Classification Algorithms

Homework 4

Amitha Narasimha Murthy (50098028), Neeti Narayan (50098029), Vivekanandh Vel Rathinam (50098075)

In this project, we have implemented Decision Trees and Naive Bayes classification algorithms and applied five-fold cross validation technique to evaluate their performance. The evaluation metric used is Accuracy.

Introduction

The goal of a Classification algorithm is to assign a class as accurately as possible to any previously unseen records, given a collection of records. A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

The general working of a classification algorithm happens in two phases:

1. Model construction: where we describe/summarize a set of pre-determined classes
2. Model Application: where we classify unseen objects

We estimate the accuracy of model using an independent test set.

In this project, we implement Decision Trees and Naïve Bayes algorithms and analyze their results in the following sections.

Data preparation

1. Some of the missing values in certain attributes are handled as special values
2. We have reduced the dimensions of the dataset and considered only those attributes that heavily influence the classification
3. The data is pre-processed to handle noise

DECISION TREES:

Decision trees is the most commonly used classification algorithm where the tree consists of decision nodes and terminating nodes where a conclusion has been reached. The arrows that come out of a node are the decisions made on that particular feature.

ID3 Algorithm:

DecisionTree(node)

1.1. If all the items in the node's dataset are in the same class, return the class label
1.2. If only one feature is remaining in the node's dataset, return the majority label
1.3. Else
1.3.1 Find the best feature to split the data and keep track of the features used to split the datasets
1.3.2 Split the dataset using the optimum feature in 1.3.1
1.3.3 Create child decision nodes with the datasets created above
1.3.4 Repeat the algorithm for the child nodes created in 1.3.3….call DecisionTree(child node)
1.3.5 Return node

Splitting the Decision Tree:

The optimum split is determined using the information gain. Information Gain measures the reduction in entropy achieved because of the split

GAIN(split) = Entropy(parent) – sum over all child nodes (n(i)/n * Entropy(i))

N(i) - Number of rows of value *i* in the split

N – Total number of rows in the dataset

Advantages:

1. Inexpensive to build
2. Its fast at classifying unknown records
3. It is easy to interpret for small sized trees
4. Accuracy is comparable to other classification techniques for many simple datasets

Disadvantages:

1. Most of the algorithms require that the target attribute will have only discrete values
2. As decision trees use the "divide and conquer" method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present
3. Number of instances at the leaf nodes could be too small to make any statistically significant decision
4. Finding an optimal decision tree is NP-hard
5. Do not generalize well to certain types of Boolean functions
6. Not expressive enough for modeling continuous variables

Feature Selection:

The below features were selected based on relevance of the features on the final result and some percentage of controlled tweaking to find the best accuracy

a) Gender
b) Age
c) Admission Type ID
d) Discharge disposition ID
e) Admission Source ID
f) Time in Hospital
g) Number of medications
h) Number of diagnosis
i) Insulin
j) Change

Tree pruning to prevent over fitting:

Pre-pruning was used to form the decision tree of a height of 5. Height of trees less than were not chosen to prevent under fitting

Height of tree	Accuracy 1	Accuracy 2	Accuracy 3	Accuracy 4	Accuracy 5
5	52.93	53.45	53.84	53.19	53.53
6	51.74	53.29	52.22	52.30	52.20
7	50.40	51.14	51.20	51.30	50.51

Handling missing values and new values:

When an attribute in a test instance has a missing value or a new value, the test instance is classified using the majority label at the decision making node in the tree. If the algorithm has traversed four nodes down the tree after making decisions and suppose at one node, there is a new value which makes the algorithm not able to traverse the tree, the algorithm classifies the test instance using the majority label at that particular node

Validation:

Cross validation was used to measure the accuracy of the ID3 decision tree algorithm and the average accuracy of the algorithm is 53.39%

Accuracy 1: 52.93%

Accuracy 2: 53.45%

Accuracy 3: 53.84%

Accuracy 4: 53.19%

Accuracy 5: 53.53%

Average Accuracy for Decision Tree: 53.39%

NAÏVE BAYES:

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. It assumes an underlying probabilistic model and allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It works under the assumption that one attribute works independently of the other attributes contained by the sample. It can solve diagnostic and predictive problems.

The Bayes Theorem:

$$P(h|D) = P(D|h)\,P(h)$$

P(D) P(h): Prior probability of hypothesis h

P(D): Prior probability of training data D

P(h|D): Probability of h given D

P(D|h): Probability of D given h

Feature Selection:

We have considered the following attributes in our implementation of Naïve Bayes:

Diag_1, Diag_2, Diag_3, Metformin, insulin, Change, DiabetesMed

The reason behind choosing these attributes is that the classification strongly depends on these and the values of these attributes are well distributed resulting in a good classification.

Implementation:

The algorithm runs in two phases:

1. Train -> where the classifier is trained to classify data using the training samples
2. Classify -> where an unseen test sample is classified using Bayesian theorem

Algorithm:

Let X be a data sample whose class label is unknown
Let H_i be the hypothesis that X belongs to a particular class C_i
To classify means to determine the *highest* $P(H_i|X)$ among all classes $C_1 \ldots C_m$
– If $P(H_1|X) > P(H_0|X)$, then X belongs to Class C_1
– If $P(H_0|X) > P(H_1|X)$, then X belongs to Class C_2

– We calculate $P(H_i|X)$ using the Bayes theorem

$P(H_i|X) = (P(H_i)\ P(X|H_i))/P(X)$

Where,
$P(H_i|X)$ = Class Posterior Probability
$P(H_i)$ = Class Prior Probability
$P(X|H_i)$ = Descriptor Posterior Probability
$P(X)$ = Descriptor Prior Probability

Advantages:

1. It is robust to noise in input data
2. Simple to implement
3. It is efficient when applied to large datasets
4. They are computationally fast

Disadvantages:

1. The attribute independence assumption might be sometimes inaccurate

Validation:

Cross validation was used to measure the accuracy of the Naïve Bayes algorithm and the average accuracy of the algorithm is 53.6%

Accuracy 1: 53.08%

Accuracy 2: 54.47%

Accuracy 3: 53.93%

Accuracy 4: 53.51%

Accuracy 5: 53.2%

Average Accuracy of Naive Bayes: 53.6%

Conclusion:

Thus, the Naïve Bayes and the Decision tree algorithms were implemented and applied on the diabetes dataset and the validations were performed using the Cross Validation method.

References:

1) http://code.activestate.com/recipes/521906-k-fold-cross-validation-partition/