Comparing Ensembles of Learners: Detecting Prostate Cancer from High Resolution MRI

Anant Madabhushi[1], Jianbo Shi[2], Michael Feldman[2], Mark Rosen[2], and John Tomaszewski[2]

[1] Rutgers The State University of New Jersey, Piscataway, NJ 08854
[2] University of Pennsylvania, Philadelphia, PA 19104
anantm@rci.rutgers.edu

Abstract. While learning ensembles have been widely used for various pattern recognition tasks, surprisingly, they have found limited application in problems related to medical image analysis and computer-aided diagnosis (CAD). In this paper we investigate the performance of several state-of-the-art machine-learning methods on a CAD method for detecting prostatic adenocarcinoma from high resolution (4 Tesla) *ex vivo* MRI studies. A total of 14 different feature ensemble methods from 4 different families of ensemble methods were compared: Bayesian learning, Boosting, Bagging, and the k-Nearest Neighbor (kNN) classifier. Quantitative comparison of the methods was done on a total of 33 2D sections obtained from 5 different 3D MRI prostate studies. The tumor ground truth was determined on histologic sections and the regions manually mapped onto the corresponding individual MRI slices. All methods considered were found to be robust to changes in parameter settings and showed significantly less classification variability compared to inter-observer agreement among 5 experts. The kNN classifier was the best in terms of accuracy and ease of training, thus validating the principle of *Occam's Razor*[1]. The success of a simple non-parametric classifier requiring minimal training is significant for medical image analysis applications where large amounts of training data are usually unavailable.

1 Introduction

Learning ensembles (Bagging [2], Boosting [3], and Bayesian averaging [4]) are methods for improving classification accuracy through aggregation of several similar classifiers' predictions and thereby reducing either the bias or variance of the individual classifiers [1]. In Adaptive Boosting (AdaBoost) proposed by Freund and Schapire [3] sequential classifiers are generated for a certain number of trials and at each iteration the weights of the training dataset are changed based on the classifiers that were previously built. The final classifier is formed using a weighted voting scheme. With the Bagging [2] algorithm proposed by Brieman, many samples are generated from the original data set via *bootstrap sampling*

[1] One should not increase, beyond what is necessary, the number of entities required to explain anything.

R.R. Beichel and M. Sonka (Eds.): CVAMIA 2006, LNCS 4241, pp. 25–36, 2006.

and then a component learner is trained from each of these samples. The predictions from each of these learners is then combined via *majority voting*. Another popular method of generating ensembles is by combining simple Bayesian classifiers [5]. The class conditional probabilities for different attributes or features can be combined using various different rules (e.g., median, max, min, majority vote, average, product, and weighted average). The drawback of Boosting, Bagging, and Bayesian learners, however, is that they require training using labeled class instances. This is an issue in most medical image analysis applications where training data is usually limited. Consequently there still remains considerable interest in simple fusion methods such as the k-Nearest Neighbor (kNN) classifier for performing general, non-parametric classification [5] which have the advantages of (1) being fast, (2) having the ability to learn from a small set of examples, and (3) can give competitive performance compared to more sophisticated methods requiring training.

While several researchers have compared machine learning methods on real world and synthetic data sets [1,7,8,9,10], these comparison studies have usually not involved medical imaging data [11]. Warfield *et al.* proposed STAPLE [6] to determine a better ground truth estimate for evaluating segmentation algorithms by combining weighted estimates of multiple expert (human or machine learners) segmentations. Other researchers have attempted to combine multiple classifiers with a view to improving classification accuracy. Attempts to compare learning ensembles have often lead to contradictory results, partly due to the fact that the data sets used in these comparisons tend to be application specific. For instance Wei *et al.* [11] found that Support Vector Machines (SVMs) outperformed Boosting in distinguishing between malignant and benign microcalcifications on digitized mammograms. Martin *et al.* [10], however, found that Boosting significantly outperformed SVMs in detecting edges in natural images. Similarly, while Quinlan [1] found that Boosting outperformed Bagging, Bauer and Kohavi [8] found that in several instances the converse was true.

In [4] we presented a computer-aided diagnosis (CAD) method for identifying lesions on high-resolution (4 Tesla (T)) *ex vivo* MRI studies of the prostate. Our methodology comprised of (i) extracting several different 3D texture features at multiple scales and orientations, (ii) estimating posterior conditional Bayesian probabilities of malignancy at every spatial location in the image, and (iii) combining the individual posterior conditional probabilities using a weighted linear combination scheme. In this paper we investigate the performance of 14 different ensembles from 4 families of machine learning methods, Bagging, Boosting, Bayesian learning, and kNN classifiers for this important CAD application. The motivations for this work were (1) to improve the performance of our CAD algorithm, (2) to investigate whether trends and behaviors of different classifiers reported in the literature [1,7,8] hold for medical imaging data sets, and (3) to analyze the weaknesses and strengths of known classifiers to this CAD problem, not only in terms of their accuracy but also in terms of training and testing speed, feature selection methods, and sensitivity to system parameters. These issues are important for (i) getting an understanding of the classification process

and (ii) because the trends observed for this CAD application may be applicable to other CAD applications as well.

This paper is organized as follows. Section 2 briefly describes the different classification methods investigated in this paper. In Section 3 we describe the experimental setup, while in Section 4 we present and discuss our main results. Concluding remarks and future directions are presented in Section 5.

2 Description of Feature Ensemble Methods

2.1 Notation

Let D=$\{(\mathbf{x}_i,\omega_i) \mid \omega_i \in \{\omega_T, \omega_{NT}\}, i\in\{1,..,N\}\}$ be a set of objects (in our case image voxels) $\mathbf{x}$ that need to be classified into either the tumor class ω_T or the non-tumor class ω_{NT}. Each object is also associated with a set of K features f_j, for $j \in \{1,..,K\}$. Using Bayes rule [5] a set of posterior conditional probabilities $P(\omega_T|\mathbf{x}, f_j)$, for $j\in\{1,..,K\}$, that object $\mathbf{x}$ belongs to class ω_T are generated. A feature ensemble $\mathbf{f}(\mathbf{x})$ assigns to $\mathbf{x}$ a combined posterior probability of belonging to ω_T, by combining either (i) the individual features $f_1, f_2, ..., f_K$, or (ii) the associated posterior conditional probabilities $P(\omega_T|\mathbf{x}, f_1)$, $P(\omega_T|\mathbf{x}, f_2)$,.., $P(\omega_T|\mathbf{x}, f_K)$ associated with $\mathbf{x}$, or (iii) other feature ensembles.

2.2 Bayesian Learners

Employing Bayes rule [5], the posterior conditional probability $P(\omega_T|\mathbf{x}, f)$ that an object $\mathbf{x}$ belongs to class ω_T given the associated feature f is given as

$$P(\omega_T|\mathbf{x}, f) = \frac{P(\omega_T)p(\mathbf{x}, f|\omega_T)}{P(\omega_T)p(\mathbf{x}, f|\omega_T) + P(\omega_{NT})p(\mathbf{x}, f|\omega_{NT})}, \tag{1}$$

where $p(\mathbf{x}, f|\omega_T)$, $p(\mathbf{x}, f|\omega_{NT})$ are the *a-priori* conditional densities (obtained via training) associated with feature f for the two classes ω_T, ω_{NT} and $P(\omega_T)$, $P(\omega_{NT})$ are the prior probabilities of observing the two classes. Owing to a limited number of training instances and due to the *minority class problem*[2] we assume $P(\omega_T)$=$P(\omega_{NT})$. Further since the denominator in Equation 1 is only a normalizing factor, the posterior conditional probabilities $P(\omega_T|\mathbf{x}, f_1)$, $P(\omega_T|\mathbf{x}, f_2)$,..., $P(\omega_T|\mathbf{x}, f_K)$ can be directly estimated from the corresponding prior conditional densities $p(\mathbf{x}, f_1|\omega_T)$, $p(\mathbf{x}, f_2|\omega_T)$,..., $p(\mathbf{x}, f_K|\omega_T)$. The individual posterior conditional probabilities $P(\omega_T|\mathbf{x}, f_j)$, for $j \in \{1,2,..,K\}$, can then be combined as an ensemble ($\mathbf{f}(\mathbf{x})$=$P(\omega_T|\mathbf{x}, f_1, f_2, ..., f_K)$) using the rules described below.

A. Sum Rule or General Ensemble Method (GEM): The ensemble $\mathbf{f}^{GEM}(\mathbf{x})$ is a weighted linear combination of the individual posterior conditional probabilities

$$\mathbf{f}^{GEM}(\mathbf{x}) = \sum_{j=1}^{K} \lambda_j P(\omega_T|\mathbf{x}, f_j), \tag{2}$$

[2] An issue where the instances belonging to the target class are a minority in the data set considered.

where λ_j, for $j \in \{1, 2, .., K\}$, corresponds to the individual feature weights. In [4] we estimated λ_j by optimizing a cost function so as to maximize the true positive area and minimize the false positive area detected as cancer by the base classifiers f_j. Bayesian Averaging ($\mathbf{f}^{AVG}$) is a special case of GEM, in which all the feature weights (λ_j) are equal.

B. Product rule or Naive Bayes: This assumes independence of the base classifiers and hence sometimes called *Naive Bayes* on account of the unrealistic assumption. For independent classifiers $P(\omega_T|\mathbf{x}, f_j)$, for $1 \leq j \leq K$, the probability of the joint decision rule is given as

$$\mathbf{f}^{PROD}(\mathbf{x}) = \prod_{j=1}^{K} P(\omega_T|\mathbf{x}, f_j). \tag{3}$$

C. Majority Voting: If for a majority of the base classifiers, $P(\omega_T|\mathbf{x}, f_j) > \theta$, where θ is a pre-determined threshold, $\mathbf{x}$ is assigned to class ω_T.

D. Median, Min, Max: According to these rules the combined likelihood that $\mathbf{x}$ belongs to ω_T are given by the median ($\mathbf{f}^{MEDIAN}(\mathbf{x})$), maximum ($\mathbf{f}^{MAX}(\mathbf{x})$), and minimum ($\mathbf{f}^{MIN}(\mathbf{x})$) of the posterior conditional probabilities $P(\omega_T|\mathbf{x}, f_j)$, for $1 \leq j \leq K$.

2.3 *k*-Nearest Neighbor

For a set of training samples $S = \{(\mathbf{x}_\alpha, \omega_\alpha) \mid \omega_\alpha \in \{\omega_T, \omega_{NT}\}, \alpha \in \{1, .., A\}\}$ the k-Nearest Neighbor (kNN) [5] decision rule requires selection from the set S of k samples which are nearest to $\mathbf{x}$ in either the feature space or the combined posterior conditional probability space. The final decision for the class label of $\mathbf{x}$ is to choose among the class label that appears most frequently among the k nearest neighbors of $\mathbf{x}$. Instead of making a hard (in our case binary) decision with respect to $\mathbf{x}$, as in the traditional kNN approach [5], we instead assign a soft likelihood that $\mathbf{x}$ belongs to class ω_T. Hence we define the classifier as

$$\mathbf{f}^{NN}(\mathbf{x}) = \frac{1}{k} \sum_{\gamma=1}^{k} e^{\frac{-||\Phi(\mathbf{x}) - \Phi(\mathbf{x}_\gamma)||}{\sigma}}, \tag{4}$$

where $\Phi(\mathbf{x})$ could be the feature vector $[f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_K(\mathbf{x})]$ or posterior conditional probability vector $[P(\omega_T|\mathbf{x}, f_1), P(\omega_T|\mathbf{x}, f_2), ..., P(\omega_T|\mathbf{x}, f_K)]$ associated with $\mathbf{x}$, $||\cdot||$ is the L_2 norm or Euclidean distance, and σ is a scale parameter that ensures that $0 \leq \mathbf{f}^{NN}(\mathbf{x}) \leq 1$.

2.4 Bagging

The Bagging algorithm (**B**ootstrap **agg**regation) [2] votes classifiers generated by different bootstrap samples (replicates). For each trial $t \in \{1,2,..,T\}$, a training set $S^t \subset D$ of size A is sampled with replacement. For each bootstrap training

set S^t a classifier $\mathbf{f}^t$ is generated and the final classifier $\mathbf{f}^{BAG}$ is formed by aggregating the T classifiers from these trials. To classify new instance $\mathbf{x}$, a vote for each class (ω_T, ω_{NT}) is recorded by every classifier $\mathbf{f}^t$ and $\mathbf{x}$ is then assigned to the class with most votes. Bagging however improves accuracy only if perturbing the training sets can cause significant changes in the predictor constructed [2]. In this paper Bagging is employed on the following base classifiers.

A. Bayes: For each training set S^t the *a-priori* conditional density $p^t(\mathbf{x}, f_j|\omega_T)$, for $j \in \{1, 2, .., K\}$, is estimated and the corresponding posterior conditional probabilities $P^t(\omega_T|\mathbf{x}, f_j)$ using Bayes rule (Equation 1) calculated. $P^t(\omega_T|\mathbf{x}, f_j)$, for $j \in \{1, 2, .., K\}$, can then be combined to obtain $\mathbf{f}^t(\mathbf{x})$ using any of the fusion rules described in Section 2.2.The *Bagged* Bayes classifier is then obtained as,

$$\mathbf{f}^{BAG,BAYES}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{f}^t(\mathbf{x}) > \theta), \tag{5}$$

where θ is a predetermined threshold, and $\mathbf{f}^{BAG,BAYES}(\mathbf{x})$ is the likelihood that the object belongs to class ω_T. Note that, for class assignment based on $\mathbf{f}^{BAG,BAYES}(\mathbf{x}) > 0.5$ we obtain the original Bagging classifier [2].

B. kNN: The stability of kNN classifiers to variations in training set makes ensemble methods obtained by bootstrapping the data ineffective [2]. In order to generate a diverse set of kNN classifiers with (possibly) uncorrelated errors we sample the feature space $\mathcal{F}$=$\{f_1, f_2, ..., f_K\}$ to which the kNN method is highly sensitive [12]. For each trial t=$\{1,2,..,T\}$, a bootstrapped feature set $F^t \subset \mathcal{F}$ of size $B \leq K$ is sampled with replacement. For each bootstrap feature set F^t a kNN classifier $\mathbf{f}^{t,NN}$ is generated (Equation 4). The final bagged kNN classifier $\mathbf{f}^{BAG,kNN}$ is formed by aggregating $\mathbf{f}^{t,NN}$, for $1 \leq t \leq T$, using Equation 5.

2.5 Adaptive Boosting

Adaptive Boosting (AdaBoost) [3] has been shown to significantly reduce the learning error of any algorithm that consistently generates classifiers whose performance is a little better than random guessing. Unlike Bagging [2], Boosting maintains a weight for each instance - the higher the weight, the more the instance influences the classifier learned. At each trial the vector of weights is adjusted to reflect the performance of the corresponding classifier. Hence the weight of misclassified instances is increased. The final classifier is obtained as a weighted combination of the individual classifiers votes, the weight of each classifier's vote being determined as a function of its accuracy.

Let $w^t_{\mathbf{x}_\gamma}$ denote the weight of instance $\mathbf{x}_\gamma \in S$, where $S \subset D$, at trial t. Initially for every $\mathbf{x}_\gamma$, we set $w^1_{\mathbf{x}_\gamma}$=$\frac{1}{A}$, where A is the number of training instances. At each trial $t \in \{1, 2, .., T\}$, a classifier $\mathbf{f}^t$ is constructed from the given instances under the distribution $w^t_{\mathbf{x}_\gamma}$. The error ϵ^t of this classifier is also measured with respect to the weights, and is the sum of the weights of the instances that it

mis-classifies. If $\epsilon^t \geq 0.5$, the trials are terminated. Otherwise, the weight vector for the next trial $(t+1)$ is generated by multiplying the weights of instances that $\mathbf{f}^t$ classifies correctly by the factor $\beta^t = \frac{\epsilon^t}{1-\epsilon^t}$ and then re-normalizing so that $\sum_{\mathbf{x}_\gamma} w_{\mathbf{x}_\gamma}^{t+1} = 1$. The Boosted classifier $\mathbf{f}^{BOOST}$ is obtained as

$$\mathbf{f}^{BOOST}(\mathbf{x}) = \sum_{t=1}^{T} \mathbf{f}^t \log(\frac{1}{\beta^t}) \tag{6}$$

In this paper the performance of Boosting was investigated using the following base learners.

A. Feature Space: At each iteration t, a classifier $\mathbf{f}^t$ is generated by selecting the feature f_j, for $1 \leq j \leq K$, which produces the minimum error with respect to the ground truth over all training instances for class ω_T.

B. Bayes: At each iteration t, classifier $\mathbf{f}^t$ is chosen as the posterior conditional probability $P^t(\omega_T|\mathbf{x}, f_j)$, for $j \in \{1, 2, .., K\}$, for which $P^t(\omega_T|\mathbf{x}, f_j) \leq \theta$ results in the least error with respect to the ground truth, where θ is a predetermined threshold which.

C. kNN: Since kNN classifiers are robust to variations of the training set, we employ Boosting on the bootstrap kNN classifiers $\mathbf{f}^{t,NN}$ generated by varying the feature set as described in Section 2.4. At each iteration t the kNN classifier with the least error with respect to the ground truth is chosen and after T iterations the selected $\mathbf{f}^{t,NN}$ are combined using Equation 6.

3 Experimental Setup

3.1 Data Description and Feature Extraction

Prostate glands obtained via radical prostatectomy were imaged using a 4 T Magnetic Resonance (MR) imager using 2D fast spin echo at the Hospital at the University of Pennsylvania. MR and histologic slices were maintained in the same plane of section. Serial sections of the gland were obtained by cutting with a rotary knife. Each histologic section corresponds roughly to 2 MR slices. The ground truth labels for tumor on the MR sections were manually generated by an expert by visually registering the MR with the histology on a per-slice basis. Our database comprised of a total of 5 prostate studies, with each MR image volume comprising between 50-64 2D image slices. Ground truth for the cancer on MR was only available on 33 2D slices from the 5 3D MR studies. Hence quantitative evaluation was only possible on these 33 2D MR sections.

After correcting for MR specific intensity artifacts [4], a total of 35 3D texture features at different scales and orientations and at every spatial location in the 3D MRI scene were extracted. The 3D texture features included: first-order statistics (intensity, median, standard and average deviation), second order Haralick features (energy, entropy, contrast, homogeneity and correlation), gradient

(directional gradient and gradient magnitude), and a total of 18 Gabor features corresponding to 6 different scales and 3 different orientations. Additional details on the feature extraction are available in [4].

3.2 Machine Training and Parameter Selection

Each of the classifiers described in Section 2 are associated with a few model parameters that need to be fine-tuned during training for best performance. For the methods based on Bayesian learning we need to estimate the prior conditional densities $p(\mathbf{x},f_j \mid \omega_T)$, for $1 \leq j \leq K$, by using labeled training instances. Changes in the number of training instances (A) can significantly affect the prior conditional densities. Other algorithmic parameters include (1) an optimal number of nearest neighbors (k) for the kNN classifier, (2) an optimal number of iterations (T) for the Bagging and Boosting methods, and (3) optimal values for the feature weights for the GEM technique which depends on the number of training samples used (A). These parameters were estimated via a *leave-one-out* cross validation procedure. Except for the Bagging and Boosting methods on the kNN classifiers where each kNN classifier was trained on $\frac{1}{6}^{\text{th}}$ (6) and $\frac{1}{3}^{\text{rd}}$ (12) of the total number of extracted features (35), all other classifiers were trained on the entire feature set. The Bayesian conditional densities were estimated using 5 training samples from the set of 33 2D MR slices for which ground truth was available. In Table 1 are listed the values of the parameters used for training the 14 different ensemble methods. The numbers in the parenthesis in Table 1 indicate the number of ensembles employed for each of the 4 families of learning methods.

Table 1. Values of parameters used for training the different ensemble methods and estimated via the *leave-one-out* cross validation procedure

Method	kNN (2)		Bayes	Bagging (2)		Boosting (3)		
	Features	Bayes	(7)	kNN	Bayes	kNN	Features	Bayes
Parameter	k=8	k=8	-	T=50,k=8	T=10	T=50,k=8	T=50	T=50
No. of features	35	35	35	6,12	35	6,12	35	35

3.3 Performance Evaluation

The different ensemble methods were compared in terms of accuracy, execution time, and parameter sensitivity. Varying the threshold θ such that an instance $\mathbf{x}$ will be classified into the target class if $\mathbf{f}(\mathbf{x}) \geq \theta$ leads to a trade-off between sensitivity and specificity. Receiver operating characteristic (ROC) curves (plot of sensitivity versus 100-specificity), were used to compare classification accuracy of the different ensembles using the 33 MR images for which ground truth was available. A larger area under the ROC curve implies higher accuracy of the classification method. The methods were also compared in terms of time required for classification and training. Precision analysis was also performed to assess possible *over-fitting* and parameter sensitivity of the methods compared against the inter-observer agreement of 5 human experts.

4 Results and Discussion

4.1 Accuracy

In Figure 1(a) are superposed the ROC curves for Boosting using (i) all 35 features, (ii) the posterior conditional probabilities associated with the features in (i), and (iii) the kNN classifiers trained on subsets of 6 and 12 features. Boosting all 35 features and the associated Bayesian learners results in significantly higher accuracy compared to Boosting the kNN classifiers. No significant difference was observed between Boosting the features and Boosting the Bayesian learners (Figure 1(b)), appearing to confirm previously reported results [8] that Boosting does not improve Bayesian learners.

Figure 2(a) reveals that Bagging Bayes learners performs better compared to Bagging kNN classifiers trained on reduced feature subsets. Figure 2(b), the ROC plot of 50 kNN classifiers trained on a subset of 6 features, with the corresponding Bagged and Boosted results overlaid, indicates that Bagging and

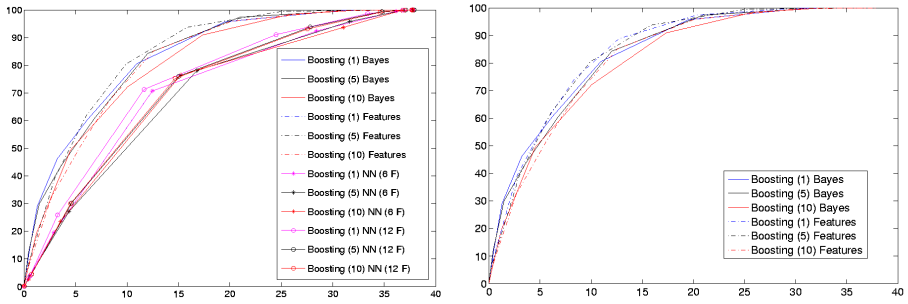


Fig. 1. ROC plots of (a) Boosting features, Bayesian learners, and kNN classifiers, (b) Boosting features and Bayesian learners. The first set of numbers (1,5,10) in the parenthesis in the figure legends indicate the number of training samples and the second set of numbers (6,12) shows the number of features used to form the kNN classifiers.

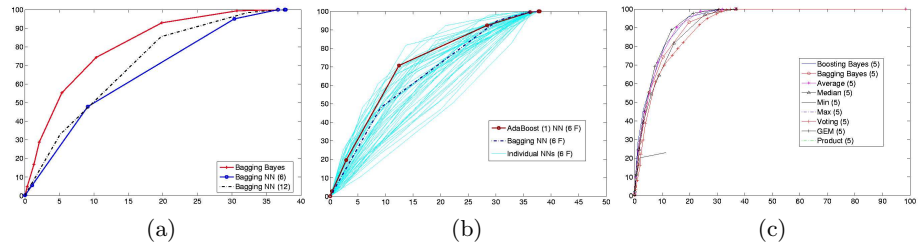
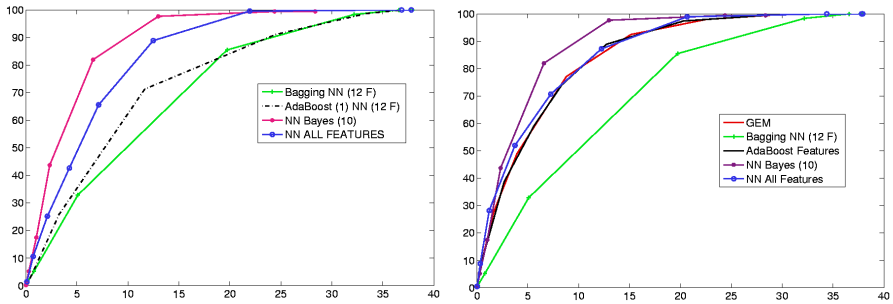


Fig. 2. ROC plots of (a) all Bagging methods, (b) 50 individual kNN classifiers trained using 6 features with the corresponding Bagged and Boosted results overlaid, and (c) different rules for combining the Bayesian learners

Boosting still perform worse than the best base kNN classifier. Figure 2(c) shows the ROC curves for the different rules for combining the Bayesian learners trained using 5 samples. Excluding the product, min, and max rules which make unrealistic assumptions about the base classifiers, all the other methods have comparable performance, with the weighted linear combination (GEM) and Boosting methods performing the best. Further, both methods outperformed Bagging. Figures 2(b), (c) suggest that on average Boosting outperforms Bagging, confirming similar trends observed by other researchers [1]. Figure 3(a) which shows kNN classifiers built using Bayesian learners perform the best, followed by kNN classifiers built using all features, followed by Boosting, and lastly Bagging. In fact Figure 3(b) which shows the ROC curves for the best ensembles from each family of methods (Bagging, Boosting, Bayes, and kNN) reveals that the kNN classifier built using Bayesian learners yields the best overall performance. This is an extremely significant result since it suggests that a simple non-parametric classifier requiring minimal training can outperform more sophisticated parametric methods that require extensive training. This is especially pertinent for CAD applications where large amounts of training data are usually unavailable. Table 2 shows A_z values (area under ROC curve) for the different ensembles.

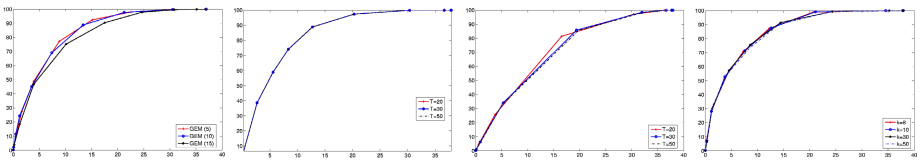
Table 2. A_z values for different ensembles from the 4 families of learning methods

Method	kNN		Bayes	Bagging		Boosting		
	Features	Bayes	(GEM)	kNN	Bayes	kNN	Features	Bayes
A_z	.943	.957	.937	.887	.925	.899	.939	.936



(a) (b)

Fig. 3. ROC curves for (a) ensembles of kNN classifiers, and (b) the best ensemble methods from each of the 4 families of classifiers: Bagging, Boosting, Bayes, and kNN



(a) (b) (c) (d)

Fig. 4. ROC curves for (a) GEM for 3 sets of training data (5, 10, 15 samples), (b) Boosting on the feature space ($T \in \{20,30,50\}$), (c) Bagging on kNN classifiers ($T \in \{20,30,50\}$, number of features=12), and (d) kNN on feature space ($k \in \{8,10,50,100\}$)

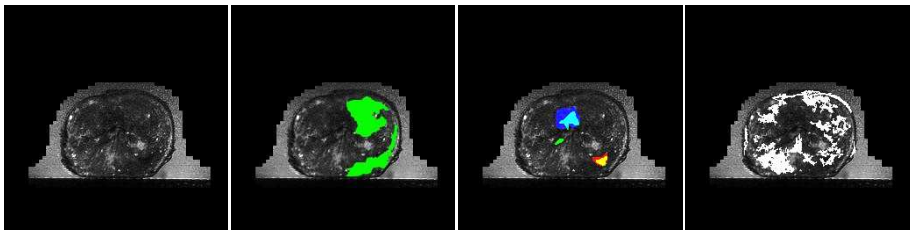
4.2 Parameter Sensitivity

The following parameter values were used for the different ensembles: (a) kNN - $k \in \{8,10,50,100\}$, (b) Bayes - 4 different training sets comprising 1, 5, 10, and 15 images from the set of 33 2D image slices for which ground truth was available, and (c) Boosting/Bagging - $T \in \{20,30,50\}$ trials. The results in Table 3 which list the standard deviation in A_z values for the 4 families of methods for different parameter settings and the plots in Figure 4 suggest that all ensembles considered are robust to changes in parameter settings, and to training. Table 3 and Figure 5 further reveal the high levels of disagreement among five human experts who independently segmented tumor on the MR slices without the aid of the corresponding histology.

Table 3. Columns 2-5 correspond to standard deviation in A_z values for the different ensembles for different parameter settings, while column 6 corresponds to the average standard deviation (%) in manual segmentation sensitivity for 5 human experts

Method	kNN	Bayes	Bagging	Boosting	Experts
Std. Deviation	1.3×10^{-3}	6.1×10^{-3}	2.7×10^{-3}	7.1×10^{-6}	20.55

Figures 5(a) corresponds to slice of a prostate MRI study and 5(b corresponds to ground truth for tumor in (a) slices obtained via histology. Figures 5(c) which represents the overlay of 5 human expert segmentations for tumor on 5(a) clearly demonstrate (i) high levels of disagreement among the experts (only the bright regions correspond to unanimous agreement), and (ii) the difficulty of the problem since all the expert segmentations had significant false negative errors. The bright areas in Figure 5(d) which represents the overlay of the kNN classification on the feature space for $k \in \{10,50,100\}$ (θ=0.5) reveals the precision of the ensemble for changes in parameter settings.



(a) (b) (c) (d)

Fig. 5. Slices from (a) a 4 T MRI study, (b) tumor ground truth on (a) determined from histology, (c) 5 expert segmentations of cancer superposed on (a), (d) result of kNN classifier for $k \in \{10,50,100\}$ (θ=0.5) superposed on (a). Note (i) lower parameter sensitivity of ensemble methods compared to inter-expert variability and (ii) higher accuracy in terms of the crucial false negative errors.

4.3 Execution Times

Table 4 shows average execution times for the ensembles for a 2D image slice from a 3D MRI volume of dimensions 256×256×50. Feature extraction times are not included. The parameter values used were: k=10, number of features K=35, training samples for Bayesian learners (A=5), and number of iterations for Bagging and Boosting T=30. All computations were performed on a 3.2 GHz Pentium IV Dell desktop computer (2 GB RAM). The kNN methods required no training, while Boosting Bayesian learners required the most amount of time to train. In terms of testing, the Boosting and kNN methods were the fastest while the Bayesian methods were the slowest. Note however that the time required to estimate the Bayesian posterior class conditional probabilities is a function of the dynamic intensity range of the different features employed, which in our case was 0-4095. Note also that columns 3, 6, and 9 do not include the time for computing the posterior class conditional probabilities.

Table 4. Execution times (training and classification) for the different ensemble methods. For brevity only one of the Bayesian methods (GEM) has been shown.

Method	kNN		Bayes	Bagging		Boosting		
	Features	Bayes	(GEM)	kNN	Bayes	kNN	Features	Bayes
Training	-	-	0.86	18.15	25.65	18.15	35.33	77.82
Classification	0.98	1.59	131.21	16.71	34.99	1.59	1.09	0.60

5 Concluding Remarks

In this paper we compared the performance of 14 ensembles from 4 families of machine learning methods: Bagging, Boosting, Bayesian learning, and kNN, for detecting prostate cancer from 4 T *ex vivo* MRI prostate studies. The kNN classifier performed the best, both in terms of accuracy and ease of training, thus

validating *Occam's Razor*. This is an especially satisfying result since an accurate non-parametric classifier requiring minimal training is ideally suited to CAD applications where large amounts of data are usually unavailable. All classifiers were found to be robust with respect to training and changes in parameter settings. By comparison the human experts had a low degree of inter-observer agreement. We also confirmed two trends previously reported in the literature, (i) Boosting consistently outperformed Bagging [1] and (ii) Boosting the Bayesian classifier did not improve performance [8]. Future work will focus on confirming our conclusions on larger data sets and with other CAD applications.

References

1. J. R. Quilan Bagging, Boosting, and C4.5 *AAAI/IAAI*, 1996, vol. 1, pp. 725-30.
2. L. Breiman, "Bagging Predictors", *Machine Learning*, vol. 24[2], pp. 123-40, 1996.
3. Y. Freund, R. Schapire, "Experiments with a new Boosting Algorithm", *National Conference on Machine Learning*, 1996, pp. 148-156.
4. A. Madabhushi, M. Feldman, D. Metaxas, J. Tomasezweski, D. Chute, "Automated Detection of Prostatic Adenocarcinoma from High Resolution Ex Vivo MRI", *IEEE Transactions on Medical Imaging*, vol. 24[12], pp. 1611-25, 2005.
5. R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, New York Wiley, 1973.
6. Simon K. Warfield, Kelly H. Zou, William M. Wells "Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation", *IEEE Trans. on Med. Imag.*, 2004, vol. 23[7], pp. 903-21.
7. T. Dietterich, "Ensemble Methods in Machine Learning", *Workshop on Multiple Classifier Systems*, pp. 1-15, 2000.
8. E. Bauer, R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, Boosting, and Variants", *Machine Learning*, vol. 36, pp. 105-42, 1999.
9. Q-L Tran, K-A Toh, D. Srinivasan, K-L Wong, S Q-C Low, "An empirical comparison of nine pattern classifiers", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 35[5], pp. 1079-91, 2005.
10. D. Martin, C. Fowlkes, J. Malik "Learning to detect natural image boundaries using local brightness, color, and texture cues", *IEEE Trans. on Pattern Anal. & Machine Intel.*, 2004, vol. 26[5], pp. 530-49.
11. L. Wei, Y. Yuang, R. M. Nishikawa, Y. Jiang, "A study on several machine-learning methods for classification of malignant and benign clustered micro-calcifications", *IEEE Trans. on Medical Imag.*, 2005, vol. 24[3], pp. 371-80.
12. S. Bay "Nearest neighbor classification from multiple feature subsets", *Intelligent Data Analysis*, vol. 3(3), pp. 191-209, 1999.