CHAPTER 6

CLASSIFIERS

This chapter presents the classifiers used in the proposed work. The section 6.1 gives the introduction of classifiers. Section 6.2 deals with Naive Bayes classifier. Section 6.3 gives details about J48 classifier. Section 6.4 describes k-NN classifier. Section 6.5 explains multilayer perceptron neural network classifier. Section 6.6 deals with Bayes Net classifier. Section 6.7 gives the details of support vector machine. Section 6.8 discusses the summary of this chapter.

6.1 Introduction

Medical image classification plays an important role in diagnosis of diseases where different imaging modalities are used. Neural networks play a vital role in classification that includes supervised and unsupervised techniques. The classification task is a supervised technique where each instance belongs to a class, that is indicated by the value of a special goal attribute or simply the class attributes. The goal attribute can take on categorical values each of them corresponding to a class [127]. Classification is typically accomplished by using a decision or discriminate function.

Medical image data classification using intelligent techniques is very important and useful technique to detect or diagnose critical diseases like brain tumor. The recent research in medical image classification uses different individual and hybrid techniques [128]. The recent trend of classification of medical image data that is being utilized by many researchers are the intelligent hybrid techniques which can be developed with the help of artificial neural network, fuzzy logic and genetic algorithm.
In a classical classifier, each object for training and testing is represented by a feature vector and a discrimination rule is applied to classify a test vector. In the image classification problem, this feature vector is usually obtained from the entire image, using the appearance-based approach (each pixel corresponds to one feature) or some type of feature extraction [129]. Due to the complexity and variability of medical images, a local representation scheme is adopted. In the classification procedure, each test image is also represented using the local features selected. To approximate the posterior probability that a certain local feature belongs to an image of a given class, the k-NN algorithm is used. This posterior probability is then used to obtain a combined decision for the set of local features.

In this research work, six types of classifiers are employed for the classification of the images. Naive Bayes classifier works on the basis of probability based technique. J48 classifier handles decision tree based method. k-Nearest Neighbor deals with distance based technique. Multilayer perceptron neural networks uses the soft computing method. Bayes Net classifier deals with probability and distance based techniques. Support vector machine handles classification and regression methodology. Different types of the classifiers are used to find the classification of the dataset.

6.2 **Naive Bayes classifier**

Naive Bayes classifier is Bayes theorem based statistical classifiers. It uses probabilistic approach to predict class of given data by matching data to class with highest posterior probability. To classify a class unknown test vector, the posterior probability of each class is calculated. The feature values present in the test vector are given and the test vector is assigned to the class that is of the highest probability. The
Bayesian classifier is a statistical classifier that has the ability to predict the probability that a given instance belongs to a particular class.

The Naive Bayes classification is given as

\[
P(C_i|V) = \frac{P(V|C_i)P(C_i)}{P(V)}
\]  

(6.1)

Where

- \( V \) - Feature values of lung image.
- \( P(C_i|V) \) - Probability of the diseases like bronchitis, emphysema, pleural effusion and pneumonia and normal image for the features selected.
- \( P(V|C_i) \) - Posterior Probability of normal lung and diseased.
- \( P(C_i) \) - Prior probability of training data \( V \) (i.e.,) feature values of lung diseases.
- \( P(V) \) - Prior probability of hypothesis.

Naive Bayes is used for classifying the extracted features in this study. The extracted features are classified to the most likely class. Learning in Naive Bayes is simplified by assuming that the features are independent of the given class.

The feature is classified as shown in equation:

\[
P(X|C) = \prod_{i=1}^{n} P(X_i | C)
\]  

(6.2)

Where

- \( X = (X_1, ..., X_n) \) is the feature vector
- \( C \) is a class
6.3 J48 classifier

J48 is the C4.5 algorithm’s earlier version developed by J. Ross Quinlan. Decision trees represent information from machine learning algorithms, providing a quick way to express the data structures [130]. A decision tree is a tree data structure consisting of decision nodes and leaves. It begins with a root node and considers other nodes as "parent". A leaf specifies a class value. A decision node specifies a test over one of the attributes, which is called the attribute selected at the node. For each possible outcome of the test, a child node is present. It is used to classify a case, which assigns a class value to a case depending on the values of the attributes of the case. In fact, a path from the root to a leaf of the decision tree can be followed based on the attribute values of the case. The class specified at the leaf is the class predicted by the decision tree. A performance measure of a decision tree over a set of cases is called classification error. It is defined as the percentage of misclassified cases.

The J48 algorithm constructs the decision tree with a divide and conquer strategy. Each node in a tree is associated with a set of cases. Also, cases are assigned weights to take into account unknown attribute values. The J48 uses tree pruning options. Many algorithms try to "prune" results and produce fewer, easily interpreted results and also corrects over fitting. The overall concept includes gradual decision tree generalization till it attains a flexibility and accuracy balance.

6.4 k-NN classifier

The Nearest Neighbor classification underlying intuitions are straightforward.

The samples based on the class of nearest neighbor are classified. As more than one neighbor is considered and so the technique is usually called k-Nearest Neighbor (k-NN) where k determines the class. As training examples are needed at run-time, it is
also called memory based classification. It is considered as lazy learning technique as induction is delayed by run time, it is also called example-based classification or case-based classification [131].

A k-NN classifier which incorporates an affinity measure is not a proper metric build. However, some performance optimizations to basic k-NN algorithm require a proper metric. Briefly, such techniques, identifies nearest neighbors of an object without comparing it to other objects, but affinity measure should be a metric, specifically, it must satisfy the triangle inequality. The formula for Minkowski distance is

$$MD_p(q_i, x_i) = \left( \sum_{i=1}^{n} |q_i - x_i|^p \right)^{\frac{1}{p}}$$

(6.3)

Where

- $p$ is index of Minkowski distance
- $q$ and $x$ are variables whose distance has to be found for an order $p$.

6.5 Multilayer perceptron Neural Network classifier

The MLP is a layered feed-forward network with a direct acyclic graph as shown in Fig.6.1. Each node represents MLP artificial neuron, and every directed arc represents a connection between two neurons and MLP signal flow direction. Labels used with arcs in graph denote the MLP’s synaptic connections strengths, also called weights. Each MLP layer consists of specific neurons, each connected with neurons of following layer to replicate synaptic connections of biological Neural Network. The biological neuron’s functions are modified by computing a differentiable nonlinear function (like a sigmoid) for MLP’s each artificial neuron [132]. A sigmoid function is biologically motivated, as it tries to account for the biological neurons’ refractory phase.
The input layer is made up of source nodes that connect the network to its environment. The second layer has only one hidden layer in the network, which applies a nonlinear transformation from the input layer to the hidden layer. The output layer is linear, supplying the response of the network to the activation pattern applied to the input layer [133].

**Fig. 6.1: Representation of a 2-layer MLP**

The MLP’s input layer neurons are determined by features chosen to represent relevant patterns in feature space for pattern classification. The input layer neurons as sensory units compute identity function, \( y = x \). Each neuron in hidden and output layers computes the sigmoidal function of input and weight values corresponding to the connections.

In this network, the calculation is done layer by layer, starting at the layer nearest to the inputs. The \( s \) value of each neuron in the first layer is calculated as the weighted sum of its neurons inputs. The activation function \( f \) then squashes \( s \) to produce the output value \( o \) for each neuron in that layer. Once the set of outputs for a layer is found, it serves as an input to the next layer. This process is repeated until the final set of network output is produced.
This is expressed mathematically as follows:

\[ s_j = \sum_{i=1}^{n} x_i V_{ij} \]  \hspace{2cm} (6.4)

\[ o_n = f(s_j) = \frac{1}{1 + \exp(-s_j)} \]  \hspace{2cm} (6.5)

Where

- \( x_i \) - \( i^{th} \) element of the input pattern
- \( V_{ij} \) - weight from \( i^{th} \) input to \( j^{th} \) neuron in the hidden layer
- \( s_j \) - activation value of the \( j^{th} \) hidden neuron

The MLP output layer neurons are determined by possible pattern classes to be dealt with, for problems of interest. The class label of output neuron producing highest output value determines the input pattern class supplied to MLP. The output of the hidden layer is given as an input to the output layer.

In this work, the number of features is equal to the number of input neurons. For example, in the proposed hybrid genetic algorithm method the number of input neuron is 27. The number of neurons in the hidden layer is two third of the number of neurons in the input layer. The Sigmoidal activation function with the initial random weights between 0 and 1 are used and the bias value is set to zero. The number of classes constituted the number of neurons in the output layer. The output will result in five classes of labels namely normal lung, bronchitis, emphysema, pleural effusion and pneumonia.

In MLP algorithm, different structures are considered for example, in the proposed M^3 feature extraction with proposed hybrid genetic algorithm the number of inputs is 27, so the number of neurons in the input layer is 27. After considering
different number of neurons in the hidden layers like 14, 18 and 20 the optimal structure is found in the 18 number of neurons in the hidden layer (i.e.,) two third of the number of input. The number of output is the five classes namely normal lung, bronchitis, emphysema, pleural effusion and pneumonia.

6.6 Bayes Net classifier

A Bayesian Network (BN) can be used to compute the conditional probability of one node, given values assigned to the other nodes hence, a BN can be used as a classifier that gives the posterior probability distribution of the classification node given the values of other attributes. When learning Bayesian networks from dataset, nodes are used to represent dataset attributes [132].

There are two ways to view a BN each suggesting a particular approach to learning. First a BN is a structure that encodes the joint distribution of the attributes. Second, the BN structure encodes a group of conditional independence relationships among the nodes, according to the concept of d-separation [134]. This suggests learning the BN structure by identifying the conditional independence relationships among the nodes. These algorithms are referred as computational intelligence based algorithms or constraint-based algorithms.

A Bayesian network defines a unique joint probability distribution over X given by

\[ P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} \theta_{X_i|x_i} \] (6.6)

The set of all Bayesian networks with n variables is given by \( B_n \). Informally, a Bayesian network encodes the independence assumptions over the component random variables of X. An edge (i, j) in E represents a direct dependency of \( X_j \) to \( X_i \). Moreover \( X_i \) is independent of its non-descendants given its parents \( \prod X_i \) in \( G \).
A Bayesian network classifier is a Bayesian network where \( X = (X_1, ..., X_n, C) \).

where

- Variables \( X_1, ..., X_n \) are called attributes
- \( C \) is called the class variable.
- Moreover, the graph structure \( G \) is such that the class variable has no parents, that is, \( \Pi_C = \emptyset \) and all attributes have at least the class variable as parent, that is, \( C \in \Pi_{X_i} \).

The corresponding classifier is defined as \( \arg \max_C P_C(C \mid X_1, ..., X_n) \) [135].

Bayes Nets or Bayesian networks are graphical representation of probabilistic relationships among a set of random variables. A Bayesian network is an annotated directed acyclic graph (DAG) \( G \) that encodes a joint probability distribution over \( X \). The nodes of the graph correspond to the random variables. The links of the graph correspond to the direct influence from one variable to the other. If there is a directed link from variable \( A \) to variable \( B \), variable \( A \) will be a parent of variable \( B \). Each node is annotated with a conditional probability distribution (CPD) that represents a link, where \( A \) denotes the parents. The pair \( (G, \text{CPD}) \) encodes the joint distribution.

Recursive function to determine the number of possible DAGs that contain \( n \) nodes is given by

\[
f(n) = \sum_{i=1}^{n} (-1)^{i+1} c_i^n 2^{i(n-i)} f(n-i) \tag{6.7}
\]

One way to construct Bayesian networks is from domain knowledge. There are three main steps to construct a Bayesian network by domain knowledge

1) Determine the number of the variables in the interested domain.

2) Determine the direct influence relationships among variables in the domain.
3) Determine the conditional probabilities given the structure of the Bayesian networks from step 2.

It is a combination of the Bayes network and decision tree based method.

6.7 SVM classifier

Support Vector Machine (SVM) is a classification and regression prediction tool, that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machine uses the hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory [136]. SVM uses pixel maps as input. It gives an accuracy comparable to sophisticated neural networks with elaborate features in a handwriting recognition task. It is also being used for many applications such as handwriting analysis, phase analysis and so forth, especially for pattern classification and regression based applications.

![Fig. 6.2 Representation of SVM](image)

The SVM is developed by Vapnik and colleagues at Bell laboratories with further algorithm improvements by others. For a binary problem the training data points are \( \{x_i, y_i\}, i=1,\ldots, l, y_i \in \{-1, 1\}, x_i \in \mathbb{R}^d \). For the linearly separable case, the
support vector algorithm simply looks for the separating hyperplane with the largest margin. This can be formulated as follows [137]:

\[
X_i \cdot w + b \geq +1 \text{ for } y_i = +1 \\
X_i \cdot w + b \leq -1 \text{ for } y_i = -1
\] (6.8)

These can be combined into one set of inequalities:

\[
y_i (X_i \cdot w + b) - 1 \geq 0 \ \forall i
\] (6.9)

An important property of the SVM classifier, called sparseness, is a number of the resulting Lagrange multipliers, \(\alpha_i\), equal to zero. As such, the sum in the resulting classifier which should only be taken over all nonzero \(\alpha_i\) values, i.e. support values, instead of all data points[138]. The corresponding vectors \(x_i\) are referred to as the support vectors. These data points are located close to the decision boundary and contribute to the construction of the separating hyperplane [139-140].

6.8 System configuration and software

The research work is performed in the Intel (R) core (TM) i5 processor - 2410M CPU @ 2.30 GHz speed with memory 4.00 GB. The MATLAB version R2010a is used to develop the proposed work.

6.9 Summary

This research work classifies lung images automatically as normal lung, bronchitis, emphysema, pleural effusion and pneumonia with the help of the classifiers. Classification is undertaken through Naive Bayes, J48, k-Nearest Neighbor, multilayer perceptron Neural Network, Bayes Net and support vector machine classifiers. The performance measures such as classification accuracy, sensitivity, specificity, precision, recall and F-measure are calculated and presented in chapter 7.