Criteria: accuracy, speed, robustness, scalability, interpretability.

Decision Tree Steps: 1. starts as a single root node containing all of the training tuples. 2. If the tuples are all from the same class, then the node becomes a leaf, labeled with that class. 3. an attribute selection method is called to determine the splitting criterion, by using a statistical measure to select the best split way. 4. A branch is grown from the node and the tuples are partitioned accordingly. O(nDlogD)

$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$ $Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$ $SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$

$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$ $gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$

gain=info - info$_a$ ratio= gain/splitinfo

gain: biased towards multivalued attributes;

Ratio: prefer unbalanced splits in which one partition is much smaller than the others

Gini: biased to multivalued attributes, has difficulty when # of classes is large, favor tests that result in equal-sized partitions and purity in both partitions;

Repetition**:** attribute is repeatedly tested; replication: duplicate subtrees.

RainForest, AVC: Attribute * value * class

Naive Bayesian: assumes class conditional independence, classify by maximizing

$P(Ci|\mathbf{X}) = \frac{P(\mathbf{X}|Ci)P(Ci)}{P(\mathbf{X})}$ $FOIL\_Gain = pos' \times (\log_2 \frac{pos'}{pos'+neg'} - \log_2 \frac{pos}{pos+neg})$

Pros: easy to implement, good results in most of the cases; Cons: class conditional independent assumption;

Discriminative Classifiers: (NN&SVM) Pros: accuracy is high, robust with errors, no overfit, fast evaluation of target function; Cons: long training time, difficult to understand, not cooperation with domain knowledge.

SVM: The complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data; The support vectors are the essential or critical training examples.If all other training examples are removed and the training is repeated, the same separating hyperplane would be found The number of support vectors found can be used to compute an (upper) bound on the expected error rate, which is independent of the data dimensionality

CB-SVM: 1. Construct the microclusters using a CF-Tree. 2. Train an SVM on the centroids of the microclusters. 3. Decluster entries near the boundary. 4. Repeat the SVM training with the additional entries. 5. Repeat the above until convergence.

Eager classification: pros: faster, constructs a generalization, weights can be assigned to attributes, cons: commit to a single hypothesis

more time training; Lazy pros: richer hypothesis space, better accuracy, less training time; cons: expensive storage costs, slower testing, equally weighted attributes

Association-based classification: 1. search for strong associations between frequent patterns and class labels. 2. Using such rules to classify. higher accuracy than DT, which consider only one attribute at a time. It uses high confidence rules to combine multiple att.

Boosting: a training set St is sampled with replacement from S. Assign weights to the tuples within that training set. After Ct is created, update the weights so that the tuples causing error have greater probability

$$\begin{aligned} accuracy &= \frac{t\_pos+t\_neg}{(pos+neg)} \\ &= \frac{t\_pos}{pos+neg} + \frac{t\_neg}{(pos+neg)} \\ &= \frac{t\_pos}{(pos+neg)} \times \frac{pos}{pos} + \frac{t\_neg}{(pos+neg)} \times \frac{neg}{neg} \\ &= sensitivity\frac{pos}{(t\_pos+t\_neg)} + specificity\frac{neg}{(t\_pos+t\_neg)}. \end{aligned}$$

DDPMine: during the recursive FP-growth mining, we use a global variable to record the most discriminative itemset. Before proceeding to construct a conditional FP-tree, first estimate the upper bound of the information gain. If the upper bound value is not greater than the current best value, we can safely skip this conditional FP-tree as well as any FP-tree recursively constructed from this one. It integrates the feature selection mechanism into the mining framework.

GA: An initial population is created consisting of randomly generated rules, each rule is represented by a string of bits. Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules and their offsprings. Slow but easily parallelizable. A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation. a discernibility matrix storing the differences between attribute values for each pair of data tuples is used to reduce the computation intensity. Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree of membership, attribute values are converted to fuzzy values. Each applicable rule contributes a vote for membership in the categories, the truth values for each predicted category are summed, and these sums are combined. Regression: model the relationship between one or more independent or predictor variables and a dependent or response variable. AdaBoosting: Given a set of d class-labeled tuples, initially, all the weights of tuples are set the same, generate k classifiers in k rounds. At round i, tuples from D are sampled with replacement to form a training set Di of the same size. Each tuple's chance of being selected is based on its weight. Its error rate is calculated using Di as a test set. If a tuple is misclssified, its weight is increased, o.w. it is decreased. Model Selection: Receiver Operating Characteristics curves: for visual comparison of classification models, Shows the trade-off between the true positive rate and the false positive rate, the area under the ROC curve is a measure of the accuracy of the model.

Partitioning: Construct various partitions and then evaluate them, k-means, k-medoids, CLARANS; Hierarchical: Create a hierarchical decomposition of the set of data, Diana, Agnes, BIRCH, ROCK, CAMELEON; Density: Based on connectivity and density functions, DBSACN, OPTICS, DenClue; Grid: based on a multiple level granularity structure, STING, wave cluster, CLIQUE; Model: A model is hypothesized for each of the clusters and tries to find the best fit of that model: em, som, cobweb; Frequent pattern: p-Cluster; Constraint: consider user specified or application specific constraints; COD (obstacles), Link: Objects are often linked together in various ways, SimRank, LinkClus

Requirements: Scalability, deal with different types of attributes, handle dynamic data, arbitrary shape, domain knowledge to determine input parameters, noise and outliers, insensitive to order of input records, high dimensionality, incorporation of user-specified constraints, interpretability and usability

k-means: spherical, k, outliers; k-medoid: spherical, k, unscalable; CLARA: spherical, k, sensitive to samples; BIRCH: spherical, n d-D points, CF tree hold only limited entries, does not always correspond to a natural cluster, sensitive to input order; ROCK**:** arbitrary, categorical data only, emphasis interconnectiv. ignores closeness. CHAMELEON: arbitrary, quadratic time; DBSCAN: arbitrary shape; MaxDis & Minpts; quadratic time. AGNES: cannot undo, O(N^2) BIRCH Steps**:** begins by partitioning objects hierarchically using tree structures, then applies iterative partitioning to refine the clusters. 1. scan DB to build an initial in-memory CF tree, 2. use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree. CHAMELEON Steps: 1. use a graph partitioning algorithm to cluster objects into a large number of relatively small sub-clusters; 2. use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters. CLIQUE: 1. partition the data space and find the number of points that lie inside each cell of the partition.2. identify the subspaces that contain clusters using the Apriori principle: determine dense units in all subspaces of interests, determine connected dense units in all subspaces of interests. 3. generate minimal description for the clusters.

pCluster: mining frequent patterns in which each pair of objects and features satisfy thresh. Feature extraction: costly and may not be effective? Using frequent patterns as “features” 1. Downward closure 2. Clusters are more homogeneous than bi-cluster (thus the name: pair-wise Cluster)

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |(d_{xa} - d_{xb}) - (d_{ya} - d_{yb})|$$

COD: Triangulation and micro-clustering

CrossClus: measure similarity between features by how they group objects into clusters, use a heuristic method to search for pertinent features, use tuple ID propagation to create feature values. Simrank: O(N^2)s O(M^2)t

$$\text{sim}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} \text{sim}(I_i(a), I_j(b))$$

Linkclus: By the power law distribution of linkages, substantially reduce the number of pairwise similarities need to be tracked, and the similarity between less similar objects will be approximated using aggregate measures. SimTree stores similarities in a multi-granularity way. 1. initialize a simtree for objects of each type, for repeat: for each tree, update the similarities between its nodes using similarities in other trees, adjust the structure of each tree. O(M lgN^2).

Bagging: Analogy: Diagnosis based on multiple doctors’ majority vote Training:Given a set D of d tuples, at each iteration i, a training set Di of d tuples is sampled with replacement from D (i.e., boostrap), A classifier model Mi is learned for each training set Di; can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple; Accuracy: Often significant better than a single classifier derived from D; For noise data: not considerably worse, more robust Boosting: Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy.Steps: 1) Weights are assigned to each training tuple 2) A series of k classifiers is iteratively learned 3) After a classifier Mi is learned, the weights are updated to allow the subsequent classifier, Mi+1, to pay more attention to the training tuples that were misclassified by Mi 4) The final M* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy . The boosting algorithm can be extended for the prediction of continuous values. Comparing with bagging: boosting tends to achieve greater accuracy, but it also risks overfitting the model to misclassified data

Linear regression: involves a response variable y and a single predictor variable x y = w0 + w1*x; Multiple linear regression: involves more than one predictor variable For 2-D data, we may have: y = w0 + w1 x1+ w2 x2; Polynomial regression: y = w0 + w1 x + w2 x2 + w3 x3; Regression trees: Trees to predict continuous values rather than class labels. Each leaf stores a continuous-valued prediction, It is the average value of the predicted attribute for the training tuples that reach the leaf

EM : An extension to k-means; Assign each object to a cluster according to a weight (prob. distribution) New means are computed based on weighted measures Algorithm converges fast but may not be in global optima