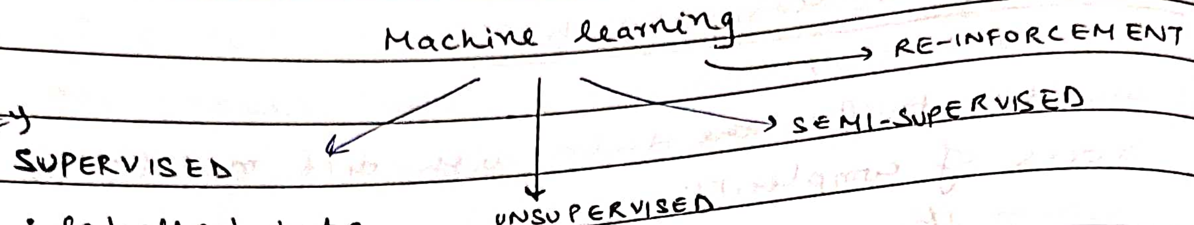
Classmate

Date 31/8/24
Page

Machine learning paradigms



Machine learning

RE-INFORCEMENT

SEMI-SUPERVISED

UNSUPERVISED

f(x)=y

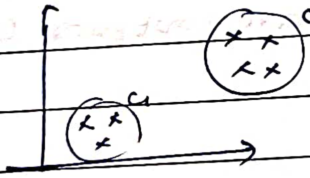
SUPERVISED

- labelled data
- Direct feedback
- Prediction

* Unsupervised learning → pattern & information.

→ It is trained with unlabelled data, only with feature vectors.

- The model tries to understand the relation b/w samples or features.
- forms clusters
- Model discovers and learns on its own, ie; it works on its own to discover pattern and information.



Out of 10 features, select the best 2.

↳ DIMENSIONALITY REDUCTION

1000 features → high computational power.

whenever dataset is unlabelled

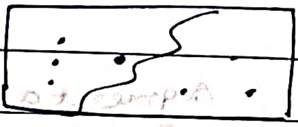
∴ UNSUPERVISED

- For a grouping of features vectors x_i, it learns a representation e_i, that is appropriate for the problum.

classmate
Date
Page

* SEMI-SUPERVISED LEARNING

1. Use small labelled sample to learn initial rules
training set $T = \{x_i\}$
unlabelled set $U = \{u_i\}$
2. we first train of T and findf()
3. Get predictions : $P = f(U)$



4. If $P_i > a$;
we add $(u_i, f(u_i))$ to T
5. Modified training set is retained.
6. Repeat till converges.

1. take only labelled data (starts from small)
find fn connecting features with value y
(we are finding decision boundary).
2. nearest to which class
update decision boundary for all samples.
* (Takes ~~[illegible]~~ unlabelled data & classies)

Depending upon what % of data is labelled
↳ like weak superwised
↳ self supervised

* REINFORCEMENT LEARNING

(self-driving cars) ⇒ AGENT → learning from its environment through trial & error.

learns from its mistake
& repeats from beginning.

Pass / fail instances For ex: ROBOTS
CONTINUOUS LEARNING

classmate
Date
Page

unsupervised → unlabelled data, don't know what to predict

obj: Find hidden structures/relations
diff clusters.

The agent learns to take actions that maximize its expected reward over-time.

(continuous actions
either penalty or reward)

Semi-supervised → partially labelled data.
~~semi +~~ unsupervised + supervised.

Reinforcement →
- No dataset
- only an agent who takes continuous actions
- learns to behave in an environment.

NOTE: Each model has its own set of parameters.

Learnable Parameters	Hyperparameters
- Estimated from the training data.	- Assigned by the programmer during training
(relations)	(INBUILT)

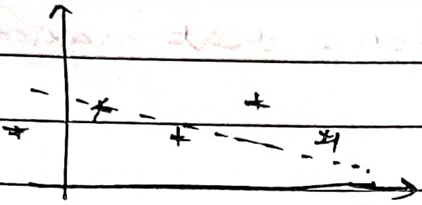
Ex: Decision Trees: Decide how many times a dec must be taken.

Neural networks: i/p layers - how many do we want?

Classification : datapoint, looking at its neighbour. how many neighbours we want?

Diff conditions we give.

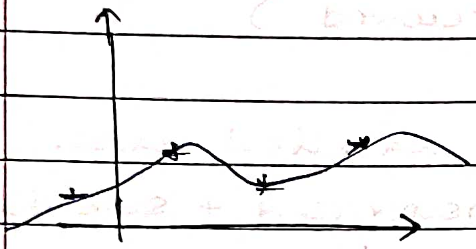
NOTE: learning ~~para~~ is modelled in parameter space.



3 ways → ① linear model.

$y = \omega^T x + b$

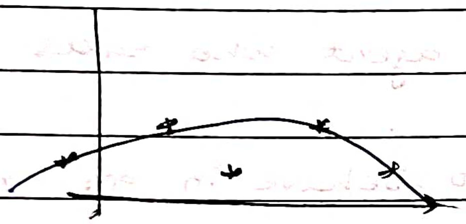
Parameter : ω, b



② Non-linear model

$y = a \sin(\phi x) + b$

parameter; θ : a, ϕ, b



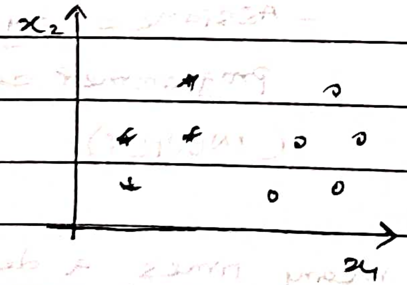
③ Quadratic Model

$y = x^T M x + \omega^T x + b$

parameter ; M, ω, b

Parameters always learnt from training.

Feature space
(Draw decision boundary.)



(linear model)

First start with random ω, b

Is it able to classify properly?

Hyper-parameters

- we need to give it a value for a model to start training.
- It is a parameter whose value is set before the model starts training.
- It can't learn by fitting the model to the data.

classmate
Date
Page

Model hyperparameters.

- · # of neighbours KNN
- · # of layers CNN

* Optimization

QUOTE: A Good choice of Hyperparameters, can really make an algo shine.

Common Algo → Grid Search
→ Random Search

Grid Search (SVM algo & classif^n) Exhaustive search.

→ Performs an exhaustive search by evaluating any candidates ~~[illegible]~~ combination.

all possible combinations of hyperparameters.

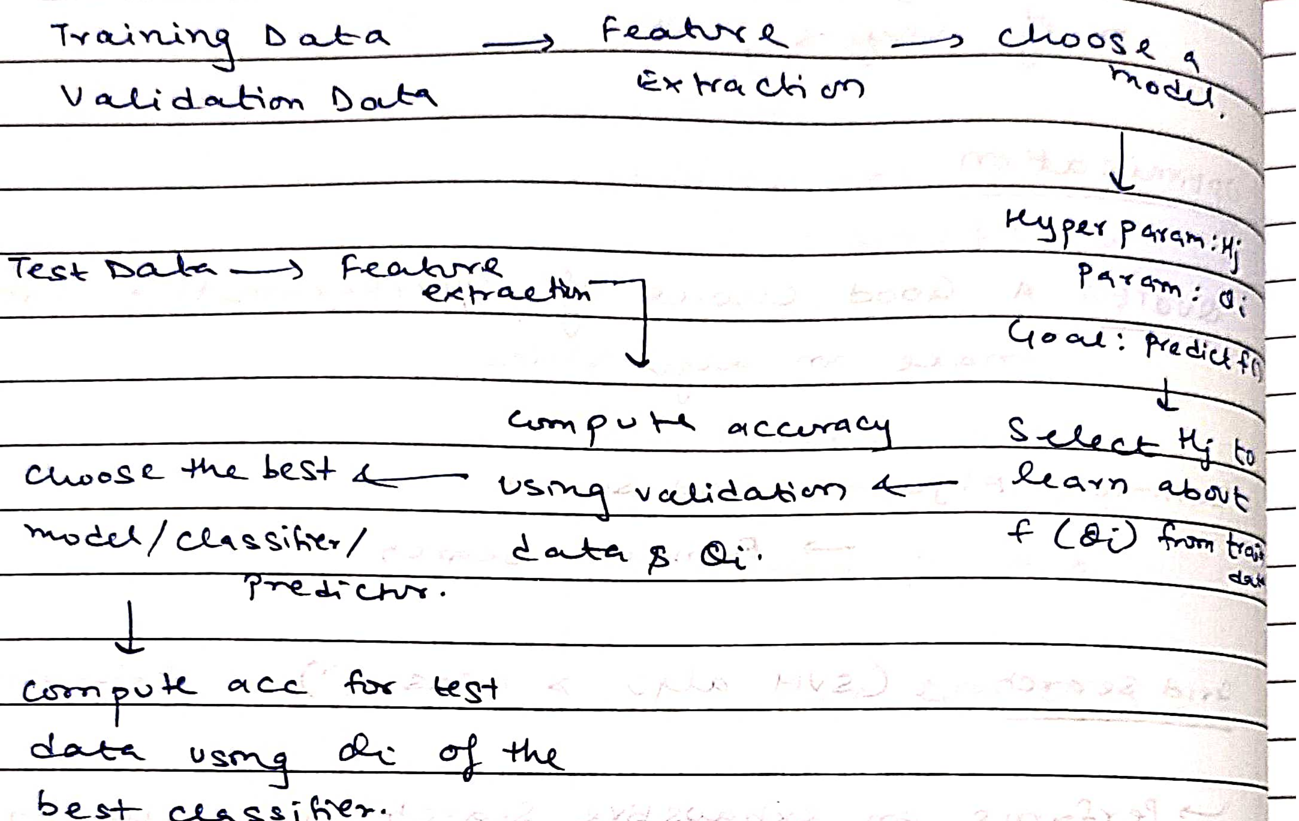
More hyperparameters, more values
⇒ High computational power
Better alternative: random search.

Random Search

- ↳ randomly picks a value, find a good/optimal value.
- ↳ May miss optimum point.
- ↳ Better in the case of huge # of ~~sampl~~ hyperparameters

Next better: Bayesian Optimization: (Based on past iter^{n}s)
learning from each iterations
& first picks randomly
depends on prev iteration

SUMMARY



Training Data
Validation Data → feature Extraction → choose a model.

↓

Hyper param: H_j
Param: θ_i
Goal: predict f()

↓

Select H_j to learn about f (θ_i) from trai dat

Test Data → feature extraction ↓

compute accuracy using validation data & θ_i. ←

choose the best ← model/classifier/ predictor.

↓

compute acc for test data using θ_i of the best classifier.

Data Quality

- The quality and quantity of traning data is the most important aspect that decides the quality of ML soln.

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

- Disadv : outliers
  missing values
  limited quantity

unwanted data ⇒ Noise

* Outliers Detection - Box plot

1. Median 25%
2. 75% population.
3. Inter Quartile
4. IQR

· outlier is an observation ie; unlike the other observation.

Caused by : measurement or i/p error
Data corruption
True outlier observation.

May cause problem during model fitting.

noise : random behavior , value scattered randomly in dataset

outlier : data follows trend , but some points do not.