Effective recognition of MCCs in mammograms using an improved neural classifier

Jinchang Ren, Dong Wang, Jianmin Jiang

Abstract

Computer-aided diagnosis is one of the most important engineering applications of artificial intelligence. In this paper, early detection of breast cancer through classification of microcalcification clusters from mammograms is emphasized. Although artificial neural network (ANN) has been widely applied in this area, the average accuracy achieved is only around 80% in terms of the area under the receiver operating characteristic curve $A_z$. This performance may become much worse when the training samples are imbalanced. As a result, an improved neural classifier is proposed, in which balanced learning with optimized decision making are introduced to enable effective learning from imbalanced samples. When the proposed learning strategy is applied to individual classifiers, the results on the DDSM database have demonstrated that the performance from has been significantly improved. An average improvement of more than 10% in the measurements of $F_1$ score and $A_z$ has fully validated the effectiveness of our proposed method for the successful classification of clustered microcalcifications.

1. Introduction

Breast cancer is the most common diagnosed cancer among women. In the United Kingdom, every year there are about 45,000 cases are diagnosed, and more than 13,200 women die from this cancer, i.e. a death rate about 29.3% (Cancer Research UK, 2009). In the United States, about 182,500 cases were diagnosed in 2008, and nearly 40,500 women die from this disease annually (American Cancer Society, 2009). Since the reasons behind are still uncertain, early detection and diagnosis is the key for improving breast cancer prognosis (Cheng et al., 2003, 2006). Among many available techniques, X-ray mammogram has been one of the most reliable methods for early detection of such disease (Hassanien, 2007). In general, it can increase the survival ratio by 20% to about 80% for patients. In England alone, around 1400 lives are saved each year via the NHS breast screening (Cancer Research UK, 2009). Other popular means for breast cancer detection include magnetic resonance imaging (MRI) (Torheim et al., 2001), electrical impedance spectroscopy (EIS) (Kerner et al., 2002), ultrasound (Joo et al., 2004), and infrared imaging (Tosteson et al., 1999).

Although mammogram contains useful information for the early detection of breast cancer, it is difficult for radiologists to make accurate and consistent judgments due to the huge amount of data and widespread screening. Consequently, about 10–30% cases are missed during the routine check (Cheng et al., 2003). With the assistance of computer-aided diagnosis (CAD), the overall sensitivity from human observers can be improved by 10% on average, which provides a promising solution in such a context.

Detection and classification of microcalcification clusters (MCCs) from mammograms plays important roles in early diagnosis of breast cancer. In early detected cases, MCCs can be found in 30–50% of the screened mammograms. This will increase to 60–80% if histological examinations of cancer cases are considered. The difficulty for the detection of MCCs is due to (i) small size but various shapes, (ii) low contrast and unclear boundary from surrounding normal tissue, etc. (Cheng et al., 2003; Sentelle et al., 2002). To solve such problems, a typical CAD system contains at least four stages including preprocessing, feature-based extraction of regions of interest (ROI), detection of MCCs, and classification. The preprocessing covers noise suppression and contrast enhancement, including histogram equalization, etc. (Papadopoulos et al., 2008), which is useful for robust extraction of features and ROIs. The features include local statistics and texture modeling (Griz et al., 2009), wavelets (Rashed et al., 2007; Soltanian-Zadeha et al., 2004; Heinlein et al., 2003; Sentelle et al., 2002), and morphological features (Kallergi, 2004), and ROIs can be determined via segmentation followed by local thresholding, spatial filtering and active contours (Cheng et al., 2003; Hassanien, 2007; Kallergi, 2004).
2. Review of ANN learning techniques

In this paper, the classification of MCCs is treated as a two-class pattern classification problem, and the two classes are referred to as "malignant" and "benign". If we denote \( \mathbf{x} \in \mathbb{R}^d \) as an input vector or pattern to be classified, and let scalar \( y \) denote its class label, i.e. \( y \in \{0, 1\} \) for ANN. The training set \( \mathbf{L} \) contains \( M \) samples, i.e. \( \mathbf{L} = \{ (\mathbf{x}_i, y_i) \} \) and \( i \in \{1, M\} \). The problem here is how to determine a classifier \( f(\mathbf{x}) \) which can make correct decision and classify the input pattern into suitable classes. In this section, brief introduction to ANN are presented, which forms the base of our improved classifier as presented in the next section.

2.1. The ANN classifier

Although there is no precise definition, ANN can be considered as an information processing system which is composed of a network of interconnected simple processing elements, i.e. neurons. Determined by the connections between these neurons and the associated parameters, ANN can exhibit complex global behavior to generate expected outputs via supervised or unsupervised learning. Inspired by the biological nervous system, the learning process is to adjust the connection strength or weights between the neurons. Each neuron forms a node in the whole network and after training each node is assigned with a determined bias or threshold. For each interconnection between two nodes, a weight is also assigned to represent the link-strength between the neurons.

For a given input vector \( \mathbf{x} = (x_1, x_2, \ldots, x_d)^T \) and weight vector \( \mathbf{w} = (w_1, w_2, \ldots, w_d)^T \), the output of a single neuron \( Z \) in Fig. 1 is determined as

\[
Z = g(\mathbf{w}^T \mathbf{x} - b) = g \left( \sum_{i=1}^{d} w_i x_i - b \right)
\]

where \( g(\cdot) \) is namely an activation function to decide whether the perceptron should fire or not. The sigmoid function \( \text{Sig}(x) = (1 + e^{-x})^{-1} \) is the most popular used activation function, and others include \( \text{tanh} \) and step functions, etc.

Using the same process as to compute the output of a single neuron, the output of the whole network can also be calculated in a topological manner. This means that for each neuron its inputs from other neurons need to be computed before determining its output. As seen, the weight vector and the bias associated to each connection and each node will influence the outputted results, and they can be determined in training or learning process as follows. First of all, the topology of the ANN needs to be specified, and the feed-forward ANN is adopted as it has been widely applied for the classification of MCCs (De Santantonio et al., 2003; Sajda et al., 2002; Verma and Zakos, 2001; Kallergi, 2004). A feed-forward ANN is a multi-layer perceptron (MLPP) which contains three or more layers of neurons, i.e. one input layer, one output layer and at least one hidden layer. With a given training set, a specified activation function and a learning ratio \( \gamma \) where \( \gamma \in (0, 1) \), the learning process for supervised training using the well-known back-propagation algorithm can be described in the following three stages.

Firstly, the initial weights and bias are set randomly between \([-1, 1]\) to attain a group of outputs \( \mathbf{z}^{(0)} \) at \( t=1 \) referring to the first round of iteration. Then, an error function is decided as \( \alpha(t) = \sum_{j=1}^{M} \gamma_j z_j^{(0)} \) using the sum squared error between the estimated output \( z \) and the target output \( y \). Finally, the error signal at the output units is propagated backwards through the whole network to update the weights using the gradient descent rule

\[
\Delta w_{ji}(t) = -\gamma \frac{\partial C(t)}{\partial w_{ji}}
\]

where \( w_{ji} \) refers to a weight between the jth node in a given layer and the ith node in the following layer. With updated weights, we can set \( t = t+1 \) to start a new iteration until the network becomes stable.
convergence. This can be measured by using a small change ratio of $c(\cdot)$ or a given number of iterations.

3. Improved neural classifier with balanced learning

In general, the performance on classification of MCCs remains unsatisfied at around 80% in terms of $A_2$ (Wei et al., 2009, 2005; El-Naqa et al., 2004; Papadopoulosab et al., 2005), and this accuracy may degrade further if the distribution of the samples is severely imbalanced (Wei et al., 2005). Unfortunately, such imbalance distribution is widely found for MCCs classification, as usually there are much more (> 4 times) benign samples than malignant ones in the training sets (Wei et al., 2005; Wu et al., 2008). Therefore, the performance of the classifier may bias to the majority class and fails for correct detection of MCCs. For this purpose, we have proposed an improved strategy, namely balanced learning, to overcome this problem.

3.1. Strategy in balanced learning

To achieve balanced learning, there are two main technical streams, i.e. data level and algorithm level methods (Kotsiantis et al., 2006; Liu et al., 2006). At the data level, balanced learning is achieved using various re-sampling solutions to balance the training data (Chawla et al., 2002). On the other hand, algorithm level solutions intend to adjust the cost function, decision threshold or the learnt probability for refined learning, such as the work reported in Tang et al. (2009), Hong et al. (2007), and Huang et al. (2006). Using Bayes optimal classifier theory, it is found that individual classifier has a fundamental performance limit which makes it little better than that of the majority class (Kotsiantis et al., 2006; Drummond and Holte, 2005; Chawla et al., 2002). Consequently, data-level solutions are preferred for balanced training in our paper.

Regarding data level solutions, there are two strategies in data re-sampling which include over-sampling of the minority class or under-sampling of majority class. Straightforward over- and under-sampling refer to random replication in the minority class and discarding samples in the majority class. Although under-sampling may reduce the size of the training set for efficiency, it may lead to serious problems in accurate modeling the majority class as most of data are ignored. On the contrary, random over-sampling seems to be a better solution despite of the increased training set.

Since random over-sampling may increase the likelihood of over-fitting in dealing with the duplicated samples, several smart sampling techniques have been presented such as synthetic over-sampling (SMOTE) (Chawla et al., 2002). In SMOTE, synthetic minority samples are generated via interpolation of one random sample and its nearest neighbors. Some other smart sampling techniques include one-sided selection, cluster-based over-sampling, Wilson's editing, etc., and details of which can be referred to the work in Hulse et al. (2007).

3.2. Proposed balanced learning strategy

According to the extensive experiments in Hulse et al. (2007), it is found that random sampling outperforms several smart sampling techniques and unaltered data set. However, the evaluation in Wei et al. (2005) indicates that random over-sampling seems not improving the performance in classification of MCCs, and similar finding is concluded in detecting sentence boundaries in Liu et al. (2006). Besides, it is indicated that SMOTE may perform down-sampling in certain cases (Liu et al., 2006). These inconsistent results need to be further clarified before applying any sampling strategies to classify MCCs for improved performance.

Fig. 2 shows a typical two-class classification problem which contains combined linear decision boundaries. This is very common in machine learning domain and the segment of the decision boundary can also be nonlinear. For the two classes marked as circle and star shapes, two pairs of same-class samples are extracted satisfying minimum neighboring distance and marked as A–B and C–D. According to the rules of smart sampling in SMOTE, synthetic samples can be generated for balanced learning. Unfortunately, the generated samples in these cases are unreliable noisy ones which may inevitably degrade the performance of training and classification.

To some degree, the analysis above can explain why smart sampling behaves well in some cases. The more complex the decision boundary is, the more noisy samples are introduced via smart sampling, and hence the worse performance may be achieved. On the other hand, smart sampling like SMOTE may work well in simpler cases such as the linear problem in detection of sentence boundary in Liu et al. (2006). Since there are much more negative samples than positive ones, the strategy here is for each positive sample in the training set to introduce additional samples. These newly introduced samples are almost replications of the original one with minor changes (increasing or decreasing at less than 1% after normalizing the range of the feature values within $[-1, 1]$) to one item of the feature values which is randomly determined. This helps to keep consistency between generated samples and the original ones for balanced learning and avoiding the problem caused by smart sampling as discussed above. Please note that it is assumed that the samples in our test set contain no noise instances thus the over-fitting caused by over-sampling in training can be avoided.

3.3. Optimized decision making

In our implemented ANN classifier, the outputs are continuous values rather than binary symbols. Conventional methods use simple thresholding in decision making. If the outputs are larger than a chosen threshold, a positive sample is detected. Otherwise, it is decided as negative. However, this simple thresholding suffers uneven distribution of the training outputs and leads to poor performance. To overcome this drawback, on the contrary, optimized decision making using optimal thresholding is proposed and described as follows.

Optimal thresholding is achieved through statistical analysis of the classifier outputs as follows. Let $z_i$ denote the predicted output...
for a given input sample $\mathbf{x}$, with a target label $y$, $y \in \{0, 1\}$, where $z_i \in (a_0, a_1)$ and the parameters $a_0$ and $a_1$ represent, respectively, the lowest and the highest boundary of the output from the classifier. Then, two conditional probabilities $p(z_i|y=0)$ and $p(z_i|y=1)$ are obtained. For a given threshold $T_0$, the optimal threshold $T_{\text{opt}}$ is determined when the minimum cost of error classification is achieved. Consequently, this optimal threshold can be used to obtain another group of classification results. The effectiveness of the proposed optimized decision making has been fully validated by our experimental results as shown in Section 5:

$$
T_{\text{opt}} = \arg\min T \text{Err}(T)
$$

### 4. Implementation and evaluation study

The data set and feature set as well as evaluation strategy are discussed in this section, along with some implementation details. These are essential for consistent evaluation of our proposed methodology to compare with others.

#### 4.1. Data set

For performance evaluation, in total 748 suspicious MCCs are collected, which contain 633 benign and 115 malignant samples where the ratio between them is nearly 5.5. These MCCs are extracted from 295 full-field mammograms in the well-known DDSM database, in which the mammography data from more than 2600 patients are scanned at 50 $\mu$m using LUMISYS (Heath et al., 1998; Heath et al., 2001). The collected MCCs are then randomly divided into two datasets for training and testing, respectively. Examples of benign and malignant cases are shown in Fig. 3.

To detect suspicious MCC regions, optimal filtering using texture measurements is employed (Wu et al., 2006, 2008). Firstly, some preprocessing is applied to remove the influence of background and several artifacts like white/black spots and scratches. Then, optimal filtering is employed using local frequency in terms of energy distribution extracted from mammograms. Finally, adaptive thresholding is utilized as post-processing for further robustness. Relevant details can be found in Wu et al. (2006).

#### 4.2. Feature set

Breast microcalcifications appear as small white specks in various patterns on the mammogram (Cheng et al., 2003). Whether their clusters are malignant or benign depends on the size, shape and geographic distribution of all microcalcification regions in a cluster, i.e. if they are tightly clustered and have certain linear structure, etc. Therefore, the extracted features need to measure these properties accordingly which include the area, the scattered degree and brightness of the regions in a cluster. In total 23 features are extracted from each of the segmented microcalcification clusters, and a list of them is shown in Table 1.

As shown, except the three single measures #1–3, the

### Table 1

<table>
<thead>
<tr>
<th>No.</th>
<th>Meanings</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Patients’ age (Cheng et al., 2003; Wu et al., 2008; Kallergi, 2004)</td>
<td>Mean</td>
</tr>
<tr>
<td>2</td>
<td>The number of regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>3</td>
<td>Least inter-distance of all regions in a cluster</td>
<td>Std.</td>
</tr>
<tr>
<td>4</td>
<td>The average intensities of all regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>5</td>
<td>The areas of all regions in a cluster</td>
<td>Std.</td>
</tr>
<tr>
<td>6</td>
<td>The compactness of all regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>7</td>
<td>The measure Fourier description FF of all regions in a cluster</td>
<td>Std.</td>
</tr>
<tr>
<td>8</td>
<td>The moment-based measure M of all regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>9</td>
<td>The eccentricity of all regions in a cluster</td>
<td>Std.</td>
</tr>
<tr>
<td>10</td>
<td>The spread of all regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>11</td>
<td>The average minimum std. of $r(\theta, l)$ of all regions in a cluster</td>
<td>Std.</td>
</tr>
<tr>
<td>12</td>
<td>Average std. of the minimum std. of $r(\theta, l)$ at various directions in all regions in a cluster</td>
<td>Mean</td>
</tr>
<tr>
<td>13</td>
<td>Average std. of the string of length $l$, starting from each point in a region at direction $\theta$</td>
<td>Std.</td>
</tr>
</tbody>
</table>

Examples of mammogram images for breast cancer detection, where samples #0014 and #1252 are shown. Malignant cases are found in the right image and also highlighted.

![Fig. 3. Examples of mammogram images for breast cancer detection, where samples #0014 and #1252 are shown. Malignant cases are found in the right image and also highlighted.](Image)
other 20 features in the feature set are composed of the mean and standard deviation values of ten features. These 20 features can be categorized into three classes including (i) intensity statistics (#4–5), (ii) shape features (#6–17), and (iii) linear structure features (#18–23). Introductions to most of these features can be found in Cheng et al. (2003), Ge et al. (2006), Papadopoulosab et al. (2005), Nemoto et al. (2006), Wu et al. (2008), Jiang et al. (2010), Kallergi (2004), Papadopoulos et al. (2008), and Bocchi et al. (2004).

The three single measures are patient’s age, the number of regions in a cluster, and the average least inter-distance (LID) of all regions. Since about 80% of the diagnosed breast cancer cases are for women over 50 years old (Cancer Research UK, 2009), age is a good indicator and has been widely used in the classification of MCCs (Cheng et al., 2003; Wu et al., 2008; Kallergi, 2004). A MCC is defined as a group of at least three microcalcifications within 1 cm², and the number of microcalcifications in a cluster is larger than its width. To measure the consistency of the intensities along the linear structures in a MCC, six features are extracted as shown in Table 1 using the mean and standard deviations of three measurements (Wu et al., 2008).

Shape features are very important indicators in this field, in which the area (size), the compactness, Fourier descriptors, moments, eccentricity, and the spread of these regions are commonly used. The definitions of these shape features can be referred to in Cheng et al. (2003), Papadopoulosab et al. (2005), Wu et al. (2008), Kallergi (2004), and Papadopoulos et al. (2008). Please note that these measures can be extracted from each microcalcification region within a candidate MCC, and the mean and standard derivation values over all regions are then determined for classification purpose.

Linear structure features form another important feature set, which has been widely used in detection and classification of MCCs (Zwiggeelaar et al., 2004; Nakayama et al., 2006; Ge et al., 2006; Wu et al., 2008). Linear structure here means a string of pixels (representing a line) with similar intensity along a certain direction, which can be denoted as r(t, l) where t; and I refer respectively to the direction and the length in the linear structure. In addition, the pixel intensities on the line are higher than that of its neighboring ones (Wei et al., 2009), which can also be used to measure the scattered degree of the distribution of the microcalcifications in a cluster. In addition, the intensity measures are also useful as high intensity is expected for the white specks in MCCs (Cheng et al., 2003; Papadopoulosab et al., 2005).

As mentioned above, all 748 MCC samples are randomly partitioned into two subsets for training and testing, respectively. All the positive samples in the training set are over-sampled for balanced learning. The trained model is then used to classify samples in the test set. This process is repeated 10 times to overcome any bias in data partition. The average performance over these 10 tests is taken as a final result for evaluations.

For a two-class problem, let us denote TP and TN as the numbers of correctly classified positive and negative samples, FP and FN for the numbers of incorrectly classified positive and negative samples, i.e. false alarms and missed positives. Several metrics can be determined below for quantitative evaluations:

\[
TP_{\text{rate}} = \frac{Recall}{TP} = \frac{TP}{TP + FN} \tag{5}
\]

\[
Precision = \frac{TP}{TP + FP} \tag{6}
\]

\[
F_{\text{rate}} = 1 - \frac{FAR}{Recall + Precision} \tag{7}
\]

To enable a single measure of performance, the \( F_1 \) score is also obtained as follows:

\[
F_1 = \frac{2 \cdot \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{8}
\]

Receiver operating characteristic (ROC) analysis and its variants are commonly used for quantitative evaluations of classifiers, especially for the detection and classification of MCCs (Cheng et al., 2003). In ROC analysis, TP vs. FP rates are adopted. Under ROC analysis, the area under the ROC curve \( A_z \) is also used as an important evaluation criterion (Cheng et al., 2003), where \( A_z = 1 \) indicates an ideal case with \( TP_{\text{rate}} = 100\% \) and \( F_{\text{rate}} = 0 \).

For the ANN classifier, the number of nodes in the hidden layer is empirically set as 15 for the better results achieved. The training process stops when the training performance keeps unchanged over a long time, say more than 4000 iterations. The performance is measured using the \( F_1 \), and the parameters which yield the highest \( F_1 \) value is stored and used for testing.

5. Results and discussions

In this section, comprehensive experiments are conducted and the results are presented. Quantitative evaluations are used to validate the effectiveness of our proposed method including balanced learning and optimal decision making.

5.1. Performance of balanced learning

First of all, the performance of balanced learning is compared with those training using the original data, and the training ratio is set as 80%, i.e. 80% of the samples for training and 20% for testing. The ROC curves are shown in Fig. 4 to show the performance of training and testing with or without balanced learning, respectively, where several facts can be summarized as follows. Firstly, training results are in general much better than testing ones. Secondly, balanced learning indeed can yield better results despite of a little higher false positive rate. Regarding training, it has generated slightly higher recall rate though its recall rate without balanced learning is already high enough. For testing, balanced learning produces much improved results. Finally, it is worth noting that balanced learning seems inferior to unbalanced learning only if the false positive rate is less than 3%. As a higher recall rate is always desirable in such applications, balanced learning still seems better than unbalanced ones.

Quantitative comparisons of the results are also reported and shown in Table 2, in which no optimized decision making is
applied in the testing. First of all, with balanced learning, the testing performance in terms of \( F_1 \) score and \( A_z \) are significantly improved, which has validated the effectiveness of balanced learning. In addition, it is found that the training performance is high in terms of all the five measures no matter balanced training is used or not. This has indicated that ANN is capable of modeling the problem accurately. However, the testing performance under balanced training is much better, in which an improvement of 12.8% is achieved in both \( F_1 \) and \( A_z \) measurements. Although the testing data is severe imbalanced, a high Specificity value is still achieved to yield a higher \( A_z \) measurement. Since the ratio between FP and TP is much larger than the ratio between FP to TN, this has led to a lower \( F_1 \) score but a higher \( A_z \) value.

5.2. Performance of optimized decision making

In Table 3, the effectiveness of our proposed optimized decision making is presented. Again, we select 80% of samples for training and 20% for testing. By comparing the results in Tables 2 and 3, we can clearly find how optimized decision making helps to generate improved performance as summarized below:

- Without balanced learning, optimized decision making contributes for ANN about 1% in \( F_1 \) score and 3.4% in \( A_z \) measurements.

### Table 2

<table>
<thead>
<tr>
<th>No balanced learning</th>
<th>Balanced learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
</tr>
<tr>
<td>Recall</td>
<td>0.957</td>
</tr>
<tr>
<td>Precision</td>
<td>1.000</td>
</tr>
<tr>
<td>Specificity</td>
<td>1.000</td>
</tr>
<tr>
<td>( F_1 )</td>
<td>0.978</td>
</tr>
<tr>
<td>( A_z )</td>
<td>0.979</td>
</tr>
</tbody>
</table>

### Table 3

Testing results from ANN under optimized decision making with or without balanced learning.

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Precision</th>
<th>Specificity</th>
<th>( F_1 )</th>
<th>( A_z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original ANN</td>
<td>0.818</td>
<td>0.643</td>
<td>0.921</td>
<td>0.720</td>
<td>0.870</td>
</tr>
<tr>
<td>Improved ANN</td>
<td>1.000</td>
<td>0.788</td>
<td>0.949</td>
<td>0.881</td>
<td>0.975</td>
</tr>
</tbody>
</table>

5.3. Performance under various training ratios

In the tests above, the training ratio is fixed at 80%. In this group of tests, the performance under various training ratios is compared. Under various training ratios, the training results and testing results with or without optimized decision making are further evaluated in terms of \( F_1 \) and \( A_z \) measurements. These results are shown in Fig. 5, where Test 2 denotes results with optimal decision making. In total three pairs of curves are plotted in each figure, in which one pair is from training and the other two are from testing without or with optimal decision making. Each pair of curves is plotted using the training ratio (changed from 50% to 90%) vs. performance of \( F_1 \) and \( A_z \) measurements and they are further discussed as follows.

Firstly, the \( F_1 \) score and \( A_z \) measurement from training with or without balanced learning are very close to each other and appear insensitive to the training ratio, which again validates that ANN is capable of accurate modeling the problem. Secondly, the testing results with balanced learning are much better that those without balanced learning. Thirdly, in most cases optimized decision making produces better results in \( F_1 \) and \( A_z \) measurements when balanced learning is employed, except the result at the training ratio of 60%. When balanced learning is not used, however, better \( F_1 \) measurement can only be yielded when the training ratio is between 70% and 80%. In addition, better \( A_z \) measurement can always be achieved from optimized decision making even without balanced learning.

5.4. Computational complexity

In comparison with conventional ANN, the proposed balanced learning and optimized decision making do need additional computations. As optimized decision making is excluded in the training iterations, it can be simply ignored. In the following, we will analyze the effect of balanced learning in such a context.

Since more samples are introduced in balanced learning, it costs additional time in learning the model. Let \( N = N_0 + N_1 \) be the total samples, where \( N_0 \) and \( N_1 \) denote, respectively, the number of negative and positive samples satisfying \( N_0 = KN_1, K > 2 \). After over-sampling of positive samples, \((K-1)N_1\) new positive samples are produced and in total we have \(2N_0\) training samples. This equals to \(2NK/(K+1)\) and less than \(2N\), which has indicated that the number of samples under balanced learning is less than twice of the number of the original samples. Under same number of
iterations, the training time should be no more than double of the one without balanced learning.

In addition, it is found that balanced learning needs no number of iterations to converge, which is about 77% of the one required for training without balanced learning. This fast converging might due to the improved distributions of training samples from our balanced learning. Consequently, the increased complexity under balanced learning becomes

$$2K/(K+1)77% - 1 = 0.54 - 1.54/(K + 1)$$

(9)

This indicates a maximum of 54% additional computing burden, which is totally acceptable for the benefit of much improved performance.

6. Conclusions

In this paper, an improved neural classifier is proposed, where balanced learning and optimized decision making are introduced for classification of benign and malignant MCCs in mammograms. The experiments are conducted on 748 samples extracted from the well-known DDSM database, and the main findings can be summarized as follows.

Firstly, balanced learning indeed has significantly improved the classification accuracy, and an average gain of more than 10% can be achieved for ANN in terms of both $F_1$ score and $A_c$ measurement. Secondly, optimized decision making yields improved results in $F_1$ score and $A_c$ no matter balanced learning is used or not. Thirdly, a training ratio between 70% and 80% is suggested due to the various performances under different training ratios. Finally, it is found that the suggested balanced training will only bring up to 54% of additional computation load, a tolerable cost for the much-improved performance. Further investigations include applying feature selection approaches for improved efficiency as well as reducing false alarms for more robustness.

Acknowledgment

Firstly, we wish to thank anonymous reviewers and the editor for their insightful comments. The authors would also like to thank Dr. Z.-Q. Wu for his help in data collection and feature extraction in this work.

References


