E Alpaydin

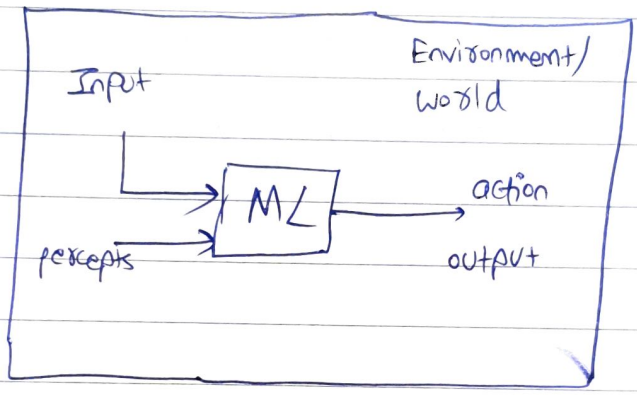
* 4) The elements of statistical learning – Hastie, Tibshirani, Friedman

19th July, 2019

Machine learning

A learning system is one which improves its performance at a task with experience:

Task:



$y = f(x)$

find this f.

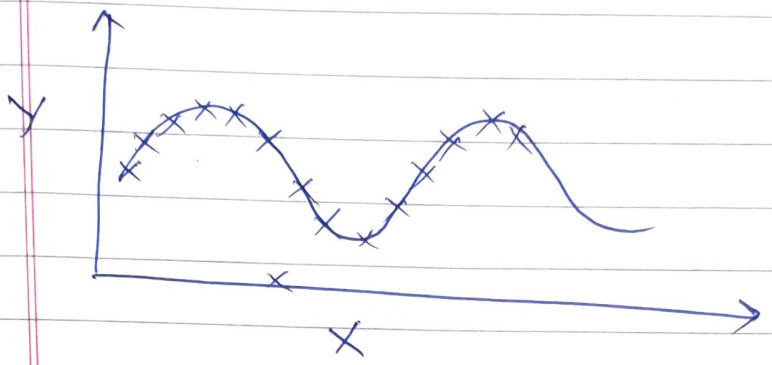
We need a performance measurement to measure the goodness of our fnction f.

Learning is changing our function f when our outputs don't match the expected outputs.

Deductive Knowledge:

general (axioms) → special (theorems)

Inductive reasoning
- special ⟶ general
⟶ Machine learning



Deductive – given a sinusoidal curve, we know how it will be, always a sinusoidal

Inductive – given a set of points (x), we need to fit a curve such that most of these points lie on that.

Types of y (outputs):

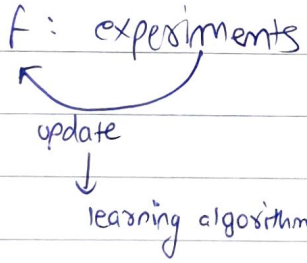
1. According to nature of y:
   - y is binary 0/1.
   - y is a finite set of integers
     1, 2, 3, 4
     A, B, C, D
     malaria, influ...

   ✓ Classification / categorization [into one or more classes]

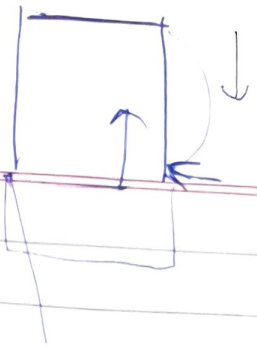
- y is a real number
  - regression problem.

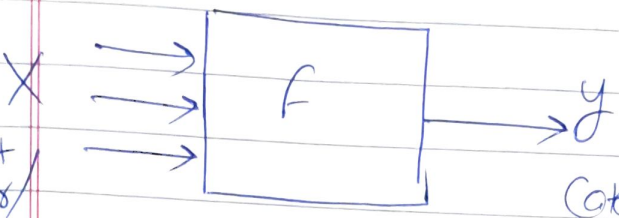
2. According to nature of x:
   - x is a vector (feature)
   - x is a sequence/signal.
   - x is a graph
   - x is a more complex object

3. According to structure of f:
   Architecture of f:



24th July





input vector / feature vector

Category / integers / classes
- classification
binary / k-class

Real
- regression

Training -
supervised learning

x_1, y_1
x_2, y_2
⋮
x_n, y_n

Unsupervised learning / exploratory learning

x_1
x_2
⋮
x_n

Semi-supervised learning

x_1
x_2
⋮
x_n
x_{n+1} y_{n+1}
x_{n+2} y_{n+2}

Reinforcement learning –

$$\begin{cases} X_1 & R(F(X_1)) \\ X_2 & R(F(X_2)) \\ X_3 & \vdots \\ X_n & R(F(X_n)) \end{cases}$$

Learning problem

Given a training set (supervised / unsupervised / reinforcement)

Find f, which generalizes well to a new X.

Optimization problem →
We decide a functional form for f.
Error function – $l(f(x), y)$

$$L = \sum_{i \in TS} l(f(x_i), y_i)$$

Find out f belonging to F, which has minimum loss L on the training set.

The choice of the function f depends on the nature of x, nature of y and how complex the mapping between them is.

If we are constrained by the resources, we would like to find the best possible solution while still being constrained.
eg- what is the best 10 layer neural network?

26th July

~~Predicting the weather on a day by~~

1) Sky Condition : [sunny, cloudy, raining]
2) Air temp [warm, cold]
3) Humidity [normal, high]
4) Wind [strong, weak]
5) Water [cool, warm]
6) Forecast [same, change]

Output Enjoy sports yes, no (binary classification)

Data table →

	Sky	Air Temp	Humidity	Wind	Water	Forecast	Output
an instance →	Sunny	Warm	normal	strong	cool	same	yes

A binary classification problem can be seen as a set learning concept.

C – target concept
h – hypothesis function

$(sky = sunny) \wedge (temp = low)$

$h \in \mathcal{H}$ ⟶ hypothesis space
cardinality of $\mathcal{H} = 4 \times 3^5$
since any feature may or may not be included in the hypothesis.

? – don't care

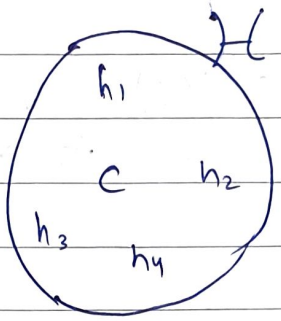
<sunny, ?, normal, ?, cool, same>
this should be read as –

$(sky = sunny) \wedge (humidity = normal) \wedge (water = Cool) \wedge (forecast = same)$

But the form to write ~~attribute~~ hypothesis :⟶
<sunny, ?, normal, ?, cool, same>

In a 6-D space, the hypothesis becomes a set as the don't cares can change about.

1) We ~~asso~~ make an assumption that the target concept $c \in H$ [Realisability assumption]



[Inductive hypothesis assumption]

2) If c matches h for all the training set, it will match for all x.

In other words, the learning set has reduced ~~with all~~ to a search problem.

A hypothesis for which c matches h over the training set, it is called a consistent h.

A consistent hypothesis is THE actual hypothesis, according to inductive hypothesis.

Given a set of Instances, find the consistent hypothesis from the family of hypothesis space.

$$T - \text{training} = \{x_1, x_2, \ldots, x_n\}$$

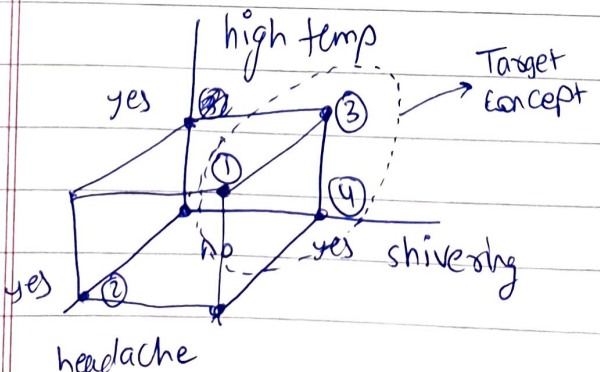
if $h(x_i) = c(x_i) \quad \forall x_i \in T$

then $h(x) = c(x) \quad \forall x$

31st July 2019

High temp	Head ache	shivering	Outcome malaria
Y	Y	Y	Y
N	Y	N	N
Y	N	Y	Y
N	N	Y	Y

Attribute space / feature space



① $(\text{high temp} = Y \wedge \text{headache} = Y \wedge \text{shivering} = Y)$

one possible formula, not the correct one.

② $(\text{shivering} = Y)$

3^3 possible hypothesis – h ⟶ hypothesis space $\mathcal{H}$

The target concept will not ~~ever~~ always be part of our hypothesis space.
If the target concept IS a part of the hypothesis space $\mathcal{H}$, then the target concept is realizable.

If $c \in \mathcal{H}$, then c is realizable using $\mathcal{H}$.

Instance set:
set of given examples (instances) with their labels is called instance set.

High temp	Head-ache	Shivering	malaria
Y	Y	Y	Y
N	N	N	N
Y	N	Y	Y

C_1 C_2 C_3 $\left\{ \begin{array}{l} \langle \text{high temp} = Y \wedge \text{shivering} = Y \rangle \\ \langle \text{high temp} = Y \rangle \\ \langle \text{shivering} = Y \rangle \end{array} \right\} \vee$

$C_1, C_2, C_3 \subset \mathcal{H}$

All these C_1, C_2, C_3 are consistent. ~~with the~~

$V \in \mathcal{H}$ is called the version space of $\mathcal{H}$, w.r.t instance set I.

Q. Set of all consistent hypothesis = version space

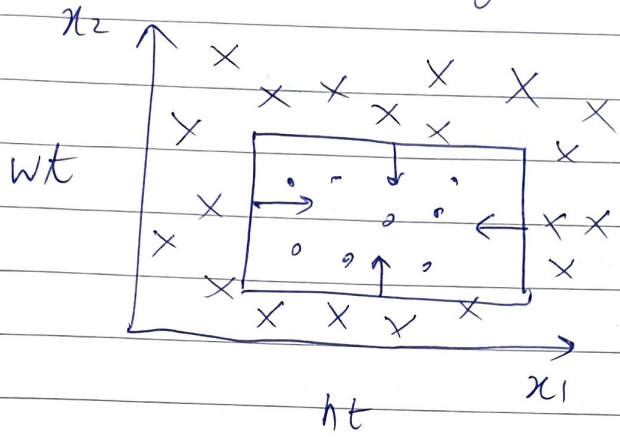
Given an instance set I and a hypothesis space $\mathcal{H}$ find a hypothesis $h^* \in \mathcal{H}$ which is the closest approximation of target concept c.

Learning problem

classmate
Date
Page

Inductive principle:

Any consistent hypothesis is a good approximation of the target concept if the training space is large and realizability holds.



---

high temp headache shivering malaria.

N	Y	N	N
Y	N	Y	Y
N	N	Y	Y

high temp = Y ∧ head ache = ~~X~~ ? ∧ shivering = Y | h_1

hight = ~~X~~ ? ∧ shivering = Y | h_2

shivering = Y | h_3

1st August

Training examples for Enjoy Sport –

Learning means learning a subset of the features space.

- If formula f1 satisfies more examples than formula f2 then f1 is said to be more general than f2.

FindSpecific algorithm:

We start with a very specific hypothesis and then make the minimum incremental changes so that the resulting hypothesis satisfies our positive examples.

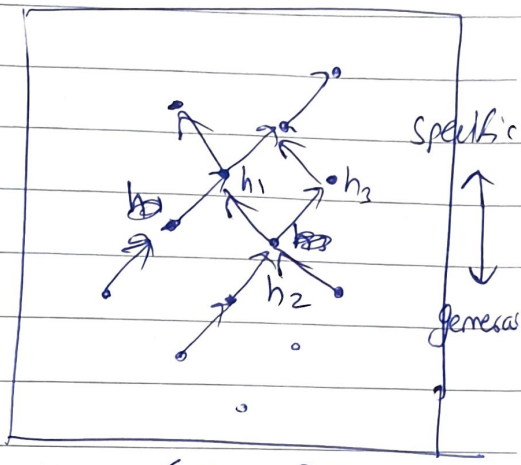
Using this algorithm, we obtain the most specific hypothesis which is consistant with the given training examples.

Sunny	Warm	Normal	Strong	Warm	Same	Yes
Su	W	High	S	W	S	Y
Rainy	Cold	H	S	W	Change	No
S	W	H	S	Cold	C	Y

Wind = Strong

classmate
Date
Page

hypothesis H



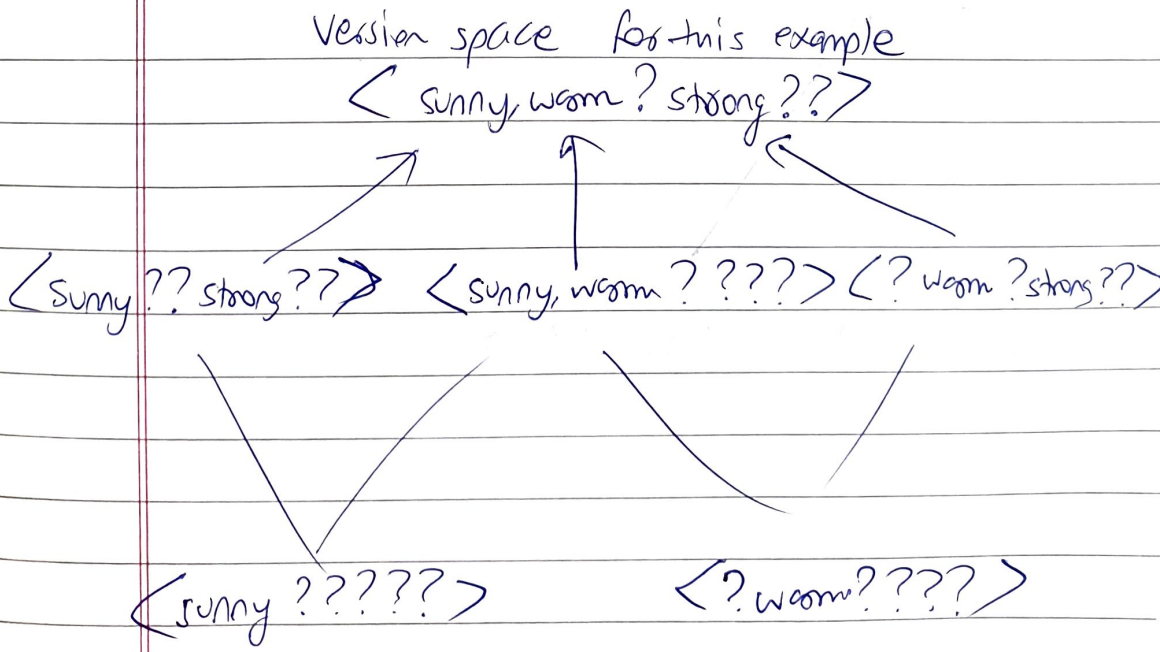
h_1 = < Sunny ?? Cool ?? >

h_2 = < Sunny ?? ? ?? >

h_3 = < Sunny ? ? ? Humid >

general ⟶ specific

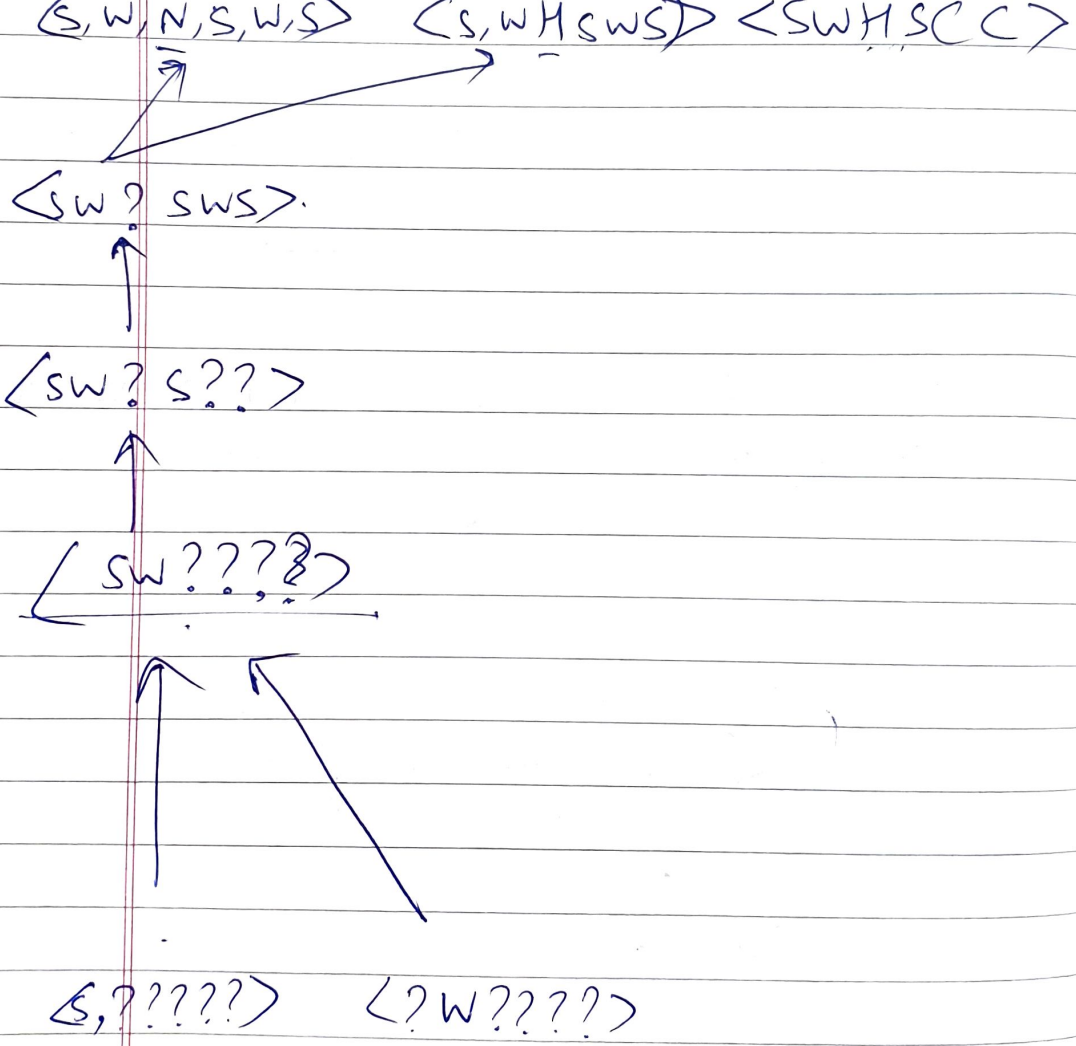
The FindSpecific algorithm will go down the lattice (depending on the order of training examples faced) and stops at the least general sol^n satisfying all examples.



(1) We are ensuring only the positive examples amount to a positive result.

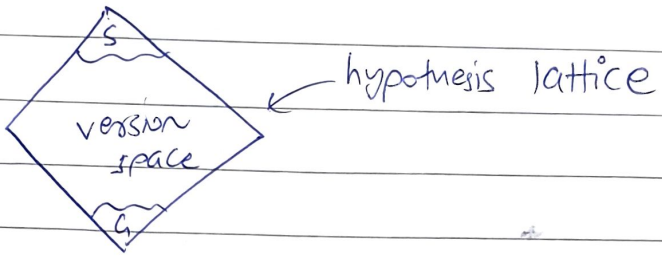
(2) We are ensuring this for all positive examples

(1) & (2) implies → all negative examples will have negative output



2nd August

To find all members of the version space, it is enough to find the most specific consistent example and the most general consistent example.

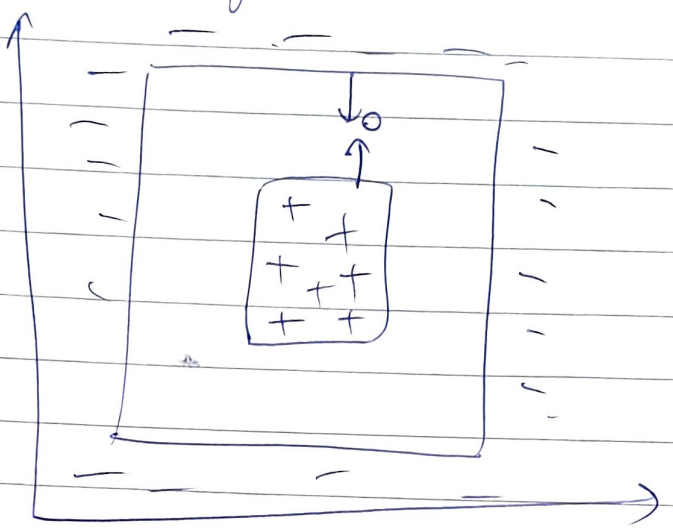


Candidate elimination algorithm →

G ← maximally general hypothesis in H
S ← maximally specific hypothesis in H

For each training example d, do
If d is positive
- remove from G every hypothesis inconsistent with d
- for each hypothesis s in S that is

To minimise the version space, we will choose a new training example which is most ambiguous.



Since we don't know if the new training example is +ive or -ive, so we choose the new example is chosen at the middle.

If there is no bias (in our example, the knowledge that the hypothesis is purely conjunctive), then we will require infinite training examples.

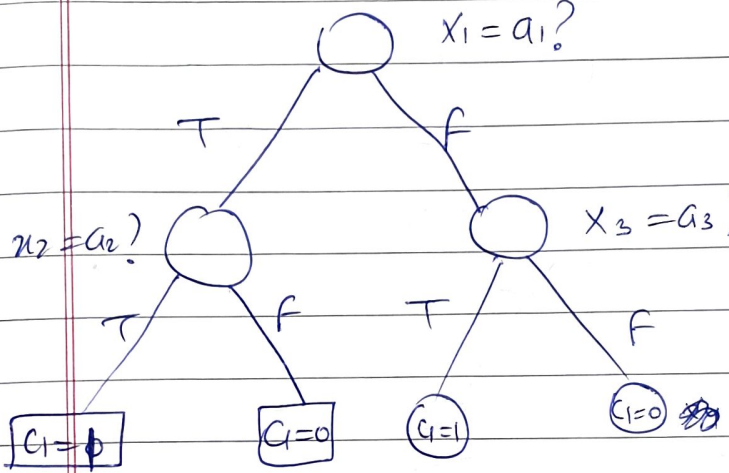
7th August '19

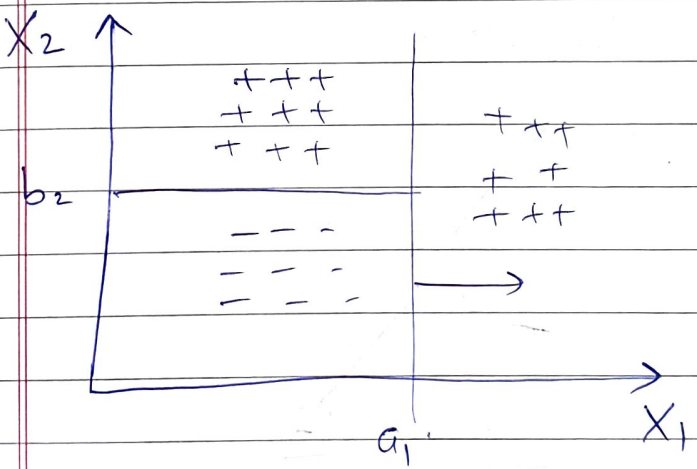
Machine learning

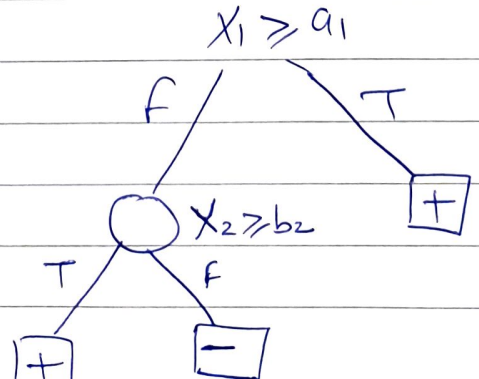
$(X_1 = a_1) \wedge (X_2 = a_2) \wedge (X_3 = a_3)$

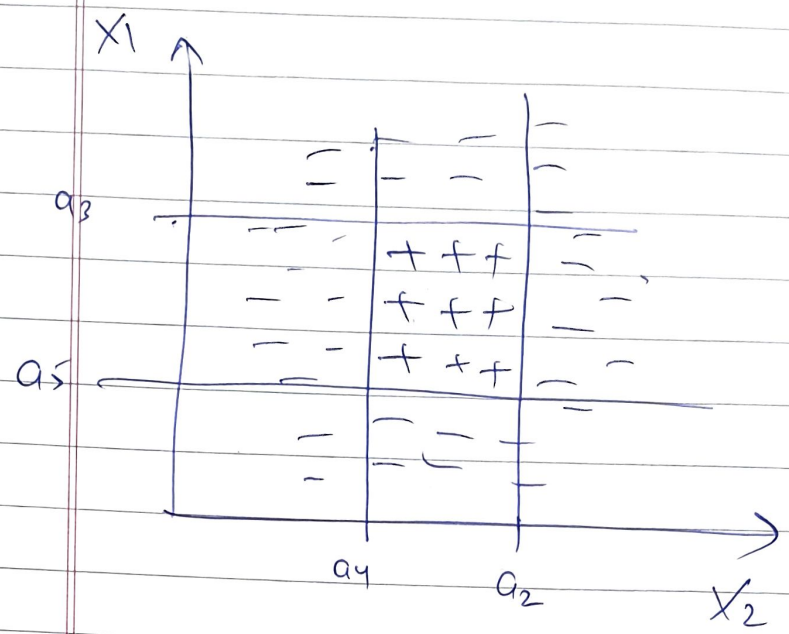
$(X_1 = a_1 \wedge X_2 = a_2) \vee (X_3 = a_3)$

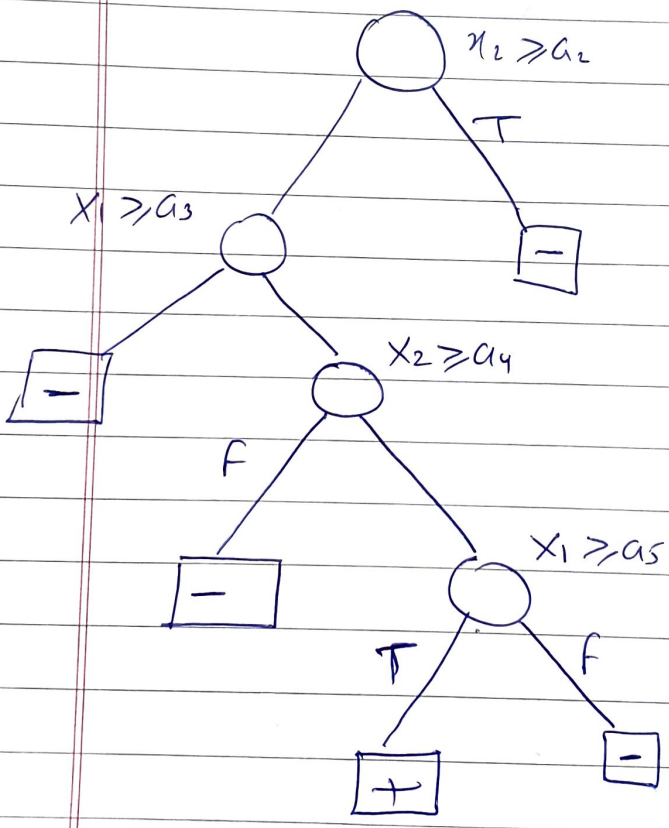
Decision tree classifier →







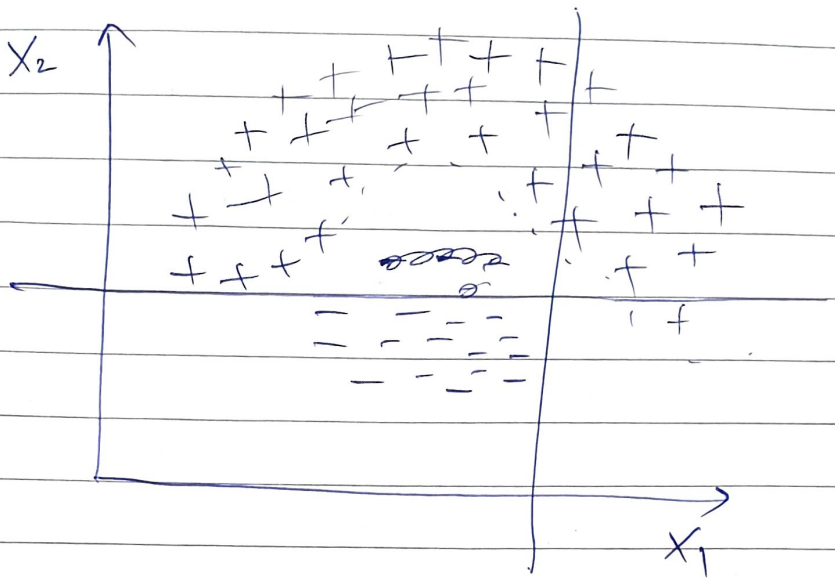




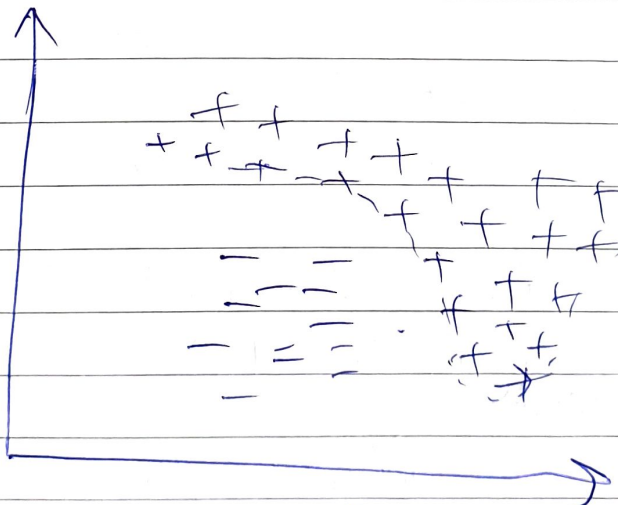
* Each node partitions the input space into 2 classes.

* We can ~~draw~~ obtain multiple decision trees for the same decision problem.

We would like to find the smallest decision tree.



* We would want to recursively divide our resultant data-set into pure and non-pure partitions.



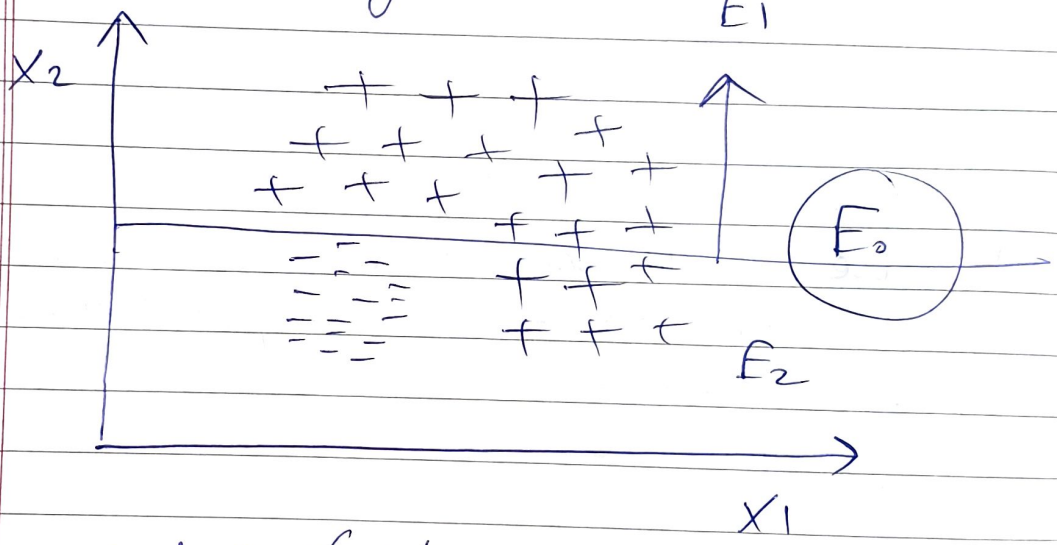
$$P_+ = \frac{n_+}{n_+ + n_-} \qquad P_- = \frac{n_-}{n_+ + n_-}$$

Measure of purity of classes ⟹ entropy

Entropy $= -p_+ \log p_+ - p_- \log p_-$

at $p_+ = 1$ and $p_- = 0$
or $p_+ = 0$ and $p_- = 1$

then entropy $= 0$



goodness of cut
= reduction in entropy

$= E_0 - (E_1 + E_2)$
$= -((E_1 + E_2) - E_0)$

information gain

We greedily choose the cut with maximum information gain.

8th August, '19

<F, I, CS, M> <F, I, EE, M>

<F, I, ?, M>

<F, ?, ?, M>.

G = {<?, ?, ?, ?>}

S = {<F, I, CS, M>, <F, I, EE, M>, <F, I, CS, M>, <F, I, ?, M>, <F, ?, CS, M>, <F, I, ?, M>, <F, ?, ?, M>, <F, ?, ?, M>, <F, ?, ?, M>}

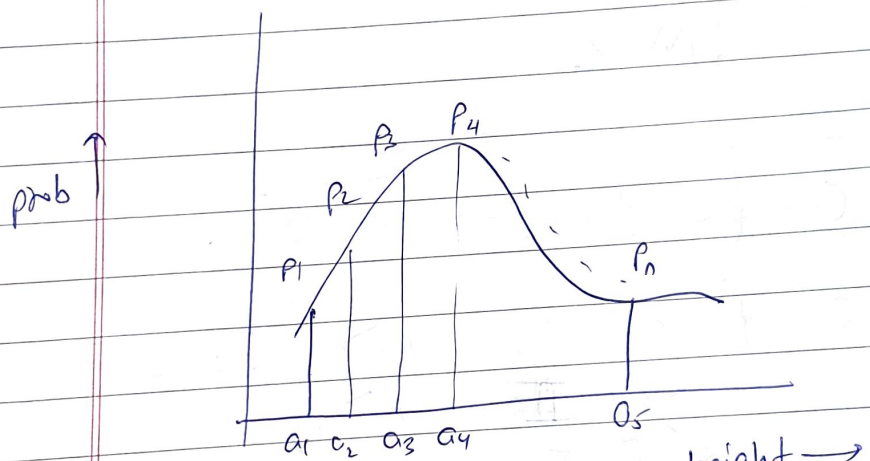
G = <?, ?, ?, M> <F, ?, ?, ?>

9th August, Friday

Information gain

⊙ Gain (S, A) : reduction in entropy after choosing attribute A.

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



$$entropy = -\sum_{i=1}^{n} P_i \log(P_i)$$

This greedy approach will not result in a decision tree of shortest length.

But ~~since~~ our algorithm is biased towards shorter trees.

* Occham's razor principle – Use the minimal instrument required to solve the problem.

Methods ~~of~~ to avoid overfitting –

* Pre-pruning
* Post-pruning

* Methods to evaluate trees to prune –

Gini index

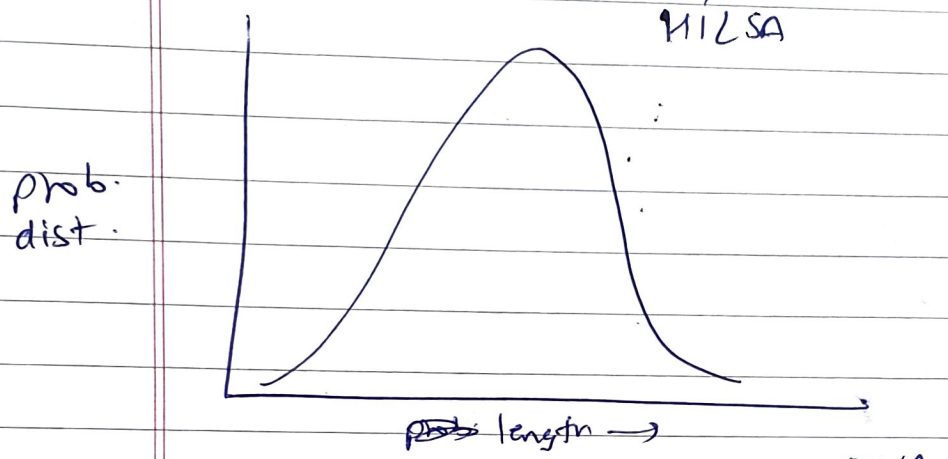
Another sensible measure of impurity

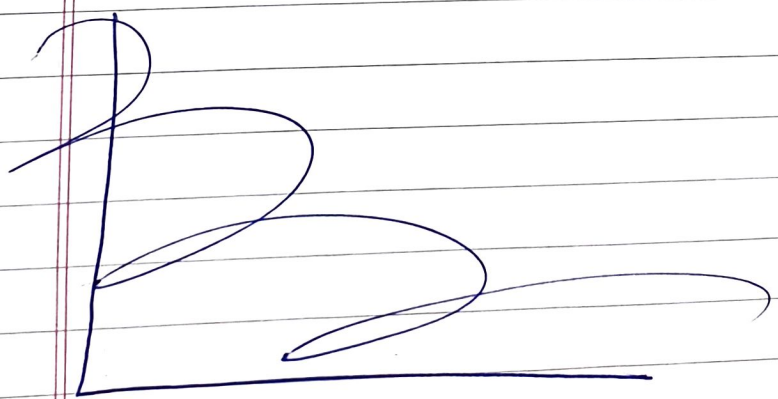
$$Gini = \sum_{i \neq j} p(i)\,p(j)$$

14th August, 2019

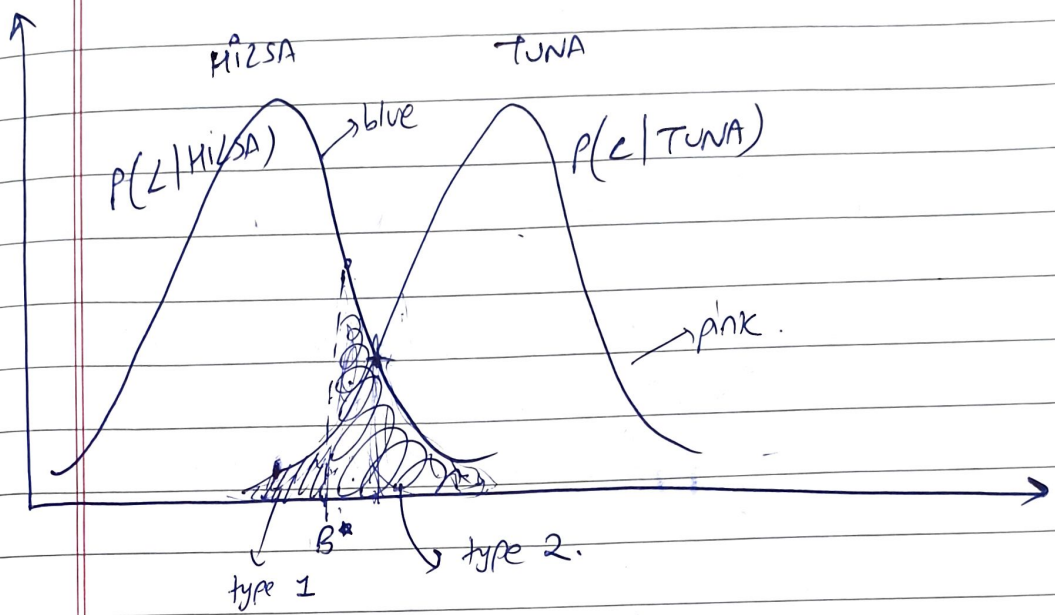
A simple species classification problem

Measure the length of ~~this~~ a fish and decide its class.





Error of decision rule -



Type 2 → Actually HILSA, classified as TUNA

Type 1 → Actually TUNA, " " HILSA.

B^* → threshold value

Minimum possible error

$$P(B^* | HILSA) = P(B^* | TUNA)$$

If type I error and type II error are different, boundary shifts.

Location / time of experiment.

In Calcutta → more hilsas

In California → more TUNAS.

Apriori probability

If we were to guess the type of fish in CCU, we would guess hilsa.

Similarly, in California, we would guess Tuna.

We multiply our ~~conditional~~ class conditional probabilities with our prior probabilities.

Posteriori ≈ apriori × class probability

Apriori

P(Hilsa)

P(Tuna)

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

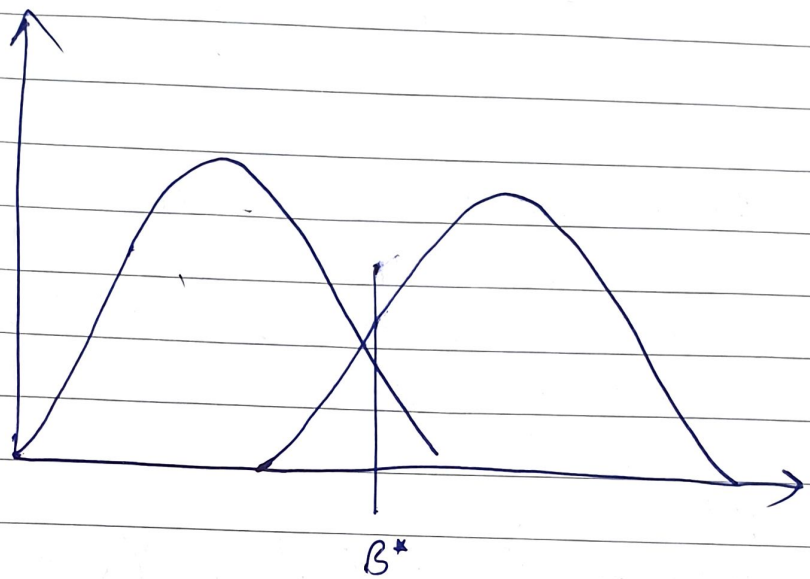
Aposteriori

$$\frac{P(L=2ft|Hilsa)\,P(Hilsa)}{P(L=2ft)}$$

$$= P(Hilsa \mid L=2ft)$$

$$\frac{P(L = 2ft \mid Tuna)\, P(Tuna)}{P(L = 2ft)}$$

$$= P(Tuna \mid L = 2ft)$$



Bayes optimal classifier

$$\text{Aposterior} = \frac{P(L \mid Hilsa)\, P(Hilsa)}{P(L)}$$

$P(L)$ → this does not change the boundary point since it is just a constant in both Hilsa and Tuna. So it is generally dropped.

15th August, 19

TM

MAP classifier

Entropy

$$H(X) = -\sum_{i=1}^{n} P(X=i) \log_2 (P(X=i))$$

Entropy $\propto$ impurity of data.

Specific conditional entropy of $H(X|Y=v)$ of X given $Y=v$

$$H(X|Y=v) = -\sum_{i=1}^{n} P(X|Y=v) \log P(X|Y=v)$$

Conditional entropy $H(X|Y)$ of X given Y:

$$H(X|Y) = \sum_{v \in values(Y)} P(Y=v) H(X|Y=v)$$

Information gain = reduction in entropy

$$= H(X) - H(X|Y)$$

16th August, 19.

Bayes classifier

EXCELLENT classmate Date Page FOR
small dataset,
~~Bayes classifier~~
large features.

$P(A \wedge B) = P(A) \cdot P(B)$ if $A \perp B$.

if $A \perp B \rightarrow$ then joint prob = prod. of marginals

$P(A_1 \wedge B \mid M) = P(A \mid M) \cdot P(B \mid M)$
if $A \perp B$.

There is no way to know if the attributes A and B are independent. We assume them suitably using our world knowledge.

~~give bijths~~

$A : [a_1, a_2, a_3, a_4] \Rightarrow$ [$a_1, a_2, a_3, a_4 \rightarrow$ independent features]

$P(A \mid M) = P(a_1 \mid M) \cdot P(a_2 \mid M) \cdot P(a_3 \mid M) P(a_4 \mid M)$

$P(A \mid M) P(M) > P(A \mid N) P(N)$

Often, we compare the log of conditional probabilities instead of comparing the actual conditional probabilities.

Smoothening –

If one of the conditional probability is 0 then the entire expression becomes 0.

Prob. estimation –

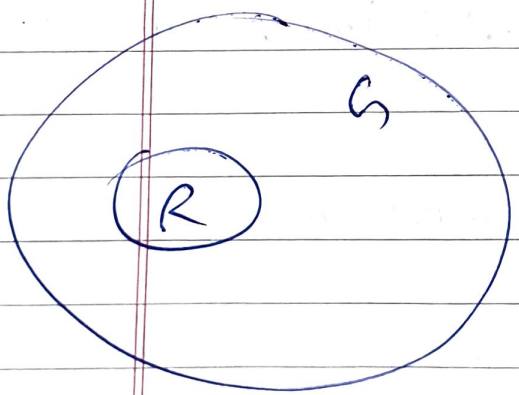
Original: $P(A_i|C) = \frac{N_{ic}}{N_c}$

Laplace: $P(A_i|C) = \frac{N_{ic}+1}{N_c+C}$ C: no. of classes

p: prior prob.

m-estimate: $P(A_i|C) = \frac{N_{ic}+mp}{N_c+m}$ m: parameter

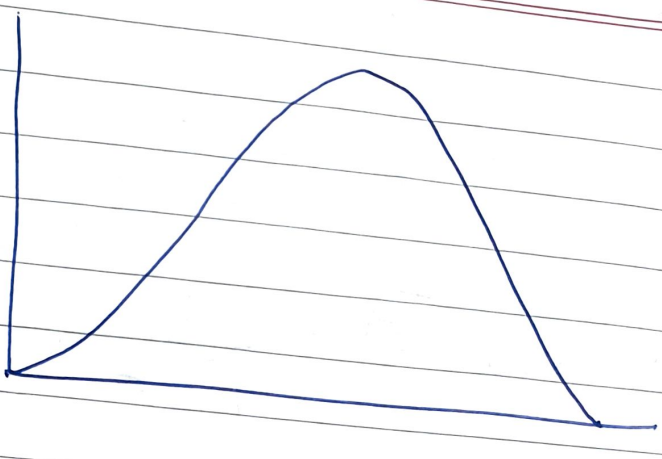
Shrinkage –



$R \subseteq G$

$\hat{P}(R) = \lambda \hat{P}(R) + (1-\lambda)\hat{P}($

$\hat{P}(G)$



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

Q. find the decision boundary when $N(x_1, \mu_1, \sigma_1)$ and $N(x_2, \mu_2, \sigma_2)$

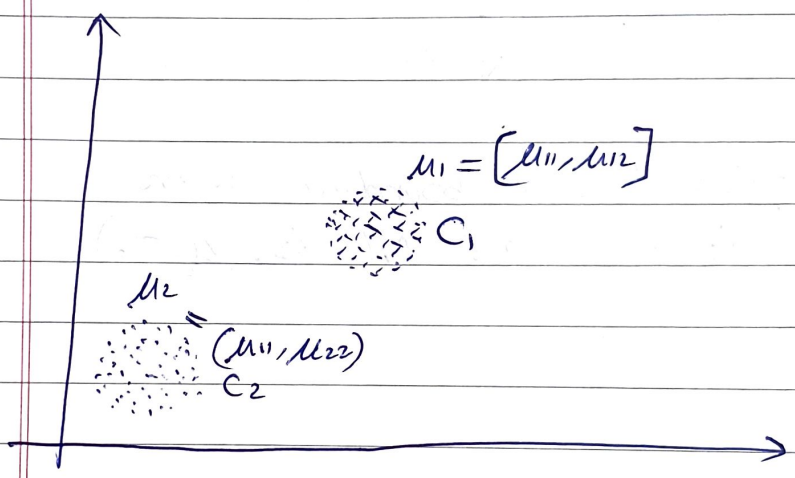
21st August, 2019

Two class Bayes classifier

Each class is a multivariate gaussian.

$$N(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$\mu =$



Covariance matrix (Σ)

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

for d-dimensional case.

$X_1 = x_{11}, x_{12}, \ldots, x_{1d}$ } C_1

$X_2 = x_{21}, x_{22}, \ldots, x_{2d}$ }

$\vdots$

$X_n = x_{n1}, x_{n2}, \ldots, x_{nd}$ } C_2

$$\mu_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i \;, \quad x_i \in C_1$$

$$= E(x_i) \;, \quad x_i \in C_1$$

~~$\sigma_{11} = E\left[(x_{i,1} - E(x_{i,2}))^2\right]$~~

~~$\sigma_{11} = E\left[(x_{i,1} - E(x_{i,1}))^2\right]$~~

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd} \end{bmatrix}_{d \times d}$$

for 2 dimensional case:-

$$\sigma_{11} = E\left[(x_{i,1} - E(x_{i,1}))^2\right]$$

$$\sigma_{22} = E\left[(x_{i,2} - \mu_2)^2\right]$$

$$\sigma_{12} = E\left[(x_{i,1} - \mu_1)(x_{i,2} - \mu_2)\right]$$

$$= E[x_{i,1} \cdot x_{i,2}] - E[x_{i,1}] \cdot E[x_{i,2}]$$

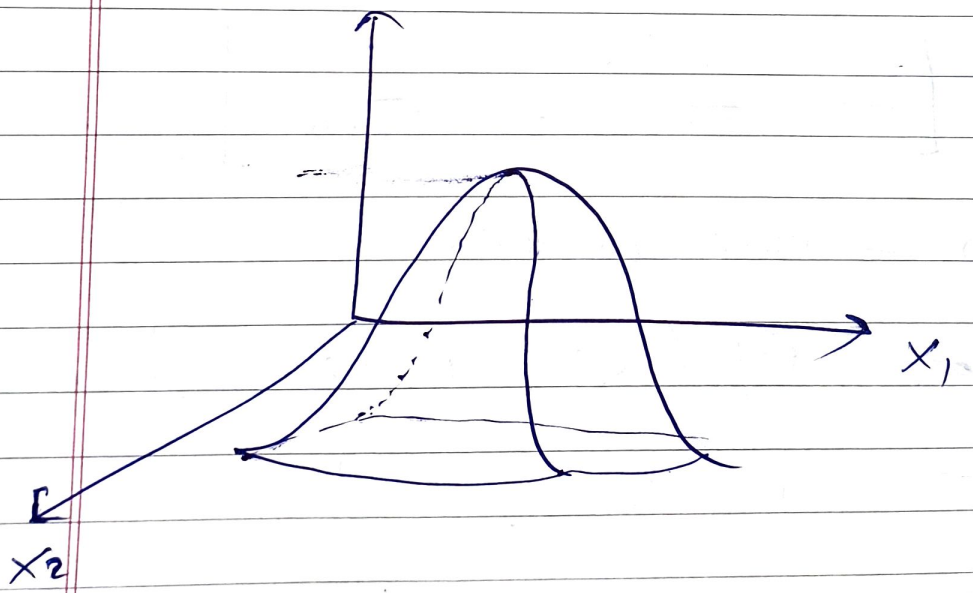
if σ_{12} → if i increases, then 2 increases is high

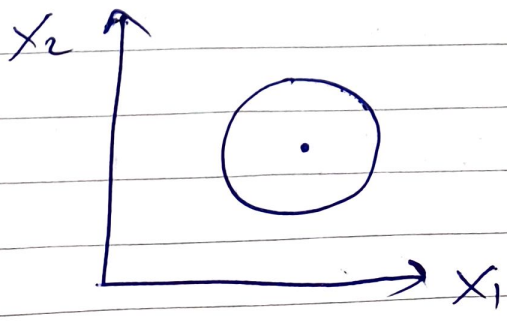
$$\frac{\text{Cov}(X_1, X_2)}{\text{Var}(X_1)\,\text{Var}(X_2)} = \rho$$

$$N(X, \mu, \Sigma)$$

$$\mu = [\mu_1, \mu_2]$$

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$ → always symmetric

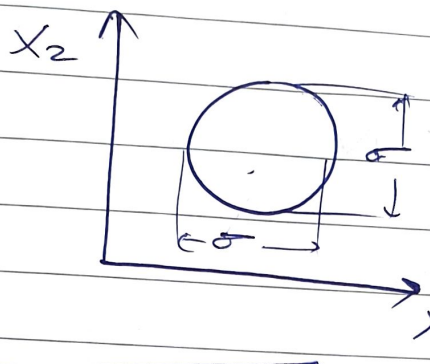




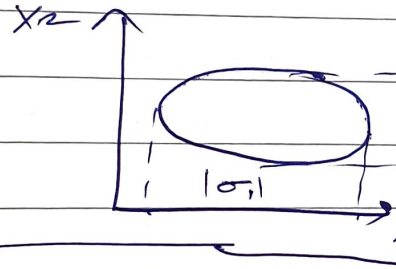
$$\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$

σ_{11} → dispersion along X_1 direction

σ_{22} → ~~direction~~ dispersion along X_2 direction.



$$\Sigma = \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \sigma_{22} \end{bmatrix}$$



axis parallel elipse.

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$$

Bayes classifier → (posterior prob. distribution)

$$C_1 = P_1(X) = N(X_\theta, \mu_1, \Sigma_1)$$

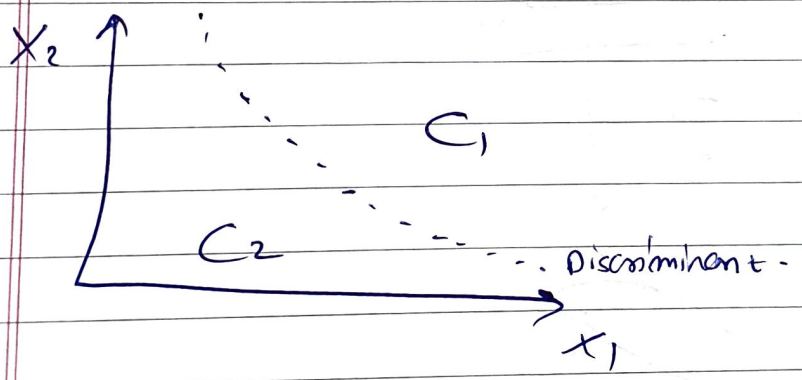
$$C_2 = P_2(X) = N(X, \mu_2, \Sigma_2)$$

log likelihood ratio.

$$\frac{P_1(X)}{P_2(X)} = \log\left[\frac{N(X_\theta, \mu_1, \Sigma_1)}{N(X_1, \mu_2, \Sigma_2)}\right]$$

$$\text{sign}\left[\log \frac{P_1(X)}{P_2(X)}\right]$$

if +ive → class 1
-ive → class 2



$$C_1 = \frac{1}{\sqrt{2\pi}} \Sigma_1^{-1} e^{-\frac{1}{2}(X_\theta - \mu_1)\Sigma_1^{-1}(X - \mu_2)}$$

$$C_2 = \frac{1}{\sqrt{2\pi}} \Sigma_2^{-1} e^{-\frac{1}{2}(X - \mu_2)\Sigma_2^{-1}(X - \mu_1)}$$

$$-\frac{1}{2}(x-\mu_1)(\Sigma_1\Sigma_1^T)^{-1}(x-\mu_1)^T$$

$$C_1 = \frac{1}{\sqrt{2\pi}}(\Sigma_1\Sigma_1^T)^{-1}\, e$$

$$-\frac{1}{2}(x-\mu_2)(\Sigma_2\Sigma_2^T)^{-1}(x-\mu_2)^T$$

$$C_2 = \frac{1}{\sqrt{2\pi}}(\Sigma_2\Sigma_2^T)^{-1}\, e$$

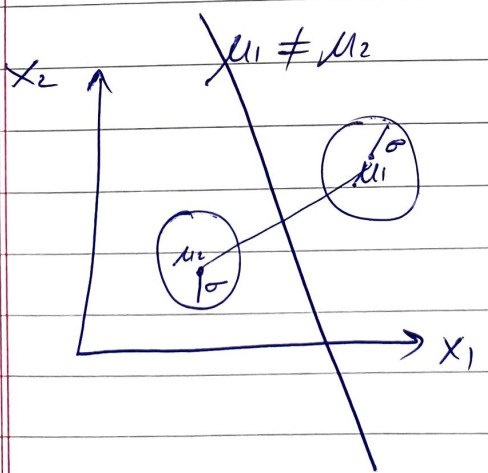
$$C_1 = C_2$$

$$\frac{-1}{2}(x-\mu_1)(\Sigma_1\Sigma_1^T)^{-1}(x-\mu_1)^T + \log\frac{1}{\sqrt{2\pi}} + \log(\Sigma_1\Sigma_1^T)^{-1}$$

$$= \frac{-1}{2}(x-\mu_2)(\Sigma_2\Sigma_2^T)^{-1}(x-\mu_2)^T + \log\frac{1}{\sqrt{2\pi}} + \log(\Sigma_2\Sigma_2^T)^{-1}$$

for Case 1:

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$



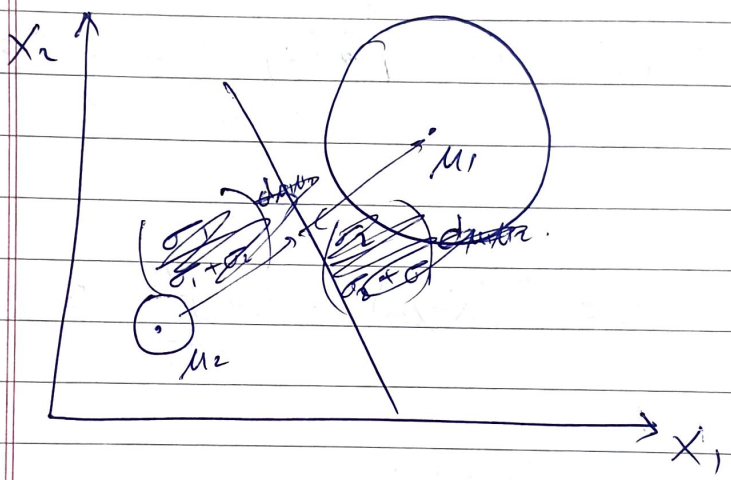
$$|x-\mu_1|^2 = |x-\mu_2|^2$$

Case 2:

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_1 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} \sigma_2 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

$\mu_1 \neq \mu_2$.

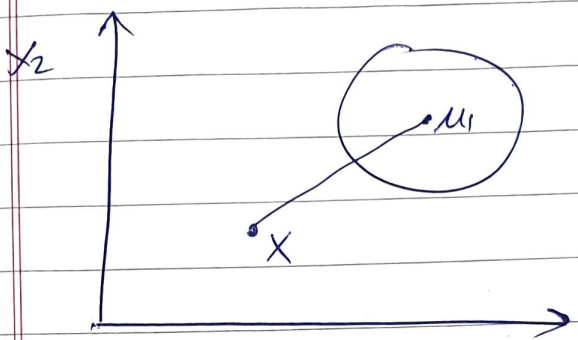


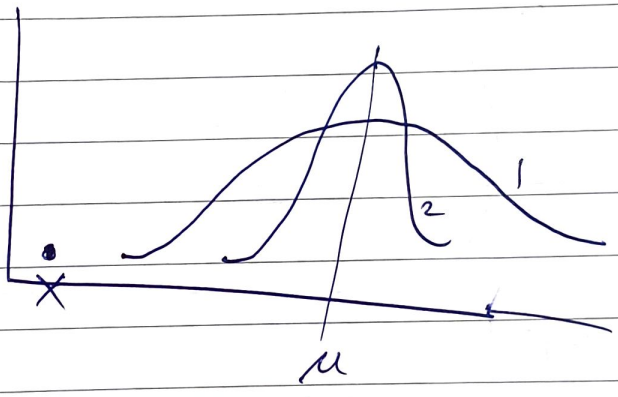
$$\left(\frac{X_1 - \mu_1}{\sigma_1}\right)^2 = \left(\frac{X_2 - \mu_2}{\sigma_2}\right)^2$$

Distance b/w points →

x_1 ——— x_2

Distance b/w a point and a distribution.





Both distributions have same μ but we can see that 2 is closer.

So using only mean is not a good metric

So we use Mahalanobis distance —

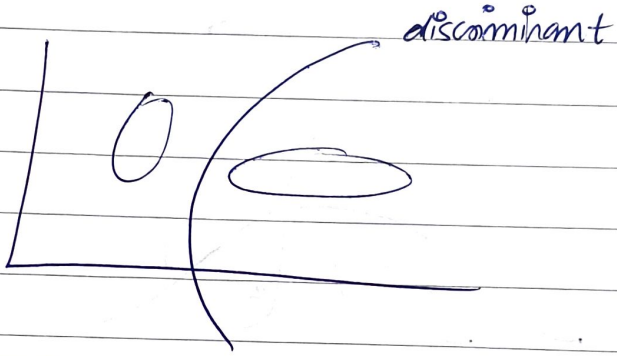
$$\frac{X_a - \mu_1}{\sigma_1}, \quad \frac{X - \mu_2}{\sigma_2}$$

Case 3:

$$\Sigma_1 = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} & \\ & \end{bmatrix}$$

Quadratic discriminant



22nd August 2019

Bayes Classifier

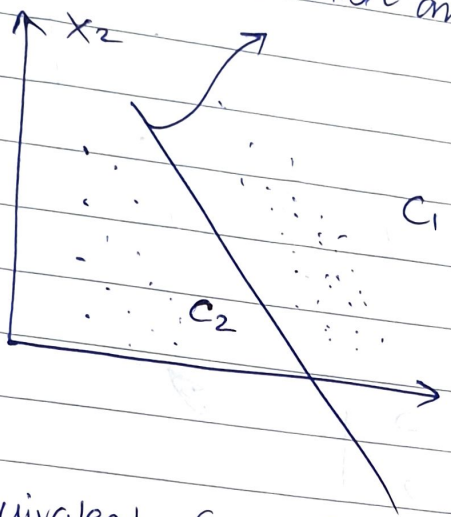
Maximum aposteriori classifier

if $P(X|C_1)\,P(C_1) > P(X|C_2)\,P(C_2)$

$\longrightarrow$ class $= C_1$ for X

o/w. $= C_2$

Discrimination function and class boundary



Equivalent form of $P(x|C_1)P(C_1) > P(x|C_2)P(C_2)$ is

$$\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)} > 1 \longrightarrow C_1$$

else $\longrightarrow C_2$.

take log

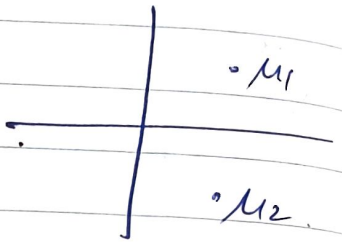
$$\frac{P(x|C_1)}{P(x|C_2)} < \frac{P(C_2)}{P(C_1)}$$

$$\log\left(P(x|C_1)P(C_1)\right) - \log\left(P(x|C_2)P(C_2)\right) > 0$$

$\log$ likelihood ratio.

$X = [X_1, X_2]$

$P(C_1) = P(C_2) = 0.5$



$\mu_1 = [\mu_{11}, \mu_{12}] = [3, 3]$

$\mu_2 = [\mu_{21}, \mu_{22}] = [3, -3]$

$\Sigma_1 * \Sigma_1^T = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$

$\Sigma_2 * \Sigma_2^T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

$P(X_1|C_1) \sim N(X_1, \mu_1, \Sigma_1)$

$P(X_2|C_2) \sim N(X_1, \mu_2, \Sigma_2)$

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

$$(\Sigma_1 \Sigma_1^T)^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix}$$

$$= \frac{1}{\sqrt{2\pi}\,|\Sigma_1|} e^{-(X-\mu_1)^T \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} (X-\mu_1)^T}$$

$1 \times 2 \quad 2 \times 2 \quad 2 \times 1$

$$= \frac{1}{\sqrt{2\pi}\,(|\Sigma_1| = 1)}\, e^{-\left[X_1-3 \;\; X_2-3\right]\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} X_1-3 \\ X_2-3 \end{bmatrix}}$$

$$= \frac{1}{\sqrt{2\pi}}\, e^{-\left[2(X_1-3) \;\; \frac{1}{2}(X_2-3)\right]\begin{bmatrix} X_1-3 \\ X_2-3 \end{bmatrix}}$$

$$= \frac{1}{\sqrt{2\pi}}\, e^{-\left[2(X_1-3)^2 + \frac{1}{2}(X_2-3)^2\right]}$$

$$= \frac{1}{\sqrt{2\pi}}\, e^{-\left[2(X_1-3)^2 + \frac{1}{2}(X_2-3)^2\right]} \checkmark$$

On equating prob dist. of N_1 and N_2.

$$(X-\mu_1)(\Sigma_1\Sigma_1^T)(X_1-\mu_1)^T + \log|\Sigma_1| =$$

$$(X-\mu_2)(\Sigma_2\Sigma_2^T)(X_2-\mu_2)^T + \log|\Sigma_2|$$

$$X^2\left((\Sigma_1\Sigma_1^T)^{-1} + (\Sigma_2\Sigma_2^T)^{-1}\right) + X\left(-\mu_1(\Sigma_1\Sigma_1^T)^{-1} - \mu_1(\Sigma_2\Sigma_2^T)^{-1}\right)$$

$$W_1X^2 + W_2X + W_0 = 0.$$

W_1?

W_2?

W_3?

$(X_0 - \mu_1)\Sigma_1^{-1}(X - \mu_1)^T$ → mahalanobis distance between X and $N(X, \mu_1, \Sigma_1)$

$X = [x_1, x_2]$

$\mu_1 = [3, 4]$

$\mu_2 = [2, -4]$

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

What is the discriminant function obtained by Bayes classifier?

We find the decision boundary so that we don't have to evaluate the probabilities for every test example.

Steps :—

1. Assume a distribution
2. Estimate parameters
3. Find decision boundary.

30th August, 2019

Classifier evaluation

- parameter tuning
- comparison

- defining a score function
- estimating a score function
- comparing score function of two classifiers.

Define score function

Confusion matrix [with test set]

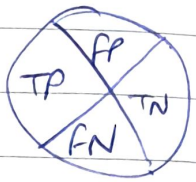
C-class problem

predictions \ actual labels	c_1	c_2	.	.	.	c_c
c_1						
c_2				n_{ij}		
.						
.						
.						
c_c						

cost of ~~mis~~ classifying an i-class example as a j-class example.

$= b_{ij}$

predictions \ actual	P	N
P	TP	FP
N	FN	TN



$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

(specificity) $$\text{Precision} = \frac{TP}{TP + FP}$$

only getting in getting most actually +ive of our +ive predictions to be actually true.

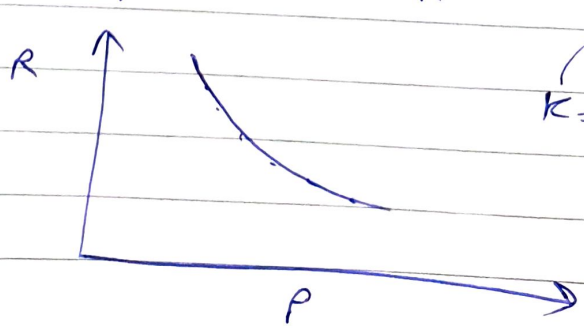
~~only care about correctly classifying +ive examples~~

$$\text{Recall (sensitivity)} = \frac{TP}{TP + FN}$$

most are classified as +ive

F - score ⇒ harmonic measure of precision, recall

$$\frac{2}{F} = \frac{1}{P} + \frac{1}{R}$$



curve which has a larger area under the ROC curve is considered to be a better classifier.

$$P_{correct} = E(Accuracy) \Big|_{|s| \to \infty}$$

s – test sample

$$\approx \hat{} \; Accuracy_{|s|=n < \infty}$$

How do we efficiently estimate E(Accuracy) using a small test set?

Estimation techniques –

- bootstrap
- hold out
- k-fold cross validation
- Leave out one estimate.

Bootstrapping

$\hat{a}$, $\hat{a}$

Instead $\hat{a}_1'$, $\hat{a}_2'$, $\hat{a}_3'$, ..., $\hat{a}_K'$

$(1 - \bar{a}_i)$ – bias

$\sigma(a_i)$ – variance

⊙ Bias is a better estimate of a than individual samples.

[Wednesday]

4th September 2019

Classifier evaluation

- Score definition
- Estimating the score

Score definition

Confusion matrix

predicted \ actual	1	0
1	TP	FP
0	FN	TN

$n_1 + n_2 = n$

$$\text{accuracy} = \frac{TP + TN}{n}$$

[as num_training_ex- accuracy denotes the prob. of a correct classification of a re test data

$n_1 \gg n_2 \Rightarrow$ Recall $= \frac{TP}{n_1}$

$n_2 \gg n_1 \Rightarrow$ Precision $= \frac{TP}{TP + FP}$

Accuracy tends to have a high-variance over separate test datasets, which is undesirable.

Cross-validation

Experiment 1 : a_1
2 : a_2
⋮
K : a_K

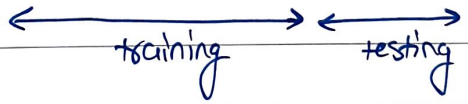
$\bar{a}$ ↑
$\sigma(a)$ ↓

Given a small training dataset, we would like to repeat our experiment several times to gauge a good estimate of the accuracy.

⇒ We resample our population K-times.

- Holdout

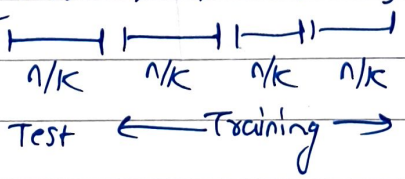
$\{x_1, x_2, x_3, \ldots, x_n\}$



- Bootstrapping
  Repeats hold-out K times [Lots of random selections have to be made]

- K-fold cross validation [Groups have to be made only once, no randomness involved]

$\{x_1, x_2, x_3, \ldots, x_n\}$



Every batch gets to be the test data once while the other batches act as training data.

Error = bias + variance

All classifiers fight the trade-off between accuracy and variance.

Probabilistic graphical model

Bayesian network.

X_1	X_2	X_3	X_4	y
1	0	0	1	1
0	0	0	1	0
1	1	1	1	?

$P(y=1) = ?$

$P(y=0) = ?$

Probability table

X_1	X_2	X_3	X_4	y	P()
1	1	1	1	1	0.2
1	1	1	1	0	0.3
0	0	0	0	1	0.1

If we can factor some possibilities, then we can save some calculations.

for
$$P(1,1,1,1|1) = P(1|1)\,P(1|1)\,P(1|1)\,P(1|1)$$

↑ 4 factors to be calculated and stored

instead, we could save

$$= P(1,1|1)\,P(1,1|1)$$

↳ only 2 factors have to be calculated and stored.