ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Survey on Data Mining Classification Methods

K. Murali
Assistant Professor,
CMR Technical Campus,
Hyderabad.

Dr. B. Rama Subba Reddy
Professor & HOD,
Dept of CSE,
SVCE, Tirupati.

Abstract:

Data mining is the process of extracting information from a data set and transform it into an understandable structure. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a major technique in data mining and widely used in various fields. Classification is a data mining (machine learning) technique used to predict group membership for data instances. In this paper, we present the basic classification techniques. Several major kinds of classification method including decision tree induction, Bayesian networks, k-nearest neighbor classifier, the goal of this study is to provide a comprehensive review of different classification techniques in data mining

Keywords:

Classification Algorithms, Decision Tree Induction, Bayesian Networks, KNN, Support vector machine, Data Mining.

I. INTRODUCTION:

Classification is used to classify each item in a set of data into one of predefined set of classes or groups. The data analysis task classification is where a model or classifier is constructed to predict categorical labels (the class label attributes). Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks. A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number, and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case. Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target.

A predictive model with a numerical target uses a regression algorithm, not a classification algorithm. The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating. In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown. Classification has many applications in customer segmentation, business modeling, marketing, credit analysis, and biomedical and drug response modeling.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Data classification is defined as two-step process shown in figure 1.

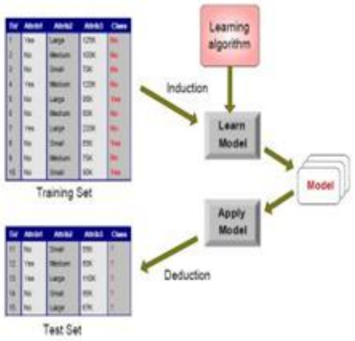


Figure: 1 Illustrating Classification Task.

Step 1: A classifier is built describing a predetermined set of data classes or concepts. (This is also known as supervised learning).

Step 2: Here, the model is used for classification. First, the predictive accuracy of the classifier is estimated. (This is also known as unsupervised learning).

II. CLASSIFICATION METHODS:

The commonly used methods for data mining classification tasks can be classified into the following groups. 1. Decision tree induction methods, 2. Rule-based methods, 3. Memory based learning, 4. Neural networks, 5. Bayesian network, 6. Support vector machines.

1. DECISION TREE INDUCTION:

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. This is illustrating in Figure 1. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

Decision tree induction algorithms are function recursively. First, an attribute must be selected as the root node. In order to create the most efficient (i.e., smallest) tree, the root node must effectively split the data. Each split attempt to pare down a set of instances (the actual data) until they all have the same classification. The best split is the one that provides what is termed the most information gain.

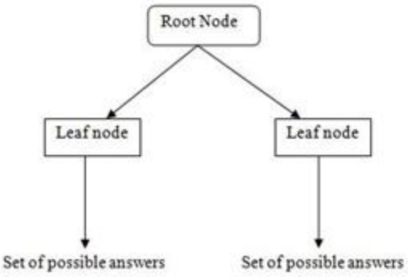


Fig 2: Decision Tree Induction

The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive, divide-and-conquer manner. The Tree induction algorithm, summarized as follows.

Step1: The algorithm operates over a set of training instances, C.

Step2: If all instances in C are in class P, create a node P and stop, otherwise select a feature or attribute F and create a decision node.

Step3: Partition the training instances in C into subsets according to the values of V.

Step4: Apply the algorithm recursively to each of the subsets C.

These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example, Hunt's algorithm, id3, c4.5, cart, sprint are greedy decision tree induction algorithms.

A. Hunt's Algorithm:

Hunt's algorithm grows a decision tree recursively by partitioning a training data set into smaller, purer subsets. This algorithm contains two steps in order to construct a decision tree.

Step 1: which is the terminating step for the recursive algorithm, checks if every record in a node is of the same class. If so, the node is labeled as a leaf node with its classification the class name of all the records within.

Step 2: If a node is not pure then selects/creates an attribute test condition to partition the data into two purer data sets. From here a child node is created for each subset.

The algorithm recurses until the all leaf nodes are found. A common means of deciding which attribute test condition should be used is the notion of the "best split." This concept boils down to choosing the test condition that results in subsets that are purer (where purity is richer when the set contains records of the same class). Three common formulas for calculating impurity is entropy, gini, and classification error.

B. (Iterative Dichotomiser 3) Algorithm

ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree. ID3 is the precursor to the C4.5 algorithm.
The ID3 algorithm can be summarized as follows:

- Take all unused attributes and count their entropy concerning test samples
- Choose attribute for which entropy is minimum (or, equivalently, information gain is maximum)
- Make node containing that attribute

C. C4.5 Algorithm

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan.[1] C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. In pseudocode, the general algorithm for building decision trees is:

- Check for base cases
- For each attribute a
- Find the normalized information gain from splitting on a
- Let a_best be the attribute with the highest normalized information gain
- Create a decision node that splits on a_best
- Recurse on the sublists obtained by splitting on a_best, and add those nodes as children of node

D. Rnd Tree (Random Forest)

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).
Each tree is grown as follows:If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sam t split on these m is used to split the node. The value of m is held constant during the forest growing.

2. RULE-BASED METHOD

Rule-based classifiers classify data by using a collection of ―if . . . then . . .‖ rules. The rule antecedent or condition is an expression made of attribute conjunctions. The rule consequent is a positive or negative classification. In order to build a rule-based classifier we can follow a direct method to extract rules directly from data. The advantages of rule-based classifiers are that they are extremely expressive since they are symbolic and operate with the attributes of the data without any transformation. Rule-based classifiers, and by extension decision trees, are easy to interpret, easy to generate and they can classify new instances efficiently.

Classify records by using a collection of
“if…then…” rules
Rule: (Condition) → y
Where Condition is a conjunction of attributes and
Y is the class label
LHS: rule antecedent or condition
RHS: rule consequent

Examples of classification rules

(Blood Type=Warm) ∧ (Lay Eggs=Yes) →Birds
(Taxable Income < 50K) ∧ (Refund=Yes) → Evade=No
rule-based classifier (example)
r1: (give birth = no) ∧ (can fly = yes) → birds
r2: (give birth = no) ∧ (live in water = yes) → fishes
r3: (give birth = yes) ∧ (blood type = warm) → mammals r4: (give birth = no) ∧ (can fly = no) → reptiles

3. MEMORY BASED LEARING

Memory-based learning is a family of learning algorithms that, instead of performing explicit generalization, compare new problem instances with instances seen in training, which have been stored in memory. it is called instance-based because it constructs hypotheses directly from the training instances themselves. this means that the hypothesis complexity can grow with the data. in the worst case, a hypothesis is a list of n training items and the computational complexity of classification a single new instance is O(n). one advantage memory-based learning has over other methods of machine learning is its ability to adapt its model to previously unseen data. a simple example of an instance-based learning algorithm is the k-nearest neighbor algorithm.

K-Nearest Neighbor and Memory-Based Reasoning

When trying to solve new problems, people often look at solutions to similar problems that they have previously solved. k-nearest neighbor (k-NN) is a classification technique that uses a version of this same method. It decides in which class to place a new case by examining some number — the ”k” in k-nearest neighbor — of the most similar cases or neighbors. it counts the number of cases for each class, and assigns the new case to the same class to which most of its neighbors belong.

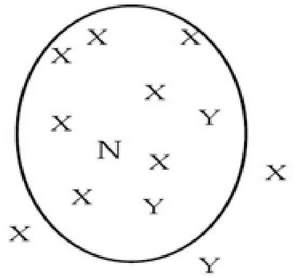


Fig 3: k-nearest neighbor. n is a new case. it would be assigned to the class x because the seven x’s within the ellipse outnumber the two y’s

K-NN models are very easy to understand when there are few predictor variables. they are also useful for building models that involve non-standard data types, such as text. the only requirement for being able to include a data type is the existence of an appropriate metric.

4. NEURAL NETWORKS:

In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in Data. Using neural networks as a tool, data warehousing firms are harvesting information from datasets in the process known as data mining. The difference between these data warehouses and an ordinary database is that there is actual manipulation and cross-fertilization of the data helping users makes more informed decisions.

A. Feed forward Neural Network:

One of the simplest feed forward neural networks (FFNN), such as in Figure 4, consists of three layers: an input layer, hidden layer and output layer. In each layer there are one or more Processing Elements (PEs).

PEs is meant to simulate the neurons in the brain and this is why they are often referred to as neurons or nodes. A PE receives inputs from either the outside world or the previous layer. There are connections between the PEs in each layer that have a weight (parameter) associated with them. This weight is adjusted during training. Information only travels in the forward direction through the network - there are no feedback loops.

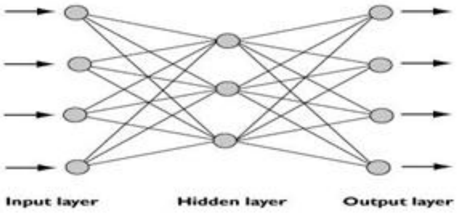


Fig 4: FIFO in Neural Networks

5. BAYESIAN NETWORK

A Bayesian network, Bayesian networks are directed acyclic graphs(DAG) whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters, or hypotheses. Edges represent conditional dependencies; nodes which are not connected represent variables which are conditionally independent of each other. Each node is associated with a probability function that takes as input a set of values for the node's parent variables and gives the probability of the variable represented by the node. For example, if the parents are m Boolean variables then the probability function could be represented by a table of 2^m possible combinations of parents being true or false. Similar ideas may be applied to undirected, and possibly cyclic, graphs; such are called Markov networks. Bayesian approaches are a fundamentally important DM technique. Given the probability distribution, Bayes classifier can provably achieve the optimal result. Bayesian method is based on the probability theory. Bayes Rule is applied here to calculate the posterior from the prior and the likelihood, because the latter two is generally easier to be calculated from a probability model One example of five attributes Bayes net is shown in figure 5.

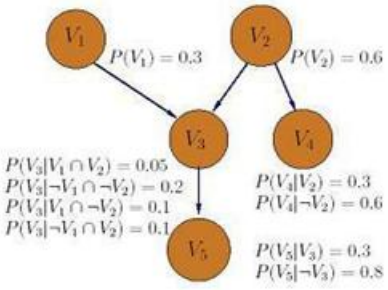


Fig 5 : A Bayes net for 5 attributes.

A general model can be followed below for build Bayes net:

Step1: Choose a set of relevant variables.

Step2: Choose an ordering for the variables.

Step3: Assume the variables are X1, X2, ..., Xn (where X1 is the first, and Xi is the ith).

Step4: for i = 1 to n:

Step5: Add the Xi vertex to the network

Step6: Set Parent(Xi) to be a minimal subset of X1, ..., Xi-1, such that we have conditional independence of Xi and all other members of X1, ..., Xi-1 given Parents(Xi).

Step7: Define the probability table of P(Xi=k | Assignments of Parent(Xi)).

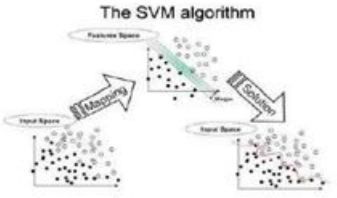
There are many choices of how to select relevant variables, as well as how to estimate the conditional probabilities. If the network as a connection of Bayes classifiers, then the probability estimation can be done applying some PDF like Gaussian. In some cases, the design of the network can be rather complicated. There are some efficient ways of getting relevant variables from the dataset attributes. Assume the coming signal to be stochastic will give a nice way of extracting the signal attributes. And normally, the likelihood weighting is another way to getting attributes.

6. SUPPORT VECTOR MACHINES

Support Vector Machines (SVM's) are learning machines that can perform binary classification and regression estimation tasks. SVM's are recognized as efficient tools for data mining and are popular because of two important factors,

- Unlike the other classification techniques, SVM's minimize the expected error rather than minimizing the classification error.
- SVM's employ the duality theory of mathematical programming to get a dual problem that admits efficient computational methods.

SVM models have similar functional form to neural networks and radial basis functions, both popular data mining techniques. However, neither of these algorithms has the well-founded theoretical approach to regularization that forms the basis of SVM. The quality of generalization and ease of training of SVM is far beyond the capacities of these more traditional methods. The support vector machine (SVM) is a training algorithm for learning classification and regression rules from data, for example the SVM can be used to learn polynomial, radial basis function (RBF) and multi-layer perceptron (MLP)classifiers.



EVALUATING THE PERFORMANCE OF CLASSIFIER

Hold – Out method

The original data with labeled examples is classified into two sets, called training set and test set. The set should not be used in testing and the test set should not be used in learning. Unseen test set provides unbiased estimate of accuracy. This method is mainly used when the data set is large.

N-Fold Cross-Validation

The available data is partitioned into n equal-size disjoint subsets. Use each subset as the training set to learn a classifier. The procedure is run n times, which given accuracies average of the n accuracies.10-fold and 5-fold cross- validations are commonly used. This method is used when the available data is not large.

Leave-one-out cross validation

This method is used when the data set is very small. It is a special case of cross-validation. Each fold of cross validation has only a single test example and all the test of the data is used in training [5]. If the original data has m examples, this is m-fold cross-validation.

Validation set

The available data is divided into three subsets, 1. Training set 2. Validation set and 3. Test set. A validation set is used frequently for estimation parameters in learning algorithm. In such cases, the values that give the best accuracy on the validation set are used as the final parameter values. Cross validation can be used for parameter estimating as well.

III. CONCLUSION:

The goal of classification algorithms is to generate more certain, precise and accurate system results. Numerous methods have been suggested for the creation of ensemble of classifiers. Classification methods are typically strong in modeling interactions. Several of the classification methods produce a set of interacting logic that best predict the phenotype. However, a straightforward application of classification methods to large numbers of markers has a potential risk picking up randomly associated markers. But still it is difficult to recommend any one technique as superior to others as the choice of a dataset. Finally, there is no single classification algorithms is best for all kind of dataset. Classification algorithms are specific in their problem domain.

REFERENCES:

[1] Jiawei Han and Micheline Kamber Data Mining: Concepts and Techniques,3rd edition.

[2] Baik, S. Bala, J, A Decision Tree Algorithm for Distributed Data Mining.

[3] Witten, I. & Frank, E. "Data Mining: Practical Machine Learning Tools And Techniques", 2nd Edition, Morgan Francisco, 2005.

[4] Friedman, N., Geiger, D. & Goldszmidt M. Bayesian Network Classifiers. Machine Learning

[5] Avirm Michael Kearns And Dana Ron,"Algorithmic Stability And Sanity-Check Bounds For Leave-One-Out Cross Validation".

[6] Cristianini, N., Shawe-Taylor: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, 2000

[7] https://en.wikipedia.org1.