Data Science Techniques

Tom Flaherty

Version 0.0, November 9, 2016

Table of Contents

1. Concepts

Definitions

Machine Learning

Constructs algorithms that can learn from data.

Statistical Learning

A branch of applied statistics that emerged in response to machine learning. It emphasizes statistical models and assessment of uncertainty.

Data Science

The extraction of knowledge from data that leverages ideas from mathematics statistics machine learning computer science engineering…

Contrast

All of these displines are very similar with different emphases.

Training Fitting Predictions

Traditional Statistics

- Domain experts work for 10 years to learn good features
- Then they bring the statistician a small clean dataset

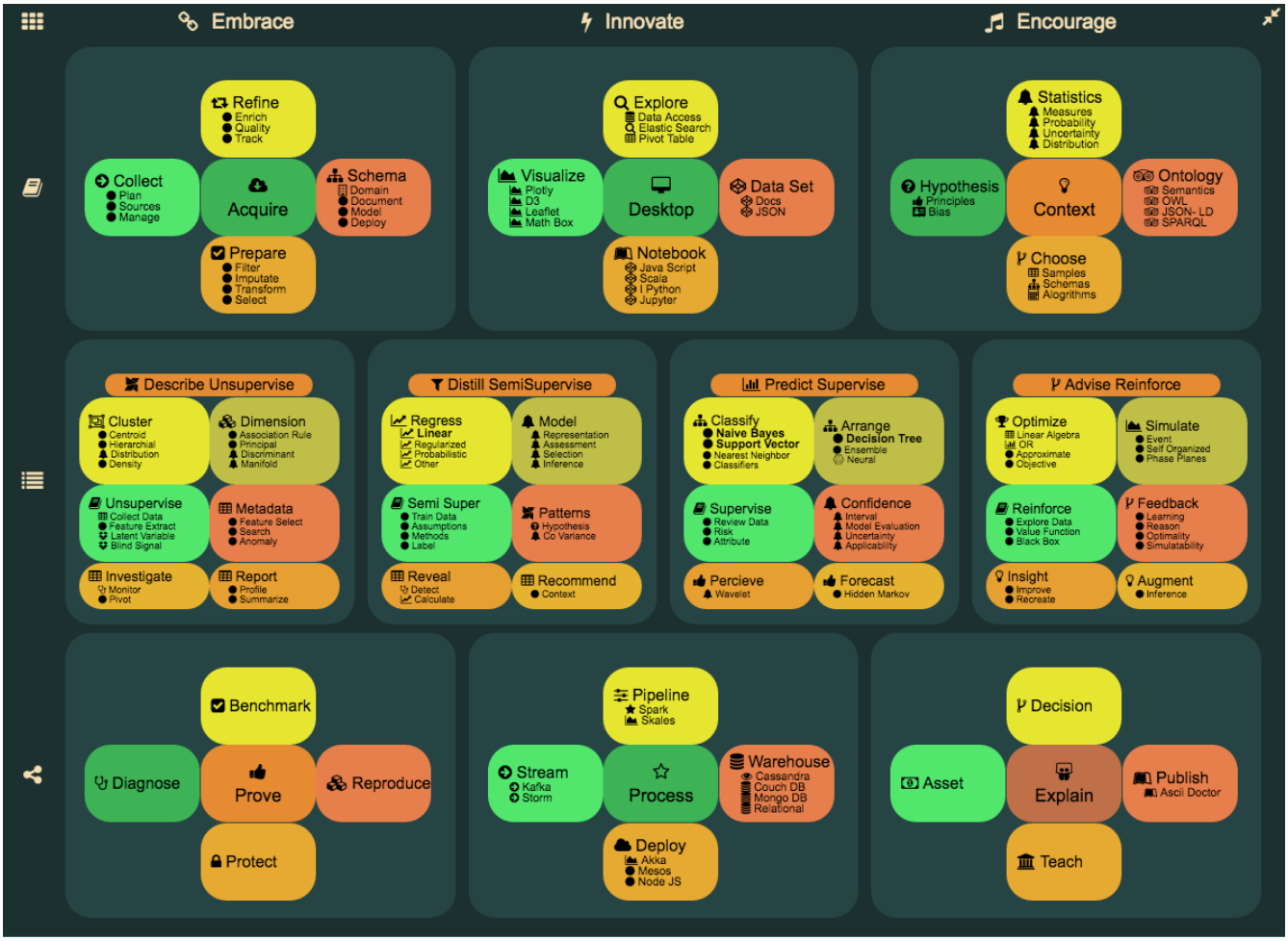
Todays Approach

- We start with a large dataset with many features.
- Then use a machine learning algorithm to find the good ones.
- A huge change.

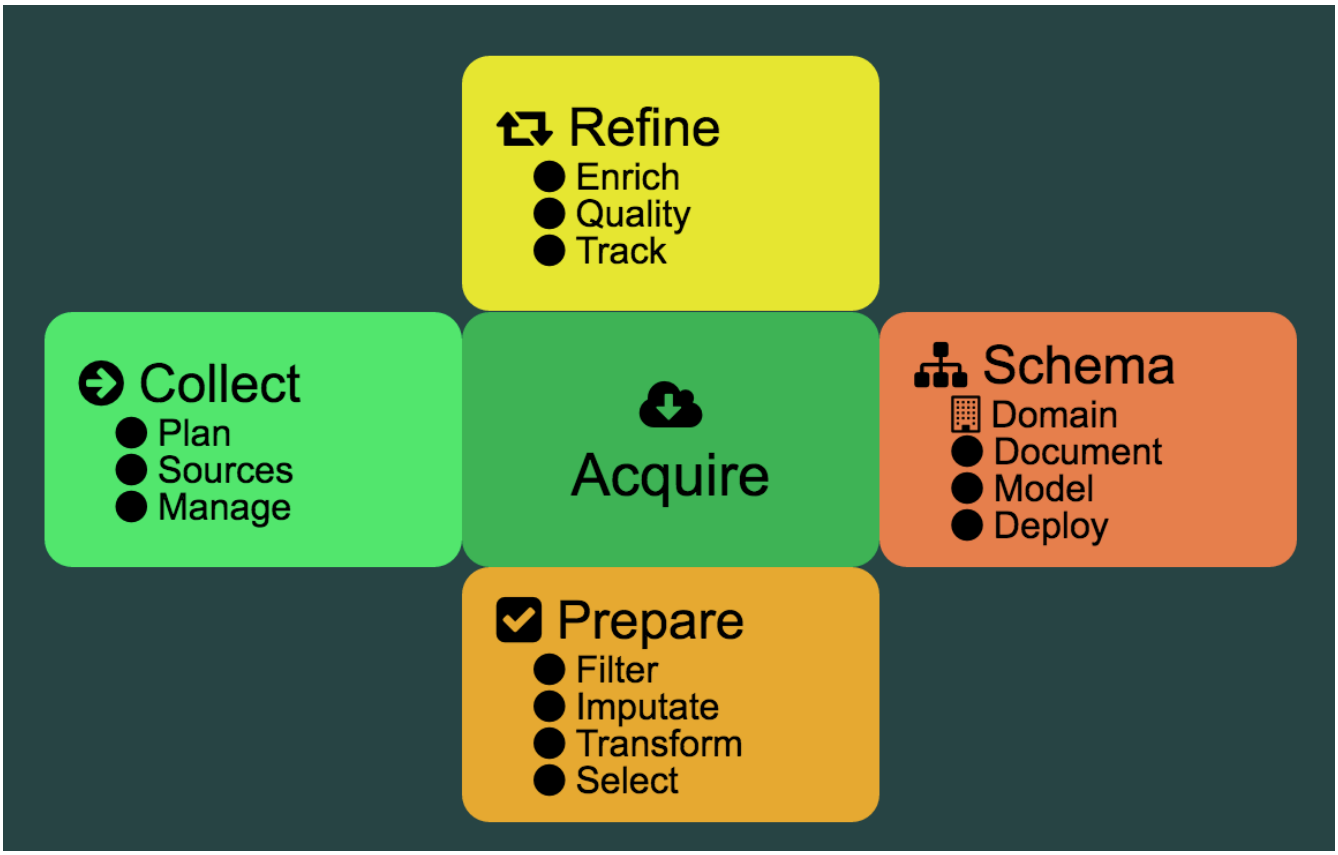
Core Data Science Practices

Practice	Learning	Algorithm	Technique	Results	
Acquire	Collect	Refine	Prepare	Schema	
Describe	Unsupervised	Cluster	Dimension	Metadata	
Distill	Semi-Supervised	Regression	Model	Patterns	
Predict	Supervised	Classification	Desision Tree	Confidence	
Advise	Reinforce	Optimize	Simulate	Feedback	

A Complete Set of Data Science Practices



2. Acquire


Refine
Enrich
Quality
Track
Collect
Plan
Sources
Manage
Acquire
Schema
Domain
Document
Model
Deploy
Prepare
Filter
Imputate
Transform
Select

Collect

Plan

Sources

Manage

● Refine

● Improve

● Quality

● Track

● Schema

● Domain

● Document

● Model

● Deploy

● Prepare

● Filter

● Imputate

● Transform

● Select

3. Describe


Describe Unsupervise
Cluster
Centroid
K Means
SOM
Affinity
Fuzzy C Means
Hierarchial
Agglomerative
Birch
Canopy
Fractal
Distribution
Expectation Max
Spectral
Correllation
Density
DBSCAN
Mean Shift
Dimension
Association Rule
Apriori
Dictionary
Principal
Independent
Correspondence
Factor Analysis
Discriminant
Linear
Quadratic
Regularized
Cross
Manifold
Isomap
Locally Linear
Laplacian
Unsupervise
Collect Data
Implicit Collect
Emotional Spaces
Density
Text Rank
Feature Extract
Hashing
Term Frequency
Latent Variable
Expectation Max
Method Of Moments
Dirichlet
Blind Signal
Principal Comp
Non Neg Matrix
Hebbian Learning
Metadata
Feature Select
Genetic
Ensemble
Signal Noise
Sum Squares
Search
Grid
Exhaustive
Best First
Annealing
Anomaly
Outlier
Local Outlier
Model Based
Smoothing
Investigate
Monitor
Detect
Investigate
Pivot
Drill Down
Roll Up
Report
Profile
Baseline
Detail
Relate
Difference
Summarize
Relational
Extraction
Abstraction

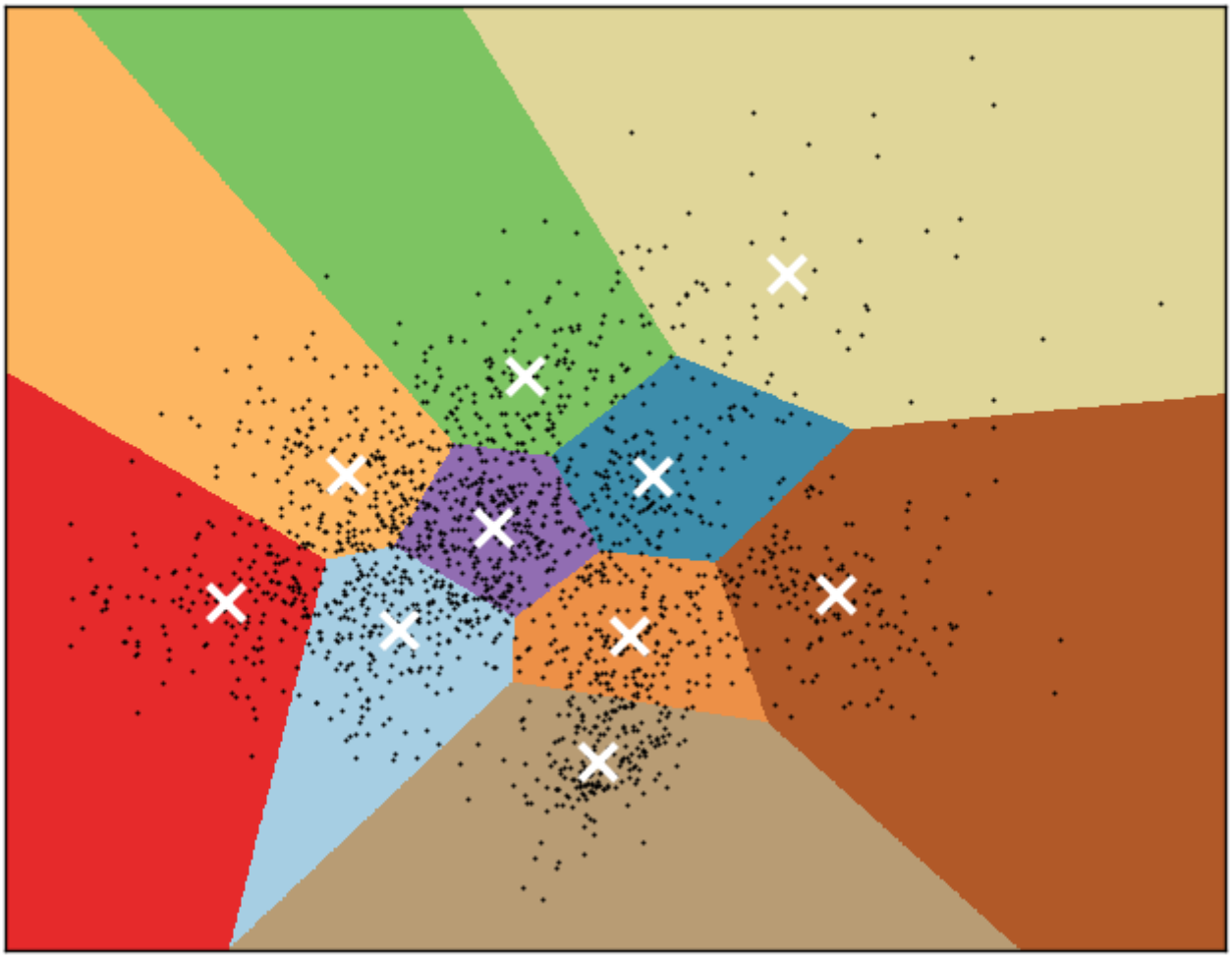
Cluster

Centroid

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k {\displaystyle k} k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

K Means

- A popular unsupervised non deterministic algorithm for cluster analysis.
- Centroid based that works like a nearest neighbor Voroni diagram.



Strengths	Tight clusters	Scalable	General purpose	Fast
Use When	Known Number	Small Number	Even Size	Flat geometry

How It Works

- The algorithm operates on a given data set through pre-defined number of clusters, k.
- The output of K Means algorithm is k clusters

Benefits

- In case of globular clusters, K-Means produces tighter clusters than hierarchical clustering.
- Given a smaller value of K, K-Means clustering computes faster than hierarchical clustering for large number of variables.

Drawbacks

- Difficult to specify K value.
- Does not work well with global cluster.
- Different initial partitions can result in different final clusters.

- It does not work well with clusters of varying size and density.

Applications

- K Means Clustering algorithm is used by most of the search engines like Yahoo, Google
- To cluster web pages by similarity and identify the ‘relevance rate’ of search results.
- This helps search engines reduce the computational time for the users.

Fuzzy C-Means

- A case where each data point has fuzzy membership in more than one cluster
- Attempts to partition data points into a collection of c fuzzy clusters with respect to some given criterion

Benefits

- Gives best result for overlapped data set and comparatively better then k-means algorithm.
- Unlike k-means where data point must exclusively belong to one cluster center here data point is assigned membership to each cluster center as a result of which data point may belong to more then one cluster center.

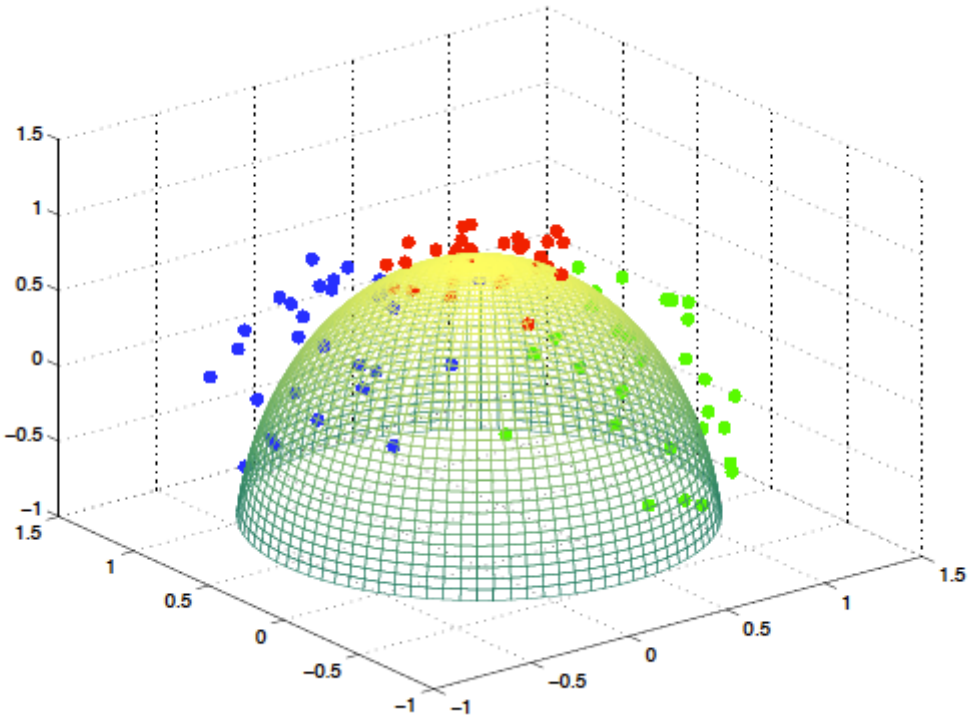
Drawbacks

- Apriori specification of the number of clusters.
- With lower value of β we get the better result but at the expense of more number of iteration.
- Euclidean distance measures can unequally weight underlying factors

---

Self Organizing Map

A self-organizing map (SOM) consists of components called nodes or neurons. Associated with each node are a weight vector of the same dimension as the input data vectors, and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher-dimensional input space to a lower-dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector.



Benefits

- Probably the best thing about SOMs that they are very easy to understand. It's very simple, if they are close together and there is grey connecting them, then they are similar. If there is a black ravine between them, then they are different. Unlike Multidimensional Scaling or N-land, people can quickly pick up on how to use them in an effective manner.
- Another great thing is that they work very well. As I have shown you they classify data well and then are easily evaluate for their own quality so you can actually calculated how good a map is and how strong the similarities between objects are.

Drawbacks

- One major problem with SOMs is getting the right data. Unfortunately you need a value for each dimension of each member of samples in order to generate a map. Sometimes this simply is not possible and often it is very difficult to acquire all of this data so this is a limiting feature to the use of SOMs often referred to as missing data.
- Another problem is that every SOM is different and finds different similarites among the sample vectors. SOMs organize sample data so that in the final product, the samples are usually surrounded by simliar samples, however similar samples are not always near each other. If you

have a lot of shades of purple, not always will you get one big group with all the purples in that cluster, sometimes the clusters will get split and there will be two groups of purple. Using colors we could tell that those two groups in reality are similar and that they just got split, but with most data, those two clusters will look totally unrelated. So a lot of maps need to be constructed in order to get one final good map.

- The final major problem with SOMs is that they are very computationally expensive which is a major drawback since as the dimensions of the data increases, dimension reduction visualization techniques become more important, but unfortunately then time to compute them also increases. For calcuating that black and white similarity map, the more neighbors you use to calculate the distance the better similarity map you will get, but the number of distances the algorithm needs to compute increases exponentially.

Steps

- Initialize the Weights
- Get Best Matching Unit
- Scale Neighbors
- Specify Learn Function

Determining the Quality

- There is a very simple method for displaying where similarities lie and where they do not. In order to compute this we go through all the weights and determine how similar the neighbors are. This is done by calculating the distance that the weight vectors make between the each weight and each of its neighbors. With an average of these distances a color is then assigned to that location.
- If the average distance were high, then the surrounding weights are very different and a dark color is assigned to the location of the weight. If the average distance is low, a lighter color is assigned. So in areas of the center of the blobs the colors are the same, so it should be white since all the neighbors are the same color. In areas between blobs where there are similarities it should be not white, but a light grey. Areas where the blobs are physically close to each other, but are not similar at all there should be black.

Hierachial

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively.

This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.

Algomerative

- The Agglomerative Clustering object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:
- Agglomerative Clustering can also scale to large number of samples when it is used jointly with a connectivity matrix, but is computationally expensive when no connectivity constraints are added between samples: it considers at each step all the possible merges.

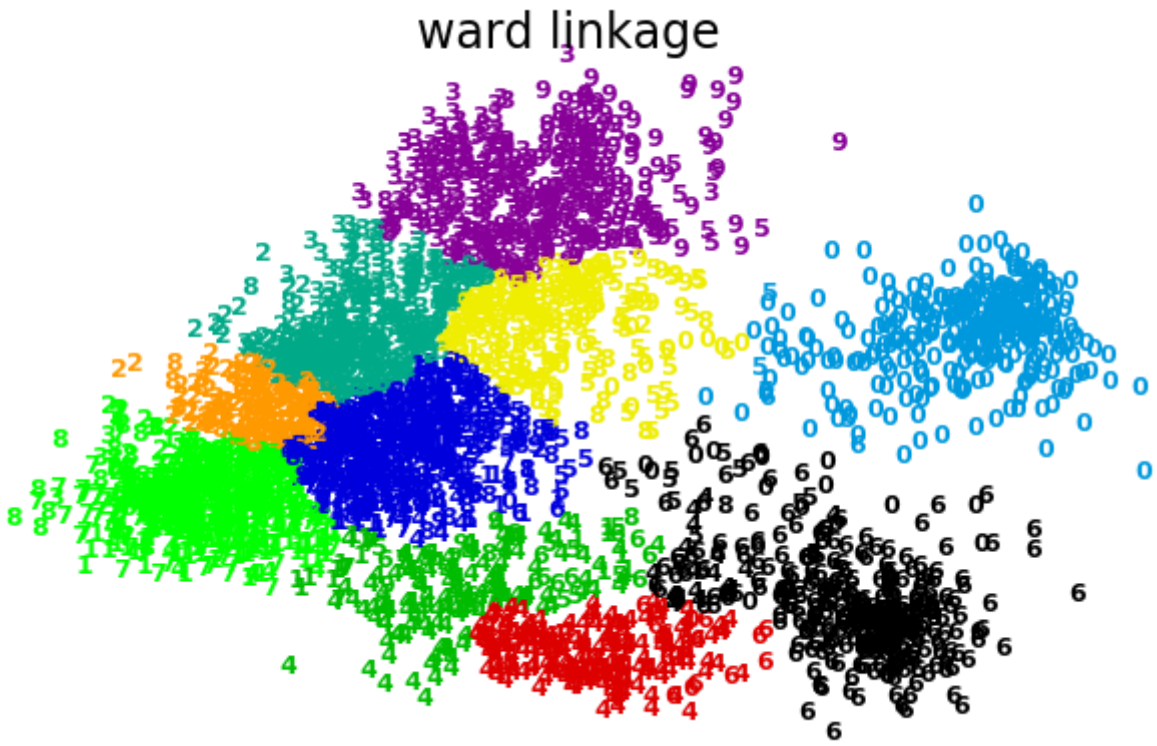
Agglomerative Feature

Groups together features that look very similar, thus decreasing the number of features. It is a dimensionality reduction tool, see Unsupervised dimensionality reduction.

- Linkages

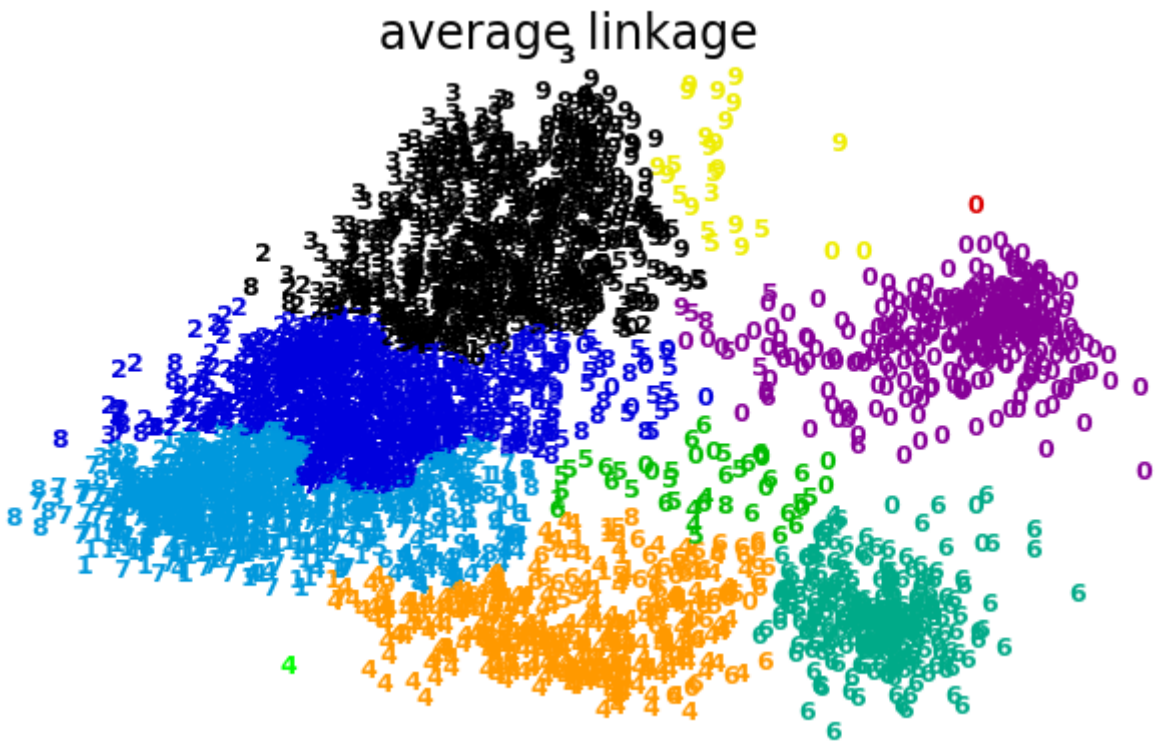
Ward

Minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.



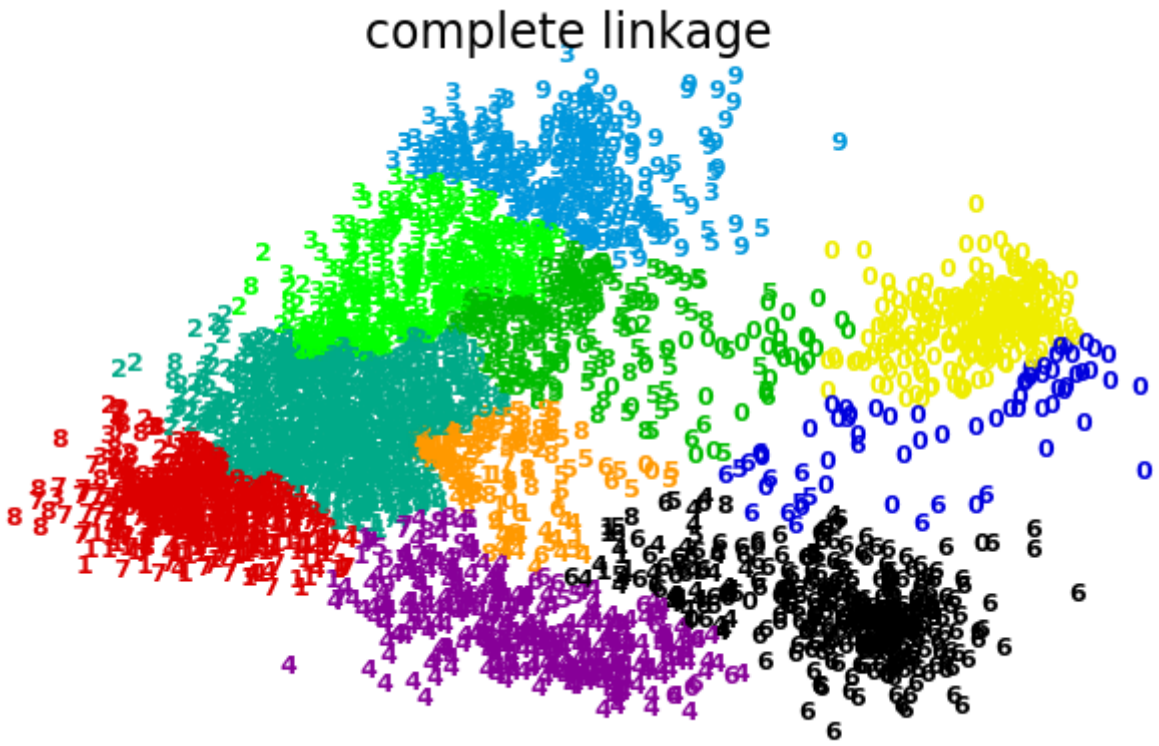
Average

Average linkage minimizes the average of the distances between all observations of pairs of clusters.



Complete

Maximum or complete linkage minimizes the maximum distance between observations of pairs of clusters.

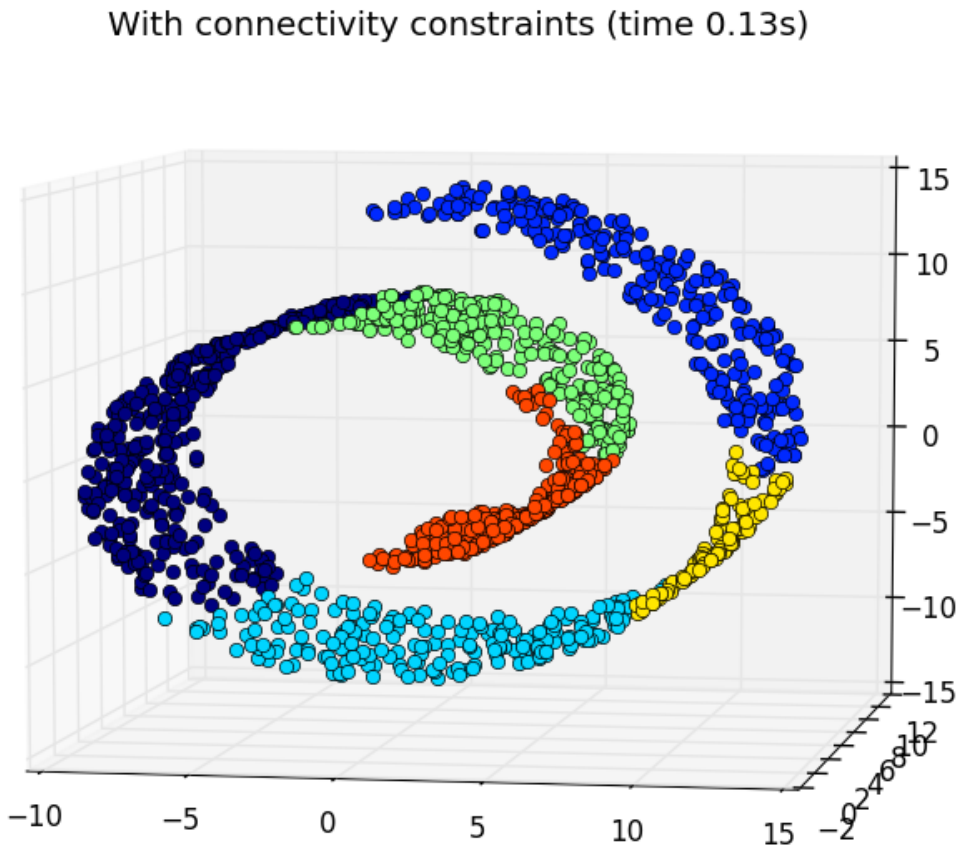


Agglomerative cluster has a “rich get richer” behavior that leads to uneven cluster sizes. In this regard, complete linkage is the worst strategy, and Ward gives the most regular sizes. However, the affinity (or distance used in clustering) cannot be varied with Ward, thus for non Euclidean metrics,

average linkage is a good alternative.

Adding ConnectivityConstraints

An interesting aspect of AgglomerativeClustering is that connectivity constraints can be added to this algorithm (only adjacent clusters can be merged together), through a connectivity matrix that defines for each sample the neighboring samples following a given structure of the data. For instance, in the swiss-roll example below, the connectivity constraints forbid the merging of points that are not adjacent on the swiss roll, and thus avoid forming clusters that extend across overlapping folds of the roll. unstructured structured



Varing Metrics

- These constraint are useful to impose a certain local structure, but they also make the algorithm faster, especially when the number of the samples is high.
- The connectivity constraints are imposed via an connectivity matrix: a scipy sparse matrix that has elements only at the intersection of a row and a column with indices of the dataset that should be connected. This matrix can be constructed from a-priori information: for instance, you may wish to cluster web pages by only merging pages with a link pointing from one to another. It can also be learned from the data,
- Average and complete linkage can be used with a variety of distances (or affinities), in particular Euclidean distance (l2), Manhattan distance (or Cityblock, or l1), cosine distance, or any precomputed affinity matrix.

    - l1 distance is often good for sparse features, or sparse noise: ie many of the features are zero, as in text mining using occurences of rare words.
    - cosine distance is interesting because it is invariant to global scalings of the signal.
- The guidelines for choosing a metric is to use one that maximizes the distance between samples in different classes, and minimizes that within each class.

Birch

The Birch builds a tree called the Characteristic Feature Tree (CFT) for the given data. The data is essentially lossy compressed to a set of Characteristic Feature nodes (CF Nodes). The CF Nodes have a number of subclusters called Characteristic Feature subclusters (CF Subclusters) and these CF Subclusters located in the non-terminal CF Nodes can have CF Nodes as children.

The CF Subclusters hold the necessary information for clustering which prevents the need to hold the entire input data in memory. This information includes:

- Number of samples in a subcluster.
- Linear Sum - A n-dimensional vector holding the sum of all samples
- Squared Sum - Sum of the squared L2 norm of all samples.
- Centroids - To avoid recalculation linear sum / n_samples.
- Squared norm of the centroids.
- The Birch algorithm has two parameters:

  Branching

  Limits the number of subclusters in a node

  Thresholdg

  Limits the distance between the entering sample and the existing subclusters

- This algorithm can be viewed as an instance or data reduction method, since it reduces the input data to a set of subclusters which are obtained directly from the leaves of the CFT. This reduced data can be further processed by feeding it into a global clusterer. This global clusterer can be set by n_clusters. If n_clusters is set to None, the subclusters from the leaves are directly read off, otherwise a global clustering step labels these subclusters into global clusters (labels) and the samples are mapped to the global label of the nearest subcluster. image::describe/Birch.png[alt="Birch",align="center"]

Algorithm

- A new sample is inserted into the root of the CF Tree which is a CF Node. It is then merged with the subcluster of the root, that has the smallest radius after merging, constrained by the threshold and branching factor conditions. If the subcluster has any child node, then this is done repeatedly till it reaches a leaf. After finding the nearest subcluster in the leaf, the properties of this subcluster and the parent subclusters are recursively updated.
- If the radius of the subcluster obtained by merging the new sample and the nearest subcluster is greater than the square of the threshold and if the number of subclusters is greater than the branching factor, then a space is temporarily allocated to this new sample. The two farthest subclusters are taken and the subclusters are divided into two groups on the basis of the distance between these subclusters.
- If this split node has a parent subcluster and there is room for a new subcluster, then the parent is split into two. If there is no room, then this node is again split into two and the process is continued recursively, till it reaches the root.

Birch or KMeans

- Birch does not scale very well to high dimensional data. As a rule of thumb if n_features is greater than twenty, it is generally better to use MiniBatchKMeans.
- If the number of instances of data needs to be reduced, or if one wants a large number of subclusters either as a preprocessing step or otherwise, Birch is more useful than MiniBatchKMeans.

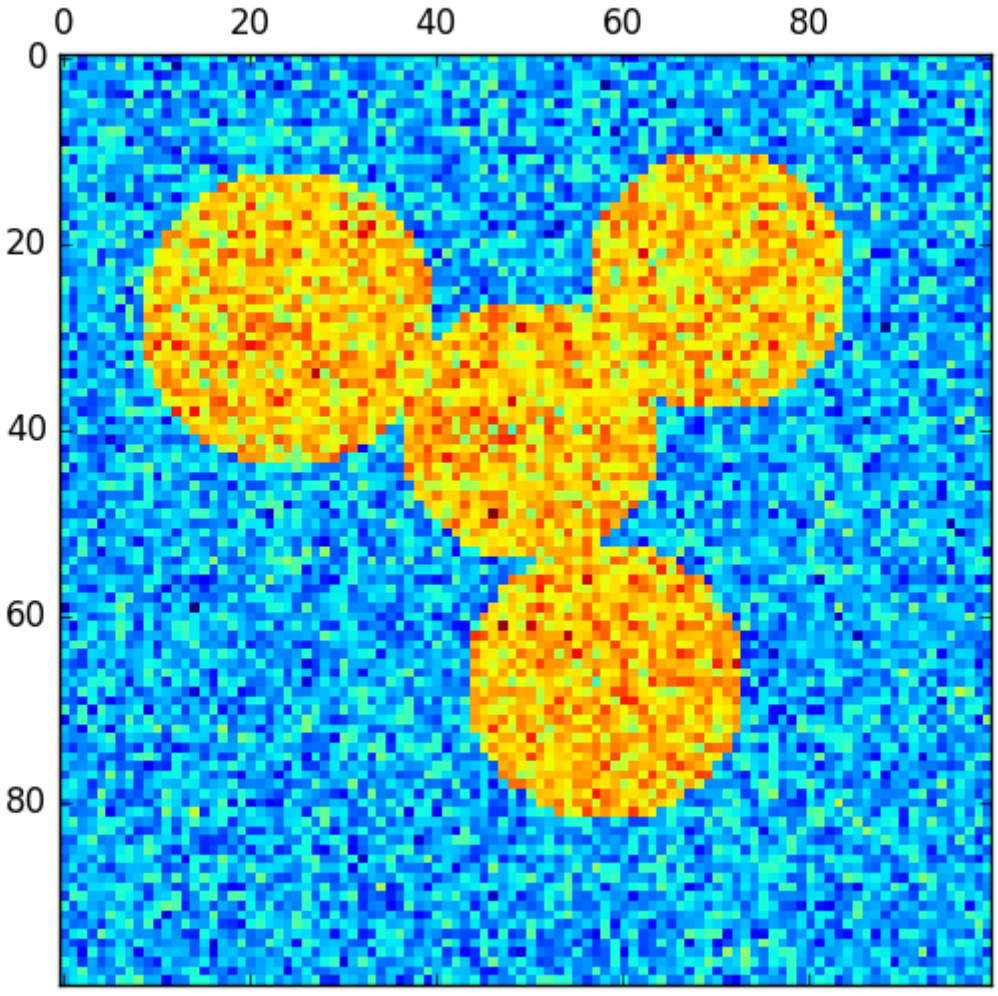
Distribution

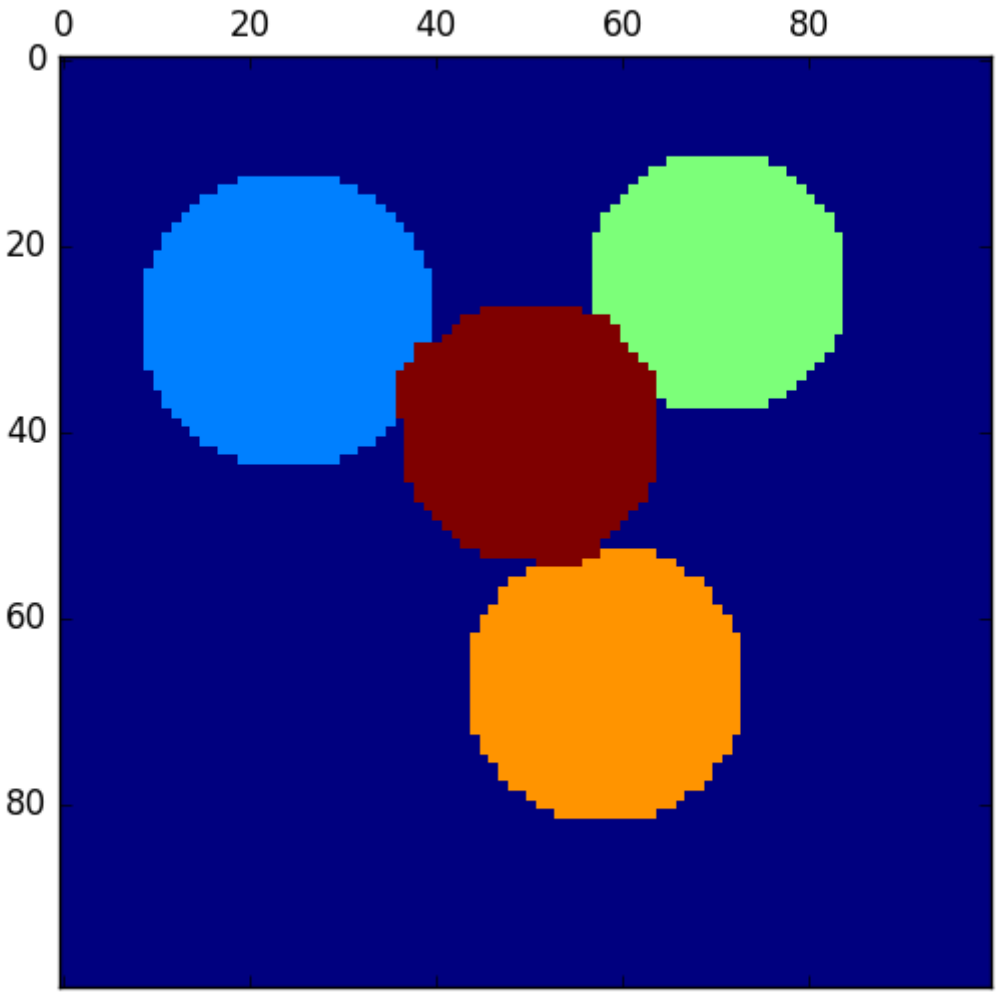
The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A convenient property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

While the theoretical foundation of these methods is excellent, they suffer from one key problem known as overfitting, unless constraints are put on the model complexity. A more complex model will usually be able to explain the data better, which makes choosing the appropriate model complexity inherently difficult.

Spectral

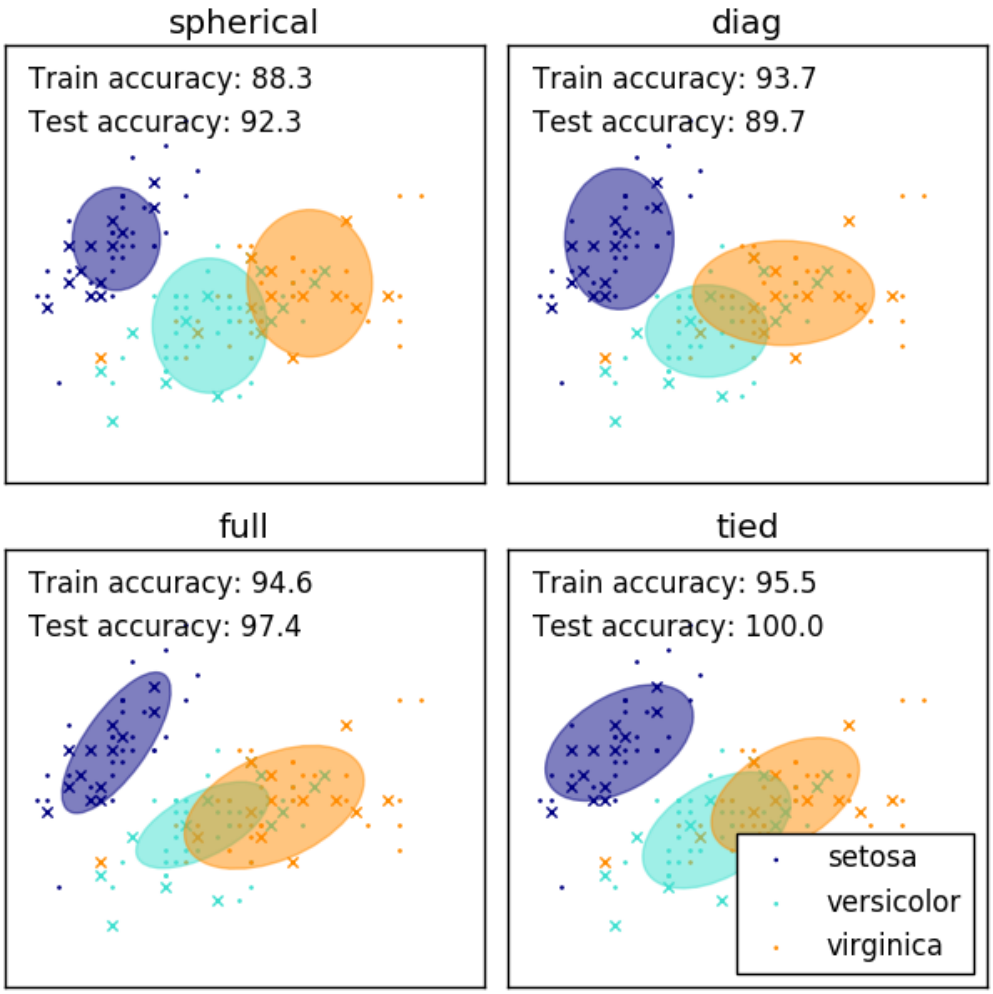
- SpectralClustering does a low-dimension embedding of the affinity matrix between samples, followed by a KMeans in the low dimensional space. It is especially efficient if the affinity matrix is sparse and the pyamg module is installed. SpectralClustering requires the number of clusters to be specified. It works well for a small number of clusters but is not advised when using many clusters.
- For two clusters, it solves a convex relaxation of the normalised cuts problem on the similarity graph: cutting the graph in two so that the weight of the edges cut is small compared to the weights of the edges inside each cluster. This criteria is especially interesting when working on images: graph vertices are pixels, and edges of the similarity graph are a function of the gradient of the image.




0
20
40
60
80
0
20
40
60
80

Expectation Max

- Simplifies difficult maximum likelihood problems.
- Generates strong statistics for interpretation.
- Not scalable



Strengths	Strong Statistics	Confidence Ellipsoids	Bayesian Criteria	x
Use For	Density Estimation	Flat geometry	Assessment	Sample Assignment

How It Works

- The expectation-maximization (EM) algorithm for fits a Gaussian Mixture models. It can also draw confidence ellipsoids for multivariate models, and compute the Bayesian Information Criterion to assess the number of clusters in the data. A Gausian Mixture fit method is provided that learns a Gaussian Mixture Model from train data. Given test data, it can assign to each sample the class of the Gaussian it mostly probably belong to using the GMM.predict method.
- The Gausie different options to constrain the covariance of the difference classes estimated: spherical, diagonal, tied or full covariance.

Benefits

Speed

It is the fastest algorithm for learning mixture models

Agnostic

This algorithm maximizes only the likelihood, it will not bias the means towards zero, or bias the cluster sizes to have specific structures that might or might not apply.

Drawbacks

Singularities

when one has insufficiently many points per mixture, estimating the covariance matrices becomes difficult, and the algorithm is known to diverge and find solutions with infinite likelihood unless one regularizes the covariances artificially.

Number of components

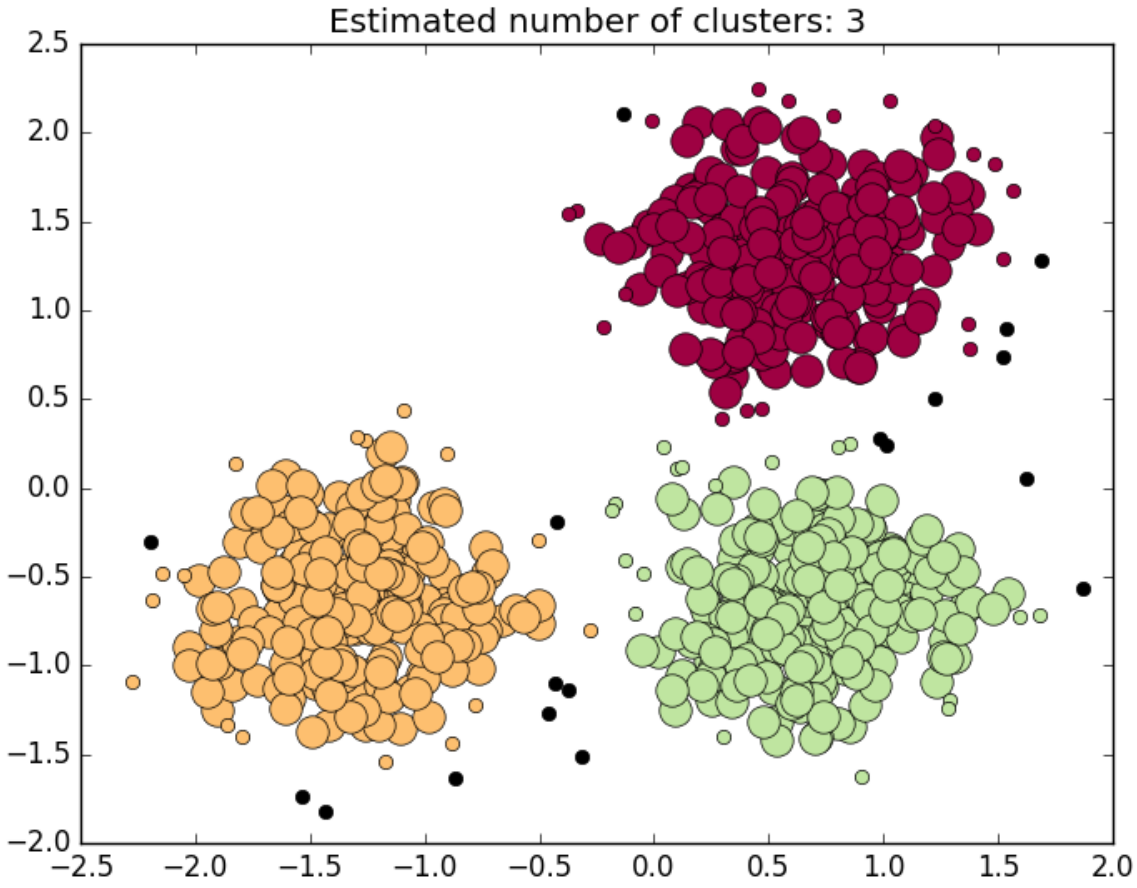
this algorithm will always use all the components it has access to, needing heldout data or information theoretical criteria to decide how many components to use in the absence of external cues.

Density

Defines clusters as connected dense regions in the data space.

DBScan

- The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped. The central component to the DBSCAN is the concept of core samples, which are samples that are in areas of high density. A cluster is therefore a set of core samples, each close to each other (measured by some distance measure) and a set of non-core samples that are close to a core sample (but are not themselves core samples). There are two parameters to the algorithm, min_samples and eps, which define formally what we mean when we say dense. Higher min_samples or lower eps indicate higher density necessary to form a cluster.
- More formally, we define a core sample as being a sample in the dataset such that there exist min_samples other samples within a distance of eps, which are defined as neighbors of the core sample. This tells us that the core sample is in a dense area of the vector space. A cluster is a set of core samples that can be built by recursively taking a core sample, finding all of its neighbors that are core samples, finding all of their neighbors that are core samples, and so on. A cluster also has a set of non-core samples, which are samples that are neighbors of a core sample in the cluster but are not themselves core samples. Intuitively, these samples are on the fringes of a cluster.
- Any core sample is part of a cluster, by definition. Any sample that is not a core sample, and is at least eps in distance from any core sample, is considered an outlier by the algorithm.
- In the figure below, the color indicates cluster membership, with large circles indicating core samples found by the algorithm. Smaller circles are non-core samples that are still part of a cluster. Moreover, the outliers are indicated by black points below.



More

The most popular[10] density based clustering method is DBSCAN.[11] In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering, it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times

Implementation

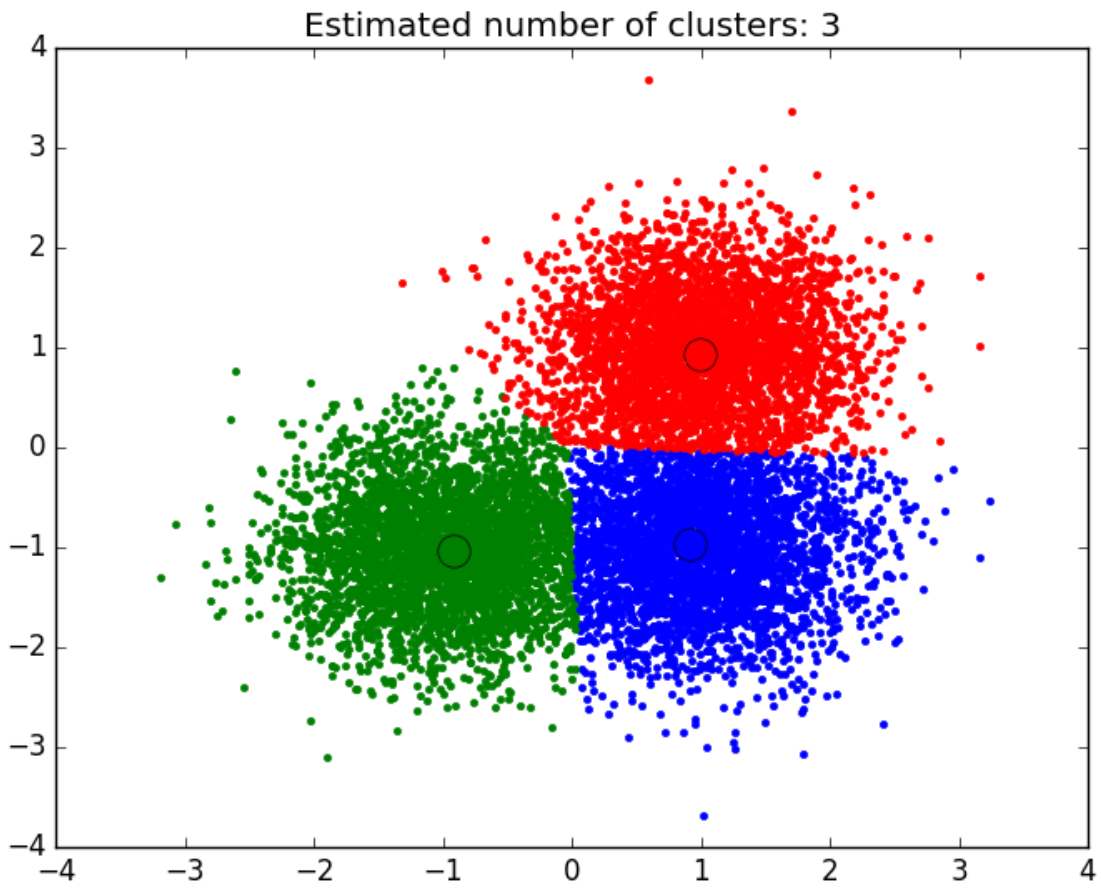
- The algorithm is non-deterministic, but the core samples will always belong to the same clusters (although the labels may be different). The non-determinism comes from deciding to which cluster a non-core sample belongs. A non-core sample can have a distance lower than eps to two core samples in different clusters. By the triangular inequality, those two core samples must be more distant than eps from each other, or they would be in the same cluster. The non-core sample is assigned to whichever cluster is generated first, where the order is determined randomly. Other than the ordering of the dataset, the algorithm is deterministic, making the results relatively stable between runs on the same data.

- The current implementation uses ball trees and kd-trees to determine the neighborhood of points, which avoids calculating the full distance matrix (as was done in scikit-learn versions before 0.14). The possibility to use custom metrics is retained; for details, see NearestNeighbors.

MeanShift

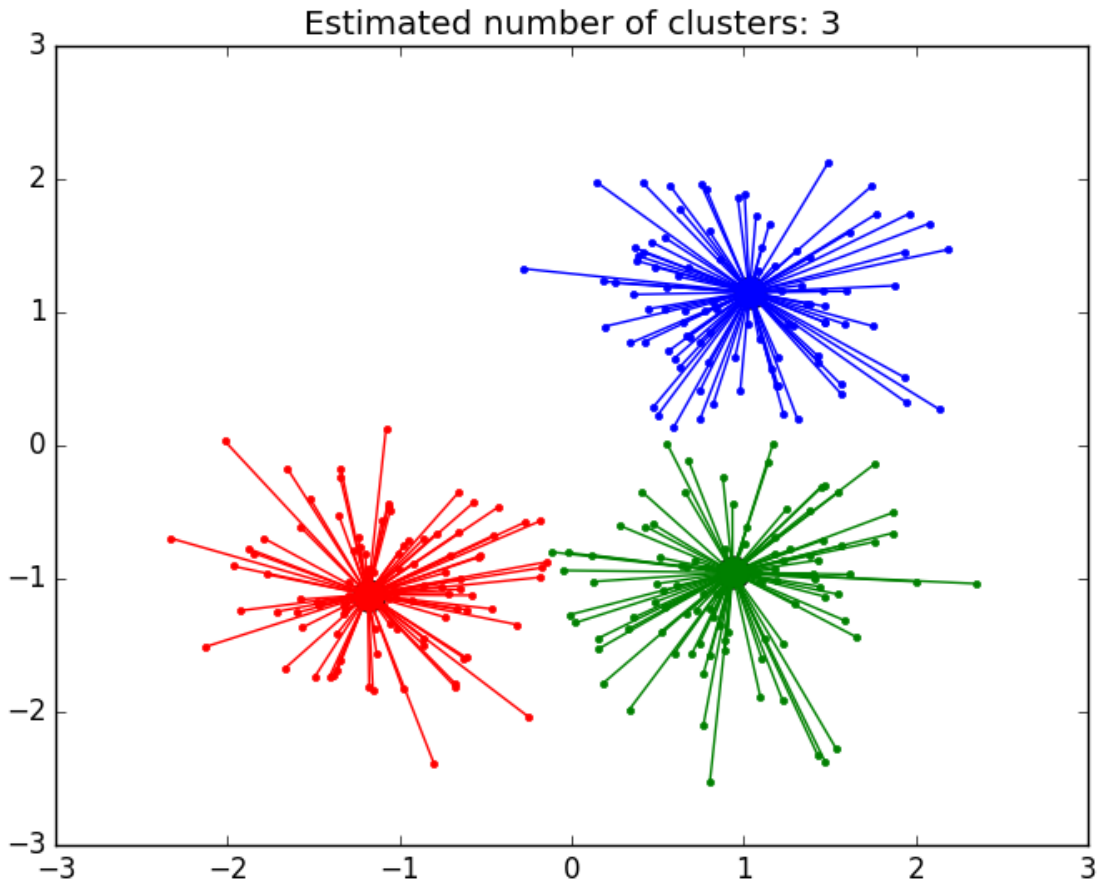
- Discover blobs in a smooth density of samples. It is a centroid based algorithm, which works by updating candidates for centroids to be the mean of the points within a given region. These candidates are then filtered in a post-processing stage to eliminate near-duplicates to form the final set of centroids.
- Given a candidate centroid x_i for iteration t, the candidate is updated according to the following equation:

- The algorithm automatically sets the number of clusters, instead of relying on a parameter bandwidth, which dictates the size of the region to search through. This parameter can be set manually, but can be estimated using the provided estimate_bandwidth function, which is called if the bandwidth is not set.
- The algorithm is not highly scalable, as it requires multiple nearest neighbor searches during the execution of the algorithm. The algorithm is guaranteed to converge, however the algorithm will stop iterating when the change in centroids is small.
- Labelling a new sample is performed by finding the nearest centroid for a given sample.



Affinity

- Creates clusters by sending messages between pairs of samples until convergence.
- A dataset is then described using a small number of exemplars, which are identified as those most representative of other samples.
- The messages sent between pairs represent the suitability for one sample to be the exemplar of the other, which is updated in response to the values from other pairs.
- This updating happens iteratively until convergence, at which point the final exemplars are chosen, and hence the final clustering is given.



Benefits

Drawbacks

- he main drawback of Affinity Propagation is its complexity.
- The algorithm has a time complexity of the order O(N^2 T),
- Further, the memory complexity is of the order O(N^2) if a dense similarity matrix is used, but reducible if a sparse similarity matrix is used.
- This makes Affinity Propagation most appropriate for small to medium sized datasets.
- A study comparing affinity propagation and Markov clustering on protein interaction graph partitioning found Markov clustering to work better for that problem

Applications

- Better for certain computer vision and computational biology tasks, e.g. clustering of pictures of human faces and identifying regulated transcripts, than k-means,[1] even when k-means was allowed many random restarts and initialized using PCA.[2]

Association Rules

Apriori

- Apriori is an unsupervised machine learning algorithm that generates association rules from data by examining lot of properties.

Strengths	Implementation	Parallization	Lots of Properties	x
Use For	Associations	Interactions	Grouped Purchases	Auto Complete

How It Works

- Association rule implies that if an item A occurs, then item B also occurs with a certain probability.
- Most of the association rules generated are in the IF_THEN format.
  - For example, IF people buy an iPad THEN they also buy an iPad Case to protect it.
  - For the algorithm to derive such conclusions, it first observes the number of people who bought an iPad case while purchasing an iPad.
  - This way a ratio is derived like out of the 100 people who purchased an iPad, 85 people also purchased an iPad case.
- If an item set occurs frequently then all the subsets of the item set, also occur frequently.
- If an item set occurs infrequently then all the supersets of the item set have infrequent occurrence.

Benefits

- It is easy to implement and can be parallelized easily.
- Apriori implementation makes use of large item set properties.

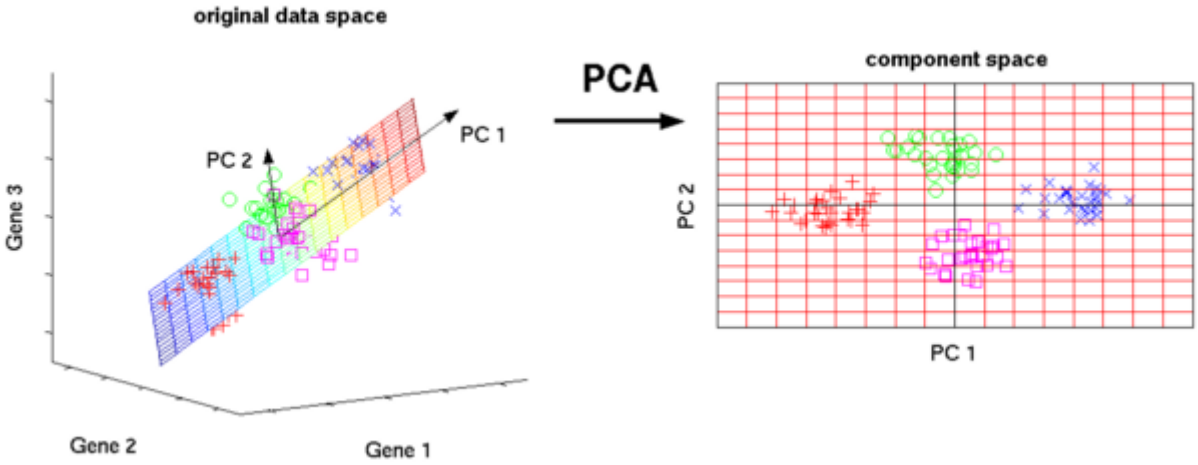
Application

- Detecting Adverse Drug Reactions

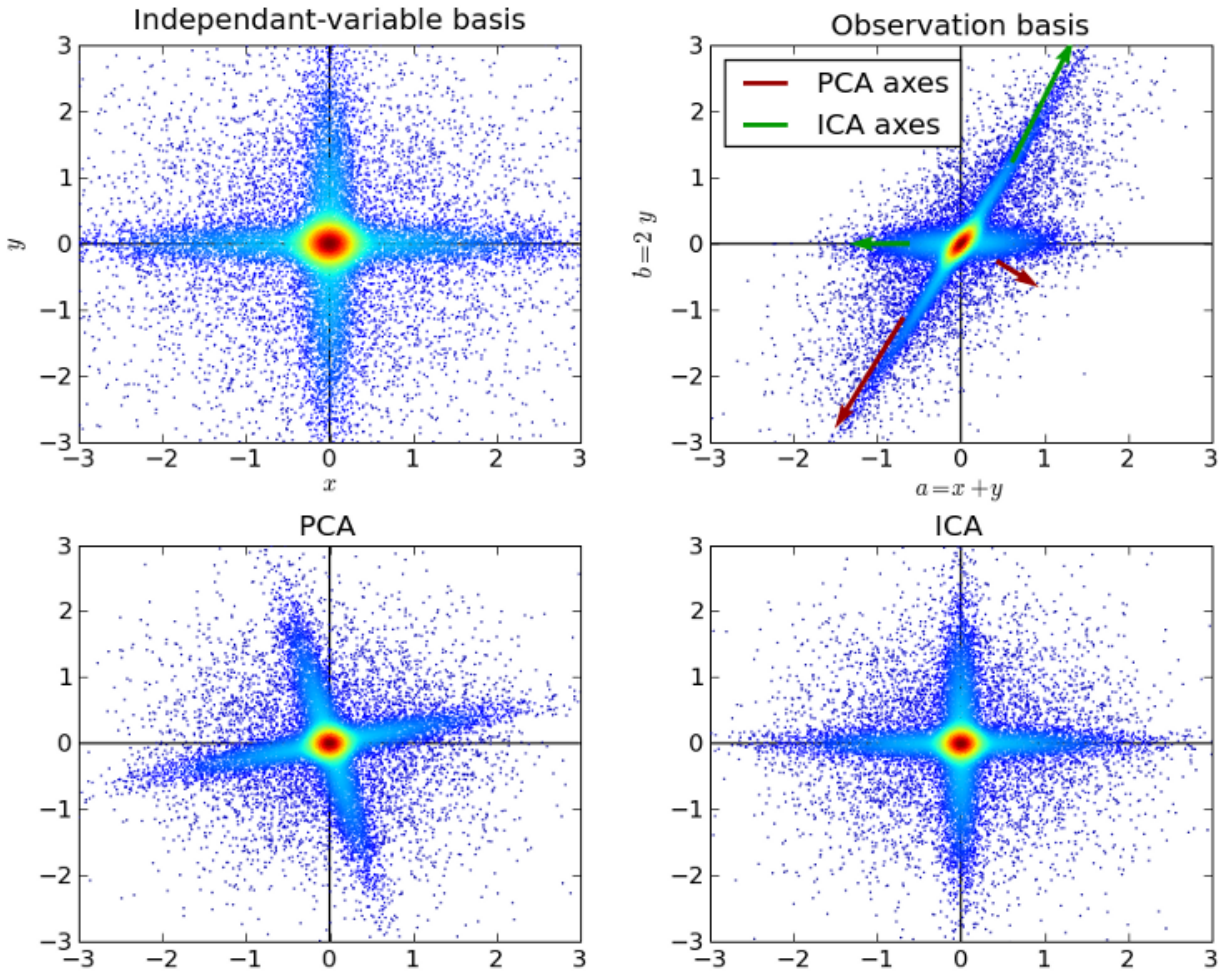
    - Apriori algorithm is used for association analysis on healthcare data like-the drugs taken by patients, characteristics of each patient, adverse ill-effects patients experience, initial diagnosis, etc.
    - This analysis produces association rules that help identify the combination of patient characteristics and medications that lead to adverse side effects of the drugs.
- Market Basket Analysis
    - Many e-commerce giants like Amazon use Apriori to draw data insights on which products are likely to be purchased together and which are most responsive to promotion.
    - For example, a retailer might use Apriori to predict that people who buy sugar and flour are likely to buy eggs to bake a cake.
- Auto-Complete Applications#
    - Google auto-complete is another popular application of Apriori
    - When the user types a word, the search engine looks for other associated words that people usually type after a specific word.

Dimensionality

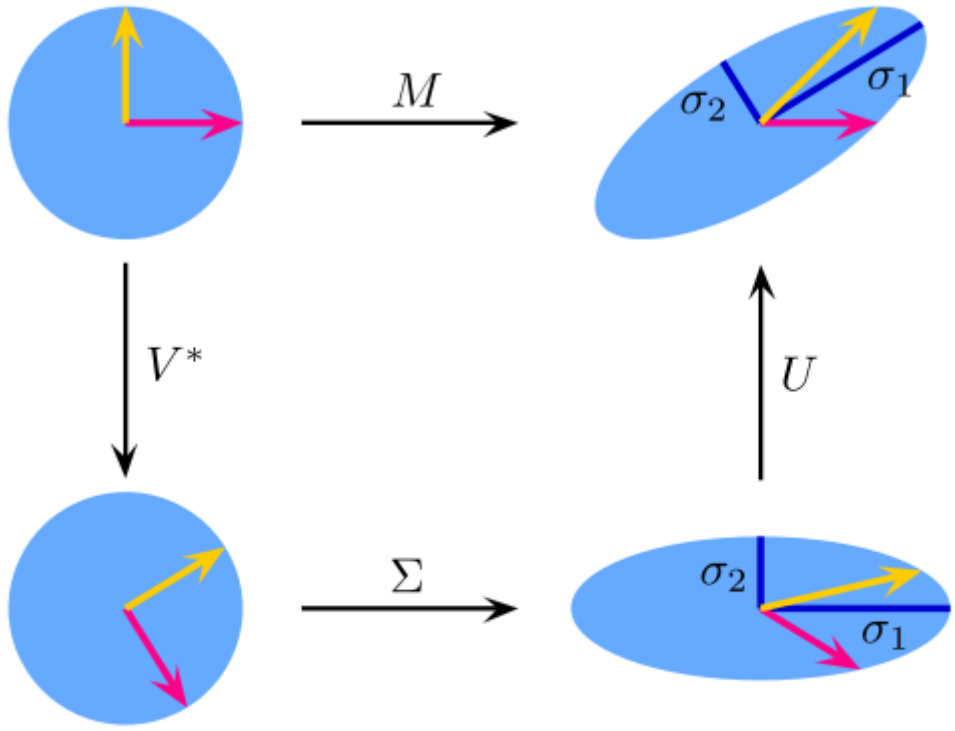
Principle Components



Independent Components



Discriminant Analysis



$$M = U \cdot \Sigma \cdot V^*$$

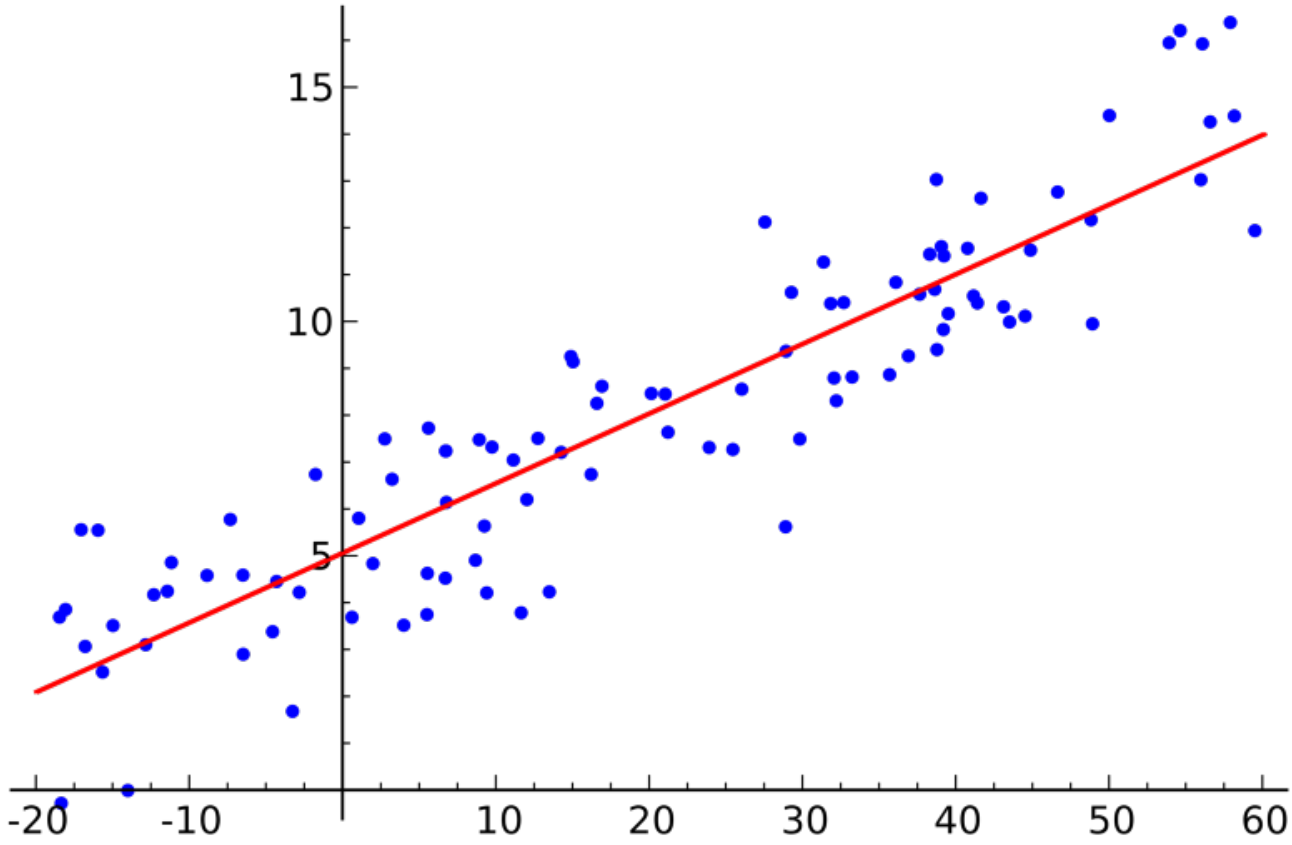
4. Distill


Distill
SemiSupervise
Regress
Linear
Linear
Logistic
Regularized
Ridge
Lasso
Elastic Net
Least Angle
Probabilistic
Bayesian
Auto Relevance
Orthogonal Match
Gausian Process
Other
Stepwise
Isotomic
Robust
Model
Representation
Smoothing
Regularization
Wavelet
Kernel
Assessment
Bias Variance
Cross Validation
Bootstrap
Selection
Estimates
Errors
Fuzzy Measure
Inference
Likelyhood
Bayesian
Expectation Max
Bagging
Stochastic
Semi Super
Train Data
Transducive
Inductive
Assumptions
Smoothness
Cluster
Manifold
Methods
Low Density
Graph Based
Heuristics
Redundancy
Label
Propagate
Spreading
Patterns
Hypothesis
T Test
Topic
Chi Squared
Co Variance
Empirical
Shrunk
Sparse Inv
Robust
Reveal
Detect
Exceptions
Clusters
Trends
Calculate
Interpolate
Extrapalate
Recommend
Context
Collab Filter
Content Based
Knowledge Based
Graph Based

Regress

Linear Regression

- Linear Regression shows the relationship between 2 variables
- Illustrates the change in one variable impacts the other.



Strengths	Simple	Interpretable	Min Tuning	Fast
Use For	Estimations	Risk	x	x

How It Works

- The algorithm shows the impact on the dependent variable on changing the independent variable.
- The independent variables are referred as explanatory variables, as they explain the factors the impact the dependent variable.
- Dependent variable is often referred to as the factor of interest or predictor.

Benefits

- It is one of the most interpretable machine learning algorithms, making it easy to explain to others.
- It is easy of use as it requires minimal tuning.
- It is the mostly widely used machine learning technique that runs fast.

Applications

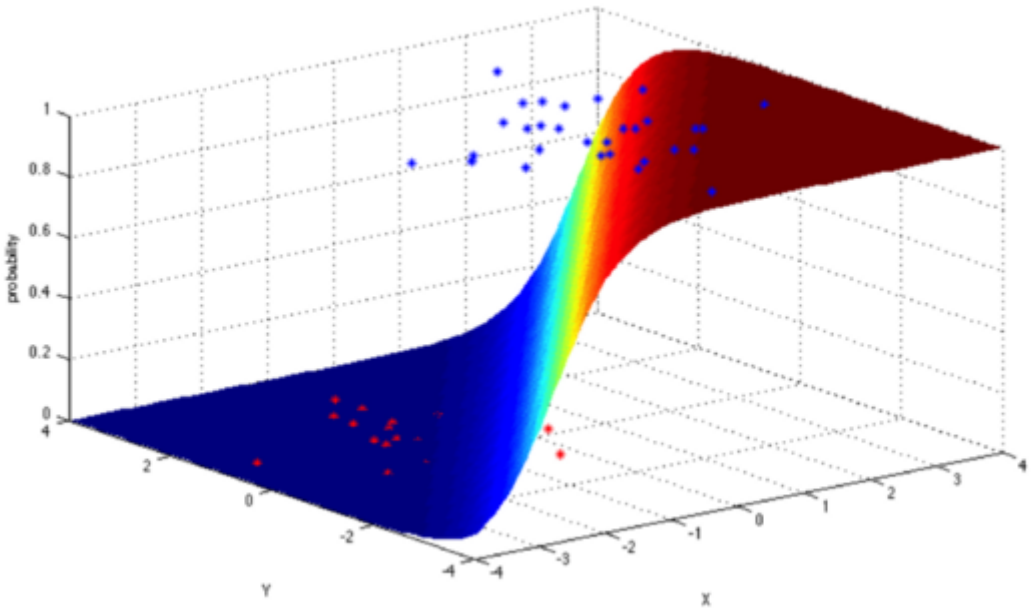
- Estimating Sales based on the trends
  - If a company observes steady increase in sales every month - a linear regression analysis of

the monthly sales data helps the company forecast sales in upcoming months.

- Risk Assessment
  - Assess riska involved in insurance or financial domain.
  - A health insurance company can do a linear regression on the number of claims per customer against age.
  - This analysis helps insurance companies find, that older customers tend to make more insurance claims.
  - Such analysis results play a vital role in important business decisions and are made to account for risk.

Logistic Regression

- Measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function
- Requires lots of data typically 20 data points per predictor.



Strengths	Non Linear	Distribution	Independence	x
Weaknesses	Identification	Discrete Outcomes	Independence	Over Confidence
Use For	Categories	Variance	x	x

How It Works

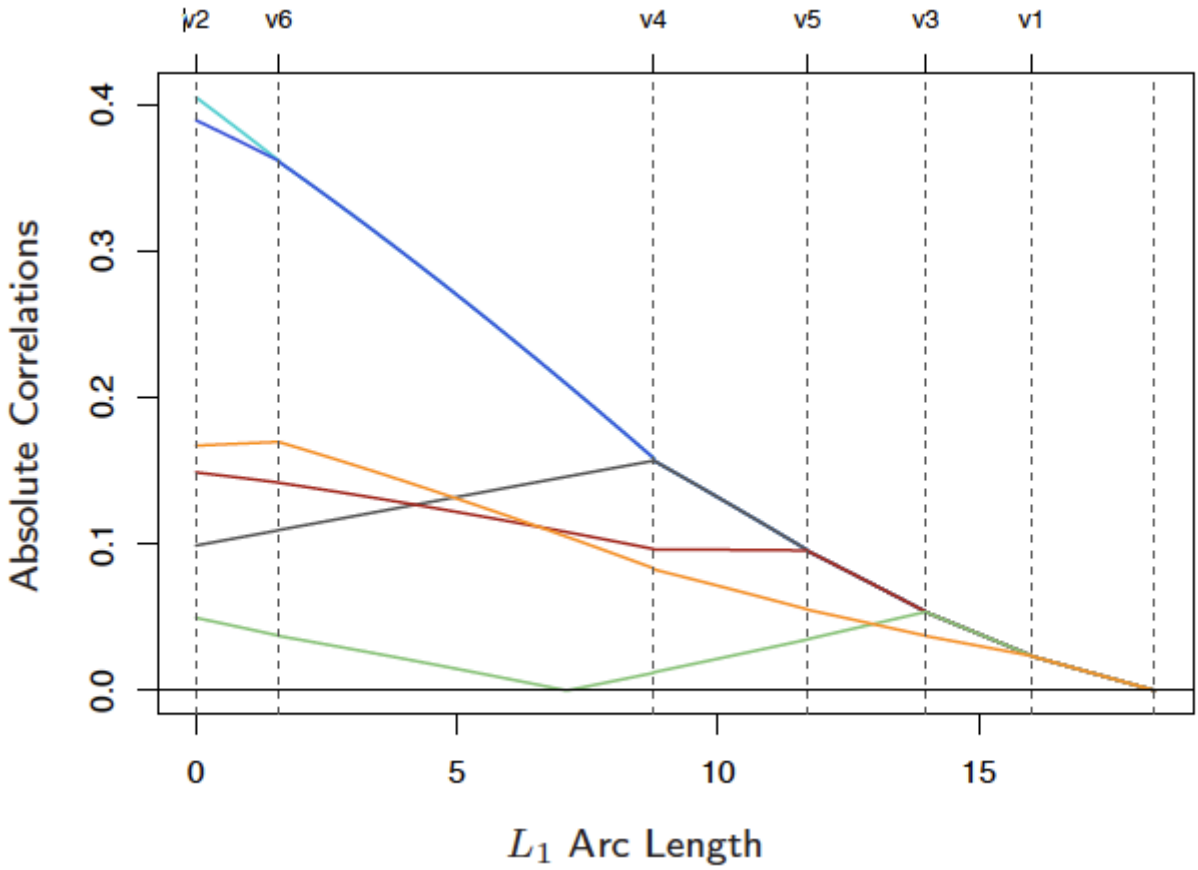
- In statistics, logistic regression is a regression model where the dependent variable (DV) is categorical.
- The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features).
- Measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function

Analogous to Linear

- A special case of the generalized linear model and thus analogous to linear regression.
- The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression.
- In particular the key differences of these two models can be seen in the following two features of logistic regression.
- First, the conditional distribution is a Bernoulli distribution rather than a Gaussian distribution because the dependent variable is binary.
- Second, the predicted values are probabilities and are therefore restricted to (0,1) through the logistic distribution function because logistic regression predicts the probability of particular outcomes.

Least Angle

- Least angle regression (LAR) can be viewed as a kind of “democratic” version of forward stepwise regression.

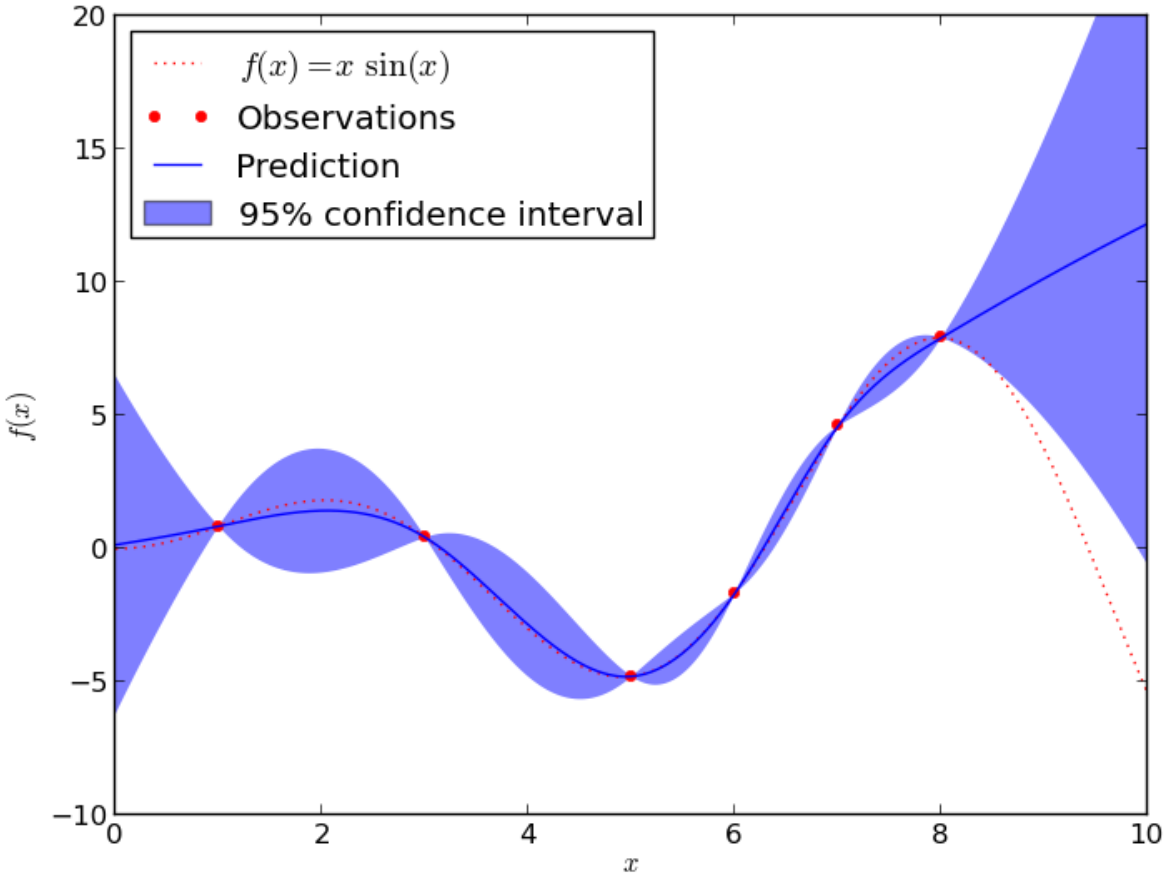


- LAR is intimately connected with the lasso, and in fact provides an extremely efficient algorithm for computing the entire lasso path

How It Works

- Forward stepwise regression builds a model sequentially, adding one variable at a time. At each step, it identifies the best variable to include in the active set, and then updates the least squares fit to include all the active variables.
- Least angle regression uses a similar strategy, but only enters “as much” of a predictor as it deserves. fit.

Gaussian Processes



Benefits

Interpolates

Observations can be interpolated (at least for regular correlation models).

Probabilistic

The prediction is probabilistic (Gaussian) so that one can compute empirical confidence intervals and exceedence probabilities that might be used to refit (online fitting, adaptive fitting) the prediction in some region of interest.

Versatile

Different linear regression models and correlation models can be specified. Common models are provided, but it is also possible to specify custom models provided they are stationary.

Drawbacks

Not Sparse

It uses the whole samples/features information to perform the prediction.

Inefficent

It loses efficiency in high dimensional spaces – namely when the number of features exceeds a few dozens. It might indeed give poor performance and it loses computational efficiency.

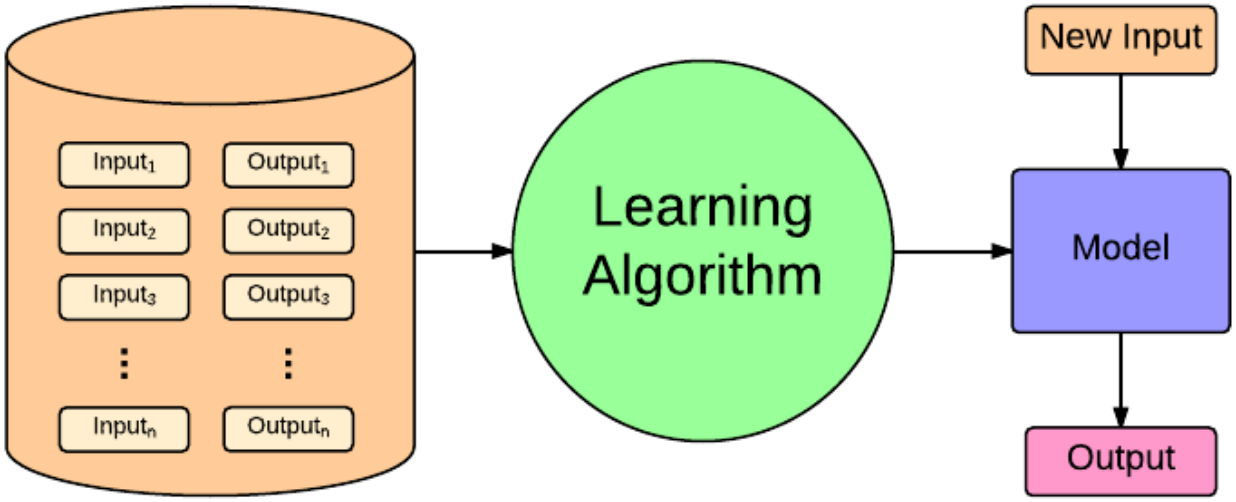
Post Classification

Classification is only a post-processing, meaning that one first need to solve a regression problem by providing the complete scalar float precision output y of the experiment one attempt to model.

5. Predict


Predict
Supervise
Classify
Naive Bayes
NB Gaussian
NB Multinomial
NB Bernoulli
Support Vector
Hyperplane
Kernel
Nearest Neighbor
K- NN
Classifiers
Multi- Class
Multi- Label
Arrange
Decision Tree
CART
C4.5
Random Forests
Ensemble
Bagging
Adaptive Boost
Gradient Boost
Neural
Perceptron
Back Prop
Deep Boltzmann
Deep Belief
Convolutional
Auto Encoders
Supervise
Review Data
Interface
Intelligence
Inference
Risk
Empirical
Structural
Noise
Non Linearity
Attribute
Represent
Dimensionality
Heterogeneity
Redundancy
Concept
Confidence
Interval
Error Bounds
Confidence
Crediable
Probability Success
Model Evaluation
Confusion Matrix
F- Measure
ROC Graph
Uncertainty
Catagories
Propagation
Applicablilty
Assumptions
Contraints
Percieve
Wavelet
Forecast
Hidden Markov

Supervised Learning



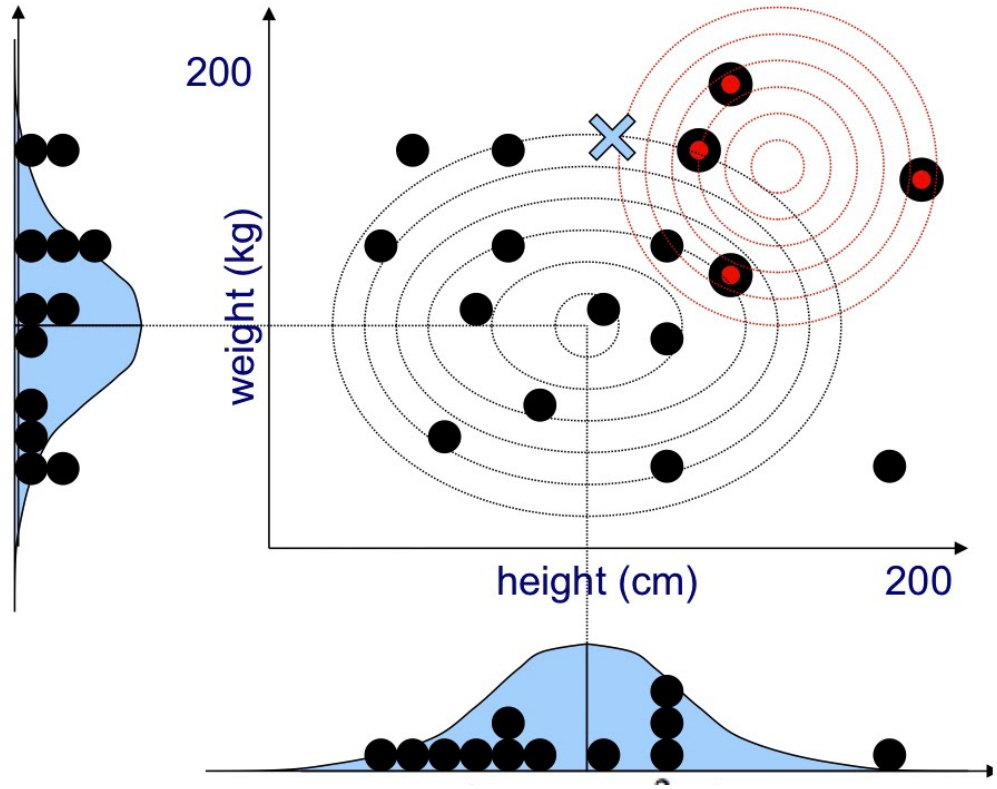
Comparison

Characteristic	Neural	SVM	Trees	MARS	k-NN
Natural	Poor	Poor	Good	Good	Poor
Missing Values	Poor	Poor	Good	Good	Good
Outliers	Poor	Poor	Good	Poor	Good
Monotone	Poor	Poor	Good	Poor	Poor
Scalability	Poor	Poor	Good	Good	Poor
Irrelevant Inputs	Poor	Poor	Good	Good	Poor
Linear Features	Good	Good	Poor	Poor	Fair
Interpretability	Poor	Poor	Fair	Good	Poor
Predictive	Good	Good	Poor	Fair	Good

Classify

Naïve Bayes

- It would be difficult and practically impossible to classify a web page, a document, an email or any other lengthy text notes manually.
- This is where Naïve Bayes Classifier machine learning algorithm comes to the rescue.



Classifier

- A classifier is a function that allocates a population's element value from one of the available categories.
- For instance, Spam Filtering is a popular application of Naïve Bayes algorithm.
- Spam filter here, is a classifier that assigns a label "Spam" or "Not Spam" to all the emails.

How it Works

Grouped by Similarities

- Naïve Bayes Classifier is amongst the most popular learning method grouped by similarities
- It works on the popular Bayes Theorem of Probability
  - to build machine learning models particularly for disease prediction and document classification. It is a simple classification of words based on Bayes Probability Theorem for subjective analysis of content.

Benefits

- Naïve Bayes Classifier algorithm performs well when the input variables are categorical.
- A Naïve Bayes classifier converges faster, requiring relatively little training data than other discriminative models like logistic regression

    - when the Naïve Bayes conditional independence assumption holds.
- it is easier to predict class of the test data set.
    - A good bet for multi class predictions as well.
    - Though it requires conditional independence assumption,
- Good performance in various application domains.

When to Use It

- If you have a moderate or large training data set.
- If the instances have several attributes.
- Attributes which describe the instances should be conditionally independent.

Applications

Sentiment Analysis

It is used at Facebook to analyse status updates expressing positive or negative emotions.

Document Categorization

Google uses document classification to index documents and find relevancy scores i.e. the PageRank. PageRank mechanism considers the pages marked as important in the databases that were parsed and classified using a document classification technique.

News Articles

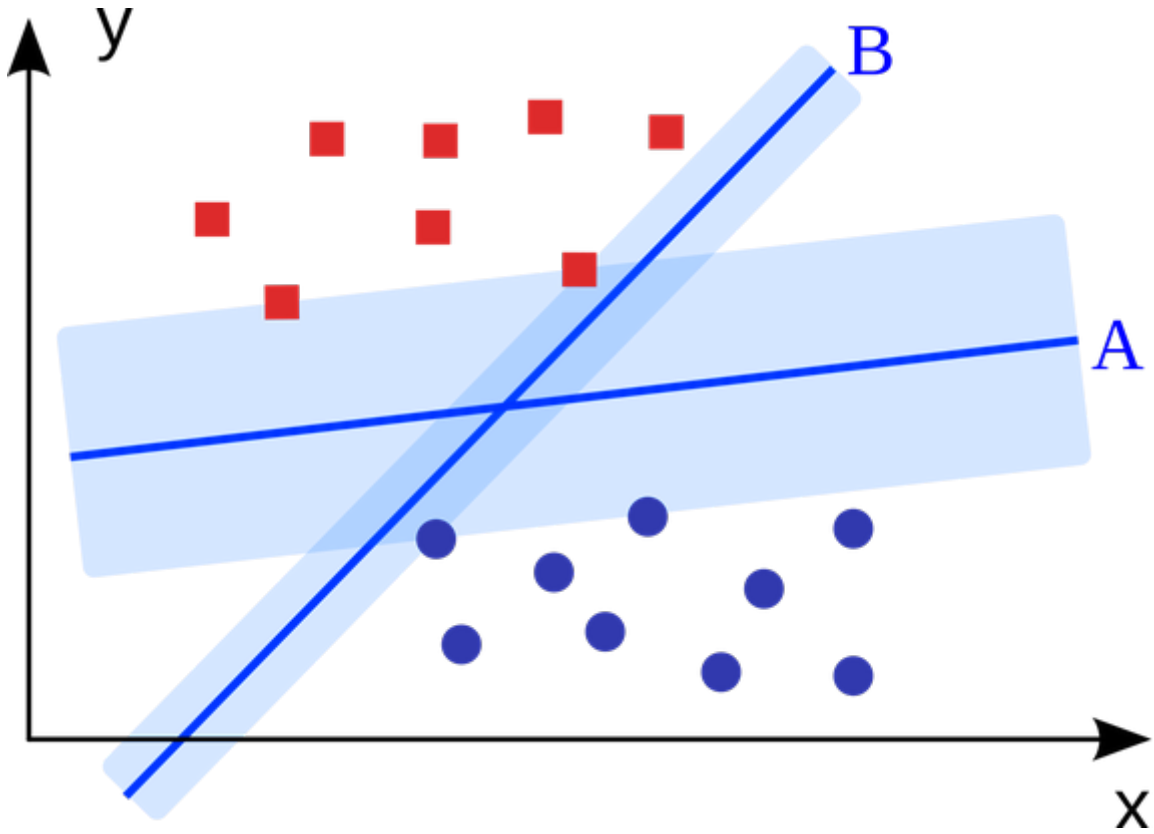
Technology, Entertainment, Sports, Politics, etc.

Email Spam Filtering

Google Mail uses Naïve Bayes algorithm to classify your emails as Spam or Not Spam

Support Vector Machine

- Support Vector Machine is a supervised machine learning algorithm for classification or regression problems
  - Where the dataset teaches SVM about the classes so that SVM can classify any new data.



How It Works

- Optimization over greedienss.
- It works by classifying the data into different classes by finding a line (hyperplane) which separates the training data set into classes.
  - As there are many such linear hyperplanes, SVM algorithm tries to maximize the distance between the various classes that are involved and this is referred as margin maximization.
  - If the line that maximizes the distance between the classes is identified, the probability to generalize well to unseen data is increased.

SVM's are classified into two categories:

- Linear SVM's – In linear SVM's the training data i.e. classifiers are separated by a hyperplane.
- Non-Linear SVM's- In non-linear SVM's it is not possible to separate the training data using a hyperplane.
  - For example, the training data for Face detection consists of group of images that are faces and another group of images that are not faces (in other words all other images in the world except faces).
  - Under such conditions, the training data is too complex that it is impossible to find a representation for every feature vector. Separating the set of faces linearly from the set of non-face is a complex task.

Benefits

Performance

SVM offers best classification performance (accuracy) on the training data.

Efficieny

SVM renders more efficiency for correct classification of the future data.

Assuptions

The best thing about SVM is that it does not make any strong assumptions on data.

No Overfit

It does not over-fit the data.

Dimensional

Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

Subset

Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Versatile

different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Drawbacks

Labeling

Requires full labeling of input data

Membership

Uncalibrated class membership probabilities

Two-class

Only directly applicable for two-class tasks.

Interpretation

Parameters of a solved model are difficult to interpret.

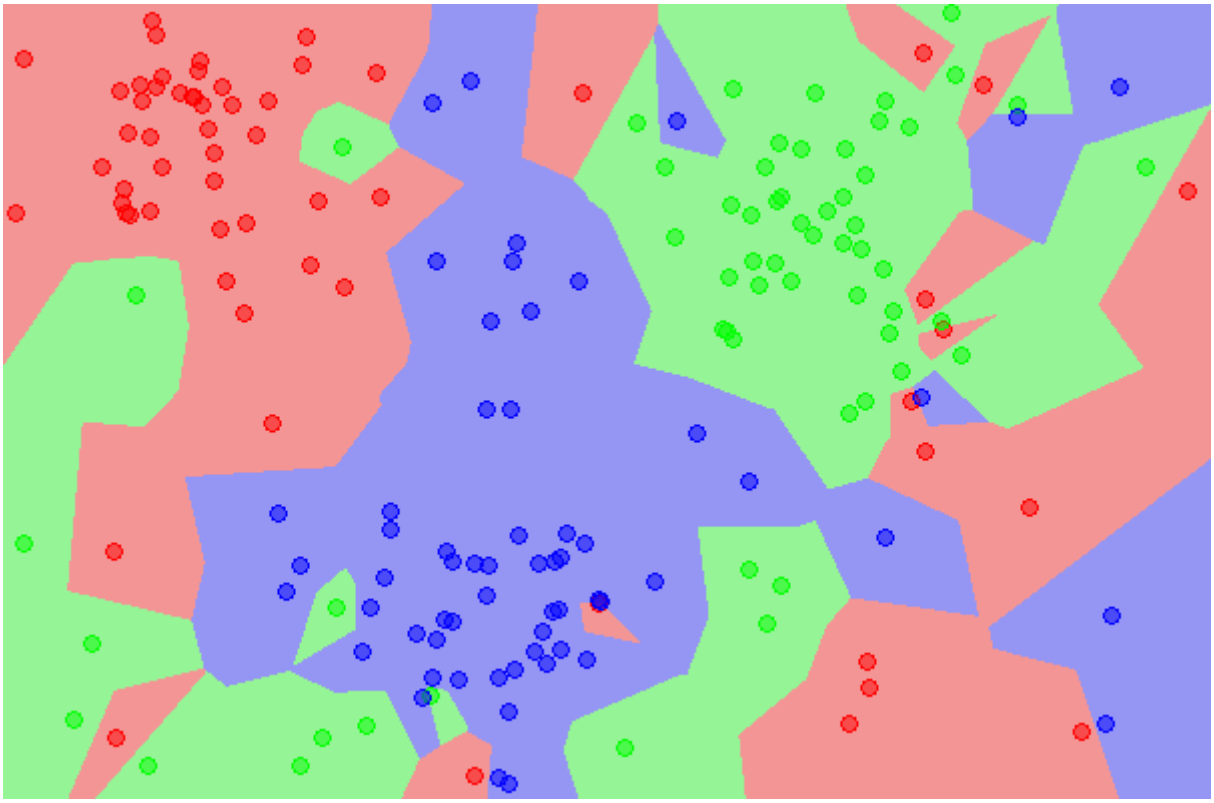
Applications

- Text and Hypertext Categoricalization
- Image Classification
- SVM is commonly used for stock market forecasting by various financial institutions.
  - For instance, it can be used to compare the relative performance of the stocks when compared to performance of other stocks in the same sector.
  - The relative comparison of stocks helps manage investment making decisions based on the

classifications made by the SVM learning algorithm.

Nearest Neighbor

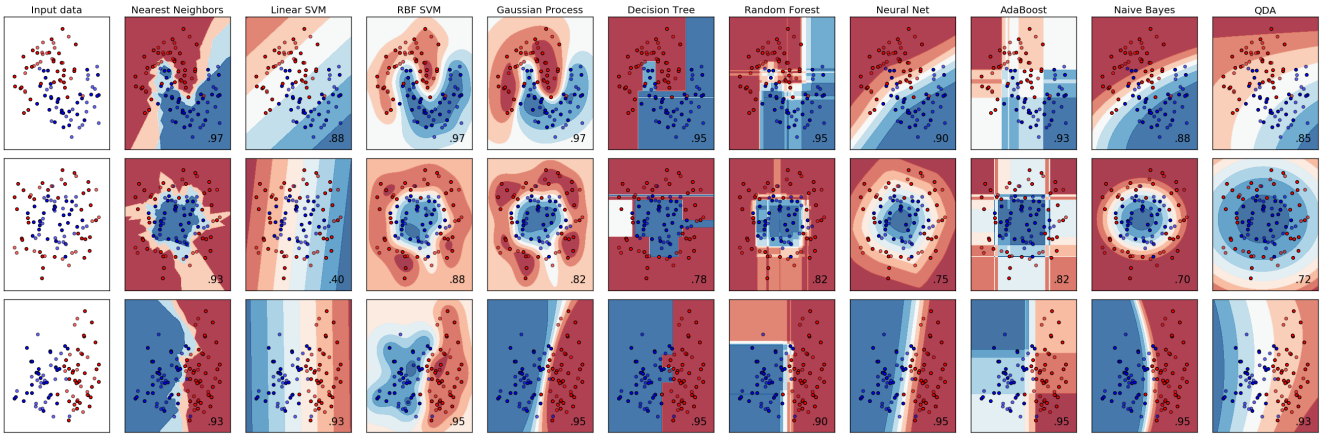
- In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method
- It can be used for classification and regression.[1]
- In both cases, the input consists of the k closest training examples in the feature space.



How It Works

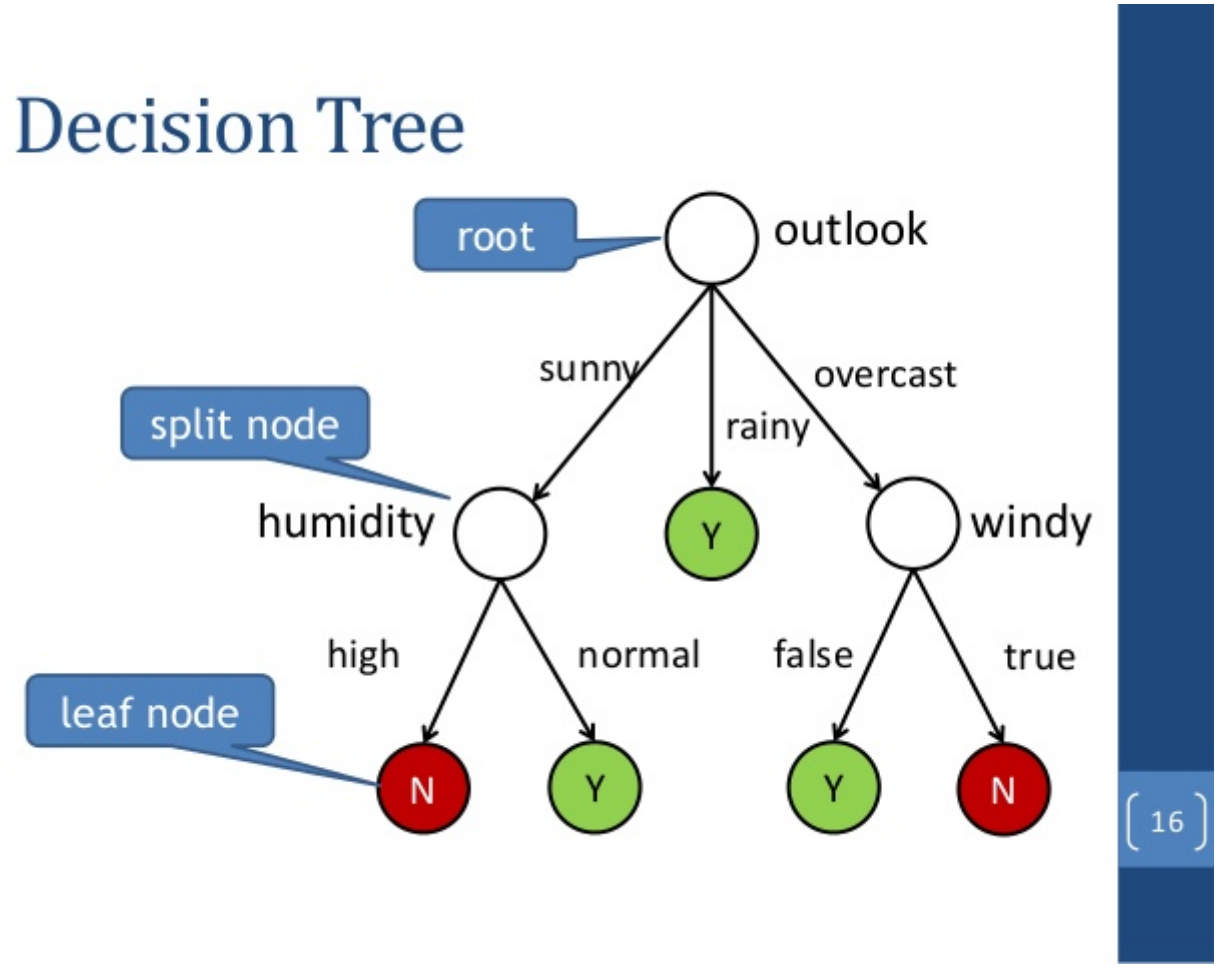
- The output depends on whether k-NN is used for classification or regression:
  - In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
  - In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.
- k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.
- Both for classification and regression, it can be useful to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.[2]
- The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

Classifiers



Arrange

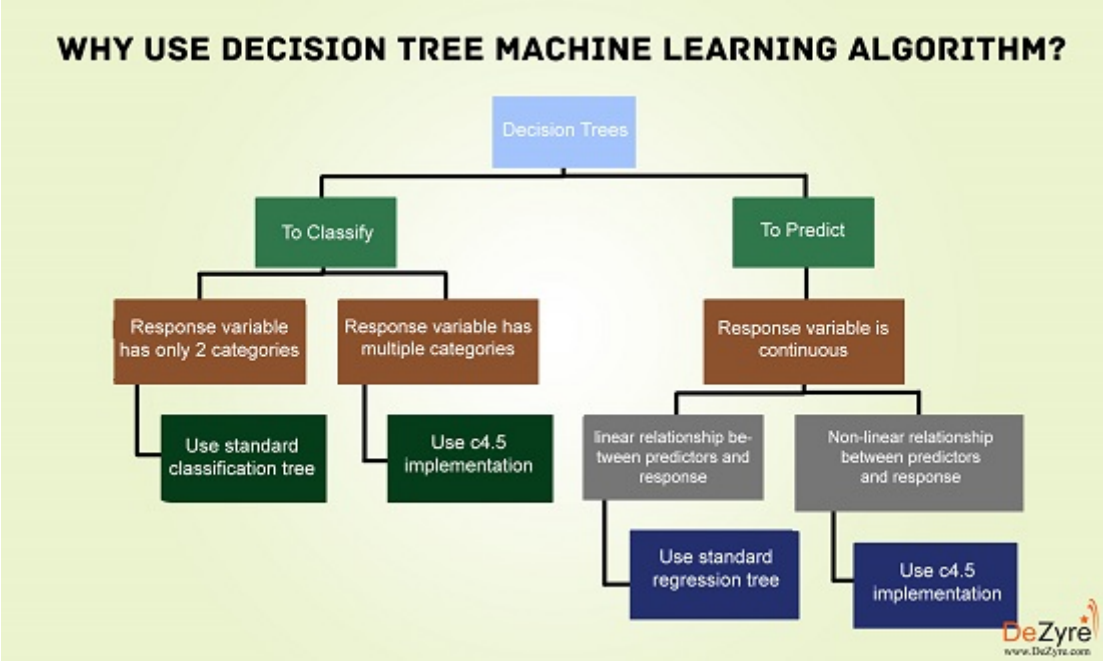
Decision Tree



In a decision tree

- The internal node represents a test on the attribute
- Each branch of the tree represents the outcome of the test
- Eac leaf node represents a particular class label i.e. the decision made after computing all of the attributes.
- The classification rules are represented through the path from root to the leaf node.

Types of Decision Trees



- Classification Trees- These are considered as the default kind of decision trees used to separate a dataset into different classes, based on the response variable. These are generally used when the response variable is categorical in nature.
- Regression Trees-When the response or target variable is continuous or numerical, regression trees are used. These are generally used in predictive type of problems when compared to classification.

Variables

Decision trees can also be classified into two types, based on the type of target variable It is the target variable that helps decide what kind of decision tree would be required for a particular problem.

- Continuous Variable Decision Trees
- Binary Variable Decision Trees.

Why should you use Decision Trees

- These machine learning algorithms help make decisions under uncertainty and help you improve communication, as they present a visual representation of a decision situation.
- Decision tree machine learning algorithms help a data scientist capture the idea that if a different decision was taken, then how the operational nature of a situation or model would have changed intensely.
- Decision tree algorithms help make optimal decisions by allowing a data scientist to traverse through forward and backward calculation paths.

When to use Decision Trees

- Decision trees are robust to errors and if the training data contains errors- decision tree algorithms will be best suited to address such problems.
- Decision trees are best suited for problems where instances are represented by attribute value pairs.

- If the training data has missing value then decision trees can be used, as they can handle missing values nicely by looking at the data in other columns.
- Decision trees are best suited when the target function has discrete output values.

Benefits

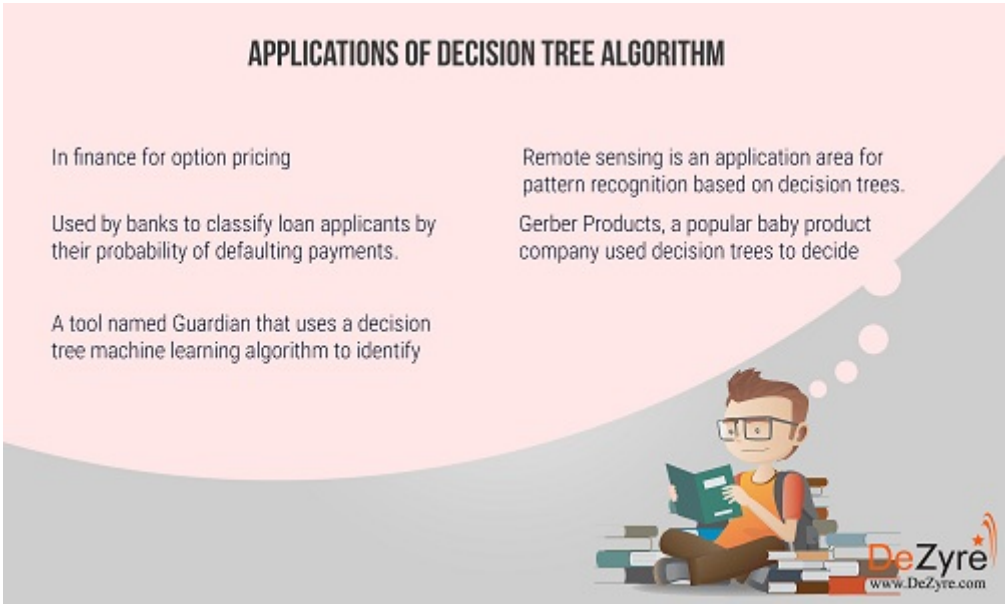
- Decision trees are very instinctual and can be explained to anyone with ease. People from a non-technical background, can also decipher the hypothesis drawn from a decision tree, as they are self-explanatory.
- When using decision tree machine learning algorithms, data type is not a constraint as they can handle both categorical and numerical variables.
- Decision tree machine learning algorithms do not require making any assumption on the linearity in the data and hence can be used in circumstances where the parameters are non-linearly related. These machine learning algorithms do not make any assumptions on the classifier structure and space distribution.
- These algorithms are useful in data exploration. Decision trees implicitly perform feature selection which is very important in predictive analytics. When a decision tree is fit to a training dataset, the nodes at the top on which the decision tree is split, are considered as important variables within a given dataset and feature selection is completed by default.
- Decision trees help save data preparation time, as they are not sensitive to missing values and outliers. Missing values will not stop you from splitting the data for building a decision tree. Outliers will also not affect the decision trees as data splitting happens based on some samples within the split range and not on exact absolute values.
- Of all the well-known learning methods, decision trees come closest to meeting the requirements for serving as an off-the-shelf procedure for data mining.
- They are relatively fast to construct and they produce interpretable models (if the trees are small).
- they naturally incorporate mixtures of numeric and categorical predictor variables andmissing values.
- They are invariant under (strictly monotone) transformations of the individual predictors.
- As a result, scaling and/or more general transformations are not an issue, and they are immune to the effects of predictor outliers.
- They perform internal feature selection as an integral part of the procedure.
- They are thereby resistant, if not completely immune, to the inclusion of many irrelevant predictor variables.

Drawbacks

- inaccuracy
- The more the number of decisions in a tree, less is the accuracy of any expected outcome.
- They seldom provide predictive accuracy comparable to the best that can be achieved with the data at hand.
- A gradient boosted model (GBM) is a generalization of tree boosting that attempts to mitigate these problems, so as to produce an accurate and effective off-the-shelf procedure for data mining.
- Boosting decision trees improves their accuracy often dramatically.

- Some advantages of trees that are sacrificed by boosting are speed, interpretability, and, for AdaBoost, robustness against overlapping class distributions and especially mislabeling of the training data.
- A major drawback of decision tree machine learning algorithms, is that the outcomes may be based on expectations. When decisions are made in real-time, the payoffs and resulting outcomes might not be the same as expected or planned. There are chances that this could lead to unrealistic decision trees leading to bad decision making. Any irrational expectations could lead to major errors and flaws in decision tree analysis, as it is not always possible to plan for all eventualities that can arise from a decision.
- Decision Trees do not fit well for continuous variables and result in instability and classification plateaus.
- Decision trees are easy to use when compared to other decision making models but creating large decision trees that contain several branches is a complex and time consuming task.
- Decision tree machine learning algorithms consider only one attribute at a time and might not be best suited for actual data in the decision space.
- Large sized decision trees with multiple branches are not comprehensible and pose several presentation difficulties.

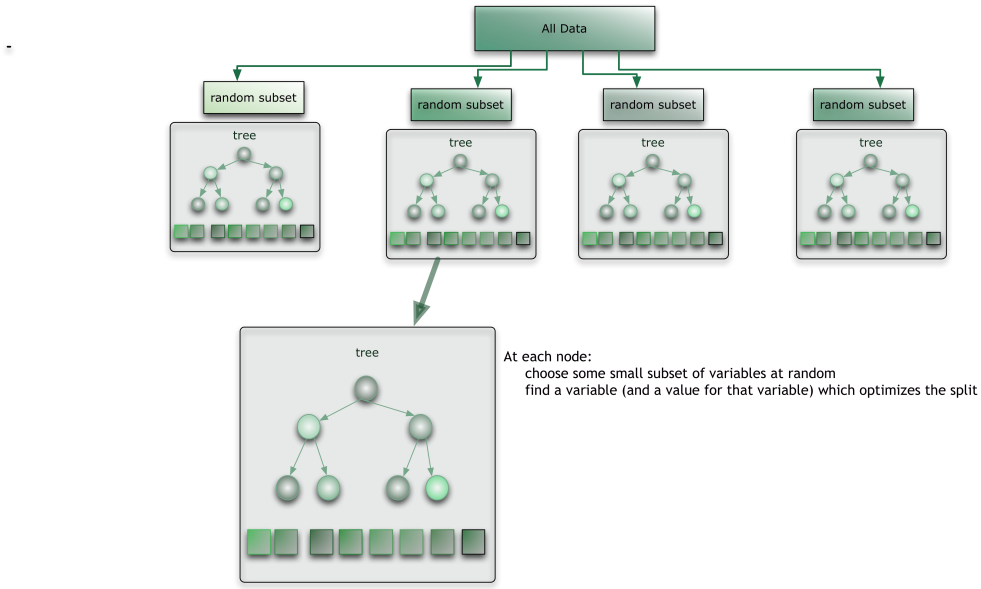
Applications



- Decision trees are among the popular machine learning algorithms that find great use in finance for option pricing.
- Remote sensing is an application area for pattern recognition based on decision trees.
- Decision tree algorithms are used by banks to classify loan applicants by their probability of defaulting payments.
- Gerber Products, a popular baby product company, used decision tree machine learning algorithm to decide whether they should continue using the plastic PVC (Poly Vinyl Chloride) in their products.
- Rush University Medical Centre has developed a tool named Guardian that uses a decision tree machine learning algorithm to identify at-risk patients and disease trends.

Random Forest

- The random forest takes Desision Trees to the next level by combining trees with the notion of an ensemble.
- Thus, in ensemble terms, the trees are weak learners and the random forest is a strong learner.



How It Works

- Random Forest uses a bagging approach
- To create a bunch of decision trees with random subset of the data.
- A model is trained several times on random sample of the dataset to achieve good prediction performance from the random forest algorithm.
- In this ensemble learning method, the output of all the decision trees in the random forest, is combined to make the final prediction.
- The final prediction of the random forest algorithm is derived by polling the results of each decision tree or just by going with a prediction that appears the most times in the decision trees.
- For instance, in the above example - if 5 friends decide that you will like restaurant R but only 2 friends decide that you will not like the restaurant then the final prediction is that, you will like restaurant R as majority always wins.

Benefits

- Overfitting is less of an issue with Random Forests, unlike decision tree machine learning algorithms.
- There is no need of pruning the random forest.
- It maintains accuracy when there is missing data and is also resistant to outliers.
- Simple to use as the basic random forest algorithm can be implemented with just a few lines of code.
  - They do not require any input preparation
  - They are capable of handling numerical, binary and categorical features, without scaling, transformation or modification.
- Implicit feature selection as it gives estimates on what variables are important in the classification.

Drawbacks

- Algorithms are fast but not in all cases.

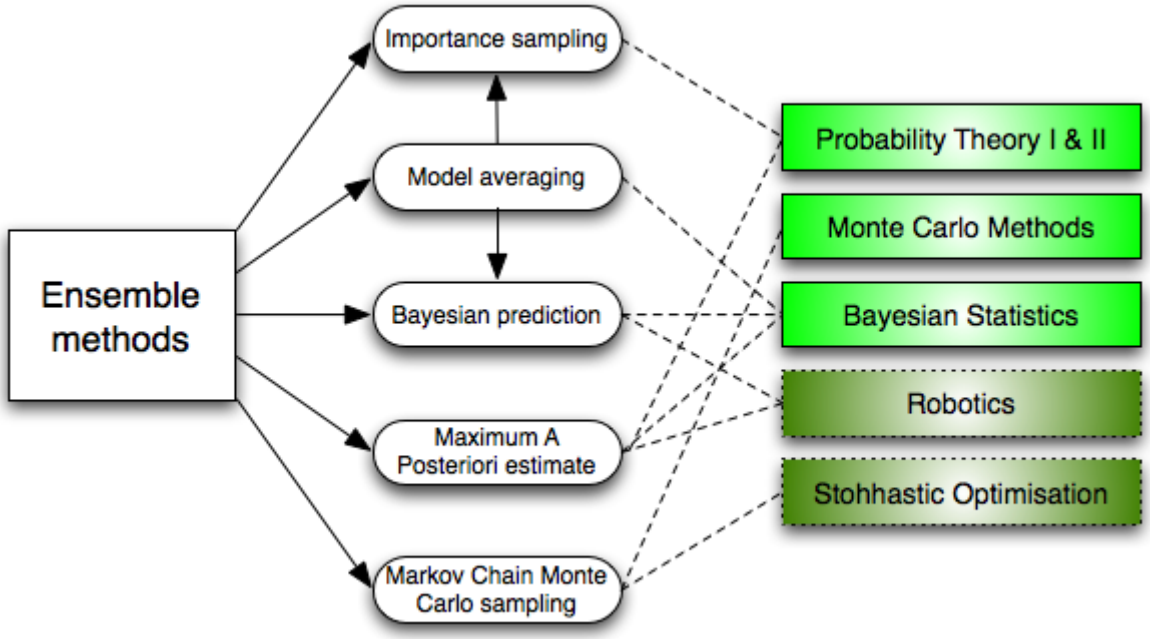
When To Use It

- Random Forest machine learning algorithms help data scientists save data preparation time

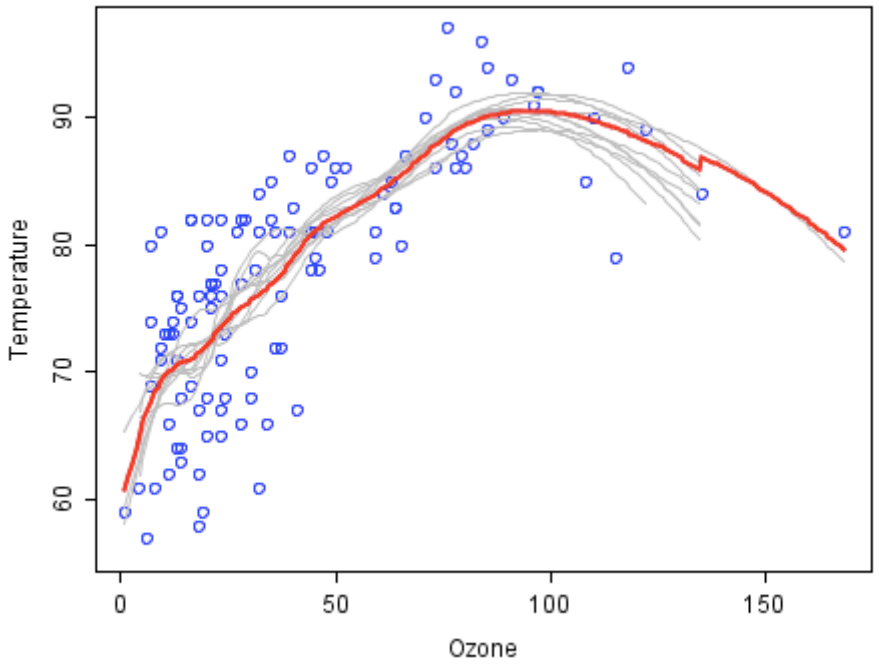
---

Ensemble Methods

- Ensembles are a divide-and-conquer approach used to cleanly delineate classifier bounderies.



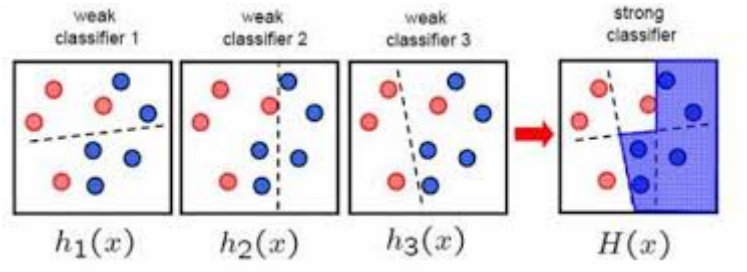
- The main principle behind ensemble methods is that a group of "weak learners" can come together to form a "strong learner".
- The figure below (taken from here) provides an example.



- Each classifier, individually, is a "weak learner," while all the classifiers taken together are a "strong learner".

Adaptive Boost

- Boosting is one of the most powerful learning ideas introduced in the last twenty years.
- It was originally designed for classification problems, but it can profitably be extended to regression as well.
- The motivation for boosting was a procedure that combines the outputs of many “weak” classifiers to produce a powerful “committee.”



How It Works

- From this perspective boosting bears a resemblance to bagging and other committee-based approaches.
- However we shall see that the connection is at best superficial and that boosting is fundamentally different.
- Problems in machine learning often suffer from the curse of dimensionality — each sample may consist of a huge number of potential features.
- Unlike neural networks and SVMs, Adaptive Boosting selects only those features known to improve the predictive power of the model, reducing dimensionality and potentially improving execution time as irrelevant features do not need to be computed.
- Boosting is a form of linear regression in which the features of each sample x are the outputs of some weak learner.
- Specifically, in the case where all weak learners are known a priori, Adaptive Boosting corresponds to a single iteration of the backfitting algorithm in which the smoothing splines are the minimizers of the cost function.

Benefits

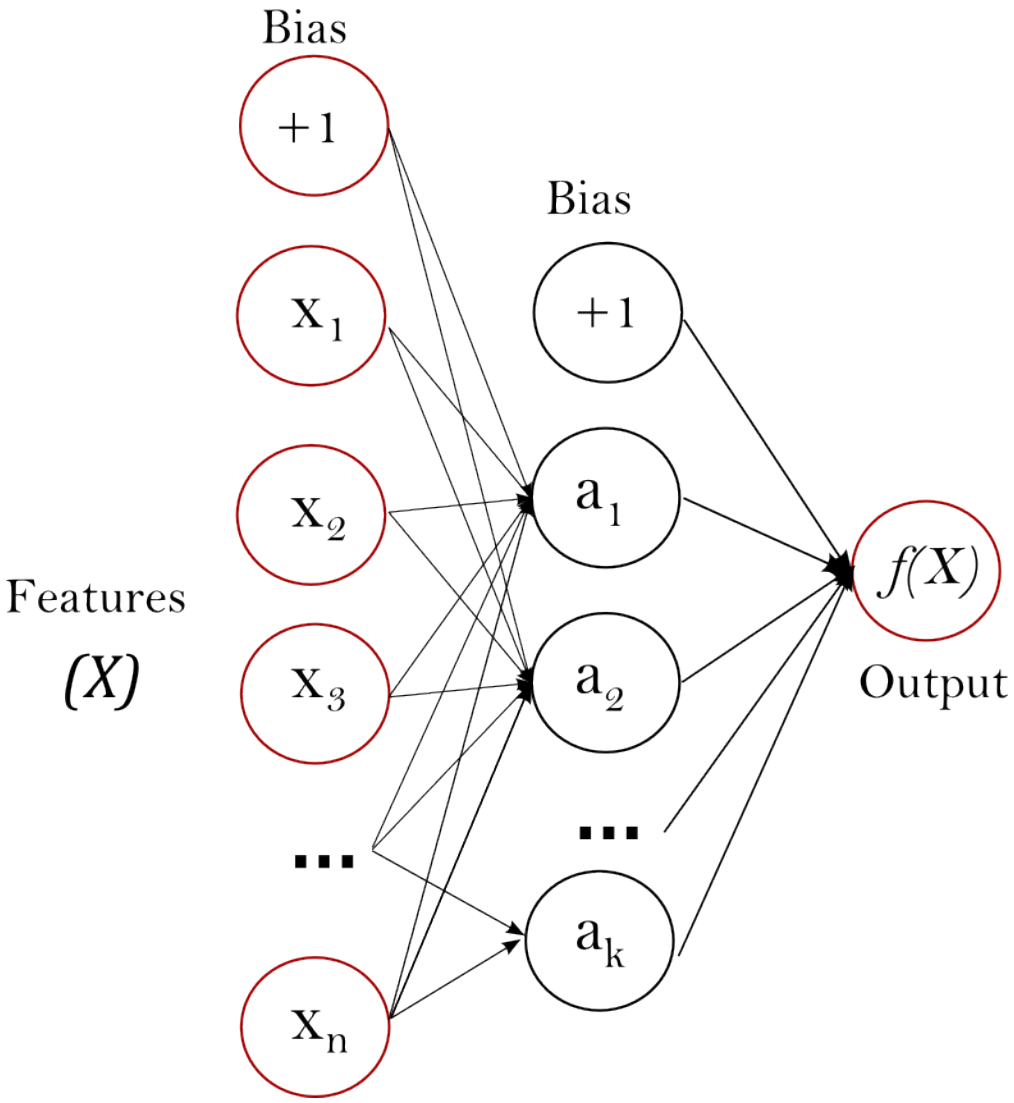
- Very simple to implement
- Feature selection on very large sets of features
- Fairly good generalization

Drawbacks

- Suboptimal solution due to greedy learning
- Can overfit in presence of noise

Neural Nets

- The term neural network has evolved to encompass a large class of models and learning methods.
- Here we describe the most widely used “vanilla” neural net, sometimes called the single hidden layer back-propagation network, or single layer perceptron.



How It Works

- There has been a great deal of hype surrounding neural networks, making them seem magical and mysterious. As we make clear in this section, they are just nonlinear statistical models, much like the projection pursuit regression model discussed above.
- A neural network is a two-stage regression or classification model, typically represented by a network diagram as in Figure 11.2. This network applies both to regression or classification. For regression, typically K = 1 and there is only one output unit Y1 at the top. However, these networks can handle multiple quantitative responses in a seamless fashion, so we will deal with the general case.

6. Advise


Advise
Reinforce
Optimize
Linear Algebra
Decomposition
Band Matrix
Sparse Matrix
OR
Scheduling
Routing
Allocation
Approximate
Genetic
Anealing
Gradient
Objective
Constraint Logic
Linear Prog
Non Linear Prog
Integer Prog
Simulate
Event
Discrete
Markov
Monte Carlo
Self Organized
Org Maps
Swarm
Agent
Phase Planes
Dynamics
Activity
Differential
Reinforce
Explore Data
Epsilon Greedy
Boltzmann
Gaussian
State Dependent
Value Function
Q Learning
SARSA
Fitted Q
Policy Gradient
Policy Reinforce
Natural Actor
Black Box
Stochastic Hill
Swarm
Natural Evolution
Fitness Expect
Feedback
Learning
Active Learning
Ensemble
Reason
Inductive
Transductive
Deductive
Optimality
Constraints
Sensitivity
Sparsity
Simulatability
Linearity
Constantability
Comprehensiveness
Separatrix
Insight
Improve
Construcual
Nutrients
Flow
Recreate
Change
Evolve
Emerge
Augment
Inference
Expert
Logical
Fuzzy

Reinforced

Optimize

Simulate

Feedback

Insight

Augment