8

ENSEMBLE LEARNING

Chapter Objectives

- Explain the basic characteristics of ensemble learning methodologies.
- Distinguish between the different implementations of combination schemes for different learners.
- Compare bagging and boosting approaches.
- Introduce AdaBoost algorithm and its advantages.

One of the primary goals of data mining is to predict an "unknown" value of a new sample from observed samples. Such a prediction is achieved by two sequential phases as shown in Figure 8.1: (a) training phase—producing a predictive model from training samples using one of the available supervised learning algorithms; and (b) testing phase—evaluating the generated predictive model using test samples that are not used in the training phase. Numerous applications of a data-mining process showed validity of the so-called "No-Free-Lunch Theorem." It states that there is no single learning algorithm that is the best and most accurate in all applications. Each algorithm determines a certain model that comes with a set of assumptions. Sometimes these assumptions hold, sometimes not; therefore, no single algorithm "wins" all the time.

Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition. Mehmed Kantardzic.
© 2011 by Institute of Electrical and Electronics Engineers. Published 2011 by John Wiley & Sons, Inc.

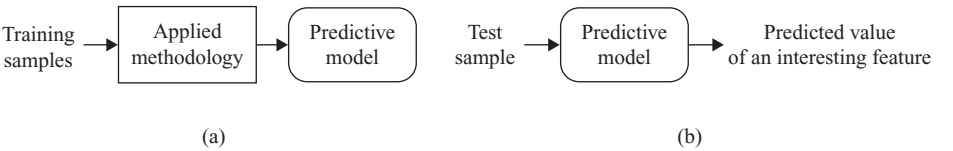

Figure 8.1. Training phase and testing phase for a predictive model. (a) Training phase; (b) testing phase.

In order to improve the accuracy of a predictive model, the promising approach called the *ensemble learning* is introduced. The idea is to combine results from various predictive models generated using training samples. The key motivation behind the proposed approach is to reduce the error rate. An initial assumption is that it will become much more unlikely that the ensemble will misclassify a new sample compared with a single predictive model. When combinining multiple, independent, and diverse "decisions makers," each of which is at least more accurate than random guessing, correct decisions should be reinforced. The idea may be demonstrated by some simple decision processes where single-human performances are compared with human ensembles. For example, given the question "How many jelly beans are in the jar?", the group average will outperform individual estimates. Or, in the TV series "Who Wants to be a Millionaire?" where audience (ensemble) vote is a support for the candidate who is not sure of the answer.

This idea is proven theoretically by Hansen and company through the statement: If N classifiers make *independent* errors and they have the error probability $e < 0.5$, then it can be shown that the error of an ensemble E is monotonically decreasing the function of N. Clearly, performances quickly decrease for dependent classifiers.

8.1 ENSEMBLE-LEARNING METHODOLOGIES

The ensemble-learning methodology consists of two sequential phases: (a) the training phase, and (b) the testing phase. In the training phase, the ensemble method generates several different predictive models from training samples as presented in Figure 8.2a. For predicting an unknown value of a test sample, the ensemble method aggregates outputs of each predictive model (Fig. 8.2b). An integrated predictive model generated by an ensemble approach consists of several predictive models (Predictive model.1, Predictive model.2, . . . , Predictive model. n) and a combining rule as shown in Figure 8.2b. We will refer to such a predictive model as an *ensemble*. The field of ensemble learning is still relatively new, and several names are used as synonyms depending on which predictive task is performed, including combination of multiple classifiers, classifier fusion, mixture of experts, or consensus aggregation.

To perform better than a single predictive model, an ensemble should consist of predictive models that are independent of each other, that is, their errors are uncorrelated, and each of them has an accuracy rate of >0.5. The outcome of each predictive model is aggregated to determine the output value of a test sample. We may analyze

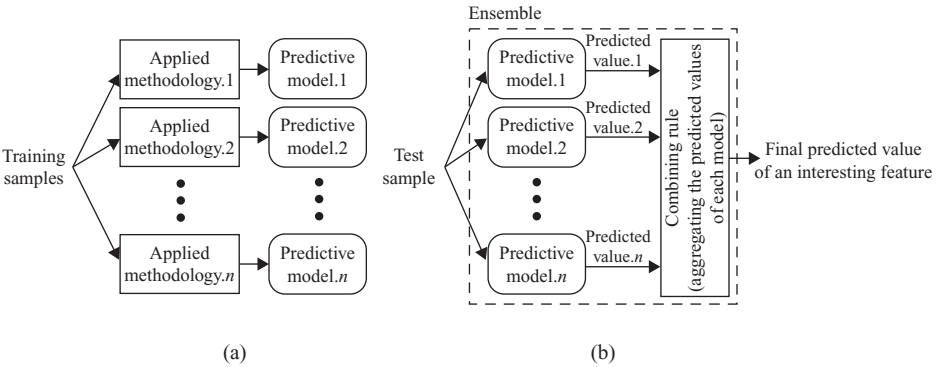

Figure 8.2. Training phase and testing phase for building an ensemble. (a) Training phase; (b) testing phase.

all steps of ensemble prediction for a classification task. For example, we may analyze a classification task where the ensemble consists of 15 classifiers, each of which classifies test samples into one of two categorical values. The ensemble decides the categorical value based on the dominant frequency of classifiers' outputs. If 15 predictive models are different from each other, and each model has the identical error rate ($\varepsilon = 0.3$), the ensemble will make a wrong prediction only if more than half of the predictive models misclassify a test sample. Therefore, the error rate of the ensemble is

$$\varepsilon_{ensemble} = \sum_{i=8}^{15} \binom{15}{i} \varepsilon^i (1-\varepsilon)^{15-i} = 0.05$$

which is considerably lower than the 0.3 error rate of a single classifier. The sum is starting with eight, and it means that eight or more models misclassified a test sample, while seven or fewer models classified the sample correctly.

Figure 8.3a shows the error rates of an ensemble, which consists of 15 predictive models (n = 15). The x-axis represents an error rate (ε) of a single classifier. The diagonal line represents the case in which all models in the ensemble are identical. The solid line represents error rates of an ensemble in which predictive models are different and independent from each other. An ensemble has a significantly lower error rate than a single predictive model only when the error rate (ε) of the members of the ensemble is lower than 0.5.

We can also analyze the effect of the number of predictive models in an ensemble. Figure 8.3b shows error-rate curves for ensembles that consist of 5, 15, 25, and 35 predictive models, respectively. Observe that when an error rate of a predictive model is lower than 0.5, the larger the number of predictive models is, the lower the error rate of an ensemble is. For example, when each predictive model of an ensemble has an error rate of 0.4, error rates of each ensemble ($n = 5$, $n = 15$, $n = 25$, and $n = 35$) are calculated as 0.317, 0.213, 0.153, and 0.114, respectively. However, this decrease in

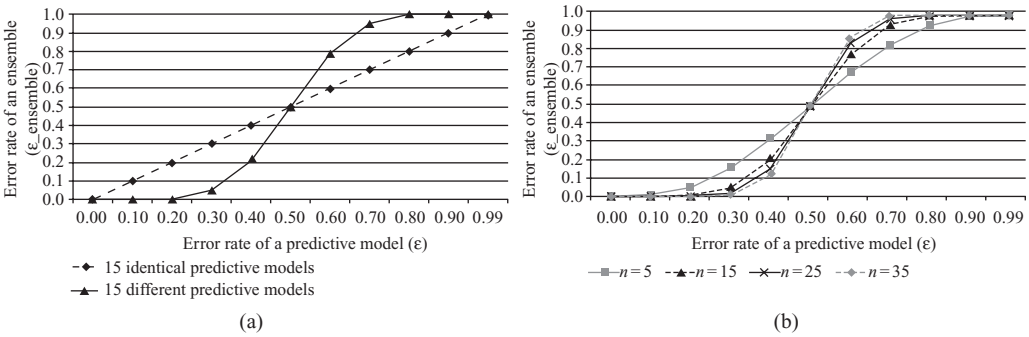

Figure 8.3. Changes in error rates of an ensemble. (a) Identical predictive models versus different predictive models in an ensemble; (b) the different number of predictive models in an ensemble.

the error rate for an ensemble is becoming less significant if the number of classifiers is very large, or when the error rate of each classifier becomes relatively small.

The basic questions in creating an ensemble learner are as follows: How to generate base learners, and how to combine the outputs from base learners? Diverse and independent learners can be generated by

(a) using different learning algorithms for different learning models such as support vector machines, decision trees, and neural networks;
(b) using different hyper-parameters in the same algorithm to tune different models (e.g., different numbers of hidden nodes in artificial neural networks);
(c) using different input representations, such as using different subsets of input features in a data set; or
(d) using different training subsets of input data to generate different models usually using the same learning methodology.

Stacked Generalization (or stacking) is a methodology that could be classified in the first group (a). Unlike other well-known techniques, stacking may be (and normally is) used to combine models of different types. One way of combining multiple models is specified by introducing the concept of a meta-learner. The learning procedure is as follows:

1. Split the training set into two disjoint sets.
2. Train several base learners on the first part.
3. Test the base learners on the second part.
4. Using the predictions from (3) as the inputs, and the correct responses as the outputs, train a higher level learner.

Note that steps (1) to (3) are the same as cross-validation, but instead of using a winner-take-all approach, the base learners are combined, possibly nonlinearly.

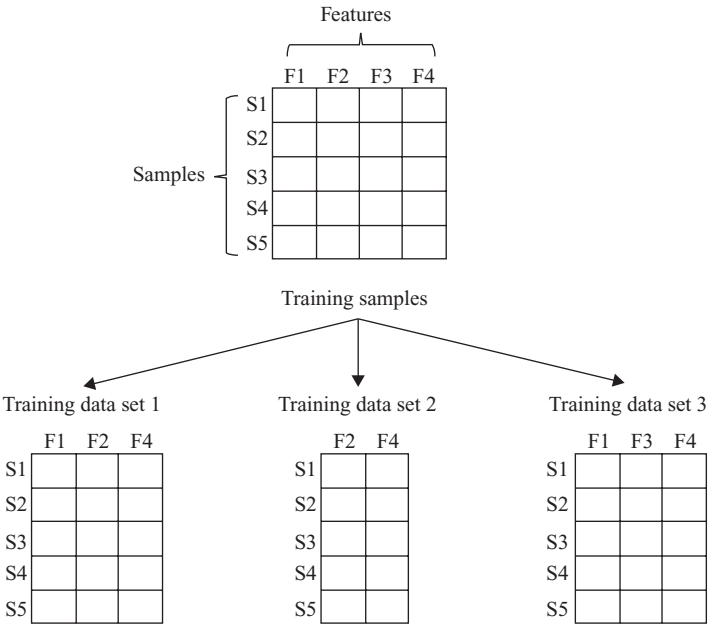

Figure 8.4. Feature selection for ensemble classifiers methodology.

Although an attractive idea, it is less theoretically analyzed and less widely used than bagging and boosting, the two most recognized ensemble-learning methodologies. Similar situation is with the second group of methodologies (b): Although a very simple approach, it is not used or analyzed intensively. Maybe the main reason is that applying the same methodology with different parameters does not guarantee independence of models.

Class (c) methodologies are based on manual or automatic feature selection/extraction that can be used for generating diverse classifiers using different feature sets. For example, subsets related to different sensors, or subsets of features computed with different algorithms, may be used. To form training data sets, different subsets of input features are chosen, and then each training sample with the selected input features becomes an element of training data sets. In Figure 8.4, there are five training samples {S1, S2, S3, S4, S5} with four features {F1, F2, F3, F4}. When the training data set 1 is generated, three features {F1, F2, F4} is randomly selected from input features {F1, F2, F3, F4}, and all training samples with those features form the first training set. Similar process is performed for the other training sets. The main requirement is that classifiers use different subsets of features that are complementary.

The *random subspace method* (RSM) is a relatively recent method of ensemble learning that is based on the theory of stochastic discrimination. Learning machines are trained on randomly chosen subspaces of the original input space and the outputs of the models are then combined. Illustrative example for the classification of movies is given in Figure 8.5. RSM works well for large feature sets with redundant features.

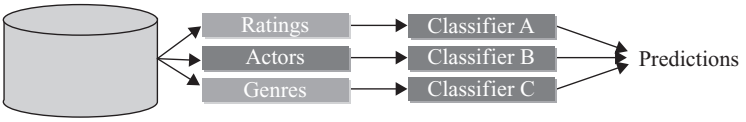

Figure 8.5. RSM approach in ensemble classifier for movie classification.

Random forest methodology, which utilizes such an approach, is implemented in many commercial data-mining tools.

Methodologies based on different training subsets of input samples (d) are the most popular approaches in ensemble learning, and corresponding techniques such as bagging and boosting are widely applied in different tools. But, before the detailed explanations of these techniques, it is necessary to explain one additional and final step in ensemble learning, and that is combining of outcomes for different learners.

8.2 COMBINATION SCHEMES FOR MULTIPLE LEARNERS

Combination schemes include:

1. *Global approach* is through learners' fusion where all learners produce an output and these outputs are combined by voting, averaging, or stacking. This represents integration (fusion) functions where for each pattern, all the classifiers contribute to the final decision.
2. *Local approach* is based on learner selection where one or more learners responsible for generating the output are selected based on their closeness to the sample. Selection function is applied where for each pattern, just one classifier, or a subset, is responsible for the final decision.
3. *Multistage combination* uses a serial approach where the next learner is trained with or tested on only instances where previous learners were inaccurate.

Voting is the simplest way of combining classifiers on a global level, and representing the result as a linear combination of outputs d_j for n learners:

$$y_i = \sum_{j=1}^{n} w_j d_j \qquad \text{where } w_j \geq 0 \text{ and } \sum_{j=1}^{n} w_j = 1$$

The result of the combination could be different depending on w_j. Alternatives for combinations are *simple sum (*equal weights*), weighted sum, median, minimum, maximum,* and *product of* d_{ij}. Voting schemes can be seen as approximations under a Bayesian framework where weights w_j approximate prior model probabilities.

Rank-level Fusion Method is applied for some classifiers that provide class "scores," or some sort of class probabilities. In general, if $\Omega = \{c1, \ldots, ck\}$ is the set of classes, each of these classifiers can provide an "ordered" (ranked) list of class labels. For

example, if probabilities of output classes are 0.10, 0.75, and 0.20, corresponding ranks for the classes will be 1, 3, and 2, respectively. The highest rank is given to the class with the highest probability. Let us check an example, where the number of classifiers is N = 3, and the number of classes k = 4, $\Omega = \{a, b, c, d\}$. For a given sample, the ranked outputs of the three classifiers are as follows:

Rank value	Classifier 1	Classifier 2	Classifier 3
4	*c*	*a*	*b*
3	*b*	*b*	*a*
2	*d*	*d*	*c*
1	*a*	*c*	*d*

In this case, final selection of the output class will be determined by accumulation of scores for each class:

$$r_a = r_a^{(1)} + r_a^{(2)} + r_a^{(3)} = 1 + 4 + 3 = 8$$
$$r_b = r_b^{(1)} + r_b^{(2)} + r_b^{(3)} = 3 + 3 + 4 = 10$$
$$r_c = r_c^{(1)} + r_c^{(2)} + r_c^{(3)} = 4 + 1 + 2 = 7$$
$$r_d = r_d^{(1)} + r_d^{(2)} + r_d^{(3)} = 2 + 3 + 1 = 5$$

The winner class is *b* because it has the maximum overall rank.

Finally, the *Dynamic Classifier Selection* (DCS) algorithm, representing a local approach, assumes the following steps:

1. Find the k nearest training samples to the test input.
2. Look at the accuracies of the base classifiers on these samples.
3. Choose one (or top N) classifiers that performs best on these samples.
4. Combine decisions for selected classifiers.

8.3 BAGGING AND BOOSTING

Bagging and boosting are well-known procedures with solid theoretical background. They belong to the class (d) of ensemble methodologies and essentially they are based on resampling of a training data set.

Bagging, a name derived from bootstrap aggregation, was the first effective method of ensemble learning and is one of the simplest methods. It was originally designed for classification and is usually applied to decision tree models, but it can be used with any

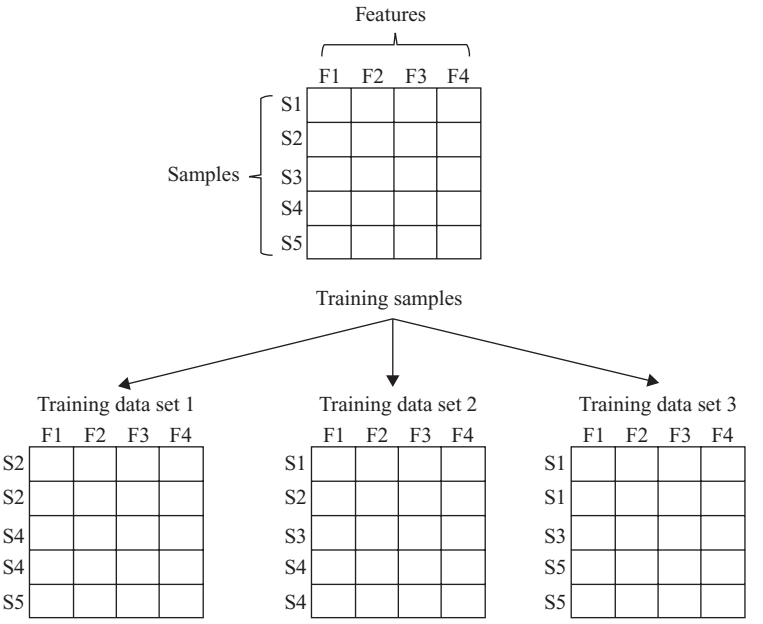

Figure 8.6. Bagging methodology distributes samples taken with replacement from initial set of samples.

type of model for classification or regression. The method uses multiple versions of a training set by using the bootstrap, that is, sampling with replacement. Each of these data sets is used to train a different model. The outputs of the models are combined by averaging (in the case of regression) or voting (in the case of classification) to create a single output.

In the bagging methodology a training data set for a predictive model consists of samples taken with replacement from an initial set of samples according to a sampling distribution. The sampling distribution determines how likely it is that a sample will be selected. For example, when the sampling distribution is predefined as the uniform distribution, all N training samples have the same probability, $1/N$, of being selected. In the same training data set, because of replacement sampling, some training samples may appear multiple times, while any training samples may not appear even once. In Figure 8.6, there are five training samples {S1, S2, S3, S4, S5} with four features {F1, F2, F3, F4}. Suppose that three training data sets are formed by samples that are randomly selected with replacement from the training samples according to the uniform distribution. Each training sample has a 1/5 probability of being selected as an element of a training data set. In the training data set 1, S2 and S4 appear twice, while S1 and S3 do not appear.

Bagging is only effective when using unstable nonlinear models where small changes in training data lead to significantly different classifiers and large changes in accuracy. It decreases error by decreasing the variance in the results of unstable learners.

Boosting is the most widely used ensemble method and one of the most powerful learning ideas introduced in the ensemble-learning community. Originally designed for classification, it can also be extended to regression. The algorithm first creates a "weak" classifier, that is, it suffices that its accuracy on the training set is slightly better than random guessing. Samples are given initial weights, and usually it starts with uniform weighting. For the following iterations, the samples are reweighted to focus the system on samples that are not correctly classified with a recently learned classifier. During each step of learning: (1) increase weights of the samples that are not correctly learned by the weak learner, and (2) decrease weights of the samples that are correctly learned by the weak learner. Final classification is based on a weighted vote of weak classifiers generated in iterations.

8.4 ADABOOST

The original boosting algorithm combined three weak learners to generate a strong, high quality learner. *AdaBoost*, short for "adaptive boosting," is the most popular boosting algorithm. AdaBoost combine "weak" learners into a highly accurate classifier to solve difficult highly nonlinear problems. Instead of sampling, as in a bagging approach, AdaBoost reweighs samples. It uses the same training set over and over again (thus it need not be large) and it may keep adding weak learners until a target training error is reached.

Given a training data set: $\{(x1, y1), \ldots, (xm, ym)\}$ where $xi \in X$ and $yi \in \{-1, +1\}$, when a weak classifier is trained with the data, for each input sample x_i the classifier will give classification $h(x_i)$ (where $h(x_i) \in \{-1, +1\}$). With these assumptions the main steps of the AdaBoost algorithm are presented in Figure 8.8.

Simplicity and easy implementation are the main reasons why AdaBoost is very popular. It can be combined with any classifiers including neural networks, decision trees, or nearest neighbor classifiers. The algorithm requires almost no parameters to tune, and is still very effective even for the most complex classification problems, but at the same time it could be sensitive to noise and outliers.

Ensemble-learning approach showed all advantages in one very famous application, Netflix $1 million competition. The Netflix prize required substantial improvement in the accuracy of predictions on how much someone is going to love a movie based on his or her previous movie preferences. Users' rating for movies was 1 to 5 stars; therefore, the problem was classification task with five classes. Most of the top-ranked competitors have used some variations of ensemble learning, showing its advantages in practice. Top competitor *BellKor* team explains ideas behind its success: "Our final solution consists of blending 107 individual predictors. Predictive accuracy is substantially improved when blending multiple predictors. Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a single technique. Consequently, our solution is an ensemble of many methods."

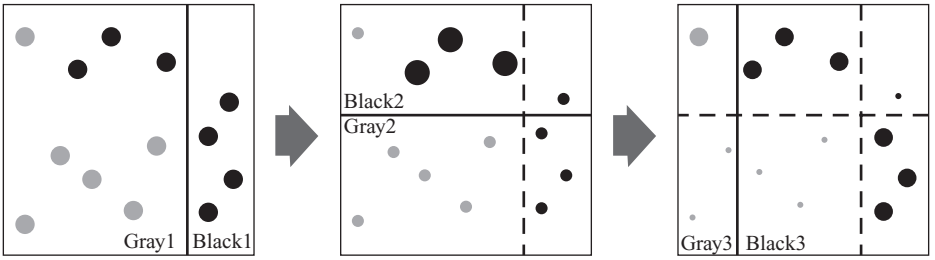

Figure 8.7. AdaBoost iterations.

- Initialize distribution over the training set $D_1(i) = 1/m$
- For $t = 1, \ldots, T$:

1. Train *Weak Learner* using distribution D_t.
2. Choose a weight (or confidence value) $\alpha_t \epsilon \mathbf{R}$.
3. Update the distribution over the training set:

$$D_{t+1}(i) = \frac{D_t(i)\, e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \qquad (2)$$

Where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution

- Final vote $H(x)$ is a weighted sum:

$$H(x) = \text{sign}\,(f(x)) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Figure 8.8. AdaBoost algorithm.

...	No Progress Prize candidates yet	...	...
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 – RMSE = 0.8712 – Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07

Figure 8.9. Top competitors in 2007/2008 for Netflix award.

8.5 REVIEW QUESTIONS AND PROBLEMS

1. Explain the basic idea of ensemble learning, and discuss why the ensemble mechanism is able to improve prediction accuracy of a model.
2. Designing an ensemble model, there are factors that directly affect the accuracy of the ensemble. Explain those factors and approaches to each of them.
3. Bagging and boosting are very famous ensemble approaches. Both of them generate a single predictive model from each different training set. Discuss the differences between bagging and boosting, and explain the advantages and disadvantages of each of them.
4. Propose the efficient boosting approach for a large data set.
5. In the bagging methodology, a subset is formed by samples that are randomly selected with replacement from training samples. On average, a subset contains approximately what percentage of training samples?
6. In Figure 8.7, draw a picture of the next distribution D_4.
7. In equation (2) of the AdaBoost algorithm (Fig. 8.8), $e^{\alpha_t y_i h_t(x_i)}$ replaces the term of $e^{-\alpha_t y_i h_t(x_i)}$. Explain how and why this change influences the AdaBoost algorithm.
8. Consider the following data set, where there are 10 samples with one dimension and two classes:

   Training samples:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
f_1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Class	1	1	1	−1	1	−1	1	−1	−1	−1

   (a) Determine ALL the best one-level binary decision trees.
   (e.g., IF $f_1 \leq 0.35$, THEN Class is 1, and IF $f_1 > 0.35$, THEN Class is −1. The accuracy of that tree is 80%)

   (b) We have the following five training data sets randomly selected from the above training samples. Apply the bagging procedure using those training data sets.
   (i) Construct the best one-level binary decision tree from each training data set.
   (ii) Predict the training samples using each constructed one-level binary decision tree.
   (iii) Combine outputs predicted by each decision tree using voting method
   (iv) What is the accuracy rate provided by bagging?

   Training Data Set 1:

	x_1	x_2	x_3	x_4	x_5	x_8	x_9	x_{10}	x_{10}	x_{10}
f_1	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1.0	1.0	1.0
Class	1	1	1	−1	1	−1	−1	−1	−1	−1

Training Data Set 2:

	x_1	x_1	x_2	x_4	x_4	x_5	x_5	x_7	x_8	x_9
f_1	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
Class	1	1	1	−1	−1	1	1	1	−1	−1

Training Data Set 3:

	x_2	x_4	x_5	x_6	x_7	x_7	x_7	x_8	x_9	x_{10}
f_1	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1.0
Class	1	−1	1	−1	1	1	1	−1	−1	−1

Training Data Set 4:

	x_1	x_2	x_5	x_5	x_5	x_7	x_7	x_8	x_9	x_{10}
f_1	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1.0
Class	1	1	1	1	1	1	1	−1	−1	−1

Training Data Set 5:

	x_1	x_1	x_1	x_1	x_3	x_3	x_8	x_8	x_9	x_9
f_1	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
Class	1	1	1	−1	1	−1	1	−1	−1	−1

(c) Applying the AdaBoost algorithm (Fig. 8.8) to the above training samples, we generate the following initial one-level binary decision tree from those samples:

$$\text{IF } f_1 \leq 0.35, \text{ THEN Class is } 1$$

$$\text{IF } f_1 > 0.35, \text{ THEN Class is } -1$$

To generate the next decision tree, what is the probability (D_2 in Fig. 8.8) that each sample is selected to the training data set? (α_t is defined as an accuracy rate of the initial decision tree on training samples.)

9. For classifying a new sample into four classes: C1, C2, C3, and C4, we have an ensemble that consists of three different classifiers: *Classifier 1*, *Classifiers 2*, and *Classifier 3*. Each of them has 0.9, 0.6, and 0.6 accuracy rate on training samples, respectively. When the new sample, X, is given, the outputs of the three classifiers are as follows:

Class Label	*Classifier 1*	*Classifier 2*	*Classifier 3*
C1	0.9	0.3	0.0
C2	0.0	0.4	0.9
C3	0.1	0.2	0.0
C4	0.0	0.1	0.1

Each number in the above table describes the probability that a classifier predicts the class of a new sample as a corresponding class. For example, the probability that *Classifier 1 predicts* the class of X as C1 is 0.9.

When the ensemble combines predictions of each of them, as a combination method:

(a) If the *simple sum* is used, which class is X classified as and why?
(b) If the *weight sum* is used, which class is X classified as and why?
(c) If the *rank-level fusion* is used, which class is X classified as and why?

10. Suppose you have a drug discovery data set, which has 1950 samples and 100,000 features. You must classify chemical compounds represented by structural molecular features as active or inactive using ensemble learning. In order to generate diverse and independent classifiers for an ensemble, which ensemble methodology would you choose? Explain the reason for selecting that methodology.

11. Which of the following is a fundamental difference between bagging and boosting?

(a) Bagging is used for supervised learning. Boosting is used with unsupervised clustering.
(b) Bagging gives varying weights to training instances. Boosting gives equal weight to all training instances.
(c) Bagging does not take the performance of previously built models into account when building a new model. With boosting each new model is built based upon the results of previous models.
(d) With boosting, each model has an equal weight in the classification of new instances. With bagging, individual models are given varying weights.

8.6 REFERENCES FOR FURTHER STUDY

Brown, G., Ensemble Learning, in *Encyclopedia of Machine Learning*, C. Sammut and Webb G. I., eds., Springer Press, New York, 2010.

"Ensemble learning refers to the procedures employed to train multiple learning machines and combine their outputs, treating them as a committee of decision makers." The principle is that the committee decision, with individual predictions combined appropriately, should have better overall accuracy, on average, than any individual committee member. Numerous empirical and theoretical studies have demonstrated that ensemble models very often attain higher accuracy than single models. Ensemble methods constitute some of the most robust and accurate learning algorithms of the past decade. A multitude of heuristics has been

developed for randomizing the ensemble parameters to generate diverse models. It is arguable that this line of investigation is rather oversubscribed nowadays, and the more interesting research is now in methods for nonstandard data.

Kuncheva, L. I., *Combining Pattern Classifiers: Methods and Algorithms*, Wiley Press, Hoboken, NJ, 2004.

Covering pattern classification methods, *Combining Classifiers: Methods and Algorithms* focuses on the important and widely studied issue of combining several classifiers together in order to achieve an improved recognition performance. It is one of the first books to provide unified, coherent, and expansive coverage of the topic and as such will be welcomed by those involved in the area. With case studies that bring the text alive and demonstrate "real-world" applications it is destined to become an essential reading.

Dietterich, T. G., Ensemble Methods in Machine Learning, in *Lecture Notes in Computer Science on Multiple Classifier Systems*, Vol. 1857, Springer, Berlin, 2000.

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions. The original ensemble method is Bayesian averaging, but more recent algorithms include error-correcting output coding, bagging, and boosting. This paper reviews these methods and explains why ensembles can often perform better than any single classifier. Some previous studies comparing ensemble methods are reviewed, and some new experiments are presented to uncover the reasons that Adaboost does not overfit rapidly.