Ensemble Methods

- An ensemble is a set of classifiers that learn a target function, and their individual predictions are combined to classify new examples.
- Ensembles generally improve the generalization performance of a set of classifiers on a domain.

Typical Ensemble Procedure

General procedure for ensemble method.

Let D denote the original training data, k denote the number of base classifiers, and T be the test data.

```
**for** $i = 1$ to $k$ **do**
    Create training set $D_i$ from $D$.
    Build a base classifier $C_i$ from D.
**end for**
**for** each test record $x \in T$ **do**
    $C^*(x) = Vote\left(C_1(\mathbf{x}), C_2(\mathbf{x}), \dots, C_k(\mathbf{x})\right)$
**end for**
```

Rationale for Ensemble Methods

- Statistical reasons
  - A set of classifiers with similar training performances may have different generalization performances.
  - Combining outputs of several classifiers *reduces the risk of selecting a poorly performing classifier*.

Rationale for Ensemble Methods

- Large volumes of data
  - If the amount of data to be analyzed is too large, a single classifier may not be able to handle it; train different classifiers on *different partitions of data*.
- Too little data
  - Ensemble systems can also be used when there is too little data; *resampling techniques*.

Rationale for Ensemble Methods

- Divide-and-conquer

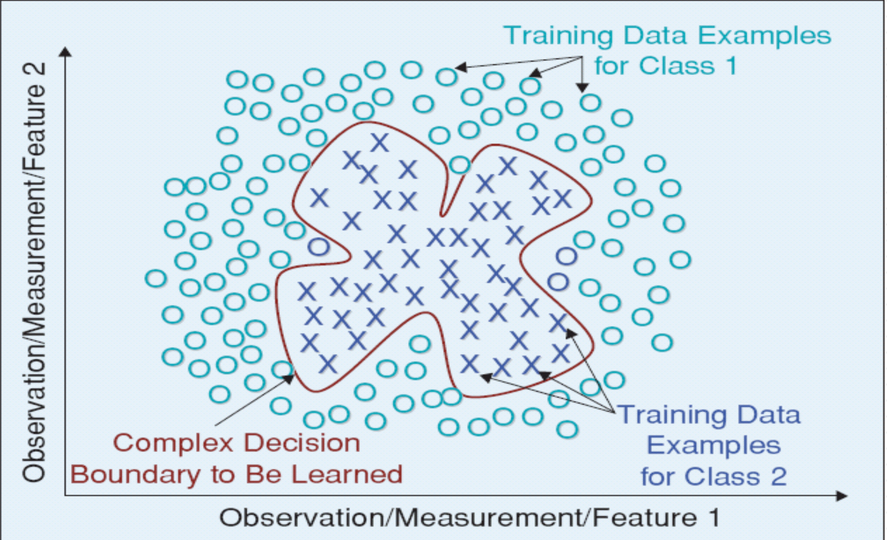

Training Data Examples for Class 1
Observation/Measurement/Feature 2
Complex Decision Boundary to Be Learned
Training Data Examples for Class 2
Observation/Measurement/Feature 1

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

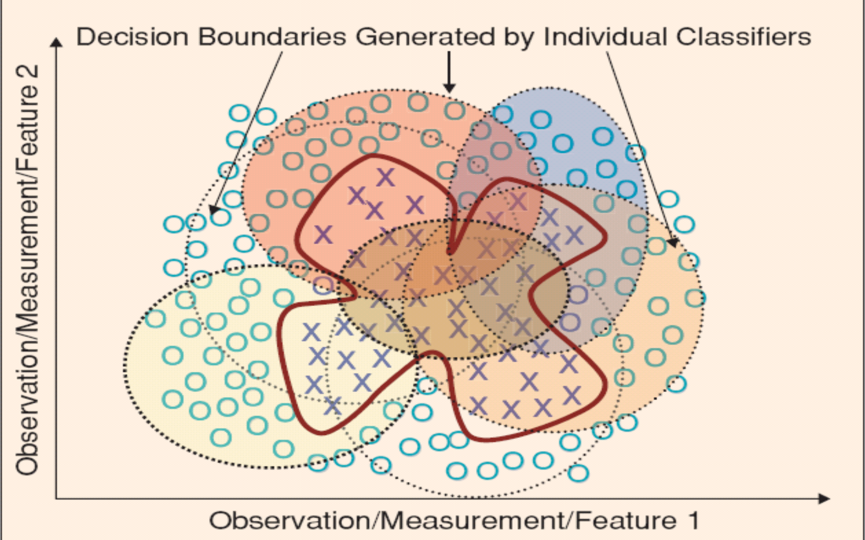

Decision Boundaries Generated by Individual Classifiers
Observation/Measurement/Feature 2
Observation/Measurement/Feature 1

Figure 2. Ensemble of classifiers spanning the decision space.

Why Ensemble Methods Work

Consider an ensemble of twenty-five binary classifiers, each of which has an error rate of $\epsilon = 0.35$. If the base classifiers are independent—i.e., their errors are uncorrelated—then the error rate of the ensemble classifier is

$$e_{\text{ensemble}} = \sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} = 0.06$$

which is considerably lower than the base classifiers.

Requirements for Ensembles

There are two necessary and sufficient conditions for an ensemble of classifiers to be more accurate than any of its individual members. Namely:

1. The classifiers must be accurate.
2. The classifiers must be diverse.

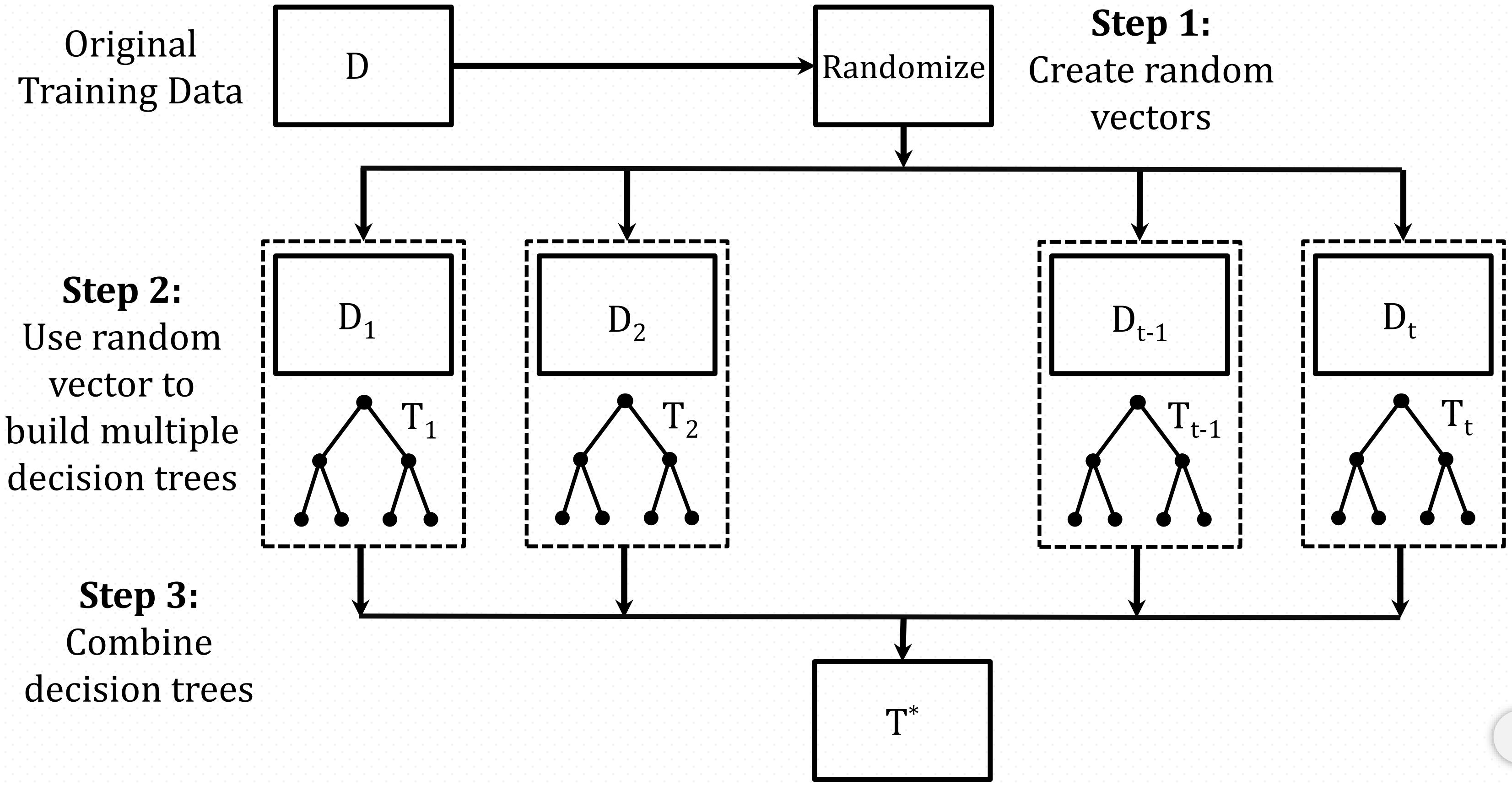
Methods for Constructing Ensembles

1. **By manipulating the training set.** Create multiple training sets by resampling the original data according to some sampling distribution.
2. **By manipulating the input features.** Choose a subset of input features to form each training set.
3. **By manipulating the class labels.** Transform the training data into a binary class problem by randomly partitioning the class labels into two disjoint subsets.
4. **By manipulating the learning algorithm.** Manipulate the learning algorithm to generate different models.

Random Forests Method

- A class of ensemble methods specifically designed for decision tree classifiers.
- Combines predictions made by many decision trees.
- Each tree is generated based on a bootstrap sample and the values of an independent set of random vectors.
- The random vectors are generated from a fixed probability distribution.

Random Forests: A Visual Explanation


Original Training Data
D
Randomize
Step 1: Create random vectors
Step 2: Use random vector to build multiple decision trees
D_1
D_2
D_{t-1}
D_t
T_1
T_2
T_{t-1}
T_t
Step 3: Combine decision trees
T^*

Randomizing Random Forests

Each decision tree uses a random vector that is generated from some fixed probability distribution. This randomness can be incorporated in many ways:

1. Randomly select F input features to split at each node (Forest-RI).
2. Create linear combinations of the input features to split at each node (Forest-RC).
3. Randomly select one of the F best splits at each node.

Why use Random Vectors?

- Recall that an underlying assumption of the ensemble process is that the base learners are *independent*.
- As the trees become more correlated (less independent), the generalization error bound tends to increase.
- Randomization helps to reduce the correlation among decision trees.

Random Forests Error Bounds

The upper bound for generalization error of random forests converges to the following expression, when the number of trees is sufficiently large

$$\text{Generalization error} \leq \frac{\bar{\rho}\left(1 - s^2\right)}{s^2},$$

where $\bar{\rho}$ is the average correlation among the trees and s is a quantity the measures the "strength" (or average performance) of the tree classifiers.

Out of Bag Error

- Assume a method for constructing a classifier from any training set. Given a specific training set T, form bootstrap training sets T_k, construct classifiers, and let these vote to form the bagged predictor. For each $y, \mathbf{x}$ in the training set, aggregate the votes only over those classifiers for which T_k does not contain $y, \mathbf{x}$. Call this the out-of-bag classifier.

Using Random Features

- Random split selection does better than bagging; introduction of random noise into the outputs also does better.
- To improve accuracy, the randomness injected has to minimize the correlation $\bar{\rho}$ while maintaining strength.

Random Forests Advantages

- Empirically comparable classification accuracies to AdaBoost.
- More robust to outliers and noise than AdaBoost.
- Runs much faster than AdaBoost.
- Produces useful internal estimates of error, strength, correlation, and variable importance.
- Simple and easily parallelized.

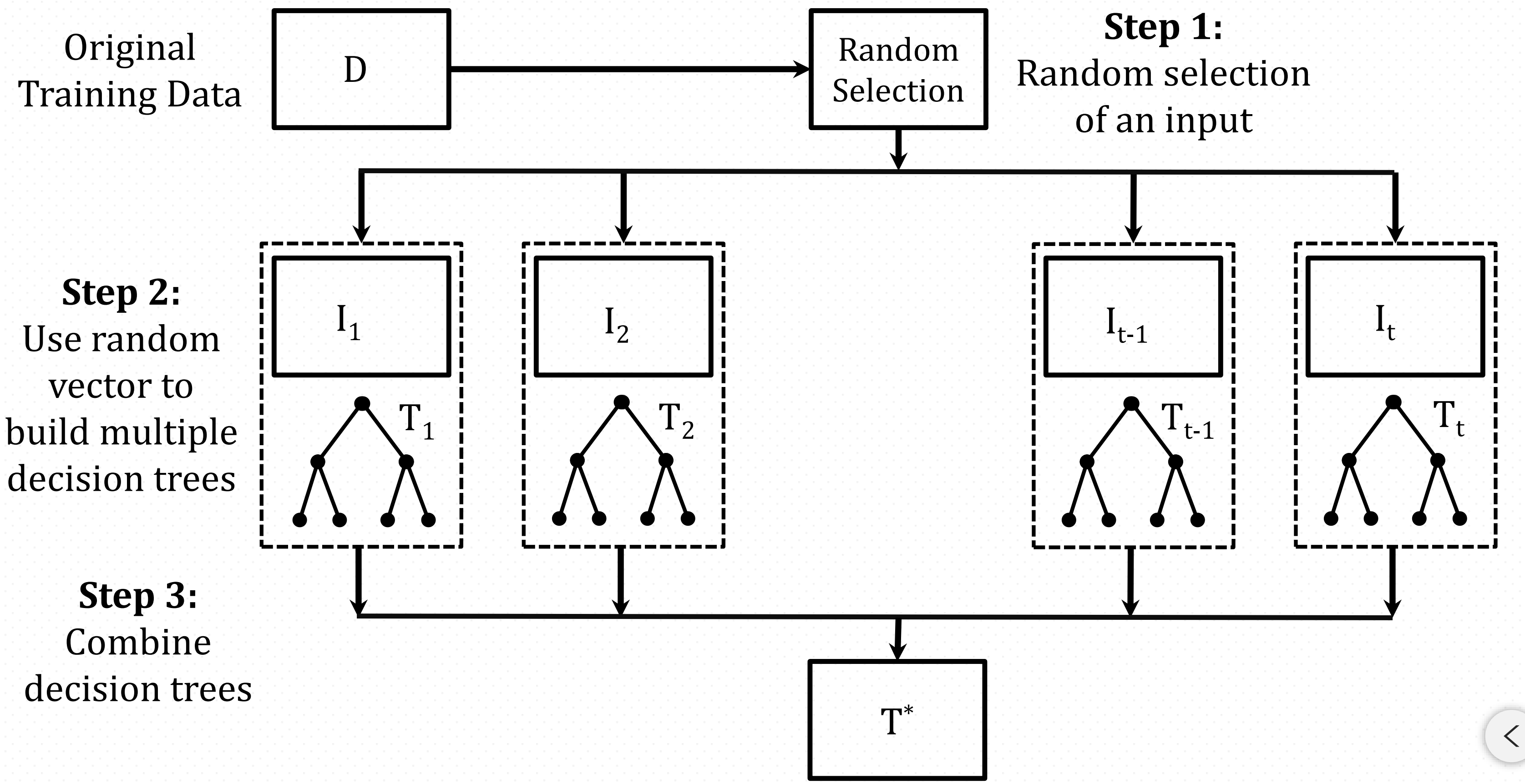
Random Subspace Method

- An ensemble classifier that consists of several classifiers trained on the same dataset using only a (random) subset of the available features.
  - A generalization of the random forest algorithm.
  - Can be composed from any underlying classifiers.

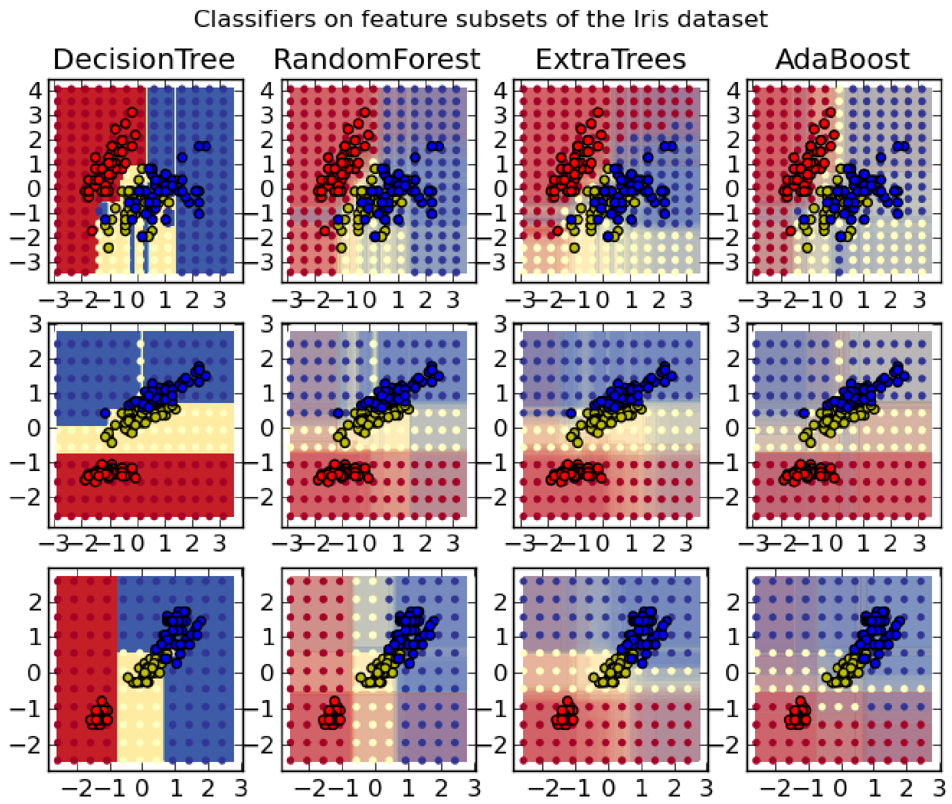
Extra-Trees Method

- **Extremely randomized trees (extra-trees)** are another class of ensemble methods specifically designed for decision tree classifiers.
- Each tree is built from the original learning sample.
- At each test node, the best split is determined among K random splits, and each one is determined by a random selection of an input (without replacement) and a threshold.

Extra-Trees: A Visual Explanation


Original Training Data
D
Random Selection
Step 1: Random selection of an input
Step 2: Use random vector to build multiple decision trees
I_1
I_2
I_{t-1}
I_t
T_1
T_2
T_{t-1}
T_t
Step 3: Combine decision trees
T^*

Ensembles of Trees: A Comparison



Summarizing Ensembles of Trees

- **Random forests** can be thought of as combining bagging and random subspaces.
  - Bootstrap aggregated (bagged) trees.
  - Random subset of the input space (random subspaces).
- **Extra-trees** combine classifiers without bagging by using random splits to generate different trees.
- Decision tree ensembles are **effective because decision trees are diverse** (display high variance).