2019

Prediction on Hepatitis dataset

DEBASISH BHOL
11609018

LOVELY PROFESSIONAL UNIVERSITY

Introduction

Life Prognosis of hepatitis is quite a challenging task in early stage due to various interdependent features. A model can be developed which can be used in prediction of life prognosis of hepatitis disease. Data mining refers to extracting or "mining" knowledge from large amounts of data. Data mining techniques have been extensively used in bioinformatics to analyze biomedical data. Data mining algorithms can be used efficiently in prediction and classification of inter-related data. The objective of this analysis is to classify and scaling the accuracy of hepatitis.
Rapid Miner is the most widely used data mining tool which support huge amount of data mining algorithms for classification and regression. Support Vector Machine (SVM) is the most widely used data mining algorithm in the field of medicine. Feature selection techniques are used to obtain dominant set of attributes.

Literature survey

In previous methods [1] which are used for prediction of hepatitis, wrapper method is used for feature selection in which attributes are removed based on trial and error method. Life Prognosis of hepatitis patients can be predicted by using classifier such as Support Vector Machine. In support vector machine, dataset is classified into training and testing data. Support vector machine analyze the training data and makes prediction on testing data. Predictive accuracy of classifiers can be enhanced by applying the techniques of feature selection. In this paper, Wrapper methods were incorporated to remove noise features before classification. After removal of noisy attributes, accuracy of the algorithm was further increased. SVM algorithm provides more and improved accuracy with the 10 attributes identified using wrapper method using a data mining tool called weak. Data mining concepts and techniques [2] provide us the how to preprocess data and handle with missing values. Preprocessing is important because the data collected in real world is incomplete, noisy and inconsistent. Preprocessing stages include data cleaning, data integration, data transformation and data reduction. Data cleaning is carried out for missing values and Noisy data. Data cleaning is carried out for missing values and Noisy data. Data integration combines data from multiple sources into a coherent data store. Inconsistencies in attribute may result in redundancies and these redundancies can be detected by correlation analysis. Data Transformation involves smoothing, aggregation, generalization, normalization and attributes construction. Data reduction includes attribute subset selection, dimensionality reduction and discretization. Feature selection is more significant for data mining algorithms for variety of reasons such as generalization performance, running time requirements and constraints.

Data Preprocessing

Dataset used in the prediction model should be more precise and accurate in order to improve the predictive accuracy of data mining algorithms. Dataset which is collected may have missing (or) irrelevant attributes. These are to be handled efficiently to obtain the optimal outcome from the data mining process.

Attribute identification

Dataset collected from UC Irvine machine learning repository which consists of 155 instances and 19 attributes with the class stating the life prognosis yes (or) no. The dataset consist of 14 nominal attribute and 6 multi-valued attributes. The attributes which are identified are

Table 1.Attributes in dataset

Attributes	Value
Class	die (1), live (2)
Age	numerical value
Sex	male (1), female (2)
Steroid	no (1), yes (2)
Antivirals	no (1), yes (2)
Fatigue	no (1), yes (2)
Malaise	no (1), yes (2)
Anorexia	no (1), yes (2)
Liver Big	no (1), yes (2)
Liver Firm	no (1), yes (2)
Spleen Palpable	no (1), yes (2)
Spiders	no (1), yes (2)
Ascites	no (1), yes (2)
Varices	no (1), yes (2)
Bilirubin	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
Alk Phosphate	33, 80, 120, 160, 200, 250
SGOT	13, 100, 200, 300, 400, 500
Albumin	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
Protime	10, 20, 30, 40, 50, 60, 70, 80, 90
Histology	no (1), yes (2)

Data cleaning and feature selection

Dataset which is collected from UCI repository may have missing values and redundant attributes. Missing values can be handled either by removing the instances or replacing them by mean, average, maximum (or) minimum. Removing the instances may further reduce the amount of data, thereby reducing the quality in prediction. Hence these missing values are replaced by zero which doesn't much affect the quality of data.

Feature selection can be done by giving weights to the attributes. Attributes are weighted by PCA, SVM attribute evaluation and Chi-Square attribute evaluation using Rapid Miner. In this

paper, Chi-Square attribute evaluation is used since it works well with efficient data mining algorithm such as Support Vector Machine.

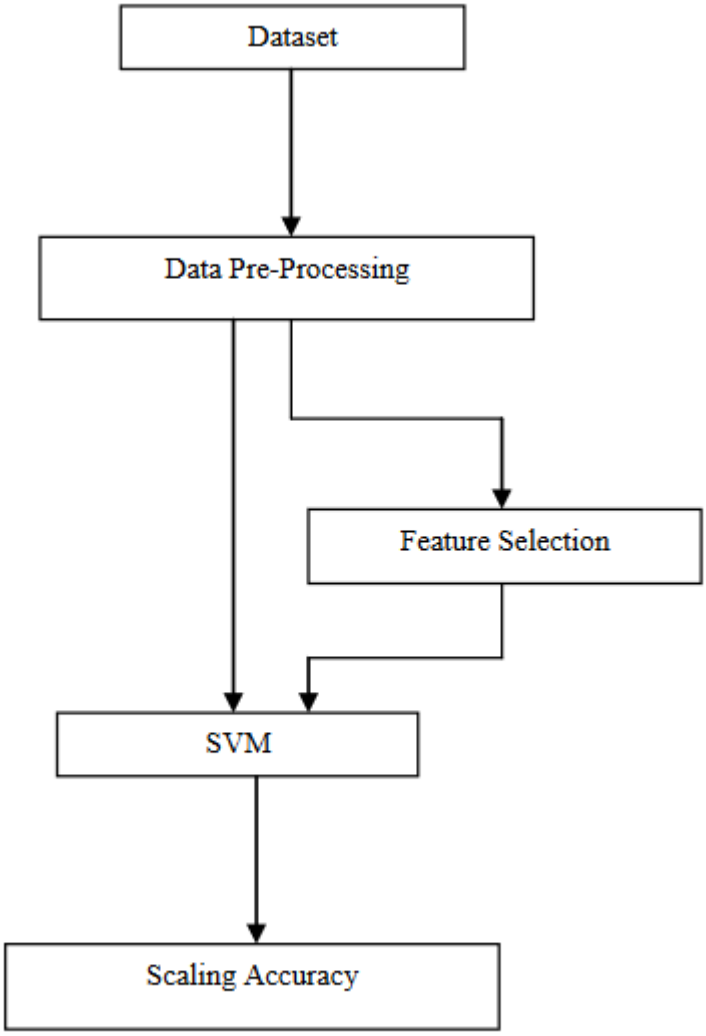

Fig 2: Architecture of proposed method

Support Vector Machine (SVM)

Support vector machine is the most widely used in bio-informatics since it minimizes the expected error rate rather than reducing the classification error rate. SVM algorithm predicts well even if the testing data is entirely different from training data. SVM attempts to determine a plane that will have smallest generalization error, among the infinite number of planes. Support vector machine chooses the plane that maximizes the margin separating two classes. Wider is the gap smaller is the generalization error.

Algorithms used

KNeighbors

In [35]:
```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 10)

knn.fit(X_train,y_train)
```

Out[35]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=10, p=2,
           weights='uniform')
```

In [53]:
```
y_pred=knn.predict(X_test)
print("Accuracy: ", accuracy_score(y_pred, y_test))
```

```
Accuracy:  0.8064516129032258
```

SVM

In [51]:
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

In [52]:
```
sv = SVC(gamma='auto')
sv.fit(X_train,y_train)
y_pred = sv.predict(X_test)
print("Accuracy: ", accuracy_score(y_pred,y_test))
```

```
Accuracy:  0.8064516129032258
```

Confusion matrix

In [39]:
```
1 y_test
```

Out[39]:
```
array([1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1], dtype=int64)
```

In [40]:
```
1 x=np.array(y_pred)
```

In [41]:
```
1 x
```

Out[41]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

In [42]:
```
1 from sklearn.metrics import confusion_matrix
```

In [43]:
```
1 cnf_matrix = confusion_matrix(y_test, y_pred)
```

In [44]:
```
1 cnf_matrix
```

Out[44]:
```
array([[ 0,  6],
       [ 0, 25]], dtype=int64)
```