CHAPTER 3

PERFORMANCE EVALUATION OF BAGGING AND BOOSTING FOR ALZHEIMER’S DISEASE

3.1 INTRODUCTION

Medical images play a vital role in ensuring information on the anatomy of human body. Because of the invention of a number of digital image equipments including Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET/CT) are proving to be inevitable in the arena of diagnosis of diseased condition. Considering medical imaging, classifying automatically and retrieval introduces a new radiograph into the archives that exist without interaction, and mostly the images that are retrieved provides a new perspective in making specific diagnoses based on image input. As the pathologic appearance of the images of a person is compared with the image database, clinician is able to make fast and accurate decision, thus reducing cost involved in medical care (Jyothi et al. 2013).

Dementia is a condition with a neurologic pathology, leading to issues with memory and thinking. It involves decline in memory leading to serious problems with the occupational, social and intellectual functions. In the U.S., dementia is one of the major issues in the elderly. The general memory deficits which are age-related are not grouped under dementia, and they do not affect their normal day-to-day activities. Dementia is a disease related to the brain which cannot be called a condition related to aging.
Dementia is a condition related to the brain which generally is progressive or chronic and higher cortical functions such as calculation, comprehension, judgment, language, learning capacity, memory, orientation are affected; whereas consciousness is not clouded (Nowotny et al. 2001). A single issue cannot be pointed out as the cause of dementia as it is multifactorial and it can be primarily and secondarily affecting the brain. The symptomatology for each individual is different based on the impact of the disease and his/her pre-morbid personality.

Alzheimer Disease (AD) is a type of dementia which is due to deterioration in the cognitive function. AD is commonly present in elderly above the age of 65 with progressive decline in language, learning capacity, memory and thinking (WHO 2012). Compared to other age-related disabilities of cognitive functional decline AD is gradual with less physical debility. Generally, the condition starts with mild symptoms and ends up with severe brain damage. The abilities of elderly with dementia are lost at differing rates. AD starts at the hippocampus region of the brain where there is necrosis of the neurons which deal with memory and learning. Then the whole brain becomes atrophic.

The characteristic feature of AD is building up protein in the brain which is termed as ‘tangles’ and ‘plaques’. Nerve cells are damaged and destroyed which makes it difficult to remember, use language and reason (Duthey 2013). An individual might become so disoriented that they will have increased difficulty in doing activities of daily living such as using phone, making meals and money handling. To be precise, AD can be defined as disease triggered by multiple molecular etiologies with varied but largely stereotyped pathogenic symptoms (Selkoe 2001). From such point of view, AD is similar to common, late life multigenic pathology that is degenerative in nature such as atherosclerosis.
A fundamental application in medical image retrieval is classification. The objective of classification of a learning algorithm is construction of classifier with a training sample set with class labels (Antonie et al. 2011). A class label is assigned by a classifier. Preprocessing and visual feature extraction are included in building classification model/classifier and visual features are extracted from labeled images (i.e. training set). Bagging and boosting classification techniques are used in this work.

Bagging, otherwise called bootstrap aggregation is an ensemble technique used to enhance unstable estimation or classification schemes. Development of boosting algorithms is done as ensemble methods. While bagging is a parallel ensemble technique, boosting is a sequential ensemble algorithm. With regards to classification, boosting is supposed to be accurate, especially with respect to AdaBoost algorithm (Bühlmann 2012).

Predictions are enhanced by bootstrap which is used by bagging. A weak classifier is improved through bootstrap aggregating procedure. Classification accuracy is improved through bagging and performance of unstable learning algorithms. Theoretical calculations are not needed and bootstrap is automatic and depends on asymptomatic results. Little code is used by boosting and variance is lowered. Weak learners are combined by boosting in locating highly accurate classifiers or better fit training set.

Boosting aims to increase a weak learning algorithm’s strength. According to a thumb rule, a weak learning algorithm must be better than random guessing. For a binary classifier, weak learning hypothesis gets 50% right (Zheng 2006). Boosting trains a weak learner many times, using training set’s reweighted original version. Boosting train’s first a weak learner with equal weight on all training set data points and then trains other weak learners based on updated weight.
3.2 CLASSIFICATION METHODS IN ALZHEIMER’S DISEASE

In every field of life classification is used. An important data mining technique is classification and is a challenging task with many applications in computer vision. Data of various kinds are classified. Every item is classified through classifying a set of data into one of predefined set of classes or groups. The problem of Classification is defined as a set of training records \( D = \{ X_1, \ldots, X_n \} \), such that each record is labeled with a class value drawn from a set of \( k \) different discrete values indexed by \( \{1\ldots k\} \). In order to construct a classification model, the training data is used, which relates the features in the underlying record to one of the class labels. A class label is predicted through a training model for instance when the class is unknown (Sravani et al. 2014).

**Support Vector Machines (SVMs)** – classification of images are done through supervised learning methods which are called Support Vector Machines (SVMs). Here, the image database are viewed as two set of vectors in a given ‘n’ dimensional space where a separating hyperplane is constructed which increases the margin between images which are relevant to query and not relevant. SVM is considered as a kernel method and kernel function present in it is vital to determine its performance (Rao et al. 2010).

The fundamental principle of SVM is considered as a maximum margin classifier. With the aid of the kernel methods, the data is first thoroughly mapped to a high-dimensional kernel space. In the kernel space, determination of maximum margin classifier is done and the decision function that corresponds in the original space can be non-linear. Through SVMs, the nonlinear data present in the feature space is classified as linear data. An optimal hyperplane is found in SVM classification method which separates
relevant and irrelevant vectors by increasing the size of its margin to the maximum (between both classes).

**Neural Networks (NN)** – As there is no need for any message with regards to the probability distribution of data and a priori probabilities of various classes, NNs are used widely in medical image classification. A supervised learning algorithm and feed forward is used by Back Propagation (BP). There are three main layers in feed-forward NN. First is the input layer, which is followed by hidden layers. There is a connection for every hidden layer from the previous layer. The output of NN is calculated by the final layer. Multiple neurons are present in every layer which will map input to the output by updating their weights with the help of gradient descent learning rule. The neurons’ weights are adjusted by the algorithm in the steepest descent direction that there is a decrease in performance function. Through back propagation of the error (Demirhan 2016), computation of the gradient of the error function relative to the hidden layer is performed.

**Decision tree** – Splitting criteria is used in this classification technique. On the basis of feature/attribute value, the instances are classified through sorting and a decision tree like flow-char is created. Every node in a decision tree is the representation of feature in an instance to be classified. The result of the test is denoted by all branches, and the class label is held by every leaf node. From the beginning the instances are classified on the basis of their feature value. The rule for classification of data tree is generated through decision tree (Sharma *et al.* 2013).

Growth phase and prune phase are the two classifiers present in a decision tree. The primary tree is built in the ‘growth phase’, while the sub-tree is created with the least estimated error in the ‘prune phase.’ The preliminary tree is pruned by the removal of small, deep nodes resulting from ‘noise’ present in the training sample, thus decreasing the challenge of over
fitting and a precise classification of unknown data is ensured. During the building of the decision tree, each node’s goal is to decide the split attribute and split point which is dependent on how well the classes are separated. The quality of the split is evaluated through numerous splitting indices that have been proposed.

**Iterative Dichotomiser 3 (ID3)** – This is a decision tree algorithm proposed by Quinlan Ross in the year 1986. There are two phases in the construction of the tree. They are tree building and pruning ID3 by using information gain measure in choosing the splitting attribute. Categorical attributes are accepted in building a tree model. Accurate results are not provided when there is noise and for the removal of noise pre-processing technique has to be used (Lavanya, Rani 2011).

While building a decision tree, information is calculated for all attributes and the root node is designated with the highest information gain to the selected attribute. The attributes are labeled as root nodes and the attribute’s possible values are propagated as arcs. Then, testing of all possible outcome instances take place to assert whether they can be classified under the same class or not. A node gets a single class name, if all instances fall under the same class, otherwise, the splitting attribute is chosen to classify instances. ID3 algorithm helps in handling continuous attributes through discretization or directly and through consideration of values best split point is found by selecting threshold on attribute values. Pruning is not supported by ID3.

**C4.5** was developed by Quinlan Ross as an extension of ID3 and is based on Hunt’s algorithm. Both categorical and continuous attributes are handled by C4.5 in order to build a decision tree. While handling continuous attributes, the attribute values are divided into two partitions on the basis of selected threshold so that the values above threshold are grouped as one entity.
and remaining as another entity. Through this, missing attribute values are handled. A decision tree is built by C4.5 using gain ratio as a method of attribute selection. The biasness of information gain is removed when there are a number of outcome values of an attribute. The gain ration is calculated for every attribute at first. The gain ratio will be at its maximum for the root node. Pessimistic pruning is used by C4.5 in order to remove unneeded branches in a decision tree to enhance the classification accuracy (Patil et al. 2015).

Breiman in the year 1984 introduced **Classification and Regression Trees (CART)**. Both classification and regression trees are built by it. The basis for it is the Hunt’s model of Decision tree construction and can be serially implemented. Gini index is used as a splitting measure in the selection of splitting attribute. A portion of the training dataset is used by CART for pruning. Both numeric and categorical attributes are used by CART in building the decision and in-built features are present which deals with missing attributes (Sujatha, Rani 2013).

Unlike other Hunt’s based algorithms, CART is used for regression analysis through regression trees. A dependent variable is forecasted in regression analysis in a set of predictor variables with specific time period. For prediction, again this is an alternative; and in implementing CART, splitting of datasets into two subgroups takes place which have an entirely different outcome. This process goes on until minimum subgroup size is reached.

**Random forests** – these are statistical inference tools which are proposed recently to derive their predictive accuracy from nonlinear nature of their constituent decision tree members and ensembles’ power. Random forest features are provided through random forest committees both predictions and model information on data proximities. It is shown through
variable importance, the variables that are associated closely with a chosen response variable, while relation of important variables are indicated through partial dependencies to the said response variable (Bhosle & Kokare 2016).

An ensemble of binary decision trees proposed by Breiman is Random forest classifier which is used in classification and regression problems. A collection of classifiers is called an ensemble where the decisions of the individual classifiers help in deciding the class of the input data. With the help of random forest, every tree is trained using random vector sampled independently from the training set with similar distribution. Majority of votes determine the class of input vector from the trees in the ensemble.

Breiman and Adele Cutler introduced Random tree which can be counted in as one of the most efficient learning algorithms. This is well-suited for large databases and 1000s of input variables can be handled without deletion of variables. Moreover, this is a very effective technique to evaluate missing data and maintain accuracy when a large data goes missing and also based on the importance of variables with regards to classification, this technique works. Proximities between pairs of cases are computed which helps in clustering and locating outliers (Hasan 2012).

Random tree can handle both classification regression problems and can be deemed as a collection of tree predictors. Feature vector is taken as the input in the classifier and classification of vector with each tree in the forest take place. Designated class is one which received majority of ‘votes’. With the help of bootstrap procedure, all trees are trained with same parameters on various training datasets. With regards to random tree, error estimation technique is not necessary, as at the time of training internally, errors are estimated.
3.3 METHODOLOGY

This section deals with wavelet features, boosting and bagging methods. With regards to experiments conducted in this work Open Access Series of Imaging Studies (OASIS) dataset was used. Feature extraction was done using wavelet texture features. Accuracy is improved by bagging and boosting. The classifiers used are K Nearest Neighbor (KNN) and naïve bayes algorithms.

3.3.1 Open Access Series of Imaging Studies (OASIS) Dataset

Through OASIS, brain MRI datasets are available freely to scientific communities. Through the compilation and distribution of MRI datasets, future inventions are facilitated in basic and clinical neuroscience. There are certain specific roles for OASIS. Primarily, for continued scientific exploration, OASIS image and related measures are considered as the datasets. To begin with, OASIS was able to provide images from more than 400 individuals with/without dementia across adult lifespan, which it is difficult even for laboratories to acquire (Akgül et al. 2009).

The initial OASIS dataset constituted an MRI data over 400 demented/non-demented individuals with varying ages (Marcus et al. 2007). The data was inclusive of longitudinal data from 150 individuals in a varying age group of 60 and 96 years including 64 individuals with varying degrees of mild-to-moderate AD was diagnosed clinically. In OASIS, the screening process included information about the patient Magnetic Resonance (MR) acquisition and subject’s cognitive performance was assessed through two clinical tests which is Mini-Mental State Examination (MMSE) and Clinical Dementia Rating (CDR) scale. MMSE is a common questionnaire, where a score of 27 / 30 is considered normal, while lower scores increasingly correlate dementia presence. CDR is an elaborate, time-consuming test
credited with discerning even very mild dementia (Marcus et al., 2010). CDR score of 0 indicates no dementia, while higher scores show dementia of increasing severity.

### 3.3.2 Wavelet Texture Features

When utilized for analysing images, wavelet transforms (Busch, Boles 2002) are calculated through the application of separable filter banks to images, as (3.1),

\[
\begin{align*}
A_n &= \left( H_x \ast \left( H_y \ast A_{n-1} \right) \right)_{41,2} \\
H_n &= \left( G_x \ast \left( H_y \ast A_{n-1} \right) \right)_{41,2} \\
V_n &= \left( G_x \ast \left( H_y \ast A_{n-1} \right) \right)_{41,2} \\
D_n &= \left( G_x \ast \left( G_y \ast A_{n-1} \right) \right)_{41,2}
\end{align*}
\]

where \( \ast \) denotes 2-dimensional convolution, \( \downarrow \) denotes down-sampling by specified factor in every dimension, while \( H \) and \( G \) are low and high pass filters, correspondingly. Approximation image \( A \) is got by low pass filtering in both directions, while detail coefficients \( H_n, V_n, \) and \( D_n \) are got by high pass filtering in one/more directions. Basic attribute extracted from wavelet coefficients is average energy of all detail images described as sum of squares of all detail images, normalized for all quantities of coefficients in images.

### 3.3.3 Bagging

Improvement of results of machine learning techniques was done through bagging and was proposed by Breiman and it is short form of “bootstrap aggregating.
During classification into two potential classes, classification algorithms create classifiers $H : D \rightarrow \{-1,1\}$ based on training sets of sample descriptions $D$. Bagging methods create sequences of classifiers $H_m$, $m=1,\ldots,M$ with regard to alterations of training sets. Classifiers are merged into compound classifiers (Machová et al. 2006). Estimation of compound classifiers are provided as weighted combinations of individual classifier estimations in (3.2):

$$H(d_i) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m H_m(d_i) \right)$$  \hspace{1cm} (3.2)

The equation may be interpreted as voting process. Sample $d_i$ is sorted to the class wherein most particular classifiers vote. Theories of classifier voting are such that variables $\alpha_m$, $m=1,\ldots,M$ are defined in a way so that accurate classifiers possess greater influence on the last estimation than less accurate classifiers. The accuracy of base classifiers $H_m$ may be only a little greater than accuracy of arbitrary classifications, which is why they are known as weak classifiers.

Bagging algorithm for classification into various classes is given in Figure 3.1:
1. Set training set D
2. For m=1,...,M
   a. Create fresh set $D_m$ of identical size as $|D|$ through arbitrary choosing of training samples from set D (few samples may be chosen more than once while others are not chosen at all).
   b. Learning of certain classifiers $H_m: D_m \rightarrow R$ through specified machine learning models on the basis of actual training set $D_m$.
3. Compound classifiers H are generated as the collection of certain classifiers $H_m$: m=1,...,M and a sample $d_i$ is sorted to class $c_i$ as per quantity of votes got from certain classifiers $H_m$.
   \[
   H(d_i, c_j) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m H_m(d_i, c_j) \right)
   \]

**Figure 3.1 Bagging Algorithm**

If there is a possibility of influencing learning processes carried out by classifiers $H_m$ in a direct manner, classification errors may also be reduced by $H_m$ while maintaining variable $\alpha_m$ as constant.

A basic prototype of bagging was denoted previously. Other bagging-like schemes were put forth training sets that were smaller in size and also with sample descriptions. A consolidation of bagging and other cross validation techniques are used up by this scheme. The division of training sets into N subsets of D/N size is denoted by cross-validation. One subset is used up as training set and others play a role of testing subsets.

### 3.3.4 Boosting

Boosting is used for classification and regression, corresponding to AdaBoost and AdaBoost R2. Sets of Neural Networks (NN) are created
sequentially where false estimation of earlier NNs play a major role in training latter networks. Through weighted averaging component estimations are merged and weighted voting for classifications where defining of weights are done through models themselves. Training sets that are utilized in boosting are utilized through re-sampling and reweighting (Zhou et al. 2002).

Resampling is utilized as NNs at the time of experiments as NNs cannot directly support weighted samples. Moreover, a weak learning algorithm is needed by boosting where errors lesser than 0.5 can be tolerated. While handling multi-class jobs, the requisite is not ensured. Bootstrap samples are created from initial training sets at the time of experimentation rather than disruption of learning procedure when fault bounds are accomplished, and can go up to a limit of 20 such samples at a given time.

Freund and Schapire proposed AdaBoost in 1995 provided a solution to several challenges of the previous boosting algorithms. Pseudo code for AdaBoost is illustrated (Figure 3.2) in a more generic format suggested by Schapire and Singer. The model accepts inputs as training sets $(x_1, y_1), \ldots, (x_m, y_m)$ wherein every $x_i$ is a part of certain domains or sample spaces $X$, and every label $y_i$ is present in certain label set $Y$. It is typically assumed $Y = \{ -1, 1 \}$. AdaBoost invokes a certain weak learning model iteratively in a series three of rounds $t = 1, \ldots, T$. A major notion of the model is the maintaining of distributions or sets of weights across the training sets (Schapire 2003). Weights of these distributions in training sample $i$, in round $t$ is represented by $D_t(i)$. Originally, all weights are initialized the same, but with every round, weights of wrongly sorted samples are incremented such that weak learners are ensured that they focus on the difficult samples in the training sets.
3.3.5 Naïve Bayes Algorithm

Bayes theory is the basis for the popular Naïve bayes classification method. On the basis of class conditional density estimation and class prior probability, derivation of test data point’s posterior class probability is done and the class with maximum posterior class probability is assigned (Ren et al. 2009). Class conditional density estimation is the major problem with Naïve Bayes. The assumption of Naïve Bayes classifier is that there is no relationship between specific feature of class to value of other features, so that (3.3):

$$P(x|C_k) = \prod_{j=1}^{d} P(x_j|C_k)$$  \hspace{1cm} (3.3)
Bayes’ theorem is the basis for Naïve Bayes classifier which is simple with strong independent assumptions. Underlying probability model can be described as “independent feature model.” Presence of absence of a specific class feature is assumed by Naïve Bayes classifier. For instance, it is an assumption that a fruit which is red, round and about a diameter of 4-inch is an apple. If such features depend on the existence of other features, all properties are considered by Naïve Bayes classifier which contributes to probability independently that the fruit is an apple.

3.3.6 K-Nearest Neighbor (KNN)

The KNN classifier is memory-based requiring no model to be fit. Consider test point \( x_{(\text{test})} \) and points in training set \( Y = \{y_1, y_2, \ldots, y_n\} \) if training set has \( n \) points. Then for test point, \( k \) training points from training set \( Y \) are found to have closest distance to \( x_{(\text{test})} \). \( x_{(\text{test})} \) is classified using majority vote among \( k \) neighbors (Shi 2012). If features are real-valued, Euclidean distance is used in feature space in (3.4):

\[
d_i = \|y_i - x_{(\text{test})}\| \tag{3.4}
\]

where \( i \) is specific index of points in training set. In terms of general pattern recognition domains, KNN pattern classifier is an effective learner. Due to its conceptual simplicity, they are quite easy to implement. Information can be included at runtime and can be used for applications which collect users’ feedback (Neo & Ventura 2012). It is proven that asymptotic error rate of K-NN rule has an upper bound that is twice that of Bayes optimal error.

3.4 RESULTS AND DISCUSSION

Samples of images from OASIS datasets are used in conducting the experiment and are compared with various techniques. 283 images with 181
images normal and 103 images with dementia are used. The naïve bayes, KNN, bagging and boosting classifiers are used. The classification accuracy, average precision and average recall as shown in Tables 3.1 to 3.3 and figures 3.3 to 3.5. The Classification Accuracy, precision, recall and F Measure is measured as in equation (3.5 to 3.7):

- **Accuracy**: It is represented as a fraction of predictions.

  \[
  \text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of predictions}} \tag{3.5}
  \]

- **Precision**: Precision is a good form of measure to define, when the costs of False Positive is high.

  \[
  \text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.6}
  \]

- **Recall**: It measures how many of the Actual Positives are captured through labelling it as Positive.

  \[
  \text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.7}
  \]

**Table 3.1 Classification Accuracy for Bagging and Boosting**

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Classification accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>0.8556</td>
</tr>
<tr>
<td>KNN</td>
<td>0.8627</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.919</td>
</tr>
<tr>
<td>Boosting</td>
<td>0.912</td>
</tr>
</tbody>
</table>
Figure 3.3 Classification Accuracy for Bagging and Boosting

Figure 3.3 shows the bagging algorithm improved classification accuracy by 7.14% when compared with naïve bayes and by 6.31% when compared with KNN. The boosting algorithm improved classification accuracy by 6.38% when compared with naïve bayes and by 5.5559% when compared with KNN.
Table 3.2 Average Precision for Bagging and Boosting

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>0.8426</td>
</tr>
<tr>
<td>KNN</td>
<td>0.8509</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.9164</td>
</tr>
<tr>
<td>Boosting</td>
<td>0.9087</td>
</tr>
</tbody>
</table>

Figure 3.4 shows the bagging algorithm improved average precision by 8.39% when compared with naïve bayes and by 7.4124% when compared with KNN. The boosting algorithm improved average precision by 7.54% when compared with naïve bayes and by 6.56% when compared with KNN.

Table 3.3 Average Recall for Bagging and Boosting
Techniques | Average Recall
--- | ---
Naïve Bayes | 0.8616
KNN | 0.8734
Bagging | 0.9072
Boosting | 0.8996

Figure 3.5 Average Recall for Bagging and Boosting

Figure 3.5 shows the bagging algorithm improved average recall by 5.156% when compared with naïve bayes and by 3.7965% when compared with KNN. The boosting algorithm improved average recall by 4.3152% when compared with naïve bayes and by 2.9554% when compared with KNN.

3.5 CONCLUSION

Same algorithm for base classifiers is used to combine a diversity of classifiers for the most popular re-sampling ensemble techniques such as bagging and boosting. With regards to noise-free data, boosting algorithms are stronger than bagging. Error of any “weak” learning algorithm is reduced
sufficiently by boosting which generates classifiers that are better slightly than random guessing. Experiments are conducted to sample images through OASIS dataset which proved that bagging and boosting provided better performance. Improved classification accuracy was seen compared to other classifiers at the end of the study. The bagging algorithm improved classification accuracy by 7.14% when compared with naïve bayes and by 6.31% when compared with KNN. The boosting algorithm improved classification accuracy by 6.38% when compared with naïve bayes and by 5.559% when compared with KNN.