Machine Learning using Python

Theory and Implementation

Vishal Kumar Jaiswal

M.Tech
CST Department , IIEST Shibpur

June 18, 2016

Summer Internship 2016

Introduction
Machine Learning
Fundamentals
Inductive learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Survey Method
Mathematical Preliminaries
Learning Algorithms
Entropy
Ensemble Learning Method

Table of Contents

What to expect

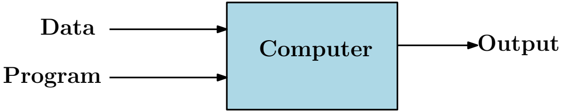
- You'll learn by doing,
- Bring machine learning to life using specified use case,
- Identify financial fraud by mining email datasets to identify writer of email.

Why machine learning?

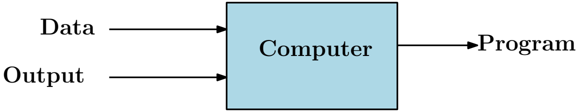
- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!
- Can help in solve the most practical society impacting problems.

Comparison with traditional programming

Traditional Programming:-



Machine Learning:-



Machine learning

- Building computational artifacts that learn over time based on experience.
- Employed in computing tasks where designing and programming explicit algorithms is infeasible.
- Build a model from example inputs in order to make data-driven predictions or decisions expressed as outputs rather than following strictly static program instructions.

Inductive Learning

Process of **learning by example** - where a system tries to induce a general rule from a set of observed instances i.e. specific to general rule.

1. **Discrete:-** Classification
2. **Continuous:-** Regression

Constructing class definitions is called **Inductive learning**.

Note:

Artificial Intelligence is based on deductive learning.

Inductive Learning

Process of **learning by example** - where a system tries to induce a general rule from a set of observed instances i.e. specific to general rule.

1. **Discrete:-** Classification
2. **Continuous:-** Regression

Constructing class definitions is called **Inductive learning**.
Note:
Artificial Intelligence is based on deductive learning.

Classification of Machine Learning

Based on feedback available to learning system, we categorize machine learning algorithms in following categories:-

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Machine Learning: Type 1

Supervised learning

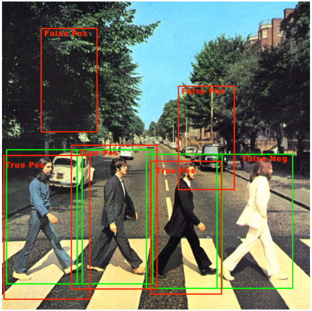
Function approximation from input and output pairs (training dataset) i.e. generalization.

Supervised Learning

Classification

Assign,to a particular input, the name of class to which it belongs.

Classification: An Example



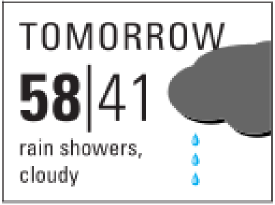
Spam filtering and Object detection

Supervised learning

Regression

Predicting a numerical value for input dataset.

Regression



stack market prediction and weather forcasting

Machine learning: Type 2

Unsupervised learning

Without any examples, summerize or describe data just by looking at the input.

e.g. naming all animals dog by a child. Describe people based on ethinicity like Punjabi, Bangali, South Indian etc.

Machine learning: Type 2

Unsupervised learning

Without any examples, summerize or describe data just by looking at the input.

e.g. naming all animals dog by a child. Describe people based on ethinicity like Punjabi, Bangali, South Indian etc.

Unsupervised Learning

Clustering

Discovering structure in data e.g. finding similiar images in google images, Collaborative filtering of news feed content by facebook.

Clustering: An Example

Google news

Machine learning: Type 3

Reinforcement learning

Learn how to behave based on delayed feedback from the environment.

e.g. Figuring out mistakes after wrong answer in test series during preparation and correcting your bias afterwards.

Note:

1. All are supplement of each other, not an alternative.
2. Fundamentally different, but trying to achieve the same goal.

Machine learning: Type 3

Reinforcement learning

Learn how to behave based on delayed feedback from the environment.

e.g. Figuring out mistakes after wrong answer in test series during preparation and correcting your bias afterwards.

Note:

1. All are supplement of each other, not an alternative.
2. Fundamentally different, but trying to achieve the same goal.

Machine learning: Type 3

Reinforcement learning

Learn how to behave based on delayed feedback from the environment.

e.g. Figuring out mistakes after wrong answer in test series during preparation and correcting your bias afterwards.

Note:

1. All are supplement of each other, not an alternative.
2. Fundamentally different, but trying to achieve the same goal.

Survey Methodology

- Collect data for machine learning by conducting survey or storing sensor output data stream.
- Preprocess data to remove inconsistencies in it.
- Before believing a survey result, check:
  1. How many people or situations are surveyed.
  2. Who is surveyed.
  3. How survey is conducted.
- Train and test machine learning algortihm on different sets of data, otherwise overfitting will happen.

Bayes' Theorem

Conditional Probability

Probability of event obtained with additional information that some other event has already occurred.

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

Assumption: We are dealing with sequential events and the new information is used to revise the probability of previous events.

Bayes' Theorem

Prior probability

Initial probability value obtained before any additional information is provided.

Posterior Probability

The probability value that has been revised by using additional information that is obtained later.

Bayes' Theorem

Definition

The probability of event A, given that event B has subsequently occurred, is

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{[P(A) \cdot P(B|A)] + [P(\bar{A}) \cdot P(B|\bar{A})]}$$

Bayes' Rule: An example

A specific cancer, C, incurs in 1% of population, $P(C)=0.01$
Pathological Test:
Probability of positive test result, if you have C,
$P(Pos|C)=90\%$ (Sensitivity)
Probability of negative test result, if you don't have C,
$P(Pos|C)=90\%$ (Specificity) So, find out

- Probability of having cancer if test positive

Bayes' Rule: An example

A specific cancer, C, incurs in 1% of population, $P(C)$=0.01
Pathological Test:
Probability of positive test result, if you have C,
$P(Pos|C)$=90% (Sensitivity)
Probability of negative test result, if you don't have C,
$P(Pos|C)$=90% (Specificity) So, find out

- Probability of having cancer if test positive

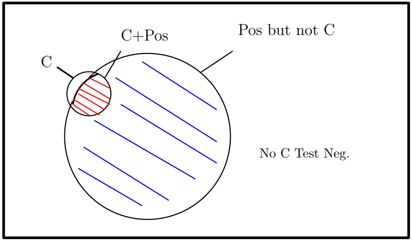
Prior Probabilities:

$P(C) = 0.01$ $\quad$ $P(\neg C) = 0.99$

$P(Pos|C) = 0.9$

$P(Neg|\neg C) = 0.9$ $\quad$ $P(Pos|\neg C) = 0.1$

All People



Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = \text{‘}0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$P(\neg C|Pos) = \dfrac{P(\neg C, Pos)}{P(Pos)} = 0.917$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$P(\neg C|Pos) = \dfrac{P(\neg C, Pos)}{P(Pos)} = 0.917$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01$ $\quad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9$ $\quad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$P(\neg C|Pos) = \dfrac{P(\neg C, Pos)}{P(Pos)} = 0.917$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = \text{‘}0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01$ $\quad P(\neg C) = 0.99$

$P(Pos|C) = $ ‘0.9

$P(Neg|\neg C) = 0.9$ $\quad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) =$ ‘0.9

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Prior Probabilities:

$P(C) = 0.01 \qquad P(\neg C) = 0.99$

$P(Pos|C) = `0.9$

$P(Neg|\neg C) = 0.9 \qquad P(Pos|\neg C) = 0.1$

Joint Probability:

$P(C, Pos) = P(C) \times P(Pos|C) = 0.009$

$P(\neg C, Pos) = P(\neg C) \times P(Pos|\neg C) = 0.099$

Normalize:

$P(Pos) = P(\neg C, Pos) + P(C, Pos) = 0.108$

Posterior: $P(C|Pos) = \dfrac{P(C, Pos)}{P(Pos)} = 0.083$

$$P(\neg C|Pos) = \frac{P(\neg C, Pos)}{P(Pos)} = 0.917$$

So, Total Probability=1

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Naive Bayes Method

- Supervised learning algorithm based on Bayes' theorem.
- The "naive" assumption of independence between every pair of features.
- works quite well for many real-world applications. e.g. document classification and spam filtering.
- extremely fast
- decent classifier, but bad estimator.

Gaussian Naive Bayes

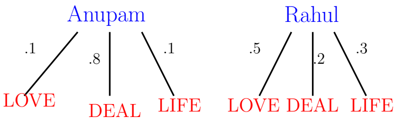
Vishal Kumar Jaiswal

Definition

The liklihood of features is:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu)^2}{2\sigma_y^2}\right)$$

Vishal Kumar Jaiswal



$P(\text{Anupam}) = 0.5$
$P(\text{Rahul}) = 0.5$

Guess: (Owner of email)

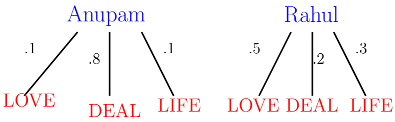
LOVE LIFE!

LIFE DEAL

Calculate

$P($ " LIFE DEAL "$|$ Anupam$)$ $0.8 \times 0.1 \times 0.5$

$P($ "LIFE DEAL" $|$ Rahul$)$ $0.2 \times 0.3 \times 0.5$



$P(\text{Anupam}) = 0.5$

$P(\text{Rahul}) = 0.5$

Guess: (Owner of email)

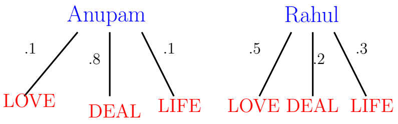
LOVE LIFE!

LIFE DEAL

Calculate

$P($ " LIFE DEAL "$|$ Anupam$)$ $0.8 \times 0.1 \times 0.5$

$P($ "LIFE DEAL" $|$ Rahul$)$ $0.2 \times 0.3 \times 0.5$



$P(\text{Anupam}) = 0.5$
$P(\text{Rahul}) = 0.5$

Guess: (Owner of email)

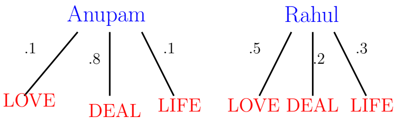
LOVE LIFE!

LIFE DEAL

Calculate

$P($ " LIFE DEAL "$|$ Anupam$)$ $0.8 \times 0.1 \times 0.5$

$P($ "LIFE DEAL" $|$ Rahul$)$ $0.2 \times 0.3 \times 0.5$



$P(\text{Anupam}) = 0.5$

$P(\text{Rahul}) = 0.5$

Guess: (Owner of email)

LOVE LIFE!

LIFE DEAL

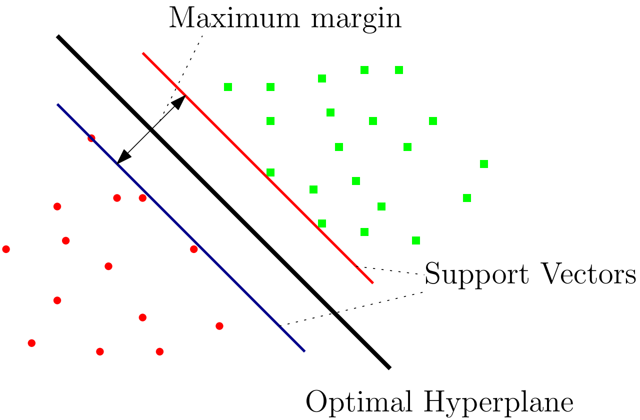
Calculate

$P($ " LIFE DEAL "$|$ Anupam$)$ $0.8 \times 0.1 \times 0.5$

$P($ "LIFE DEAL" $|$ Rahul$)$ $0.2 \times 0.3 \times 0.5$

Support Vector Machine

- Supervised learning used for classification, regression analysis and outlier detection.
- Often used as a black box and for comparative study.
- Support Vectors are set of frontier datapoints of training dataset (may be linear).
- Effective in high dimensional space.



Pictorial representation of support vector machine

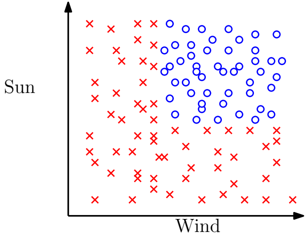
```
# Support vector classification from sklearn module
from sklearn.svm import SVC
# Kernel: default: 'rbf',  alternativess: 'linear','poly',
clf=SVC(kernel="linear",gamma=1000.0,C=1)
#Fit SVM model to according to the training data
clf.fit(features_train,labels_train)
# Perform classification on test set
pred=clf.predict(features_test)
```

Decision Trees

- Supervised algorithm
- Most popular, simple and extremely robust.
- Non-linear decision making with simple linear surfaces.

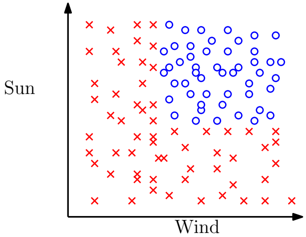
Disadvantage:

- over-complex trees that do not generalize well on datapoints(overfitting). Pruning is required.
- Learning optimal decision tree is NP-complete.



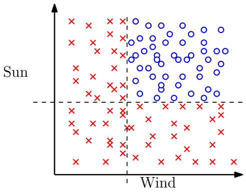
Weather pattern for Windsurf

Decision trees allows us to use multiple linear boundaries to seperate two non-linearly seperable regions.

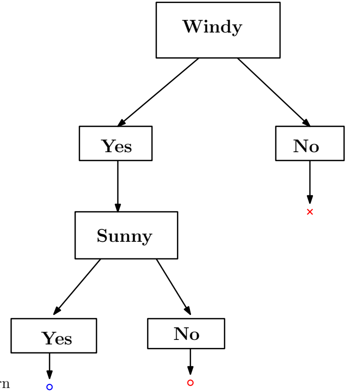


Weather pattern for Windsurf

Decision trees allows us to use multiple linear boundaries to seperate two non-linearly seperable regions.



Multiple linear decision boundary for weather pattern



Symbolic representation of decision tree for weather data

Classification

`DecisionTreeClassifier` is a class capable of performing multi-class classification on a dataset.

```
from sklearn import tree
X=[[0,0],[1,1]]
Y=[0,1]
clf=tree.DecisionTreeClassifier(min_samples_split=50)
clf=clf.fit(X,Y)
clf.predict([[2.,2.]])
```

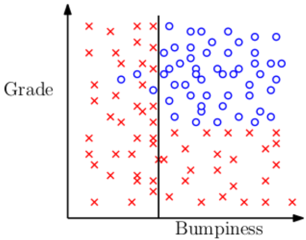
Entropy

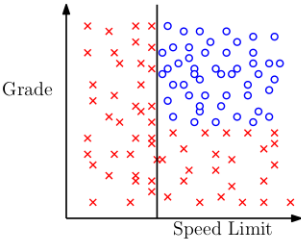
Definition

Measure of impurity in a bunch of examples.

$$\text{Entropy}, H(P) = \sum -P_i log_2 P_i$$

Controls how a decision tree decides to split the data, e.g.





Information Gain

Definition

The change in information entropy H from a prior state to a state that takes some information as given:

$$IG(T, a) = H(T) - H(T|a)$$

Decision trees tries to maximize information gain.

K Nearest Neighbors Classification Algorithm

- k closest training examples in the feature space are provided as input.
- Classification is done by a majority vote of neighbors.
- The object is assigned to the class most common among its k neighbors.

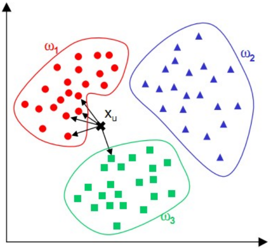


Illustration of K nearest neighbors

K Nearest Neighbors Classification Example

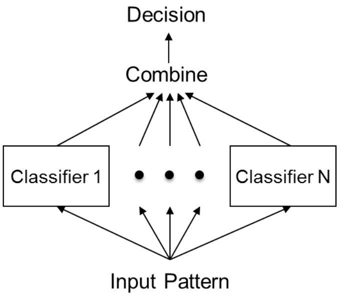
```
# Support vector classification from sklearn module
from sklearn.svm import SVC
# Kernel: default: 'rbf',  alternativess: 'linear','poly',
clf=SVC(kernel="linear",gamma=1000.0,C=1)
#Fit SVM model to according to the training data
clf.fit(features_train,labels_train)
# Perform classification on test set
pred=clf.predict(features_test)
```

Ensemble Learning Method

- Supervised Learning Methodology
- Uses multiple learning methods to obtain better performance.
- Prediction requires more computational overhead.
- Some unsupervised learning methods include consensus clustering and anamoly detection.

Random Forest

- Ensemble learning method
- Avoids overfitting of decision trees by building multiple deep decision trees
- Parameters include number of estimators.



Ensemble Learning Methodology

Adaptive Boosting, Adaboost

- Best known classifier
- Adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.
- Sensitive to noisy data and outliers
- AdaboostClassifier

- Numerical e.g. Salary info.
- Categorical e.g. Job Title
- Time Series e.g. time stamps on emails
- Text e.g. Contents of emails

References

Bayes' Theorem.

UD262: Machine Learning.

Ali Farhadi.
Cse446: Machine learning.
University of Washington.

Sebastian Thrun Katie Malone.
Ud102: Intro to machine learning.
Udacity.

Thank You