SUPERVISED AND UNSUPERVISED LEARNING

MACHINE LEARNING

Goals of ML:

- Learning from data
- Establishing relationships between mutliple features.
- Extracting statistical patterns
- Reasoning under uncertainity

Just like the BRAIN!

MACHINE LEARNING

Application Areas:

- Statistics

- Engineering

- Computer Science

- Cognitive Science

...

MACHINE LEARNING

Types of ML:

- Supervised Learning: Find the class labels or value of the new input, given the dataset.

- Reinforcement learning: Learn to act in a way that maximizes the future rewards (or minimizes a cost function)

- In game theory: Learn to act in a way that maximized the future rewards, in an environment that contains other machines.

- Unsupervised Learning: contains neither targert outputs or reward from its environment.

SUPERVISED LEARNING

	INPUTS					OUTPUT
	Gender	**Married**	**Job**	**Age**	**Salary**	**Trust**
Customer 1	Male	No	Teacher	43	1500	good
Customer 2	Female	No	Lawyer	55	2500	good
Customer 3	Male	Yes	Doctor	26	1700	bad
...	...	...		...	...	...
Customer n	Male	Yes	Lawyer	35	1600	???
Customer n+1	Female	No	Doctor	30	1400	???
Customer n+2	Male	Yes	Retired	60	2000	???

Instances are n-dimensional points in space, and the features of the instances correspond to the dimensions of that space.

SUPERVISED LEARNING

- Features can be:
  - continuous
  - categorical
  - binary

- Training set: The output of each data point is known.

- Training Algorithms...

- Test set: The output of each data point is estimated.

- Output can be:
  - a class label
  - a real number

(UNSUPERVISED LEARNING)

- No supervised target outputs

- No rewards from the environment

- No feedback

SO?

- Build representations of the inputs
- Find patterns in the inputs
- Decision making
- Predict future inputs

(UNSUPERVISED LEARNING)

- Extract information from **unlabelled** data.

- Learn a probabilistic model of the data.

  This can be useful for:
  - Outlier detection
  - Classification
  - Data compression

- Bayes Rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

(To have beliefs about the world, we trust the statistics.)

SUPERVISED LEARNING

- Dataset Collection

- Feature Selection

- Algorithm Selection

- Training

SUPERVISED LEARNING:

COLLECTING THE DATASET

Brute-force method: Measuring everything available in the hope that the relevant & informative features can be isolated.

(-) contains a lot of noisy data
(-) missing features
(-) requires significant data pre-processing
(+) simple

OR: An expert decides which features to measure and use.

COLLECTING THE DATASET

Possible problems in a dataset:

- handling the missing data

- outlier (noise) detection

- instance selection (in the case of large datasets)

- feature subset selection (in the case of redundant features and high dimensionality)

- feature construction/transformation

SUPERVISED LEARNING:

ALGORITHM SELECTION

Performance of the algorithm is determined by the **prediction accuracy**, given by:

$$\frac{\textbf{\% correct prediction}}{\textbf{\% all predictions}}$$

3 ways to calculate it:

- **2/3** training & **1/3** estimating performance

- **Cross validation** (training set is divided into mutually exclusive and equal sized subsets, and error rates of the subsets are averaged.)

- **Leave-one-out** validation is a special case of cross validation. (Every subset has only 1 instance.)

Unstability: Small changes in the training set result in large changes.

SUPERVISED LEARNING:

LOGIC BASED ALGORITHMS:

DECISION TREES

Decision trees are trees that classify instances by sorting them based on feature values.

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Table 2. Training Set

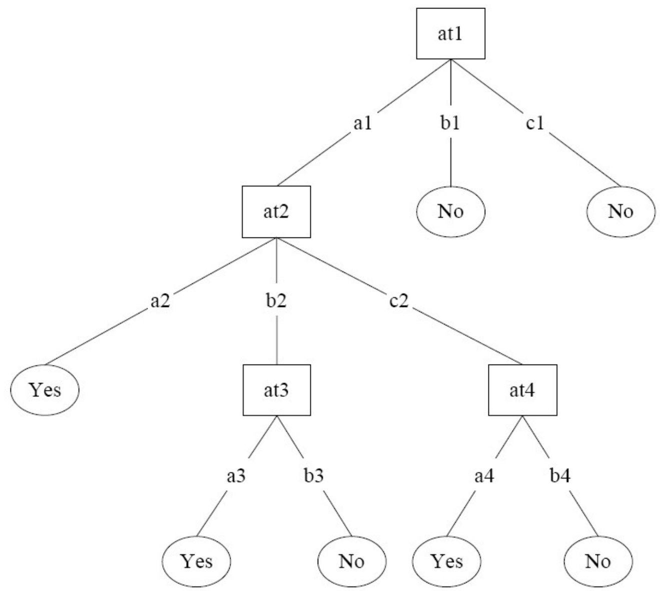


Figure 2. A decision tree

SUPERVISED LEARNING:

LOGIC BASED ALGORITHMS:

DECISION TREES

The feature that best divides the training data would be the root node of the tree

To avoid overfitting :

i) Stop the training before perfect fitting

ii) Prune the induced decision tree. The tree with fewest number of leaves is preferred.

Zheng (2000) created at-least *M of-N features. An instance is true if at least M of its* conditions is true, otherwise it is false.

SUPERVISED LEARNING:

PERCEPTRON BASED ALGORITHMS

* Dataset:

- x_1 to x_n are the input feature values.
- w_1 to w_n are the connection weights / **prediction vector**.

* Perceptron computes the weighted sum: $\sum(x_i * w_i)$

* Sum < treshold => 1

  Sum < treshold => 0

* Run the algorithm repeatedly over the training set, until it finds a **prediction vector** that is correct on all the training set.

SUPERVISED LEARNING:

PERCEPTRON BASED ALGORITHMS

* Can only classify linearly separable sets of instances.

* Binary => In the case of multiclass problems, the problem must be reduced to a set of multiple binary classification problems.

* Anytime online! (Can produce a useful answer regardless of how long they run.)

* Superior time complexity when dealing with irrelevant features.

INSTANCE-BASED LEARNING

K-NN Algorithm:
Assign the same label according to the nearest neighbours (if K>1, do majority voting)

It is a Lazy-learning algorithm! Which means:

- No generalization process until classification is performed

- Require *less* computation time during the *training phase* than eager-learning algorithm(such as decision trees, neural and Bayes nets) but *more* computation time during the *classification process*.

SUPERVISED LEARNING:

INSTANCE-BASED LEARNING

K-NN Algorithm:

Different Distance Metrics to compare feature vectors:

Minkowsky: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert^r\right)^{1/r}$
Manhattan: $D(x,y)=\sum_{i=1}^{m}\lvert x_i - y_i\rvert$
Chebychev: $D(x,y)=\max_{i=1}^{m}\lvert x_i - y_i\rvert$
Euclidean: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert^2\right)^{1/2}$
Camberra: $D(x,y)=\sum_{i=1}^{m}\frac{\lvert x_i - y_i\rvert}{\lvert x_i + y_i\rvert}$
Kendall's Rank Correlation: $D(x,y)=1-\frac{2}{m(m-1)}\sum_{i=j}^{m}\sum_{j=1}^{i-1} sign(x_i - x_j)sign(y_i - y_j)$

Table 3. Approaches to define the distance between instances (x and y)

INSTANCE-BASED LEARNING

K-NN Algorithm:

i) they have large storage requirements

ii) they are sensitive to the choice of the distance metric

iii) hard to choose the best k

SUPERVISED LEARNING:

SUPPORT VECTOR MACHINES

An optimization problem:

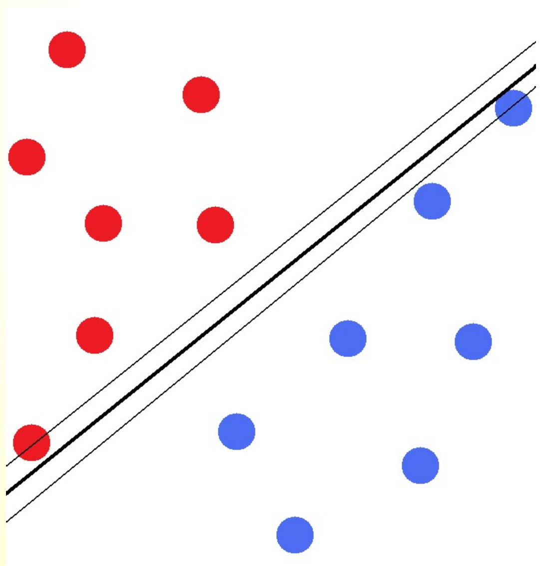
Find a hyperplane that separate the sample space which:

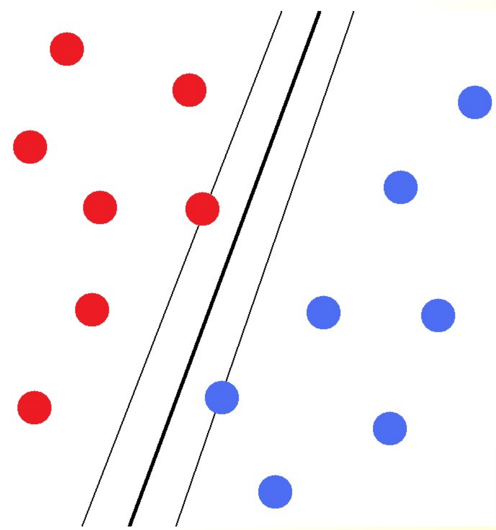
1) Maximizes the separation of the classes
2) Maximize the distance of the hyperplane to the closest samples on each side

SUPERVISED LEARNING:

SUPPORT VECTOR MACHINES

A separation with a higher margin is preferred for generalization purposes.





SUPERVISED LEARNING:

SUPPORT VECTOR MACHINES

If the training data is not **Linearly Separable:**

Kernel Trick is applied to map the input space to a higher dimensional space where the data is now Linearly separable

Some popular kernels are the following:

(1) $K(x, y) = (x \cdot y + 1)^P$,

(2) $K(x, y) = e^{-\|x - y\|^2 / 2\sigma^2}$,

(3) $K(x, y) = \tanh(\kappa x \cdot y - \delta)^P$

UNSUPERVISED LEARNING

- Extract information from **unlabelled** data.

- Learn a probabilistic model of the data.

  This can be useful for:
  - Outlier detection
  - Classification
  - Data compression

- Bayes Rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

(To have beliefs about the world, we trust the statistics.)

UNSUPERVISED LEARNING:

LATENT VARIABLE MODELS

Logic Based Algorithms
Perceptron Based Techniques
Statistical Learning Algorithms
Instance Based Learning
Support Vector Machines