# Ensembles of Classifiers

## Lecture 3

# Reasons for using Ensembles

- **Statistical reasons**:
  - Combining the output of several classifiers may reduce the risk of an unfortunate selection of a poorly performing classifier

# Reasons for using Ensembles

- **Large Volumes of Data**:
  - Sometimes, the amount of data to be analyzed can be too large to be handled by a single classifier. Thus, we can:
    - Partition the data into smaller subsets;
    - Train different classifiers;
    - Combine their outputs using a combination rule

# Reasons for using Ensembles

- **Too Little Data**:
  - A reasonable sized set of training data is crucial to learn the underlying data distribution. When available data is scarce, we can:
    - Draw overlapping random subsets of the available data using resampling techniques
    - Train different classifiers, creating the ensemble

# Reasons for using Ensembles

- **Divide and Conquer**:
  - The given task may be too complex, or lie outside the space of functions that can be implemented by the chosen classifier method (e.g.: non-linear problem, and linear classifiers)
  - Appropriate combinations of simple (e.g., linear) classifiers can learn complex (e.g., non-linear) boundaries

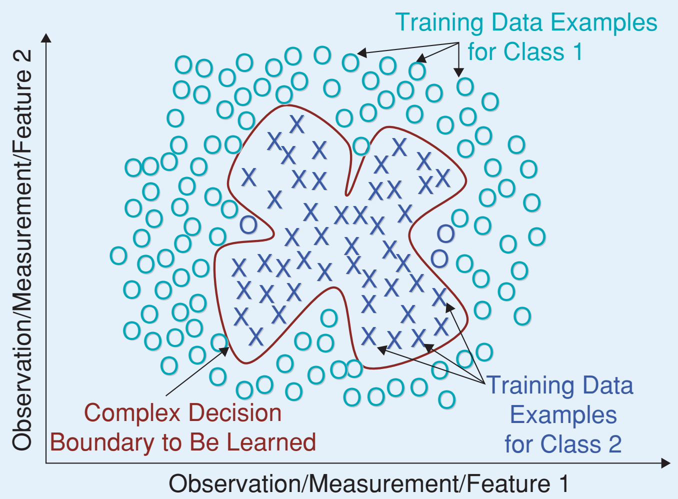


**Figure 1.** Complex decision boundary that cannot be learned by linear or circular classifiers.

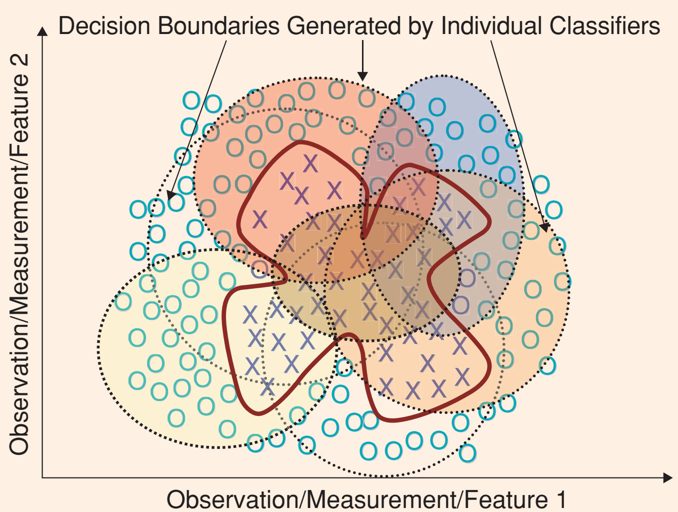


**Figure 2.** Ensemble of classifiers spanning the decision space.

# Reasons for using Ensembles

- **Data Fusion**:
  - Several sets of data obtained from different sources, where the nature of features is different (e.g.: categorical and numerical features)
  - Data from each source can be used to train a different classifier, thus creating an ensemble

# Components of an Ensemble

* Two key components:
  * A method to generate the individual classifiers of the ensemble
  * A method for combining the outputs of these classifiers

# Diversity: The Key Feature

- The individual classifiers must be diverse, i.e., they make errors on different data
- Intuition: if they make the same errors, such mistakes will be carried into the final prediction
- Thus: the errors the classifiers make should be uncorrelated

# Accuracy

- The component classifiers need to be “reasonably accurate” to avoid poor classifiers to obtain the majority of votes.
- Intuition: If the components of the ensemble are poor classifiers, they make a lot of errors, and those errors are carried out to the final prediction.

# Accuracy and Diversity

- Requirements for accuracy and diversity have been quantified:
  - Under simple majority voting and *independent error conditions*, if all classifiers have the same probability of error of *less than 50%*, then the error of the ensemble decreases monotonically with an increasing number of classifiers.

# How to achieve diversity

- Use different training data sets to train individual classifiers
- Such data sets are often obtained through resampling techniques (***bootstrapping*** or ***bagging***): training data subsets are drawn randomly, usually with replacement, from the entire training data

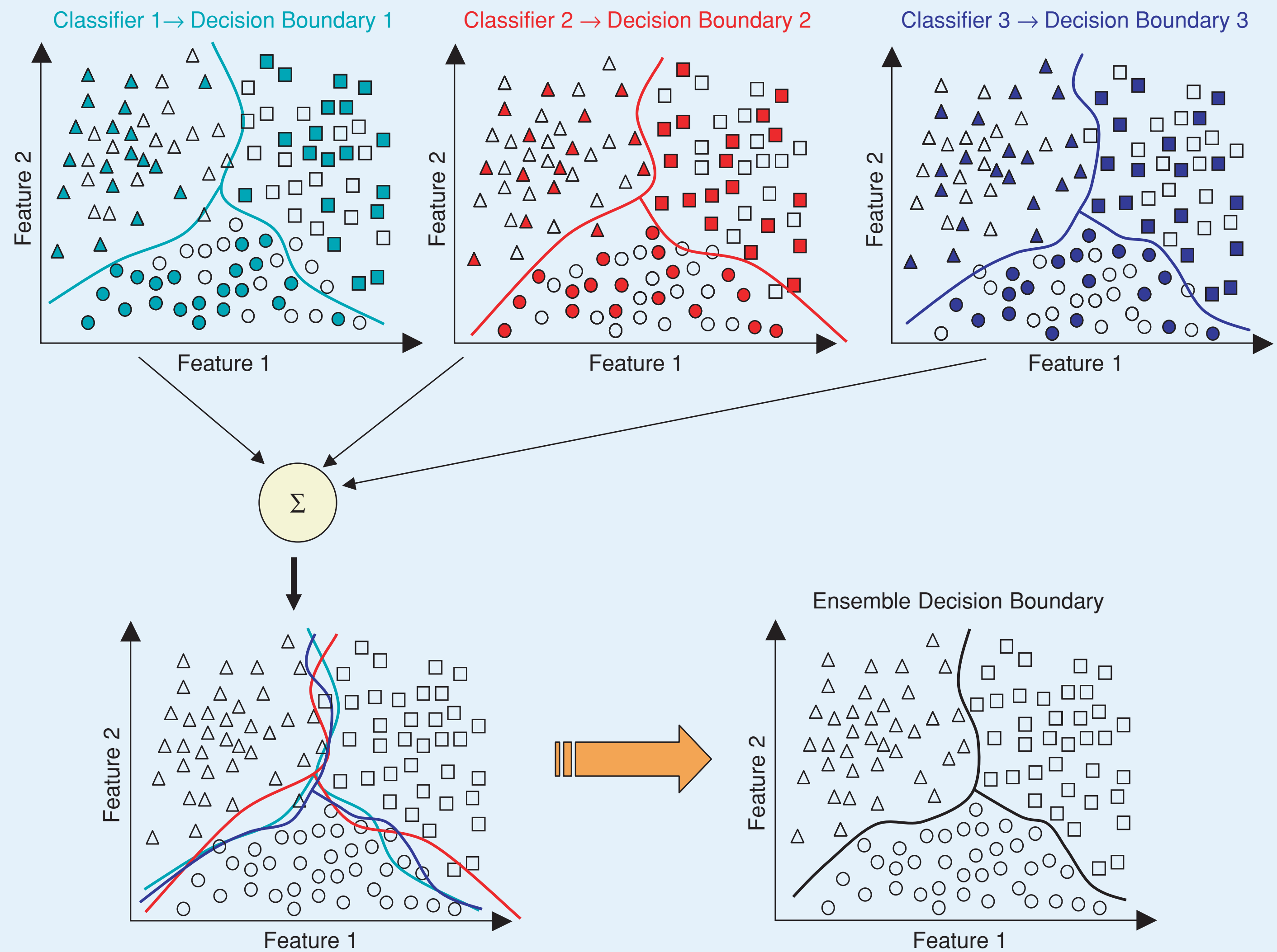


**Figure 3.** Combining classifiers that are trained on different subsets of the training data.

# How to achieve diversity

- Use different training data sets to train individual classifiers
- If the training data subsets are drawn without replacement, the procedure is also called ***jackknife*** or ***k-fold*** data split: the entire data set is split into k blocks, and each classifier is trained only on k-1 of them. A different subset of k blocks is selected for each classifier

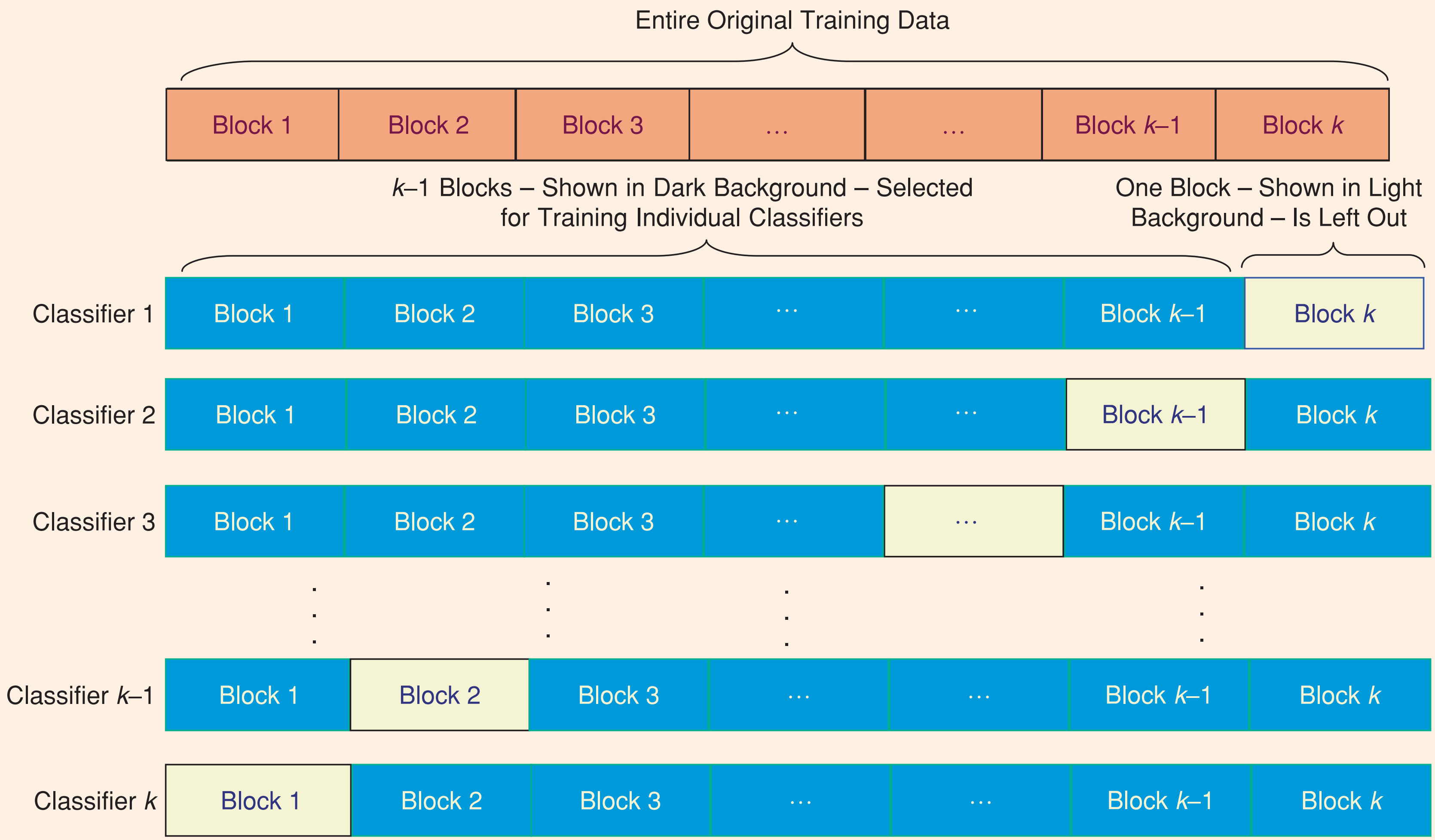


**Figure 4.** *k*-fold data splitting for generating different, but overlapping, training datasets.

# How to achieve diversity

- When is bagging (bootstrapping) effective?
- To ensure diverse classifiers, the base classifier should be ***unstable***, that is, *small changes* in the training set should lead to *large changes* in the classifier output.

# How to achieve diversity

- When is bagging (bootstrapping) effective?
- Large error reductions have been observed with *decision trees* and bagging. This is because decision trees are highly sensitive to small perturbations of the training data.

# How to achieve diversity

- When is bagging (bootstrapping) effective?
- Bagging is not effective with *nearest neighbor classifiers*. Why? NN classifiers are highly stable with respect to variations of the training data
- It has been shown that the probability that any given training point is included in a data set bootstrapped by bagging is approximately 63.2%. It follows that the nearest neighbor will be the same in 63.2% of the classifiers
- Thus, the errors are highly correlated, and bagging becomes ineffective

# How to achieve diversity

- Use different training parameters for different classifiers
- E.g., ensemble of neural networks trained with different weight initialization, or different number of layers/nodes
- If the base classifier is unstable with respect to the tuning parameters, diverse classifiers can be generated

# How to achieve diversity

- Use different type of classifiers
- E.g., an ensemble of neural networks, decision trees, nearest neighbor classifiers, and support vector machines

# How to achieve diversity

- Use different subsets of features to train the individual classifiers
- E.g., random feature subsets (random subspace method)
- This approach is effective with nearest neighbor (NN) methods, because NN techniques are highly sensitive to the chosen features

# Bagging

**bootstrap aggregating**

# Bagging

- Intuitive and simple
- Achieves good performance
- Diversity is obtained by **bootstrapping replicas of the training data**:
  - **different subsets of data are randomly drawn with replacement from the entire training data**
- Each resulting training data is used to train a different classifier of the same type.

# Bagging

- Given a test point, individual classifiers are combined by taking **a majority vote** of their decisions.
- That is: for any given instance, the class chosen by most classifiers is the **ensemble decision**.

**Algorithm: Bagging**

**Input:**

- Training data $S$ with correct labels $\omega_i \in \Omega = \{\omega_1, \ldots, \omega_C\}$ representing $C$ classes
- Weak learning algorithm **WeakLearn,**
- Integer $T$ specifying number of iterations.
- Percent (or fraction) $F$ to create bootstrapped training data

**Do** $t = 1, \ldots, T$

1. Take a bootstrapped replica $S_t$ by randomly drawing $F$ percent of $S$.
2. Call **WeakLearn** with $S_t$ and receive the hypothesis (classifier) $h_t$.
3. Add $h_t$ to the ensemble, $\mathtt{E}$.

**End**

**Test: Simple Majority Voting –** Given unlabeled instance **x**

1. Evaluate the ensemble $\mathtt{E} = \{h_1, \ldots, h_T\}$ on **x**.
2. Let $$v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$
   be the vote given to class $\omega_j$ by classifier $h_t$.
3. Obtain total vote received by each class
   $$V_j = \sum_{t=1}^{T} v_{t,j}, \quad j = 1, \ldots, C \quad (9)$$
4. Choose the class that receives the highest total vote as the final classification.

# Bagging

- Particularly appealing when data available is of limited size
- To ensure that there are sufficient training samples in each subset, relatively large portions of the samples (75% to 100%) are drawn into each subset

# Bagging

- To ensure diversity under this scenario, an **unstable** learning method is used so that different decision boundaries can be obtained with small perturbations in different training data sets

- Neural networks and decision trees are unstable, and are good candidates for bagging

- K nearest methods are stable. They are **not** good candidates for bagging

# Experiments
from
# *Bagging Predictors*
by Leo Breiman

Machine Learning, 24:123-140, 1996

# Bagging Classification Trees

**DATA SETS**

| Data Set | # Samples | # Variables | # Classes |
|---|---|---|---|
| waveform | 300 | 21 | 3 |
| heart | 1395 | 16 | 2 |
| breast cancer | 699 | 9 | 2 |
| ionosphere | 351 | 34 | 2 |
| diabetes | 768 | 8 | 2 |
| glass | 214 | 9 | 6 |
| soybean | 683 | 35 | 19 |

# Bagging Classification Trees

**MISCLASSIFICATION RATES (%)**

| **Data Set** | $\bar{e}_S$ | $\bar{e}_B$ | **Decrease** |
|---|---|---|---|
| waveform | 29.1 | 19.3 | 34% |
| heart | 4.9 | 2.8 | 43% |
| breast cancer | 5.9 | 3.7 | 37% |
| ionosphere | 11.2 | 7.9 | 29% |
| diabetes | 25.3 | 23.9 | 6% |
| glass | 30.4 | 23.6 | 22% |
| soybean | 8.6 | 6.8 | 21% |

# Bagging Classification Trees

## LARGER DATA SETS

| Data Set | #Training | #Variables | #Classes | #Test Set |
|---|---|---|---|---|
| letters | 15,000 | 16 | 26 | 5000 |
| satellite | 4,435 | 36 | 6 | 2000 |
| shuttle | 43,500 | 9 | 7 | 14,500 |
| DNA | 2,000 | 60 | 3 | 1186 |

# Bagging Classification Trees

## TEST SET MISCLASSIFICATION RATES (%)

| **Data Set** | $e_S$ | $e_B$ | **Decrease** |
|---|---|---|---|
| letters | 12.6 | 6.4 | 49% |
| satellite | 14.8 | 10.3 | 30% |
| shuttle | .062 | .014 | 77% |
| DNA | 6.2 | 5.0 | 19% |

# Bagging Class Probability Estimates

- Some classification methods estimate probabilities:

$$\hat{p}(j|\mathbf{x})$$

- Decision rule: $\arg\max_j \hat{p}(j|\mathbf{x})$
- A natural competitor to bagging by voting is to average the $\hat{p}(j|\mathbf{x})$ over all the bootstrap replications: $\hat{p}_B(j|\mathbf{x})$
- Final decision: $\arg\max_j \hat{p}_B(j|\mathbf{x})$

# How Many Bootstrap Replicates are Enough?

## BAGGED MISCLASSIFICATION RATES

| No. Bootstrap Replicates | Misclassification Rate |
|---|---|
| 10 | 21.8 |
| 25 | 19.4 |
| 50 | 19.3 |
| 100 | 19.3 |

# How Big Should the Bootstrap Learning Set Be?

- In the previous runs, the size of the bootstrap replicates was the same as the initial learning set
- While a bootstrap replicate may have 2,3,... duplicates of a given instance, it also leaves out about .37 of the instances.
- One can increase the size of the bootstrap replicates
- Diversity may decrease

# Bagging Nearest Neighbor Classifiers

## MISCLASSIFICATION RATES FOR NEAREST NEIGHBOR

| Data Set | $\bar{e}_S$ | $\bar{e}_B$ |
|---|---|---|
| waveform | 26.1 | 26.1 |
| heart | 5.1 | 5.1 |
| breast cancer | 4.4 | 4.4 |
| ionosphere | 36.5 | 36.5 |
| diabetes | 29.3 | 29.3 |
| glass | 30.1 | 30.1 |

# Variations of Bagging

# Pasting Small Votes

- Unlike bagging, pasting small votes is designed to be used with **large** data sets
- A large data set is partitioned into smaller subsets, called **bites**, each of which is used to train a different classifier
- Two variations: subsets are created at random (**Rvotes**); subsets are created based on the importance of instances (**Ivotes**)

# Pasting Small **I**votes

- Each classifier focuses on the most **important** (or most **informative**) instances
- Classifiers are added to the ensemble in an incremental and sequential fashion
- Current ensemble is evaluated on instances not used during training (**out-of-bag** classifiers)
- If an instance is misclassified by a majority vote, it is placed in the training set of the next classifier; otherwise, it is placed in the training set with a certain probability

**Algorithm: Pasting Small Votes (Ivotes)**

**Input:**

- Training data $S$ with correct labels $\omega_i \in \Omega = \{\omega_1,..., \omega_C\}$ representing $C$ classes;
- Weak learning algorithm **WeakLearn;**
- Integer $T$ specifying number of iterations;
- *Bitesize M*, indicating the size of individual training subsets to be created.

**Initialize**

1. Choose a random subset $S_0$ of size $M$ from $S$.
2. Call **WeakLearn** with $S_0$, and receive the hypothesis (classifier) $h_0$.
3. Evaluate $h_0$ on a validation dataset, and obtain error $\varepsilon_0$ of $h_0$..
4. If $\varepsilon_0 > 1/2$, return to step 1.

**Do** $t=1, \ldots, T$

1. Randomly draw an instance $\mathbf{x}$ from $S$ according to uniform distribution.
2. Evaluate $\mathbf{x}$ using majority vote of out-of-bag classifiers in the current ensemble $\mathtt{E_t}$ .
3. If $\mathbf{x}$ is misclassified, place $\mathbf{x}$ in $S_t$. Otherwise, place $\mathbf{x}$ in $S_t$ with probability $p$

$$p = \frac{\varepsilon_{t-1}}{(1-\varepsilon_{t-1})}. \quad (10)$$

   Repeat Steps 1-3 until $S_t$ has $M$ such instances.
4. Call **WeakLearn** with $S_t$ and receive the hypothesis $h_t$.
5. Evaluate $h_t$ on a validation dataset, and obtain error $\varepsilon_t$ of $h_t$.. If $\varepsilon_t > 1/2$, return to step 4.
6. Add $h_t$ to the ensemble to obtain $\mathtt{E_t}$ .

**End**

**Test –** Use simple majority voting on test data.

# Boosting

# Boosting

- Similar to bagging, boosting also creates an ensemble of classifiers by resampling the data, which are then combined by majority voting
- In boosting, though, the resampling strategy is geared to provide the **most informative** training data for each consecutive classifier

# Boosting (Adaboost.M1)

## Freund and Schapire, 1996

- Generates a set of classifiers, and combines them through weighted majority voting of the classes predicted by the individual classifiers
- Classifiers are trained using instances drawn from an iteratively updated distribution of the training data
- The distribution ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier
- Thus, consecutive classifiers' training data are more geared towards increasingly hard-to-classify instances

**Algorithm AdaBoost.M1**

**Input**:

- Sequence of $N$ examples $S = [(\mathbf{x}_i, y_i)], i = 1, \cdots, N$ with labels $y_i \in \Omega, \Omega = \{\omega_1, \ldots, \omega_C\}$;
- Weak learning algorithm **WeakLearn**;
- Integer $T$ specifying number of iterations.

**Initialize** $D_1(i) = \frac{1}{N}., i = 1, \cdots, N$ (11)

**Do for** $t = 1, 2, \ldots, T$:

1. Select a training data subset $S_t$, drawn from the distribution $D_t$.
2. Train **WeakLearn** with $S_t$, receive hypothesis $h_t$.
3. Calculate the error of $h_t$: $\varepsilon_t = \sum_{i:h_t(\mathbf{x}_i) \neq y_i} D_t(i).$ (12)

   If $\varepsilon_t > 1/2$, **abort**.
4. Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$. (13)
5. Update distribution

$$D_t : D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \textit{if } h_t(\mathbf{x}_i) = y_i \\ 1, & \textit{otherwise} \end{cases} \quad (14)$$

   where $Z_t = \sum_i D_t(i)$ is a normalization constant chosen so that $D_{t+1}$ becomes a proper distribution function.

**Test – Weighted Majority Voting:** Given an unlabeled instance $\mathbf{x}$,

1. Obtain total vote received by each class

$$V_j = \sum_{t:h_t(\mathbf{x}) = \omega_j} \log \frac{1}{\beta_t}, j = 1, \ldots, C. \quad (15)$$

2. Choose the class that receives the highest total vote as the final classification.

**Figure 8.** The AdaBoost.M1 algorithm.

# Boosting (property)

- Freund and Schapire proved that, provided that is always $\epsilon_t < 0.5$, the error rate of boosting on a given training data set, under the original uniform distribution, approaches zero exponentially fast as T increases.

# Boosting (property)

- Thus, a succession of weak classifiers can be boosted to a strong classifier that is at least as accurate as, and usually more accurate than, the best weak classifier on the training data.
- Of course, this gives no guarantee on the generalization performance on unseen instances.

# Experiments

from

*Bagging, Boosting, and C4.5*

by J. R. Quinlan

National Conference on Artificial Intelligence, 1996

# Description of data sets

| Name | Cases | Classes | Attributes | |
|---|---|---|---|---|
| | | | Cont | Discr |
| anneal | 898 | 6 | 9 | 29 |
| audiology | 226 | 6 | – | 69 |
| auto | 205 | 6 | 15 | 10 |
| breast-w | 699 | 2 | 9 | – |
| chess | 551 | 2 | – | 39 |
| colic | 368 | 2 | 10 | 12 |
| credit-a | 690 | 2 | 6 | 9 |
| credit-g | 1,000 | 2 | 7 | 13 |
| diabetes | 768 | 2 | 8 | – |
| glass | 214 | 6 | 9 | – |
| heart-c | 303 | 2 | 8 | 5 |
| heart-h | 294 | 2 | 8 | 5 |
| hepatitis | 155 | 2 | 6 | 13 |
| hypo | 3,772 | 5 | 7 | 22 |
| iris | 150 | 3 | 4 | – |
| labor | 57 | 2 | 8 | 8 |
| letter | 20,000 | 26 | 16 | – |
| lymph | 148 | 4 | – | 18 |
| phoneme | 5,438 | 47 | – | 7 |
| segment | 2,310 | 7 | 19 | – |
| sick | 3,772 | 2 | 7 | 22 |
| sonar | 208 | 2 | 60 | – |
| soybean | 683 | 19 | – | 35 |
| splice | 3,190 | 3 | – | 62 |
| vehicle | 846 | 4 | 18 | – |
| vote | 435 | 2 | – | 16 |
| waveform | 300 | 3 | 21 | – |

# C4.5, and its bagged and boosted versions

| | C4.5 | Bagged C4.5 vs C4.5 | | | Boosted C4.5 vs C4.5 | | | Boosting vs Bagging | |
|---|---|---|---|---|---|---|---|---|---|
| | err (%) | err (%) | w-l | ratio | err (%) | w-l | ratio | w-l | ratio |
| anneal | 7.67 | 6.25 | 10-0 | .814 | 4.73 | 10-0 | .617 | 10-0 | .758 |
| audiology | 22.12 | 19.29 | 9-0 | .872 | 15.71 | 10-0 | .710 | 10-0 | .814 |
| auto | 17.66 | 19.66 | 2-8 | 1.113 | 15.22 | 9-1 | .862 | 9-1 | .774 |
| breast-w | 5.28 | 4.23 | 9-0 | .802 | 4.09 | 9-0 | .775 | 7-2 | .966 |
| chess | 8.55 | 8.33 | 6-2 | .975 | 4.59 | 10-0 | .537 | 10-0 | .551 |
| colic | 14.92 | 15.19 | 0-6 | 1.018 | 18.83 | 0-10 | 1.262 | 0-10 | 1.240 |
| credit-a | 14.70 | 14.13 | 8-2 | .962 | 15.64 | 1-9 | 1.064 | 0-10 | 1.107 |
| credit-g | 28.44 | 25.81 | 10-0 | .908 | 29.14 | 2-8 | 1.025 | 0-10 | 1.129 |
| diabetes | 25.39 | 23.63 | 9-1 | .931 | 28.18 | 0-10 | 1.110 | 0-10 | 1.192 |
| glass | 32.48 | 27.01 | 10-0 | .832 | 23.55 | 10-0 | .725 | 9-1 | .872 |
| heart-c | 22.94 | 21.52 | 7-2 | .938 | 21.39 | 8-0 | .932 | 5-4 | .994 |
| heart-h | 21.53 | 20.31 | 8-1 | .943 | 21.05 | 5-4 | .978 | 3-6 | 1.037 |
| hepatitis | 20.39 | 18.52 | 9-0 | .908 | 17.68 | 10-0 | .867 | 6-1 | .955 |
| hypo | .48 | .45 | 7-2 | .928 | .36 | 9-1 | .746 | 9-1 | .804 |
| iris | 4.80 | 5.13 | 2-6 | 1.069 | 6.53 | 0-10 | 1.361 | 0-8 | 1.273 |
| labor | 19.12 | 14.39 | 10-0 | .752 | 13.86 | 9-1 | .725 | 5-3 | .963 |
| letter | 11.99 | 7.51 | 10-0 | .626 | 4.66 | 10-0 | .389 | 10-0 | .621 |
| lymphography | 21.69 | 20.41 | 8-2 | .941 | 17.43 | 10-0 | .804 | 10-0 | .854 |
| phoneme | 19.44 | 18.73 | 10-0 | .964 | 16.36 | 10-0 | .842 | 10-0 | .873 |
| segment | 3.21 | 2.74 | 9-1 | .853 | 1.87 | 10-0 | .583 | 10-0 | .684 |
| sick | 1.34 | 1.22 | 7-1 | .907 | 1.05 | 10-0 | .781 | 9-1 | .861 |
| sonar | 25.62 | 23.80 | 7-1 | .929 | 19.62 | 10-0 | .766 | 10-0 | .824 |
| soybean | 7.73 | 7.58 | 6-3 | .981 | 7.16 | 8-2 | .926 | 8-1 | .944 |
| splice | 5.91 | 5.58 | 9-1 | .943 | 5.43 | 9-0 | .919 | 6-4 | .974 |
| vehicle | 27.09 | 25.54 | 10-0 | .943 | 22.72 | 10-0 | .839 | 10-0 | .889 |
| vote | 5.06 | 4.37 | 9-0 | .864 | 5.29 | 3-6 | 1.046 | 1-9 | 1.211 |
| waveform | 27.33 | 19.77 | 10-0 | .723 | 18.53 | 10-0 | .678 | 8-2 | .938 |
| *average* | *15.66* | *14.11* | | *.905* | *13.36* | | *.847* | | *.930* |

# C4.5, and its bagged and boosted versions

| | C4.5 | Bagged C4.5 vs C4.5 | | | Boosted C4.5 vs C4.5 | | | Boosting vs Bagging | |
|---|---|---|---|---|---|---|---|---|---|
| | err (%) | err (%) | w-l | ratio | err (%) | w-l | ratio | w-l | ratio |
| anneal | 7.67 | 6.25 | 10-0 | .814 | 4.73 | 10-0 | .617 | 10-0 | .758 |
| audiology | 22.12 | 19.29 | 9-0 | .872 | 15.71 | 10-0 | .710 | 10-0 | .814 |
| auto | 17.66 | 19.66 | 2-8 | $\boxed{1.113}$ | 15.22 | 9-1 | .862 | 9-1 | .774 |
| breast-w | 5.28 | 4.23 | 9-0 | .802 | 4.09 | 9-0 | .775 | 7-2 | .966 |
| chess | 8.55 | 8.33 | 6-2 | .975 | 4.59 | 10-0 | .537 | 10-0 | .551 |
| colic | 14.92 | 15.19 | 0-6 | $\boxed{1.018}$ | 18.83 | 0-10 | $\boxed{1.262}$ | 0-10 | 1.240 |
| credit-a | 14.70 | 14.13 | 8-2 | .962 | 15.64 | 1-9 | $\boxed{1.064}$ | 0-10 | 1.107 |
| credit-g | 28.44 | 25.81 | 10-0 | .908 | 29.14 | 2-8 | $\boxed{1.025}$ | 0-10 | 1.129 |
| diabetes | 25.39 | 23.63 | 9-1 | .931 | 28.18 | 0-10 | $\boxed{1.110}$ | 0-10 | 1.192 |
| glass | 32.48 | 27.01 | 10-0 | .832 | 23.55 | 10-0 | .725 | 9-1 | .872 |
| heart-c | 22.94 | 21.52 | 7-2 | .938 | 21.39 | 8-0 | .932 | 5-4 | .994 |
| heart-h | 21.53 | 20.31 | 8-1 | .943 | 21.05 | 5-4 | .978 | 3-6 | 1.037 |
| hepatitis | 20.39 | 18.52 | 9-0 | .908 | 17.68 | 10-0 | .867 | 6-1 | .955 |
| hypo | .48 | .45 | 7-2 | .928 | .36 | 9-1 | .746 | 9-1 | .804 |
| iris | 4.80 | 5.13 | 2-6 | $\boxed{1.069}$ | 6.53 | 0-10 | $\boxed{1.361}$ | 0-8 | 1.273 |
| labor | 19.12 | 14.39 | 10-0 | .752 | 13.86 | 9-1 | .725 | 5-3 | .963 |
| letter | 11.99 | 7.51 | 10-0 | .626 | 4.66 | 10-0 | .389 | 10-0 | .621 |
| lymphography | 21.69 | 20.41 | 8-2 | .941 | 17.43 | 10-0 | .804 | 10-0 | .854 |
| phoneme | 19.44 | 18.73 | 10-0 | .964 | 16.36 | 10-0 | .842 | 10-0 | .873 |
| segment | 3.21 | 2.74 | 9-1 | .853 | 1.87 | 10-0 | .583 | 10-0 | .684 |
| sick | 1.34 | 1.22 | 7-1 | .907 | 1.05 | 10-0 | .781 | 9-1 | .861 |
| sonar | 25.62 | 23.80 | 7-1 | .929 | 19.62 | 10-0 | .766 | 10-0 | .824 |
| soybean | 7.73 | 7.58 | 6-3 | .981 | 7.16 | 8-2 | .926 | 8-1 | .944 |
| splice | 5.91 | 5.58 | 9-1 | .943 | 5.43 | 9-0 | .919 | 6-4 | .974 |
| vehicle | 27.09 | 25.54 | 10-0 | .943 | 22.72 | 10-0 | .839 | 10-0 | .889 |
| vote | 5.06 | 4.37 | 9-0 | .864 | 5.29 | 3-6 | $\boxed{1.046}$ | 1-9 | 1.211 |
| waveform | 27.33 | 19.77 | 10-0 | .723 | 18.53 | 10-0 | .678 | 8-2 | .938 |
| *average* | *15.66* | *14.11* | | *.905* | *13.36* | | *.847* | | *.930* |

# Comparison of Bagging and Boosting on chess and colic data sets