An adaptive incremental approach to constructing ensemble classifiers: Application in an information-theoretic computer-aided decision system for detection of masses in mammograms

Maciej A. Mazurowski^{a)}

Department of Radiology, Carl E. Ravin Advanced Imaging Laboratories, Duke University Medical Center, Durham, North Carolina 27705

Jacek M. Zurada

Department of Electrical and Computer Engineering, Computational Intelligence Laboratory, University of Louisville, Louisville, Kentucky 40292

Georgia D. Tourassi

Department of Radiology, Carl E. Ravin Advanced Imaging Laboratories, Duke University Medical Center, Durham, North Carolina 27705

(Received 23 September 2008; revised 18 April 2009; accepted for publication 20 April 2009; published 9 June 2009)

Ensemble classifiers have been shown efficient in multiple applications. In this article, the authors explore the effectiveness of ensemble classifiers in a case-based computer-aided diagnosis system for detection of masses in mammograms. They evaluate two general ways of constructing subclassifiers by resampling of the available development dataset: Random division and random selection. Furthermore, they discuss the problem of selecting the ensemble size and propose two adaptive incremental techniques that automatically select the size for the problem at hand. All the techniques are evaluated with respect to a previously proposed information-theoretic CAD system (IT-CAD). The experimental results show that the examined ensemble techniques provide a statistically significant improvement ($AUC=0.905 \pm 0.024$) in performance as compared to the original IT-CAD system ($AUC=0.865 \pm 0.029$). Some of the techniques allow for a notable reduction in the total number of examples stored in the case base (to 1.3% of the original size), which, in turn, results in lower storage requirements and a shorter response time of the system. Among the methods examined in this article, the two proposed adaptive techniques are by far the most effective for this purpose. Furthermore, the authors provide some discussion and guidance for choosing the ensemble parameters. © 2009 American Association of Physicists in Medicine. [DOI: [10.1118/1.3132304](https://doi.org/10.1118/1.3132304)]

I. INTRODUCTION

In recent years, computational intelligence (CI) and machine learning (ML) applications in medical decision support are gaining popularity, specifically in computer-aided analysis of medical images.¹⁻⁷ One of the most popular tasks employing CI and ML techniques is the classification task where a label has to be assigned to an incoming query case determining its membership in one of the predefined classes (e.g., benign vs malignant). Classification is made based on the characteristics of the query case.

Traditionally, a single classifier, such as linear discriminant analysis (LDA),⁸ artificial neural network (ANN),⁹ or k -nearest neighbor classifier (k-NN),¹⁰ is used for the task. Such classifiers use all previously acquired examples (clinical cases) to develop the system, typically through training. This approach has been shown to be quite effective and robust. However, it has disadvantages. A single classifier can be susceptible to initialization and training parameters and it is always dependent on the particular training dataset that may or may not properly reflect characteristics of the underlying population. Overtraining can be a significant detrimental factor when the training parameters are not carefully optimized. In the context of these difficulties, ensemble classifiers (also called multiclassifier systems) are becoming

a sensible and increasingly popular choice.¹¹ They are known to offer a more efficient use of the available data and, as a result, improved performance. Ensemble techniques have been examined extensively in the machine learning literature¹¹⁻¹³ and they have been shown effective in multiple applications.¹⁴⁻¹⁸ Applications of ensemble techniques in computer-aided diagnosis (CAD) have been explored to a lower extent. However, one can find some studies on the topic.¹⁹⁻²⁷

The underlying idea of ensemble classifiers is to construct multiple subclassifiers and then develop a combiner that summarizes the predictions provided by all the subclassifiers into one final decision. Various approaches have been proposed for both subclassifier construction and subclassifier combination.¹¹ In order to construct multiple subclassifiers one can (i) assign different training/development parameters to each subclassifier, (ii) use different types of subclassifiers, or (iii) use different subsets of the development dataset for each subclassifier. When the subclassifiers are created, a combiner must be designed that uses predictions of all the subclassifiers to make a final decision. Two general ways of combining subclassifiers are (i) classifier fusion and (ii) classifier selection. In classifier fusion (also called cooperative approach), all the predictions are merged into one final deci-

sion. In classifier selection (also called competitive approach), the best/most appropriate subclassifier is selected and its prediction is used as the final decision.

Even though ensemble classifiers are often more effective than single classifiers, they impose certain challenges such as selecting multiple parameters needed to construct an effective ensemble. While researchers typically concentrate their attention on various ways of constructing and combining classifiers, the issue of ensemble size is mostly left out. Only a few studies tackle the topic (for an overview see Ref. 11). These studies typically propose an “overproduce and select” method where an initial large set of subclassifiers is constructed and then it is reduced using various methods. Giacinto and Roli,^{28,29} for example, used clustering of the constructed neural network-based subclassifiers and selected the optimal subset based on their diversity and performance. Margineantu and Dietterich³⁰ used kappa statistic to prune the adaptive boosting ensemble to a prespecified size. A similar approach called “thinning the ensemble” can be found in the study by Banfield *et al.*³¹ All these techniques start with a set of subclassifiers and select a subset to construct the final ensemble. Note that the size of the initial ensemble needs to be specified by the designer. Furthermore, all these techniques make use of the concept of diversity to make a selection. In this article we propose adaptive incremental techniques that start with a single classifier and then gradually add subclassifiers based on the overall performance of the resulting system. The proposed techniques do not require the designer to specify the size of ensemble but determine it automatically.

In this study we focus on building ensembles for case-based classifiers.³² Instead of extracting rules from the available training data, case-based classifiers store the actual examples in the database of the system (called case base). When a new unknown query arrives, its similarity to the case base examples is assessed and based on these similarities a decision regarding the query is made. Therefore, case base size in the system directly impacts the system’s storage requirements and response time. These two issues are of particular importance in the information-theoretic CAD (IT-CAD) (Ref. 33) system employed in this study. IT-CAD stores the previously acquired examples as entire images (large storage requirement) and uses a computationally expensive mutual information index as a measure of similarity between examples.³⁴

In most studies, the only purpose of using ensembles is to improve the classification performance of the system. However, if the subclassifiers are case based, then the total number of examples used in the system may be a concern since it determines the response time and storage requirement of the system. In this article, we explore the possibility of using ensemble techniques to reduce the total number of examples stored in IT-CAD, while at the same time improving the system’s performance.

The article is organized as follows. In Sec. II, we describe briefly the IT-CAD system employed in this study and describe the examined ensemble techniques in more detail. In Sec. III we discuss the database used in the study as well as

the experimental design. In Sec. IV, we present the experimental results. Section V concludes the article with a final discussion.

II. METHODS

II.A. Information-theoretic computer-aided decision system

We based our study on the information-theoretic CAD system proposed by Tourassi *et al.*³³ for the classification of mammographic regions of interest (ROIs) as depicting masses or normal tissue. IT-CAD is a case-based classifier. To assess a similarity between images, the information-theoretic concept of mutual information is used. Mutual information $I(X, Y)$ between two random variables X and Y is defined as follows:

$$I(X, Y) = \sum_x \sum_y P_{XY}(x, y) \log_2 \frac{P_{XY}(x, y)}{P_X(x)P_Y(y)}. \quad (1)$$

To apply this concept to images, the probability distributions $P_X(x)$ and $P_Y(y)$ are replaced with intensity histograms of images X and Y and the joint probability distribution $P_{XY}(x, y)$ is replaced with a joint histogram of the two images. Specifically to quantify the similarity between two images, normalized mutual information (NMI) measure is used here,

$$\text{NMI} = \frac{2I(X, Y)}{H(X) + H(Y)}, \quad (2)$$

where H is the image entropy. Given the quantified similarities of the query image Q to all the images in the case base, a decision index d is calculated for the query,

$$d(Q) = \frac{1}{m} \sum_{i=1}^m \text{NMI}(Q, M_i) - \frac{1}{n} \sum_{j=1}^n \text{NMI}(Q, N_j), \quad (3)$$

where M_i are ROI examples depicting masses, N_j are ROI examples depicting normal tissue, m is a number of mass examples, and n is a number of normal examples in the case base. To classify the query image Q , a decision threshold T is applied such that Q is classified as positive (mass) if $d(Q) > T$ and as negative if $d(Q) \leq T$.

Note that $d(Q)$ can be seen as a special type of k -nearest neighbor classifier, where k is equal to the number of examples in the case base. In the original study on IT-CAD,³³ the impact of k on classification performance was evaluated and it was shown that the system’s performance is as good using all examples as it is when using a carefully optimized set of k -nearest neighbors. Since there is no performance or computational benefit in carefully optimizing k (i.e., still all possible NMIs need to be calculated and rank ordered), we used the $k=|S|$ configuration for our study.

II.B. Constructing the ensemble

As described before, the process of constructing an ensemble consists of two primary steps: Constructing subclassifiers and constructing a combiner. The process is depicted

FIG. 1. A diagram showing the general idea of building an ensemble with a case-based system.

in Fig. 1. This section describes both steps in detail and is concluded by a description of the proposed adaptive incremental way of building ensembles.

II.B.1. Constructing subclassifiers

Since we are using a featureless case-based classifier, there are no natural parameters (e.g., number of features) that can be varied and there is no training involved in the process of constructing IT-CAD. Thus, we concentrate on resampling the development dataset as a technique for constructing subclassifiers. Therefore, each subclassifier in this study is simply an IT-CAD based on a certain subset of the development dataset. In this study we compare two different ways of creating these subsets.

Random division. In this approach, the development dataset T is divided randomly into L mutually exclusive subsets. The sum of such subsets is equal to the original development dataset. The subsets have the same (or approximately the same if the total number of examples does not allow for an equal division) size. Note that in this approach the total number of examples used in the resulting ensemble is the same as in the original classifier.

Random selection. In random selection, for each subclassifier, a set of N examples is selected randomly from the development dataset T . The selected subsets can overlap. The maximum total number of distinct examples stored in the system for this approach is $L \cdot N$ (L is a number of classifiers). Note that this is an upper bound for the total number of examples and the actual number can be lower due to the overlap in the selected subsets. In this approach, the total number of distinct examples stored in the ensemble can be significantly lower than in the development dataset available for the original classifier and thus, random selection could offer a case base reduction benefit.

Random selection approach is similar to “bootstrap aggregating” (bagging) proposed by Breiman.¹⁵ In bagging method, bootstrap sampling is used to construct each subclassifier. Even though bagging method is shown to provide considerable improvement in performance as compared to single classifiers, it allows for a very limited, if any, reduction in the total number of examples used by the ensemble.

II.B.2. Combining subclassifiers

To combine decisions of subclassifiers, fusion based on linear discriminant analysis was used in the following way. We chose LDA since it is simple, effective, and popular particularly in medical decision support research. LDA fusion is implemented as follows. Each example $x^{(i)}$ is presented to each subclassifier in the ensemble $D = \{D_1, D_2, \dots, D_L\}$ resulting in a vector of decision values $(d_1^{(i)}, d_2^{(i)}, \dots, d_L^{(i)})$. Such vector, together with the ground truth information for the example $f(x^{(i)})$ is used as a training example for the second-level LDA combiner. Therefore the number of examples used for training the LDA is always equal to the number of examples in the entire development dataset. When $x^{(i)}$ is included in the case base of one or more subclassifiers, this example is temporarily removed from such case base while calculating the decision to avoid a bias. This method is very similar to stacked generalization used by Wolpert.³⁵

II.B.3. An incremental strategy

In most of the previously proposed ensemble techniques, the classifier designer must decide *a priori* the number of subclassifiers. Too few subclassifiers can result in a poor performance because of lack of diversity in the ensemble.¹¹ In contrast, too many subclassifiers can pose problems as well. When a trainable second-level classifier is used as the combiner, the number of inputs in the combiner is equal to the number of subclassifiers. It is well known that a high ratio of number of inputs to the number of examples can deteriorate the classifier performance due to overtraining (i.e., “curse of dimensionality”).⁸ Furthermore, in the random selection approach, more subclassifiers mean higher time complexity of the classification algorithm when classifying new incoming queries.

In this article, we propose an adaptive incremental strategy of building ensembles utilizing clinically relevant performance measures as optimization criteria. The proposed methods are based on the idea of monitoring the ensemble performance during the building stage. Specifically, two different methods are proposed to adapt the number of subclassifiers to the problem at hand. Additionally, we explore a nonadaptive incremental method for comparison.

The two adaptive methods use the same following algorithm of evaluating the candidate ensemble performance. Given the ensemble $D = \{D_1, D_2, \dots, D_L\}$, the development dataset is divided into an internal training (90% examples) and internal test (10% examples) sets. Then all examples in the training are fed into all the subclassifier and the individual subclassifier responses are used to construct an LDA combiner in the way described in Sec. II B 2. To test the

classifier, all examples from the internal test set are classified using the ensemble. The performance of the ensemble evaluated on the internal test set is used to assess generalization capabilities of the ensemble. Various methods of evaluating performance could be used at this stage. In this study we use the receiver operator characteristic (ROC) analysis and the area under the ROC curve (AUC)^{36–38} as it is clinically relevant. Note that in this internal LDA training and testing processes, all examples included in the subclassifiers remain there independently on whether they belong to internal training or internal test set. However, in either of these processes, if an example that is presented to the ensemble belongs to one or more subclassifiers, it is temporarily removed from such subclassifier(s) while calculating the subclassifier responses. This internal train and test procedure is repeated ten times implementing a tenfold crossvalidation scheme. The average internal test AUC performance is used for performance evaluation of the ensemble D . We will denote it as $AUC(D)$. This performance measure is assumed to approximate the generalizing abilities of the ensemble. Given such performance measure, we evaluate three incremental ways of constructing the ensemble.

Building ensemble with “no-control” method. In the first method (we will call it no-control method), the process of constructing an ensemble is initiated with a single subclassifier based on a randomly selected set of examples. Then, another subclassifier, also based on the randomly selected set of examples, is added. The subclassifiers do not have to satisfy any criteria to be added. This process is stopped by the system designer by specifying the number of sub-classifiers to be included in the ensemble. Therefore, this method is nonadaptive.

Building ensemble with “add-if-better” method. In the second method (we will call it add-if-better method), the ensemble is initiated as an empty set $D = \emptyset$. Then at each step k , one subclassifier is added to the ensemble. The new subclassifier becomes an element of the ensemble if it improves the performance of the ensemble $AUC(D(k))$ by a certain margin. Formally, the ensemble is extended by adding the new subclassifier if it satisfies

$$AUC(D(k)) > AUC(D(k-1)) + 0.001, \quad (4)$$

where $D(k)$ is the candidate ensemble in step k . The margin of 0.001 minimal improvement in performance is added to avoid unnecessary expansion of the ensemble with only marginal improvement in the performance. If the new subclassifier does not satisfy this condition, it is not included in the ensemble. In such case another candidate subclassifier is evaluated. This process is repeated up to 50 times in each step. If none of the 50 evaluated candidate subclassifiers satisfy the condition, the process of ensemble building is terminated.

Building ensemble with “pick-best” method. In the third method (we will call it pick-best method), the ensemble is also initiated as an empty set $D = \emptyset$. Then in each step, 50 candidate subclassifiers are constructed and the subclassifier that, when added to the ensemble, provides the best improve-

FIG. 2. A diagram showing different kinds of ensemble techniques used in this article.

ment in the ensemble performance is included in the ensemble. The process is terminated when none of the 50 subclassifiers satisfies condition (4).

Figure 2 presents all the ensemble techniques used in this article in a diagram. Note that the proposed adaptive incremental approach and thus the two methods presented above (add-if-better and pick-best methods) are applicable only to the random selection approach. In random division, as well as in random selection with no-control method, the number of subclassifiers must be determined *a priori*.

III. DATABASES AND EXPERIMENTAL DESIGN

III.A. Databases

In order to evaluate experimentally the performance of the ensemble techniques, we used the digital database of screening mammography.^{39,40} We used Lumisys scans of the original mammograms to 12 bit images at 50 $\mu\text{m}/\text{pixel}$. From these images, a set of 512×512 pixels ROI was extracted. A total of 1820 ROIs were used. 901 mass ROIs (489 malignant and 412 benign masses) were extracted based on the physicians annotations. 919 normal ROIs were extracted randomly by sampling the breast region of normal mammograms (without overlap). The task of the examined CAD system is to distinguish between ROIs depicting masses and those depicting normal tissue.

Database 1 consisted of 1500 ROIs randomly selected from the 1820 available examples. 738 of them depicted masses and 762 depicted normal tissue. Database 1 was used in the main part of the experimental evaluation. Database 2 consisted of the remaining 320 ROIs. 163 of them depicted masses and 157 depicted normal tissue. Database 2 was used for additional validation.

III.B. Experimental design

In the experiments we evaluated the performance of all proposed techniques and compared it to the performance of the original IT-CAD system. We focused on three factors that affect the ensemble's performance: The way of creating subclassifiers, number of subclassifiers, and (in the case of random selection) number of examples in each subclassifier.

For the random division approach, we evaluated the performance of the system with the number of subclassifiers varying from 2 to 500. The number of examples in each

subclassifier strictly depended on the number of subclassifiers ($N=N_{\text{dev}}/L$, where N_{dev} is the total number of examples in the development dataset).

For the random selection approach, we evaluated the ensemble performance for $N=2, 10, 20, 50, 100, 200$. As described in Sec. II, we used three methods for incremental construction of ensembles: No-control (nonadaptive), add-if-better (adaptive), and pick-best (adaptive) methods. For the no-control method, we evaluated the performance of the system in a wide range of L . For the other two methods, we simply executed the development process and recorded the performance of the system as well as the number of subclassifiers after the process was terminated.

To ensure an accurate estimation of the system performance, we used a hold-out data handling scheme in the following way. Database 1 was divided randomly into a development dataset (90% of database 1, 1350 examples) and a testing set (10% of database 1, 150 examples). The entire process of building an ensemble was executed using the development set. The testing set was used only for the final validation. The process was repeated 200 times. We report the average test performance and its variability over these 200 splits/run. To compare the performance of ensemble to the performance of the original system, a paired t-test based on these 200 splits was used.

Note that this data handling is similar to tenfold crossvalidation. It allows for incorporating three sources of performance variability in the statistical analysis: test set (test set is resampled multiple times), training set (training set is resampled multiple times), and random component of the algorithm (the algorithm is run multiple times). The hold-out data handling used in our experiments allows for a better estimation of the average performance and of the performance variability coming from these three different sources since the entire data split, training, and testing procedure is repeated 200 times (as compared to ten times in tenfold crossvalidation).

As an additional validation, the entire database 1 was used as a development dataset and database 2 was used for testing. The development and testing was repeated only once for all the techniques and examined parameters. This emulates a real-life system development and testing scenario. In all our experiments, for consistency, we used the Wilcoxon nonparametric estimation of the area under the ROC curve.

IV. EXPERIMENTAL RESULTS

We present the results in two subsections. In Sec. IV A we concentrate on the scenario when the system developer's only goal is to maximize system performance. In Sec. IV B, we focus on reducing the total number of examples stored in the system. We also evaluate the impact of the number of subclassifiers L and the number of examples N in each subclassifier on the obtained performance as well as the total number of distinct examples included in an ensemble system.

TABLE I. Maximum AUC test performance for the examined ensemble techniques.

N	Random selection			Random division
	No-control	Add-if-better	Pick-best	
2	0.903 ± 0.024	0.902 ± 0.023	0.898 ± 0.025	0.905 ± 0.024 $L=100$ $N=13-14$
5	0.903 ± 0.023	0.901 ± 0.024	0.899 ± 0.024	
10	0.903 ± 0.024	0.901 ± 0.024	0.899 ± 0.023	
20	0.904 ± 0.024	0.900 ± 0.025	0.901 ± 0.025	
50	0.904 ± 0.024	0.902 ± 0.024	0.901 ± 0.024	
100	0.904 ± 0.024	0.902 ± 0.023	0.900 ± 0.024	
200	0.903 ± 0.023	0.901 ± 0.023	0.901 ± 0.024	

IV.A. Improving performance

The average AUC test performance (for 200 crossvalidation splits) for the examined techniques is presented in Table I with respect to the number of examples N in each subclassifier. In this context, N is an algorithm parameter that has to be decided by the system designer. For the nonadaptive methods (random division and random selection with no-control method), we reported the maximum test average performance obtained when varying ensemble sizes L . For the adaptive methods (random selection with add-if-better and pick-best methods), we simply reported the average performance obtained for the method. The performance of the original, single IT-CAD classifier system was 0.865 ± 0.029 . We established that all the ensemble techniques examined in this article provided a similar and statistically significant ($p < 0.001$) improvement in performance as compared to the original IT-CAD. Furthermore, we observed that for random selection approach performance improvement was independent of the number of examples N included in each subclassifier.

IV.B. Reducing the total number of examples stored in the system

Figure 3 presents the relationship between the number of subclassifiers L and the average test ROC performance for database 1. For the techniques where the number of subclassifiers L is determined *a priori* by the designer, this relationship is represented by a curve to show how the performance depends on the choice of L . For the techniques where L is determined automatically by the algorithm, it is represented by a single point (a square or a circle). The solid curves show the performance for the random selection approach with no control for $N=2, 20, 200$. Only these three subclassifier sizes were selected to keep the graph simpler since the results for $N=5, 10, 50, 100$ followed the same trends. The dotted curve shows the performance of random division. The performance of random selection with add-if-better and pick-best methods is represented by squares and circles, respectively.

Several conclusions can be drawn from Fig. 3. When the designer selects the number of subclassifiers L (i.e., random selection with no-control method and random division), the performance is clearly dependent on L . Specifically, the AUC performance index initially increases and then, after reaching

FIG. 3. A relation between AUC performance of the system and the number of subclassifiers.

its maximum, it deteriorates with increasing L . The initial improvement in the performance ($L=1$ to $L \approx 100$) can be explained by the fact that the response of each additional subclassifier to a query can be treated as an additional feature of that query and thus can be useful in the decision process and, in turn, improve the overall performance. On the other hand, the drop in performance for $L > 150$ can be explained by the fact that adding subclassifiers corresponds to adding inputs to a second level combiner which given a limited number of training examples causes overtraining. The number of examples in each subclassifier N had no impact on susceptibility of the ensemble to overtraining. The classifier designer needs to remember that the value of L for which the performance reaches its maximum may depend on the number of examples available for the development dataset and needs to be determined with a considerable experimental effort. This calls for reliable algorithms to select the size of ensembles automatically such as the two adaptive methods proposed in this article (random selection with add-if-better and pick-best methods).

As the next step, we examined the described techniques based on if and how their performance depends on the total number of distinct examples used in the ensemble. This issue is important not only to select ensemble parameters providing the best achievable performance but also in a scenario when the total number of examples used in a system is a concern. For random division approach, the total number of distinct examples is always equal to the number of examples in the original development dataset. Therefore only random selection approach can potentially offer a reduction in the case base used in the system. Furthermore, note that in random selection the total number of distinct examples used in the system can be lower than the product of number of subclassifiers and number of examples in each subclassifier

TABLE II. Total number of distinct examples selected for the random selection approach.

N	Random selection		
	No-control	Add-if-better	Pick-best
2	219.8 ± 4.3	45.7 ± 10.9	17.7 ± 5.3
5	484.1 ± 8.4	106.9 ± 21.9	46.2 ± 12.5
10	796.5 ± 10.4	176.7 ± 36.9	84.8 ± 21.8
20	874.0 ± 11.0	293.3 ± 57.0	140.6 ± 34.7
50	1335.1 ± 3.8	556.5 ± 113.1	257.6 ± 73.6
100	1349.4 ± 0.8	841.6 ± 142.1	429.7 ± 132.7
200	1350.0 ± 0.0	1149.7 ± 124.4	730.6 ± 197.1

($L \times N$). This is the case because the same example can occur in more than one subclassifier. Therefore, $L \times N$ constitutes an upper bound for the total number of distinct examples stored in the system.

The total number of distinct examples used in the ensemble is presented in Table II. For the methods where the number of subclassifiers is determined automatically by the method, the total number of distinct examples is averaged over 200 splits. For the method where L is decided *a priori* by the system designer (no-control method), the number of distinct examples is presented for L that provided maximum average test performance (i.e., an optimal choice of the ensemble size). Since the total number of distinct examples for a given number of subclassifiers L may slightly vary, the number of examples averaged over 200 splits is provided in the table.

For the no-control method, the resulting total number of distinct examples highly depends on the number of examples N in each subclassifier. Specifically, increasing N resulted in increasing total number of distinct examples. This method offered a notable reduction in the case base for $N=2-20$. The reduction for $N=50-100$ was marginal and there was no reduction at all for $N=200$.

As for the methods where L is determined automatically (random selection with add-if-better and pick-best methods), Table II indicates that they provide a large reduction to the total number of distinct examples used in the system. For add-if-better as well as for pick-best-methods, a notable reduction was obtained for all N . It is clear that for both techniques it is beneficial to use very low number of examples in each subclassifier N if case base reduction is a concern.

IV.C. Final validation

To further validate the obtained results we conducted an additional experiment. We used the entire database 1 as a development dataset and database 2 as a testing dataset. It simulates a real-life scenario when a limited dataset is available to the system designer and the constructed system is tested on new, unknown examples. The baseline AUC performance (original IT-CAD) for this scenario was 0.871 ± 0.020 (estimate based on 5000 bootstrap samples). The best test AUC performance provided by random division

technique was 0.914 ± 0.016 and was obtained for $L=90$ resulting in $N=16.7$ (1500/90) examples in a subclassifier in average. The obtained performance was statistically significantly better than the baseline performance ($p < 0.001$).

Random selection with no-control method resulted in the maximum test AUC performance (estimate based on 5000 bootstrap samples) of 0.909 ± 0.016 ($L=160$) for $N=2$, 0.912 ± 0.016 ($L=60$) for $N=5$, 0.903 ± 0.017 ($L=60$) for $N=10$, 0.908 ± 0.016 ($L=60$) for $N=20$, 0.918 ± 0.015 ($L=30$) for $N=50$, 0.912 ± 0.016 ($L=50$) for $N=100$, and 0.909 ± 0.016 ($L=18$) for $N=200$. The resulting AUC performance for all N was statistically significantly better than the baseline (two-tailed $p < 0.001$). To further validate our conclusions concerning the drop in performance for high L caused by overtraining of the combiner, we examined the performance for $L=500$. It was considerably lower than the maximum performance for all values of N and was equal to 0.881 ± 0.019 for $N=2$, 0.860 ± 0.021 for $N=5$, 0.875 ± 0.020 for $N=10$, 0.874 ± 0.019 for $N=20$, 0.882 ± 0.019 for $N=50$, 0.878 ± 0.020 for $N=100$, and 0.874 ± 0.019 for $N=200$.

Applying random selection with add-if-better method provided the following test AUC performance: 0.903 ± 0.017 for $N=2$ ($L=22$), 0.916 ± 0.015 for $N=5$ ($L=17$), 0.914 ± 0.015 for $N=10$ ($L=24$), 0.907 ± 0.016 for $N=20$ ($L=19$), 0.906 ± 0.017 for $N=50$ ($L=13$), 0.896 ± 0.017 for $N=100$ ($L=12$), and 0.901 ± 0.017 for $N=200$ ($L=9$). The performance was statistically significantly better (two-tailed $p < 0.001$) for all N . Applying random selection with pick-best method also statistically significantly improved the performance of the original IT-CAD for all N . The test AUC performance was equal to 0.901 ± 0.017 for $N=2$ ($L=11$), 0.904 ± 0.017 for $N=5$ ($L=14$), 0.913 ± 0.016 for $N=10$ ($L=9$), 0.906 ± 0.016 for $N=20$ ($L=7$), 0.899 ± 0.017 for $N=50$ ($L=4$), 0.900 ± 0.017 for $N=100$ ($L=6$), and 0.903 ± 0.017 for $N=200$ ($L=11$). It is apparent that random selection with add-if-better as well as with pick-best methods allowed for a significant reduction of the total number of examples. Overall, the final validation results were consistent with the conclusions drawn in the primary experiment.

IV.D. Comparison to other techniques

To further validate our efforts in proposing new techniques for building ensembles, we compared their performance to some other well established techniques in machine learning. The two algorithms that we implemented for the comparison are boosting¹⁴ and bootstrap aggregating¹⁵ (bagging). First, we evaluated the performance of the boosting algorithm called AdaBoost.M1.¹⁴ The idea of this algorithm is to reinforce the training examples that are often misclassified by providing them with a higher weight in a subclassifier (boosting by reweighting) or a higher probability of being included in the training of a subclassifier (boosting by resampling). In our experiments we use boosting by reweighting. For the details of how to incorporate weights in our IT-CAD system, please see our recent publication.⁴¹

AdaBoost results in a binary ensemble classifier. We made a natural extension such that the resulting ensemble classifier returns a continuous decision value so that we can perform ROC analysis. In order to do so, when calculating the final decision of the ensemble, we utilized the actual outputs on IT-CAD based subclassifiers as opposed to the labels obtained by applying a threshold. We applied the AdaBoost.M1 algorithm on database 1 using the same crossvalidation as in the main experiments (200 random splits). For AdaBoost.M1, we obtained AUC performance of 0.873 ± 0.028 , a small improvement as compared to the original, single IT-CAD (AUC= 0.865 ± 0.029). The obtained performance, however, is still inferior compared to the performance of the methods proposed in this study.

Furthermore, we tested the Breiman's bagging algorithm. The idea of this algorithm is to create multiple subclassifiers by resampling (bootstrap) the available set of examples. Then, subclassifiers are combined using simple averaging of individual decision indices. Database 1 with 200 random splits was used to evaluate performance as in the previous experiment. For bagging, we obtained AUC performance of 0.865 ± 0.029 which indicates no improvement as compared to the original IT-CAD and far inferior performance as compared to our techniques. This result suggests that second-level classifiers are more effective as opposed to averaging in the step of combining decisions of subclassifiers.

Note that in their original implementation, AdaBoost.M1 and Breiman's bagging do not offer reduction in the case base size. In the following experiments, we preceded the two ensemble algorithms with a step of case base reduction. We used a classical machine learning technique called edited nearest neighbor (ENN) proposed by Wilson.⁴² This algorithm simply removes all examples that are misclassified by their neighbors. These misclassified examples are considered outliers. Although there are multiple case base reduction in machine learning,⁴³ we decided to use ENN as it is simple and well established in the field. Furthermore, ENN allows for automatic selection of the selected subset size. Although manual selection of the subset size can be beneficial (see the techniques evaluated in our recent paper solely devoted to the problem of optimizing case bases⁴⁷), in this study it would be impractical to introduce another degree of freedom to the analysis. A slightly modified version of the edited nearest neighbor algorithm has been also applied to reduction of the reference database in a problem of false positive reduction in mammography by another group.⁴⁴ In our experiments we used database 1 with the same crossvalidation scheme as in the main experiments (200 random splits).

The ENN reduction alone resulted in a moderate decrease in performance as compared to the original IT-CAD system (AUC= 0.832 ± 0.032 vs AUC= 0.865 ± 0.029). The benefit of this step is a reduction in the case base size to the average of 1053.6 ± 7.6 examples (reduction by 22%). Following the example selection step with AdaBoost.M1 resulted in performance of AUC= 0.853 ± 0.027 . This is a moderate increase as compared to the selection step alone. However, the result-

ing performance is still below the level of the original IT-CAD system and notably below the performance of our methods. Following the example selection step with bagging resulted in AUC performance of 0.832 ± 0.032 , i.e., the same as for the selection step alone. This finding is consistent with the results for applying bagging without the selection step (no change in performance).

In conclusion, the techniques proposed in this study are superior, as applied to our IT-CAD system, than the two well established ensemble algorithms in machine learning: Ada-Boost.M1 and Breiman's bagging. Even though some reduction can be obtained by applying an example selection algorithm prior to building ensemble, the reduction is notably worse than in our techniques and the hybrid (reduction + ensemble) system performs worse than our methods. Future studies can evaluate performance of combination of example selection and ensemble learning while other selection techniques are applied.

V. CONCLUSIONS AND DISCUSSION

In this article, we evaluated the effectiveness of ensemble techniques with the information-theoretic CAD system previously proposed by our group for the detection of masses in mammograms. We discussed advantages and limitations of these techniques. We compared two general approaches to constructing subclassifiers: Random division and random selection and we used LDA classifier as a second-level combiner. In response to one of the limitations of ensemble classifiers, specifically the fact that it is often not clear how many subclassifiers should be included in the ensemble, we proposed two adaptive methods that determine the number of subclassifiers automatically.

The study results allow us to draw the following conclusions:

- All examined ensemble methods provide a similar and statistically significant improvement in performance ($\text{AUC}=0.898 \pm 0.026$ to $\text{AUC}=0.905 \pm 0.024$) as compared to the original system ($\text{AUC}=0.865 \pm 0.029$).
- For the methods where the number of subclassifiers L is determined *a priori* by the system designer (random selection with no-control and random division), the obtained performance is highly dependent on this choice.
- When random selection is used to construct subclassifiers, the achieved performance is not affected by the number of examples N in each subclassifier.
- Ensemble techniques can be used to notably reduce the total number of examples used in the case-based system.
- The two adaptive methods (random selection with add-if-better and pick-best methods) proposed in this article turn out to be the most effective if a case base reduction is a concern.

- (a) The total number of distinct examples can be reduced to as few as 45.7 ± 10.9 examples (3.4% of the original case base size) for random selection with add-if-better method.

- (b) The total number of distinct examples can be reduced even further to 17.7 ± 5.3 examples (1.3% of the original case base size) if random selection with pick-best method is used.

- The proposed methods are superior to the previously proposed in the machine learning literature bagging and boosting methods as applied to our CAD system.

The finding that the total number of examples can be reduced to as few as 17.7 ± 5.3 examples or 1.3% of the original case base without compromising the system's performance may seem controversial since in many studies the reference database used for the system is very large. However, it is consistent with multiple recent studies showing that when intelligent techniques are applied for the selection of examples, a dramatic case base reduction can be obtained. For example, Skalak *et al.*⁴⁵ showed that the case base of a k -nearest neighbor system can be reduced to only 1% of its original size. Similarly, Wilson⁴³ demonstrated that the case base can be reduced to 1% of its original size without compromising the performance of the system. In a very recent study Pekalska *et al.*⁴⁶ reduced the database of a case-based system to only 20 examples without compromising the performance of the original system. In one of our recent studies,⁴⁷ we also showed that when such techniques as random mutation hill climbing or a selection technique based on genetic algorithms are applied to our IT-CAD in mammography, the case base can be reduced to 2%–4% of the original size without compromising performance. In this study we show that ensemble techniques can also be used for this purpose.

Note also that the proposed techniques have been applied only to a case-based system. However, it can be adapted to rule-based systems such as support vector machines or neural networks which are popular choices for computer-assisted mass detection schemes. The resulting ensembles could also provide some insight into which examples are the most useful in such systems. This can be a part of future research.

To conclude, ensemble techniques turned out to be very effective in our system. The two proposed incremental algorithms of constructing ensembles are preferable since in addition to a significant improvement in performance, they allow for substantial reduction of case base size and they automatically adapt the number of subclassifiers to the classification problem at hand.

ACKNOWLEDGMENTS

This work was supported in part by Grant No. R01 CA101911 from the National Cancer Institute and the University of Louisville Grosscurth Fellowship.

^{a)}Electronic mail: maciej.mazurowski@duke.edu

¹K. Doi, "Current status and future potential of computer-aided diagnosis in medical imaging," *Br. J. Radiol.* **78**, S3–S19 (2005).

²M. P. Sampat, M. K. Markey, and A. C. Bovik, *Handbook of Image and Video Processing* (Academic, New York, 2005), pp. 1195–1217.

³J. Y. Lo, A. O. Bilska-Wolak, M. K. Markey, G. D. Tourassi, J. A. Baker, and C. E. Floyd, Jr., *Recent Advances In Breast Imaging, Mammography, and Computer-Aided Diagnosis of Breast Cancer* (SPIE, Bellingham,

- 2006), pp. 871–900.
- ⁴I. Sluimer, A. Schilham, M. Prokop, and B. van Ginneken, “Computer analysis of computed tomography scans of the lung: A survey,” *IEEE Trans. Med. Imaging* **25**, 385–405 (2006).
 - ⁵S. Katsuragawa and K. Doi, “Computer-aided diagnosis in chest radiography,” *Comput. Med. Imaging Graph.* **31**, 212–223 (2007).
 - ⁶K. Doi, “Computer-aided diagnosis in medical imaging: Historical review, current status and future potential,” *Comput. Med. Imaging Graph.* **31**, 198–211 (2007).
 - ⁷J. J. Perumpillichira, H. Yoshida, and D. V. Sahani, “Computer-aided detection for virtual colonoscopy,” *Cancer Imaging* **5**, 11–16 (2005).
 - ⁸R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification* (Wiley-Interscience, New York, 2000).
 - ⁹J. M. Zurada, *Introduction to Artificial Neural Systems* (West Publishing Co., St. Paul, 1992).
 - ¹⁰T. Mitchell, *Machine Learning* (McGraw-Hill, New York, 1997).
 - ¹¹L. I. Kuncheva, *Combining Pattern Classifiers* (Wiley-Interscience, New York, 2004).
 - ¹²J. Kittler, M. Hatef, R. P. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 226–239 (1998).
 - ¹³R. Ranawana and V. Palade, “Multi-classifier systems: Review and a roadmap for developers,” *International Journal of Hybrid Intelligent Systems* **3**, 35–61 (2006).
 - ¹⁴Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” *Proceedings of the 13th International Conference on Machine Learning* (Springer, New York, 1996).
 - ¹⁵L. Breiman, “Bagging predictors,” *Mach. Learn.* **24**, 123–140 (1996).
 - ¹⁶K. Woods, W. P. Kegelmeyer, Jr., and K. Bowyer, “Combination of multiple classifiers using local accuracy estimates,” *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 405–410 (1997).
 - ¹⁷L. I. Kuncheva, J. C. Bezdek, and R. P. Duin, “Decision templates for multiple classifier fusion: An experimental comparison,” *Pattern Recogn.* **34**, 299–314 (2001).
 - ¹⁸L. I. Kuncheva, “Switching between selection and fusion in combining classifiers: An experiment,” *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.* **32**, 146–156 (2002).
 - ¹⁹I. Bloch, “Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account,” *Pattern Recogn. Lett.* **17**, 905–919 (1996).
 - ²⁰Z.-H. Zhou and Y. Jiang, “Medical diagnosis with c4.5 rule preceded by artificial neural network ensemble,” *IEEE Trans. Inf. Technol. Biomed.* **7**, 37–42 (2003).
 - ²¹D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, “Ensemble clustering in medical diagnostics,” in *Proceedings of 17th IEEE Symposium on Computer-Based Medical Systems (CBMS 2004)* (IEEE, Piscataway, 2004), pp. 576–581.
 - ²²D. West, P. Mangiameli, R. Rampal, and V. West, “Ensemble strategies for a medical diagnostic decision support system: A breast cancer diagnosis application,” *Eur. J. Oper. Res.* **162**, 532–551 (2005).
 - ²³M. Raza, I. Gondal, and R. L. C. David Green, “Classifier fusion using Dempster-Shafer theory of evidence to predict breast cancer tumors,” in *2006 IEEE Region 10 Conference (TENCON 2006)* (IEEE, Piscataway, 2006), pp. 1–4.
 - ²⁴J. L. Jesneck, L. W. Nolte, J. A. Baker, C. E. Floyd, and J. Y. Lo, “Optimized approach to decision fusion of heterogeneous data for breast cancer diagnosis,” *Med. Phys.* **33**, 2945–2954 (2006).
 - ²⁵M. A. Mazurowski, J. M. Zurada, and G. D. Tourassi, “Database decomposition of a knowledge base CAD system in mammography; an ensemble approach to improve detection performance,” in *Proceedings of the SPIE, Medical Imaging 2008: Computer-Aided Diagnosis* (SPIE, Bellingham, 2008), Vol. 6915, p. 69151K.
 - ²⁶M. A. Mazurowski, J. M. Zurada, and G. D. Tourassi, “Reliability assessment of ensemble classifiers: Application in mammography,” in *International Workshop on Digital Mammography* (Springer, New York, 2008).
 - ²⁷B. Zheng, Y. H. Chang, W. F. Good, and D. Gur, “Performance gain in computer-assisted detection schemes by averaging scores generated from artificial neural networks with adaptive filtering,” *Med. Phys.* **28**, 2302–2308 (2001).
 - ²⁸G. Giacinto and F. Roli, “Design of effective neural network ensembles for image classification purposes,” *Image Vis. Comput.* **19**, 699–707 (2001).
 - ²⁹G. Giacinto and F. Roli, “An approach to the automatic design of multiple classifier systems,” *Pattern Recogn. Lett.* **22**, 25–33 (2001).
 - ³⁰D. D. Margineantu and T. G. Dietterich, “Pruning adaptive boosting,” in *Proceedings of the 14th International Conference on Machine Learning* (Springer, New York, 1997), pp. 378–387.
 - ³¹R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, “A new ensemble diversity measure applied to thinning ensembles,” in *Proceedings of the Fourth International Workshop on Multiple Classifier Systems (MCS 2003)*, LNCS 2709 (Springer, New York, 2003), pp. 306–316.
 - ³²D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.* **6**, 37–66 (1991).
 - ³³G. D. Tourassi, R. Vargas-Voracek, D. M. Catarious, and C. E. Floyd, “Computer-assisted detection of mammographic masses: A template matching scheme based on mutual information,” *Med. Phys.* **30**, 2123–2130 (2003).
 - ³⁴G. D. Tourassi, B. Haarawood, S. Singh, J. Y. Lo, and C. E. Floyd, “Evaluation of information-theoretic similarity measures for content-based retrieval and detection of masses in mammograms,” *Med. Phys.* **34**, 140–150 (2007).
 - ³⁵D. H. Wolpert, “Stacked generalization,” *Neural Networks* **5**, 241–259 (1992).
 - ³⁶A. P. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *Pattern Recogn.* **30**, 1145–1159 (1997).
 - ³⁷T. Fawcett, “An introduction to ROC analysis,” *Pattern Recogn. Lett.* **27**, 861–874 (2006).
 - ³⁸N. A. Obuchowski, “Receiver operating characteristic curves and their use in radiology,” *Radiology* **229**, 3–8 (2003).
 - ³⁹M. Heath, K. Bowyer, D. Kopans, W. P. Kegelmeyer, R. Moore, K. Chang, and S. MunishKumaran, in *Proceedings of the Fourth International Workshop on Digital Mammography* (Kluwer Academic, Dordrecht, 1998), pp. 457–460.
 - ⁴⁰M. Heath, K. Bowyer, D. Kopans, R. Moore, and W. P. Kegelmeyer, “The digital database for screening mammography,” in *Proceedings of the Fifth International Workshop on Digital Mammography* (Springer, New York, 2001), pp. 212–218.
 - ⁴¹M. A. Mazurowski, P. A. Habas, J. M. Zurada, and G. D. Tourassi, “Decision optimization of case-based computer aided decision systems using genetic algorithms with application to mammography,” *Phys. Med. Biol.* **53**, 895–908 (2008).
 - ⁴²D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Trans. Syst. Man Cybern.* **2**, 408–421 (1972).
 - ⁴³R. Wilson and T. R. Martinez, “Reduction techniques for instance-based learning algorithms,” *Mach. Learn.* **38**, 257–286 (2000).
 - ⁴⁴S. C. Park, R. Sukthankar, L. Mummert, M. Satyanarayanan, and B. Zheng, “Optimization of reference library used in content-based medical image retrieval scheme,” *Med. Phys.* **34**, 4331–4339 (2007).
 - ⁴⁵D. B. Skalak, “Prototype and feature selection by sampling and random mutation hill climbing algorithms,” in *Proceedings of the 11th International Conference on Machine Learning* (Springer, New York, 1994), pp. 293–301.
 - ⁴⁶E. Pekalska, R. P. Duin, and P. Paclik, “Prototype selection for dissimilarity-based classifiers,” *Pattern Recogn.* **39**, 189–208 (2006).
 - ⁴⁷M. A. Mazurowski, J. M. Zurada, and G. D. Tourassi, “Selection of examples in case-based computer-aided decision systems,” *Phys. Med. Biol.* **53**, 6079–6096 (2008).