

# **Data Mining**

## **Classification: Alternative Techniques**

---

Ensemble Methods

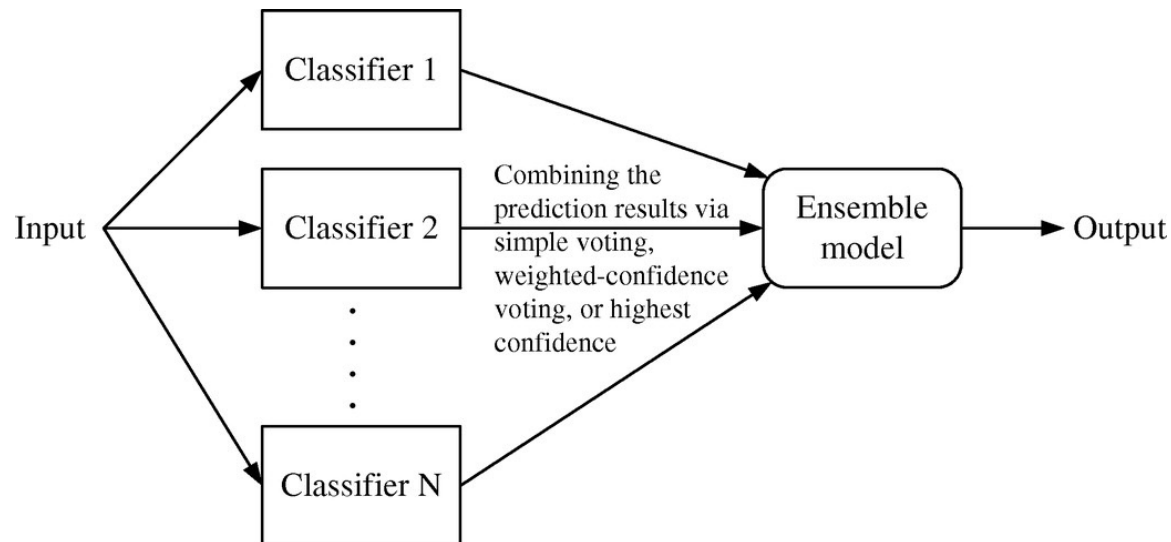
Introduction to Data Mining

Fang Zhou

# Ensemble Methods

---

- Construct a set of classifiers from the training data
- Predict class label of test records by combining the predictions made by multiple classifiers



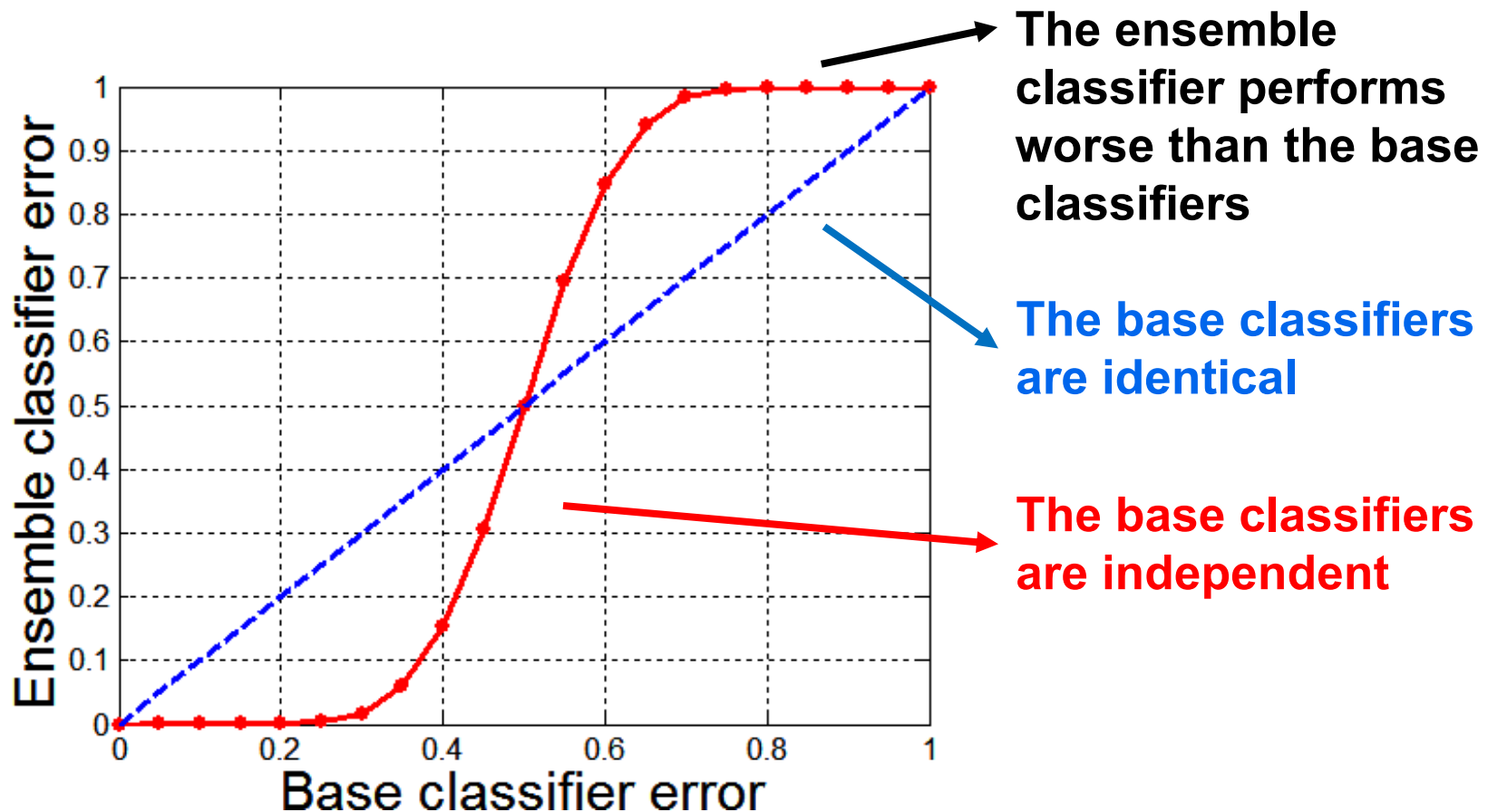
# Why Ensemble Methods work?

---

- Suppose there are 25 base classifiers
  - Each classifier has error rate,  $\varepsilon = 0.35$
  - Assume errors made by classifiers are uncorrelated
  - Probability that the ensemble classifier makes a wrong prediction:

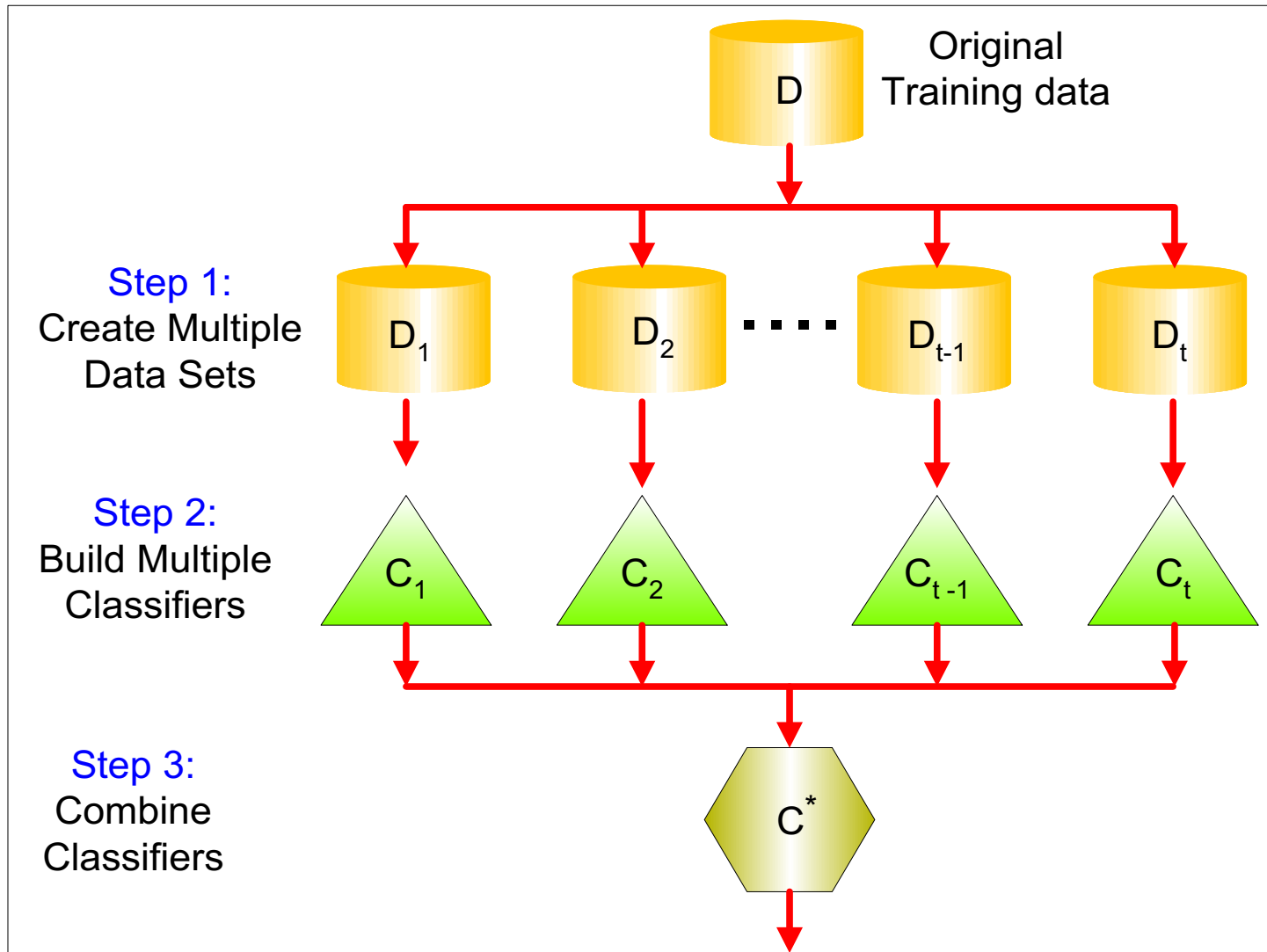
$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1 - \varepsilon)^{25-i} = 0.06$$

# Why Ensemble Methods work?



- The base classifiers should be independent of each other
- The base classifiers should do better than a classifier that performs random guessing

# General Approach



# Types of Ensemble Methods

---

- Manipulate the training set
  - Resampling the original data according to some sampling distribution
  - Example: bagging, boosting
- Manipulate the input features
  - A subset of input features is chosen to form each training set
  - Example: random forest

# Types of Ensemble Methods

---

- Manipulate the class labels
  - When the number of classes is large, the training data is transformed into a binary class problem by randomly partitioning the class labels into two disjoint subset
- Manipulate the learning algorithm

# Ensemble method

---

## Algorithm 5.5 General procedure for ensemble method.

---

```
1: Let  $D$  denote the original training data,  $k$  denote the number of base classifiers,
   and  $T$  be the test data.
2: for  $i = 1$  to  $k$  do
3:   Create training set,  $D_i$  from  $D$ .
4:   Build a base classifier  $C_i$  from  $D_i$ .
5: end for
6: for each test record  $x \in T$  do
7:    $C^*(x) = \text{Vote}(C_1(\mathbf{x}), C_2(\mathbf{x}), \dots, C_k(\mathbf{x}))$ 
8: end for
```

can be generated sequentially  
or in parallel

- Ensemble methods work better with **unstable classifiers**
  - Base classifiers that are sensitive to minor perturbations in the training set
    - For example, decision tree or ANN

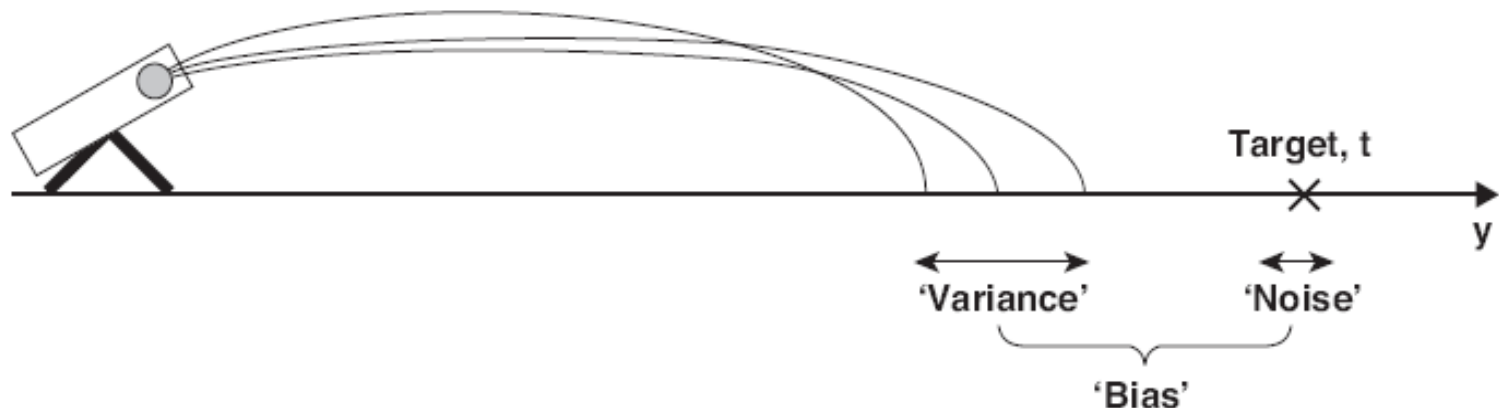


# Bias-Variance Decomposition

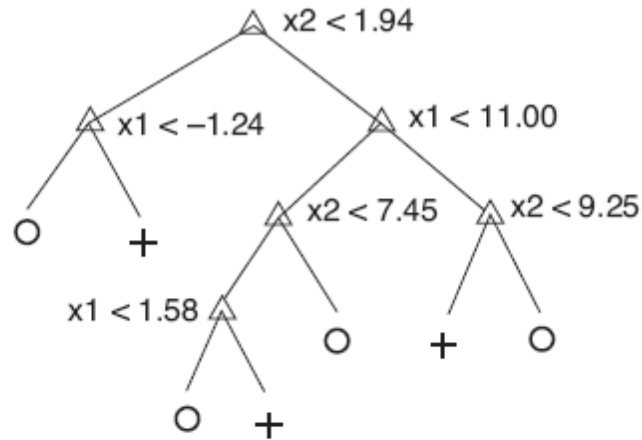
- Consider the trajectories of a projectile launched at a particular angle. The observed distance can be divided into 3 components.

$$d_{f,\theta}(y, t) = \text{Bias}_{\theta} + \text{Variance}_f + \text{Noise}_t$$

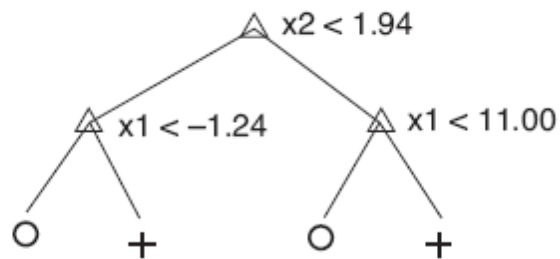
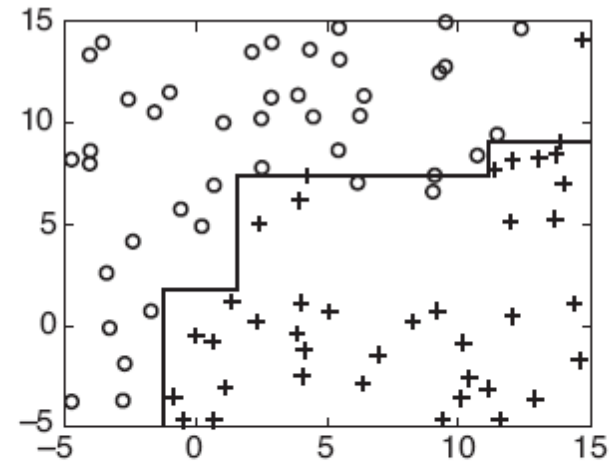
- Forth ( $f$ ) and angle( $\theta$ )
- Suppose the target is  $t$ , the projectile hits at  $x$  at a distance  $d$  away from  $t$ .



# Two decision trees



(a) Decision tree  $T_1$



(b) Decision tree  $T_2$

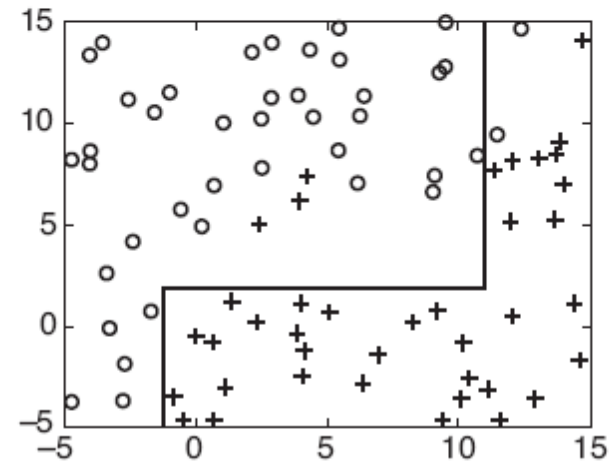
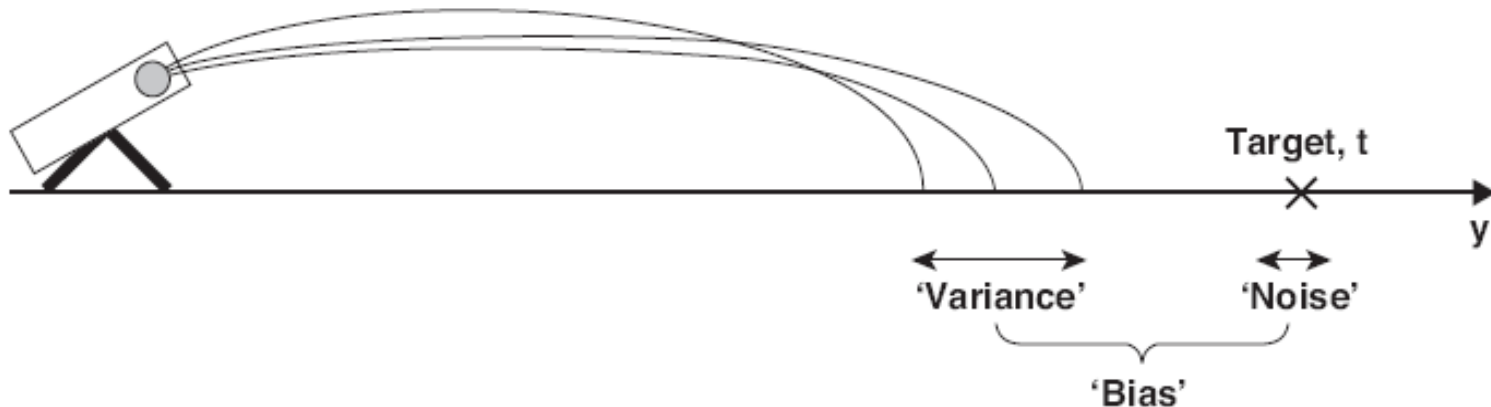


Figure 5.33. Two decision trees with different complexities induced from the same training data.

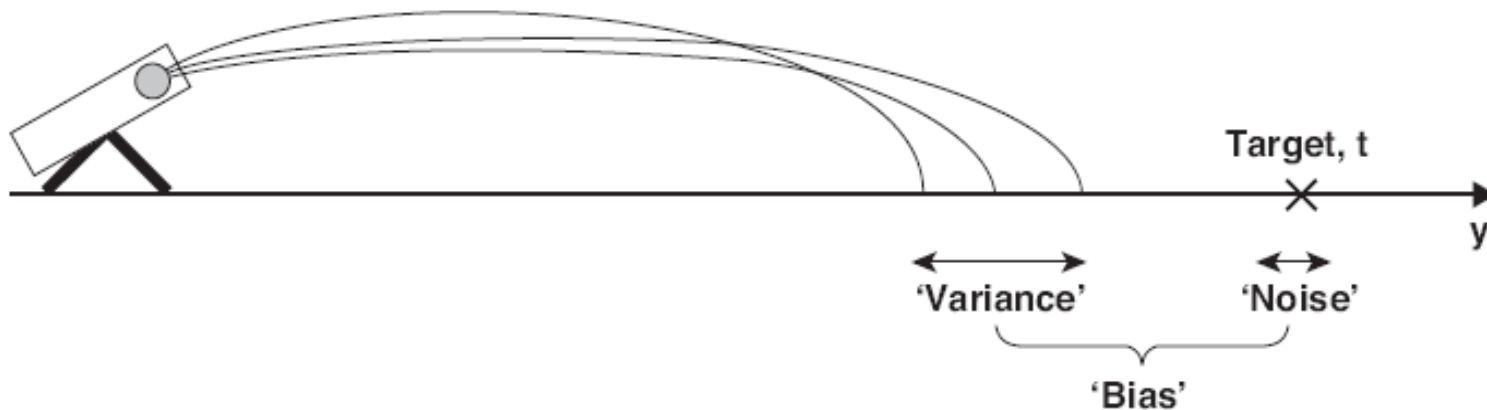
# Two decision trees

- **Bias:** The stronger the assumptions made by a classifier about the nature of its decision boundary, the larger the classifier's bias will be.
  - A smaller tree has a stronger assumption.
  - An algorithm can not learn the target.



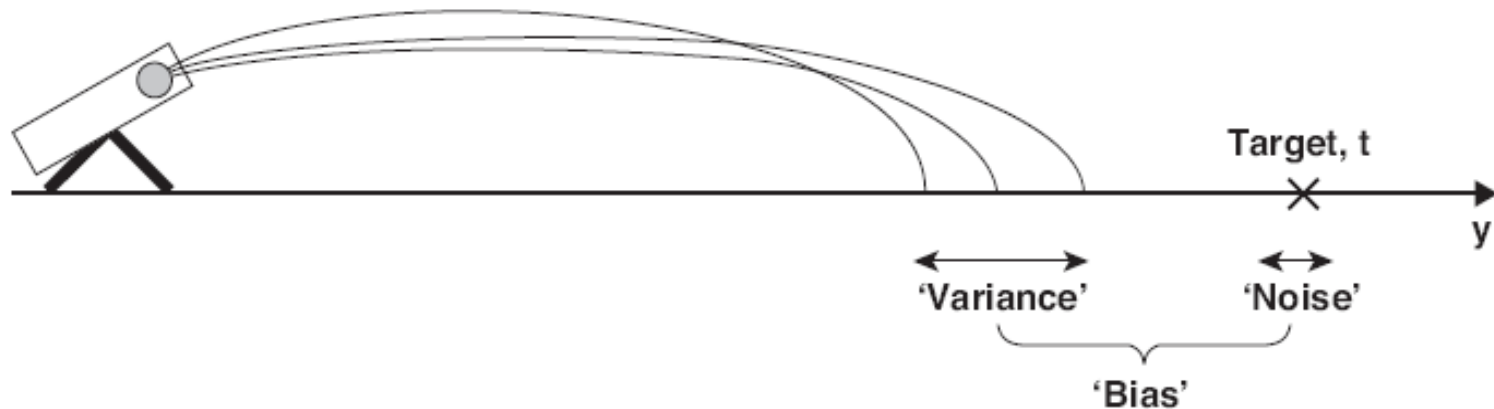
# Two decision trees

- **Variance:** Variability in the training data affects the expected error, because different compositions of the training set may lead to different decision boundaries.

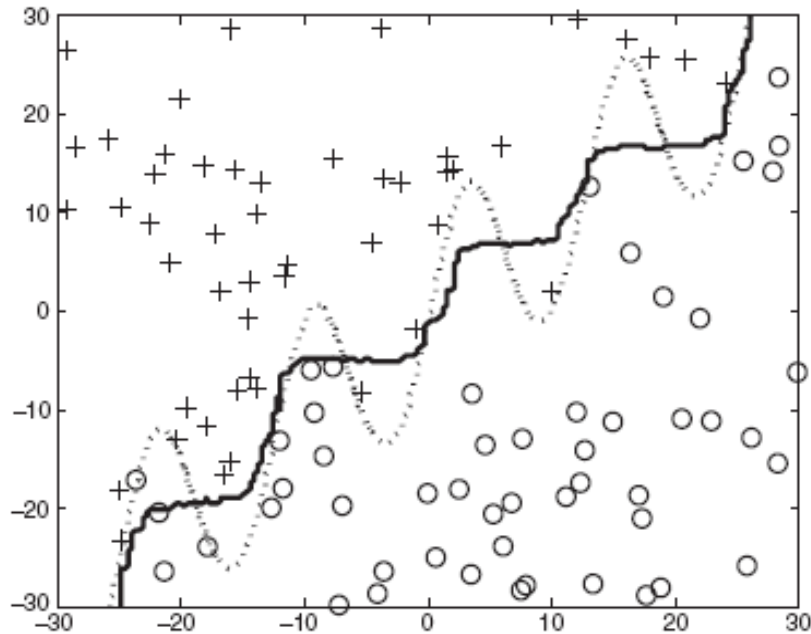


# Two decision trees

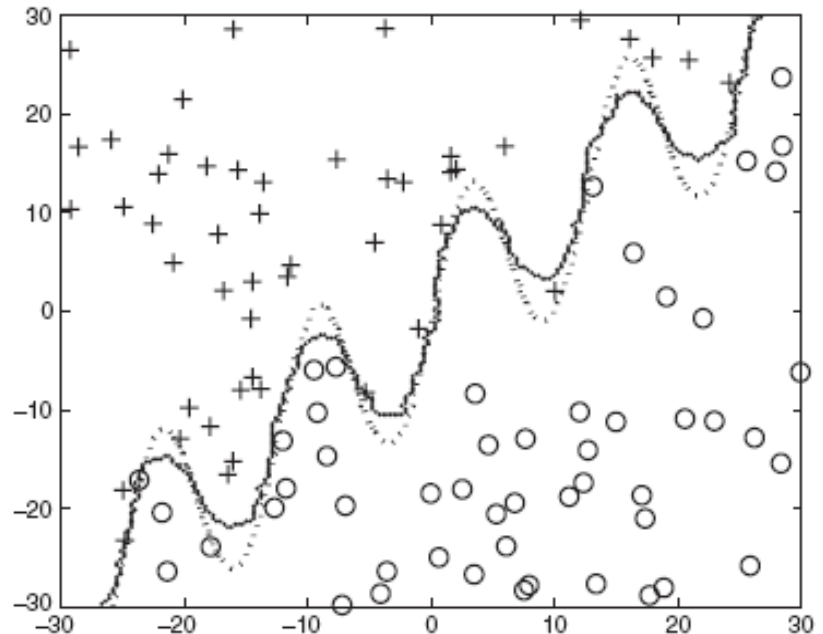
- Intrinsic **noise** in the target class
  - Target class can be non-deterministic
  - Same attributes values with different class labels



# Bias of decision tree and 1-nn



(a) Decision boundary for decision tree.



(b) Decision boundary for 1-nearest neighbor.

**The bias of a 1-nearest neighbor classifier is lower than the bias of a decision tree classifier.**

# Examples of Ensemble Methods

---

- How to generate an ensemble of classifiers?
  - Bagging
  - Boosting

# Bagging

---

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample
- Each sample has probability  $(1 - 1/n)^n$  of being selected. When  $n$  is large, a bootstrap sample contains about 63.2% of the training data.



# Summary on bagging

---

- Bagging improves generalization error by reducing the bias/variance? of the base classifiers.

# Summary on bagging

---

- Bagging improves generalization error by reducing the variance of the base classifiers.
- The performance of bagging depends on the stability of the base classifier.
  - If a base classifier is unstable, bagging helps to reduce the errors associated with random fluctuations in the training data.
  - If a base classifier is stable, then the error of the ensemble is primarily caused by bias in the base classifier. Bagging may degrade the classifier's performance, as the sample size is 37% smaller.
- Does not focus on any particular instance
  - Less susceptible to model overfitting when applied to noisy data.

# Boosting

---

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Boosting assigns a weight to each training example
  - Initially, all  $N$  records are assigned equal weights
  - Unlike bagging, weights may change at the end of each boosting round

# Summary on boosting

---

- Because of its tendency to focus on training examples that are wrongly classified, the boosting technique can be quite susceptible to overfitting.


# Random forests

---

Training Data

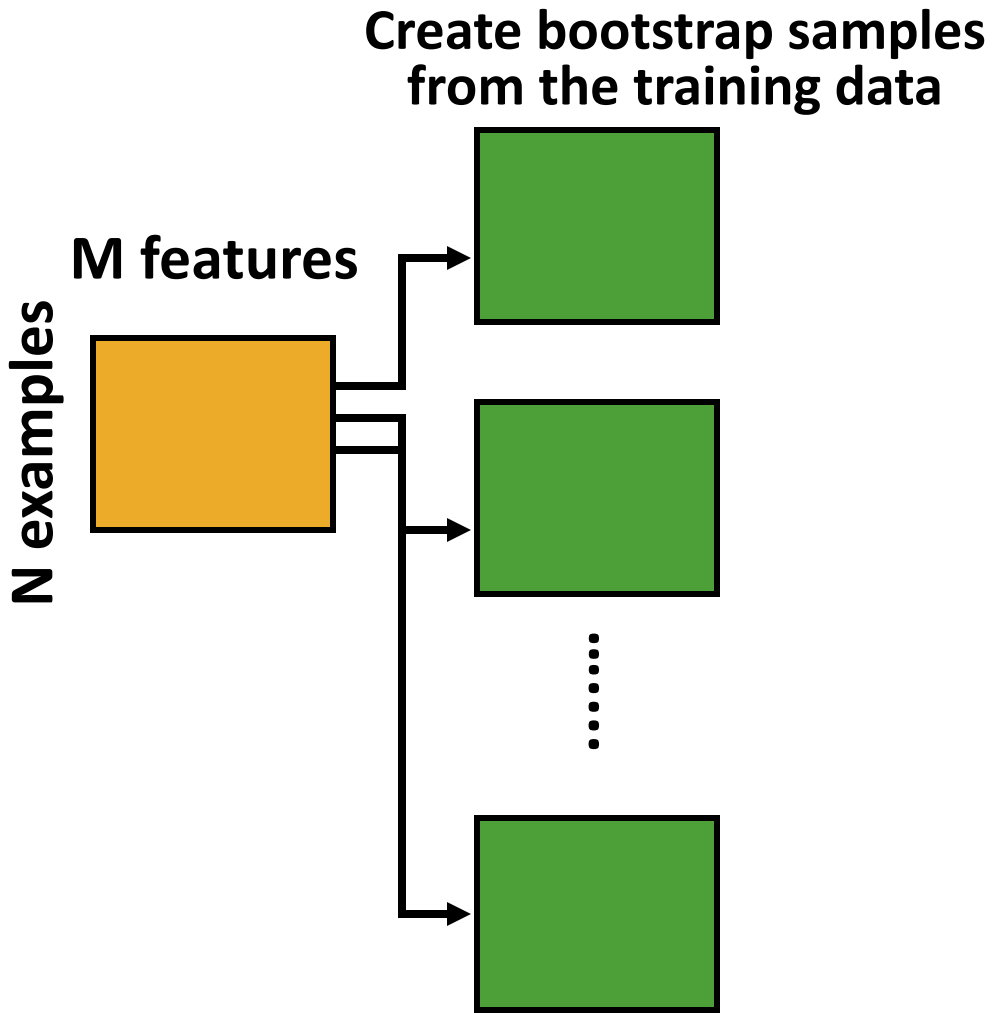
N examples

M features

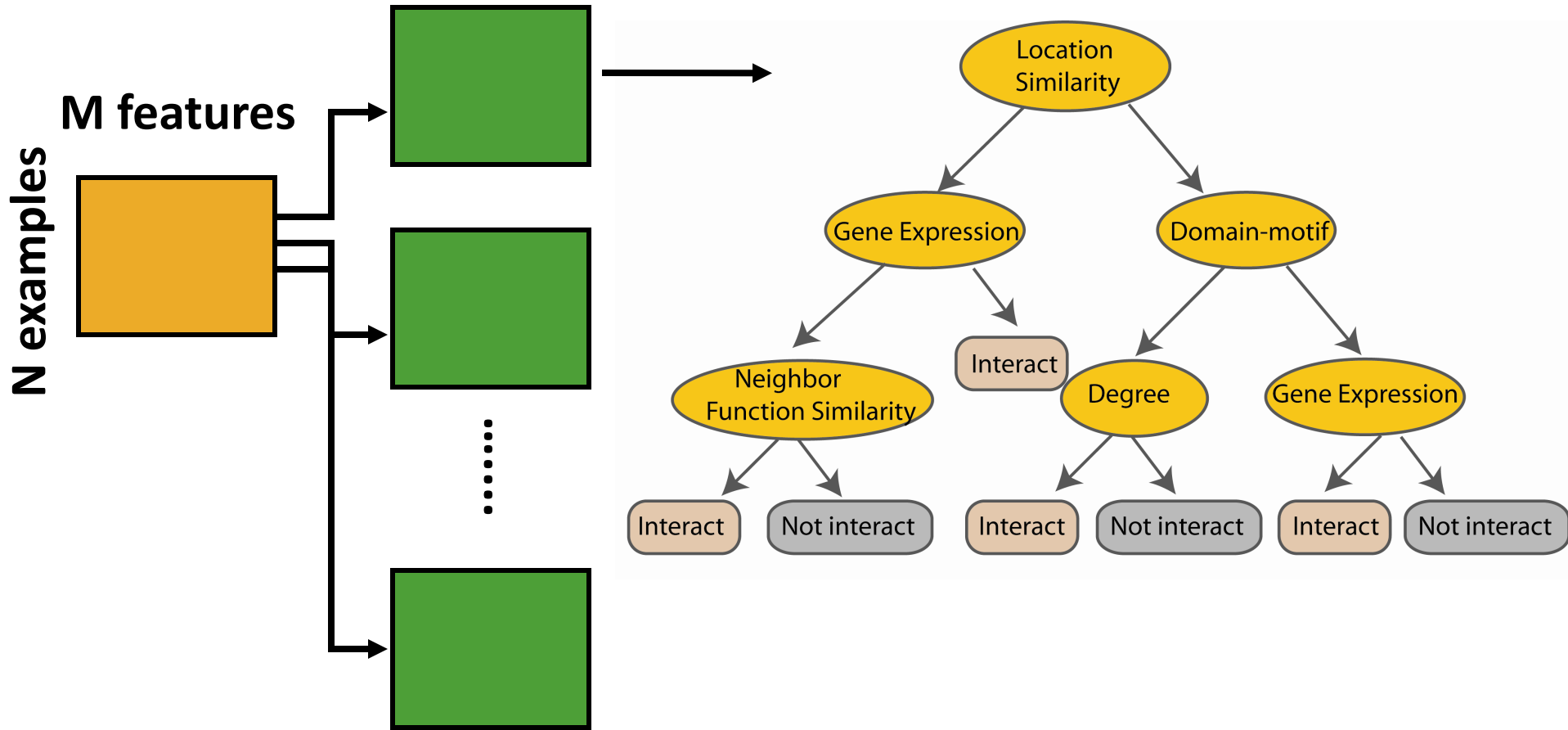


# Random forests

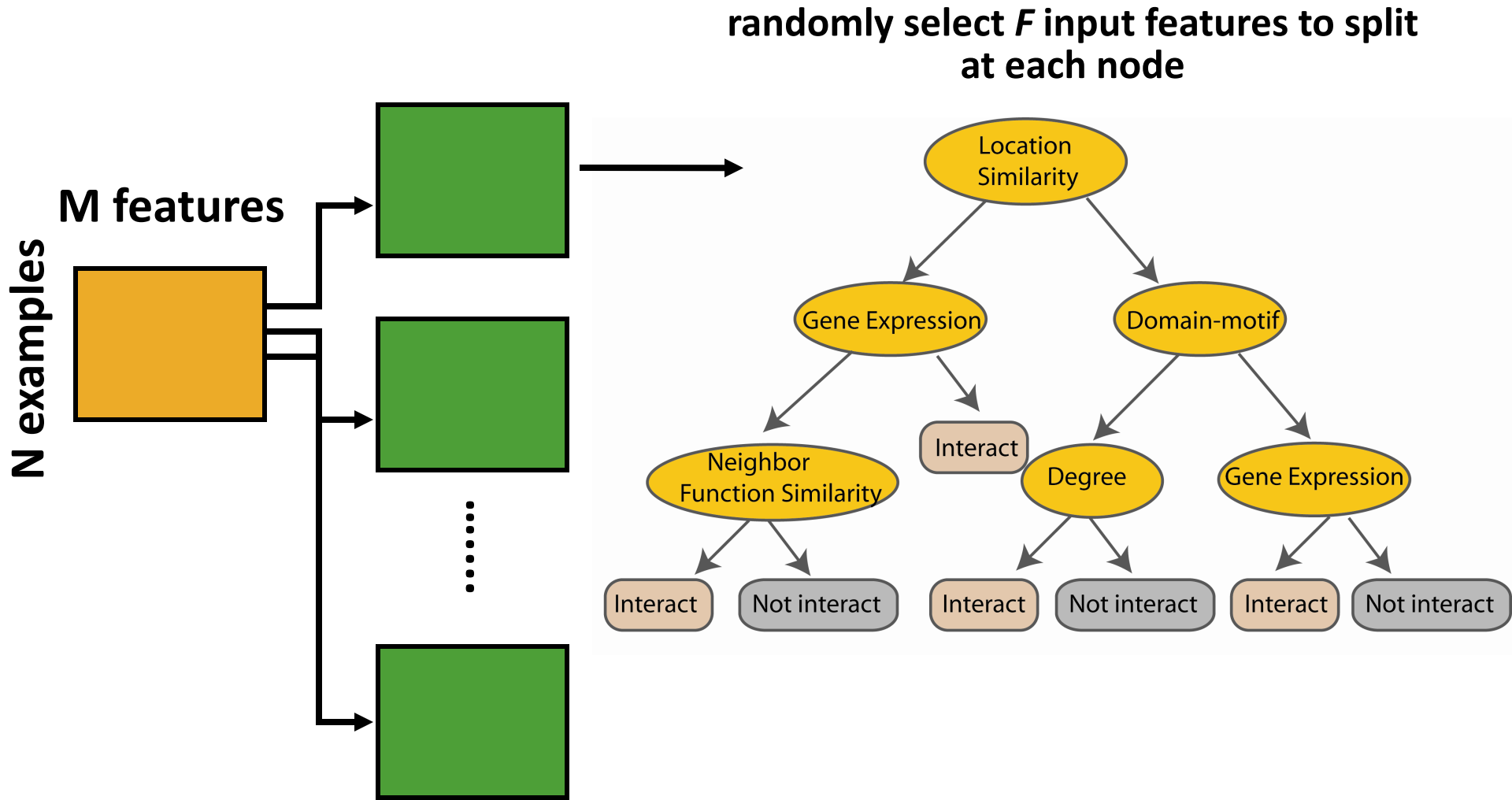
---



# Random forests

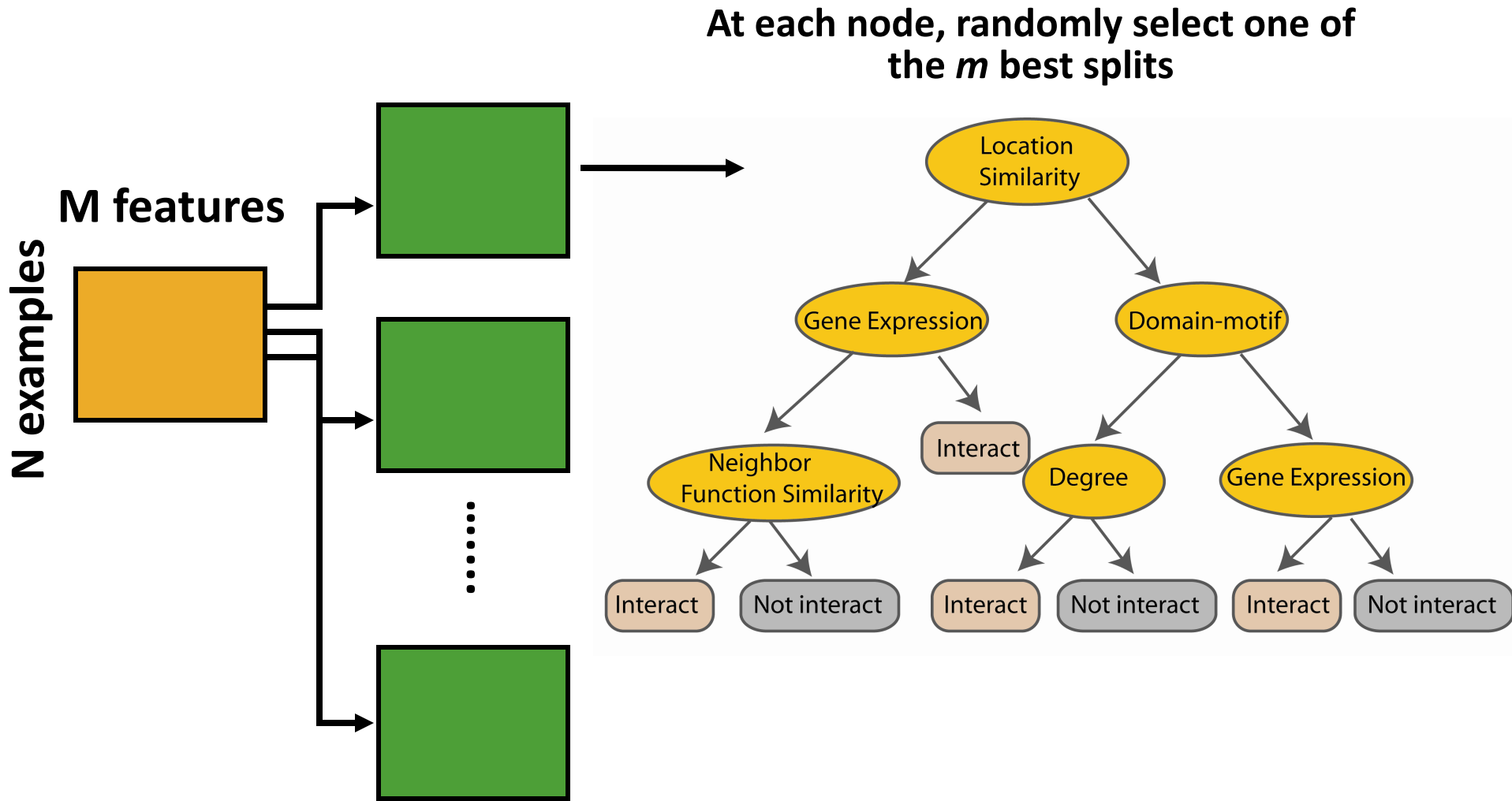


# Random forests

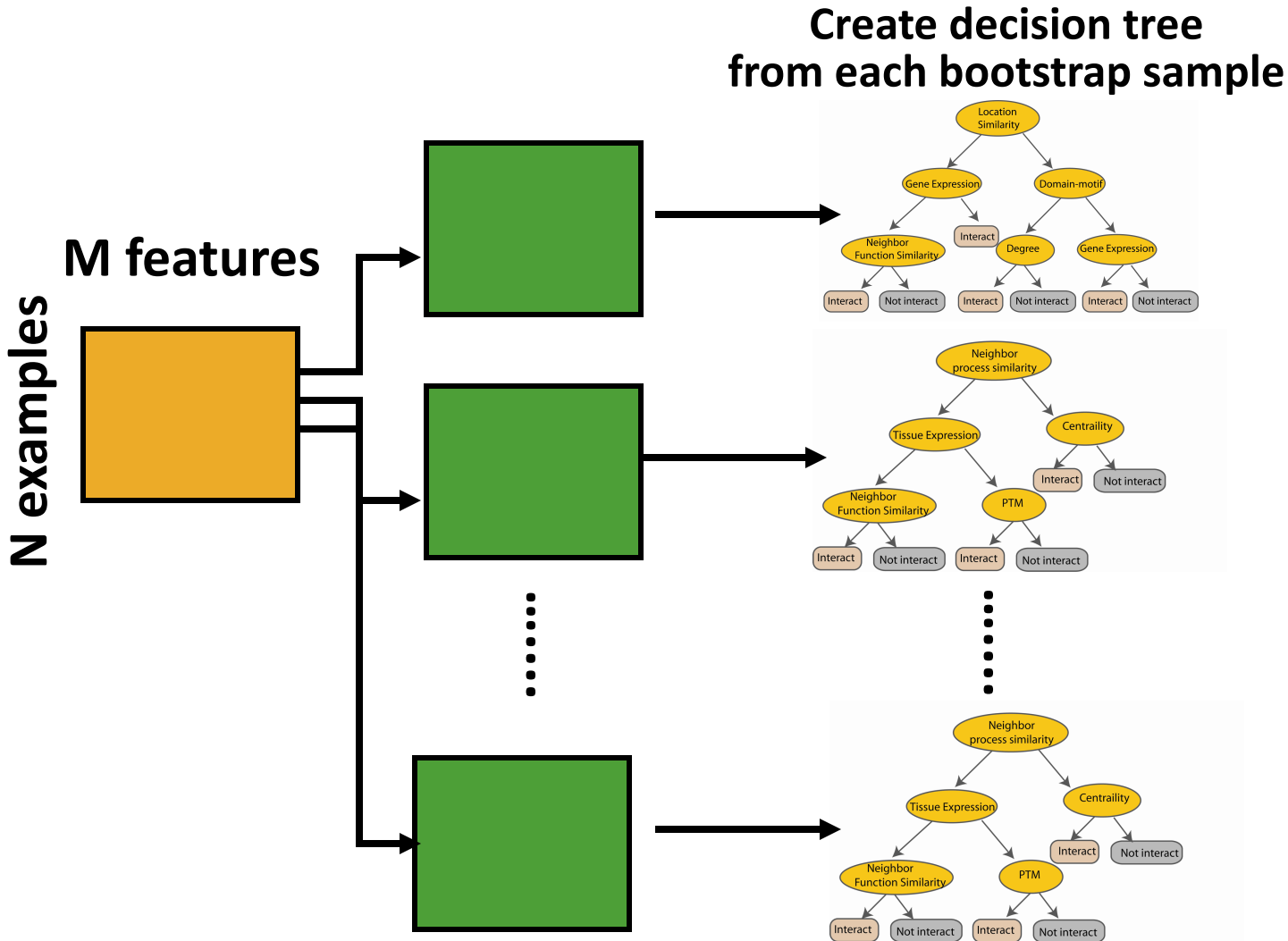




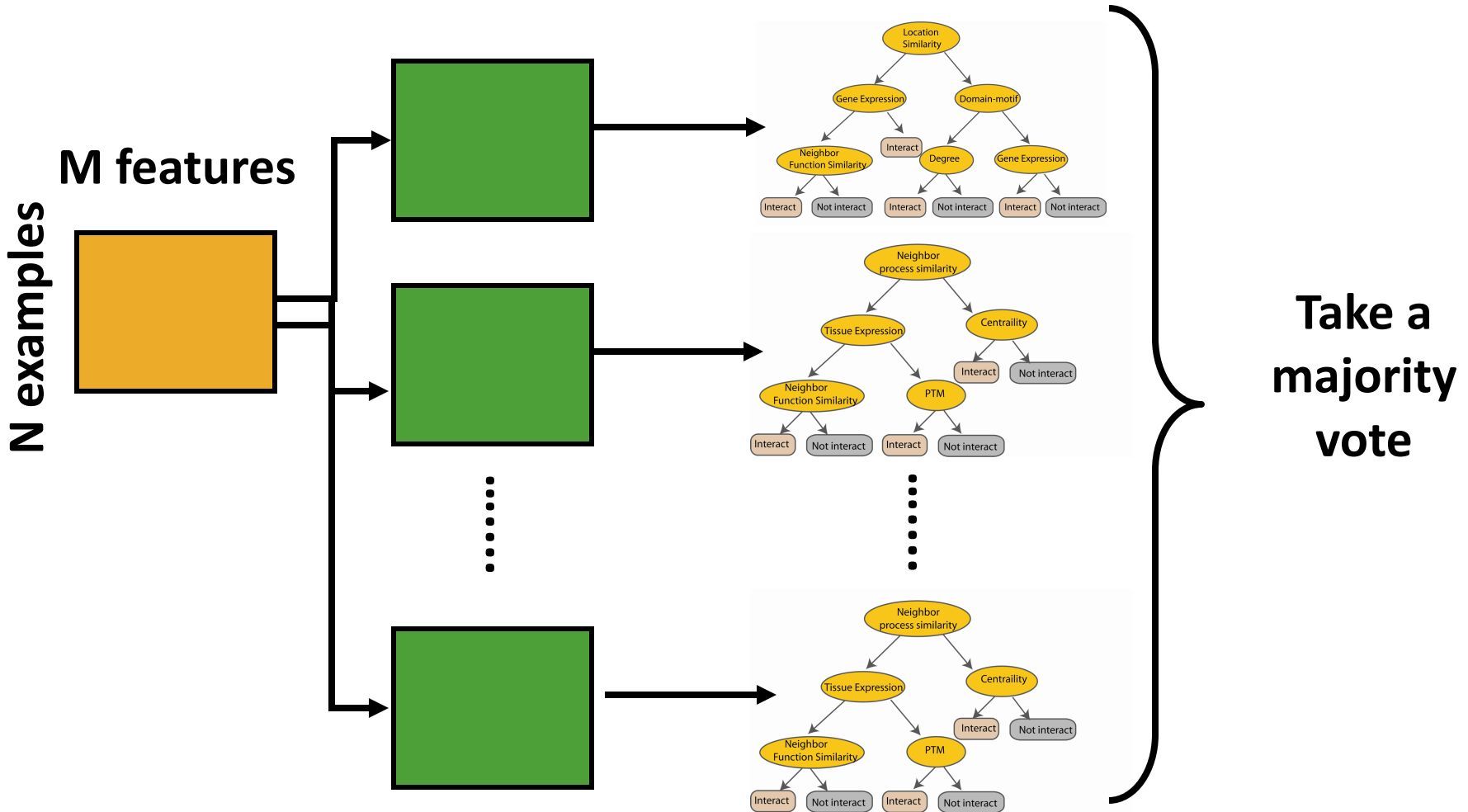
# Random forests



# Random forests



# Random forests



# **Data Mining**

## **Classification: Alternative Techniques**

---

Imbalanced Class Problem

Introduction to Data Mining

Fang Zhou

# Class Imbalance Problem

---

- Lots of classification problems where the classes are skewed (more records from one class than another)
  - Credit card fraud
  - Intrusion detection
  - Defective products in manufacturing assembly line

# Challenges

---

- Evaluation measures such as accuracy is not well-suited for imbalanced class
- Detecting the rare class is like finding needle in a haystack

# Outline

---

- Alternative metrics
- Cost-sensitive learning
- Sampling-based methods

# Confusion Matrix

- Confusion Matrix:

		Predicted Class	
		Class = +	Class = -
Actual Class	Class = +	a	b
	Class = -	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)



# Accuracy

		Predicted Class	
		Class = +	Class = -
Actual Class	Class = +	a (TP)	b (FN)
	Class = -	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Problem with Accuracy

---

- Consider a 2-class problem
  - Number of Class - examples = 990
  - Number of Class + examples = 10
- If a model predicts everything to be class -, accuracy is  $990/1000 = 99\%$ 
  - This is misleading because the model does not detect any class + example

The accuracy measure treats every class as equally important, so it may not be suitable for analyzing imbalanced data sets.

# Alternative Measures

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	TP	FN
	Class = -	FP	TN

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

$$\text{Recall, } r = \frac{TP}{TP + FN} \quad (\text{True positive rate})$$

$$F_1 = \frac{2rp}{r + p} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}}$$

# Alternative Measures

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	10	0
	Class = -	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2 * 1 * 0.5}{1 + 0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

# Alternative Measures

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	10	0
	Class = -	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2*1*0.5}{1+0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	1	9
	Class = -	0	990

$$\text{Precision (p)} = \frac{1}{1+0} = 1$$

$$\text{Recall (r)} = \frac{1}{1+9} = 0.1$$

$$\text{F - measure (F)} = \frac{2*0.1*1}{1+0.1} = 0.18$$

$$\text{Accuracy} = \frac{991}{1000} = 0.991$$

# Alternative Measures

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	40	10
	Class = -	10	40

Precision (p) = 0.8

Recall (r) = 0.8

F - measure (F) = 0.8

Accuracy = 0.8

# Alternative Measures

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	40	10
	Class = -	10	40

Precision (p) = 0.8

Recall (r) = 0.8

F - measure (F) = 0.8

Accuracy = 0.8

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	40	10
	Class = -	1000	4000

Precision (p) = ~ 0.04

Recall (r) = 0.8

F - measure (F) = ~ 0.08

Accuracy = ~ 0.8

# Measures of Classification Performance

		Predicted class	
		Class = +	Class = -
Actual Class	Class = +	TP	FN
	Class = -	FP	TN

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$ErrorRate = 1 - accuracy$$

$$Precision = \text{Positive Predictive Value} = \frac{TP}{TP + FP}$$

$$Recall = \text{Sensitivity} = TP \text{ Rate} = \frac{TP}{TP + FN}$$

$$Specificity = TN \text{ Rate} = \frac{TN}{TN + FP}$$

$$FP \text{ Rate} = \alpha = \frac{FP}{TN + FP} = 1 - specificity$$

$$FN \text{ Rate} = \beta = \frac{FN}{FN + TP} = 1 - sensitivity$$

$$Power = sensitivity = 1 - \beta$$



# Handling Class Imbalanced Problem

---

- Cost-sensitive classification
  - Misclassifying rare class as majority class is more expensive than misclassifying majority as rare class
- Sampling-based approaches

# Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class= +	Class= -
	Class= +	$f(+, +)$	$f(+, -)$
	Class= -	$f(-, +)$	$f(-, -)$

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	$C(i, j)$	Class= +	Class= -
	Class= +	$C(+, +)$	$C(+, -)$ False negative
	Class= -	$C(-, +)$ False alarm	$C(-, -)$

$C(i,j)$ : Cost of misclassifying class  $i$  example as class  $j$

$$\text{Cost} = \sum C(i, j) \times f(i, j)$$

# Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
	C(i,j)	+	-
	+	-1	100
	-	1	0

Model $M_1$	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%

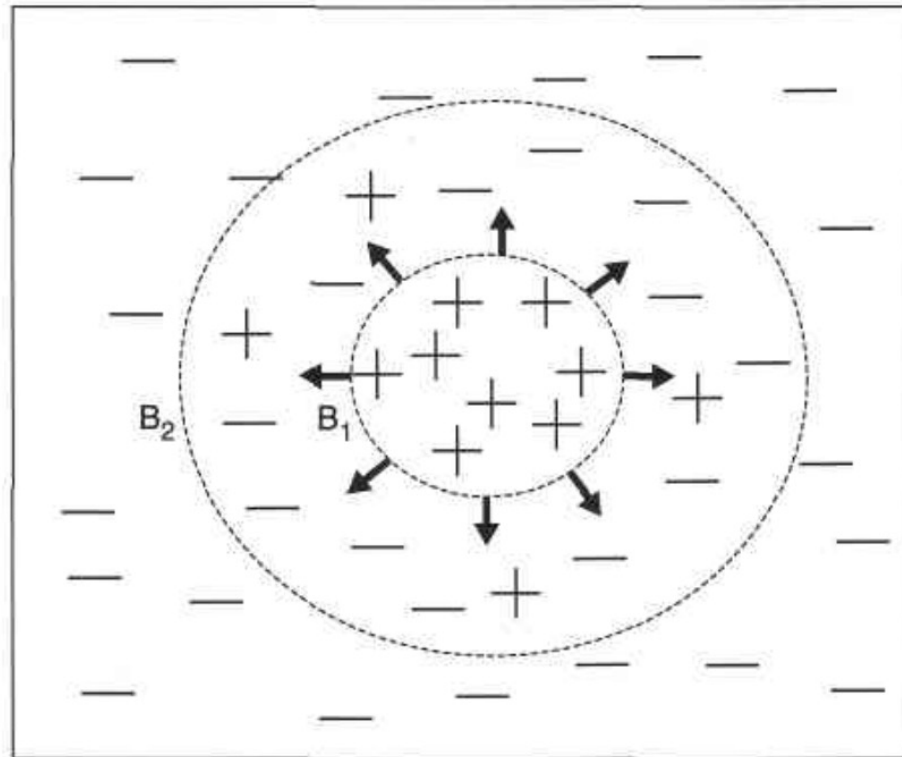
Cost = 3910

Model $M_2$	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

# Cost Sensitive Classification



**Figure 5.44.** Modifying the decision boundary (from  $B_1$  to  $B_2$ ) to reduce the false negative errors of a classifier.

# Cost Sensitive Classification

---

- A typical decision rule:
  - Assign the positive class to node  $t$

$$\begin{aligned} & p(+|t) > p(-|t) \\ \implies & p(+|t) > (1 - p(+|t)) \\ \implies & 2p(+|t) > 1 \\ \implies & p(+|t) > 0.5. \end{aligned}$$

**The misclassification costs are identical for both positive and negative examples**

# Cost Sensitive Classification

---

- Given the cost matrix, an example should be classified into the class that has the minimum expected cost.
- a cost-sensitive algorithm assigns the class label  $i$  to node  $t$  if it minimizes the following expression:

$$C(i|t) = \sum_j P(j|t)C(j, i)$$

$P(i|t)$  denotes the fraction of training records from class  $i$  that belong to the leaf node  $t$ .

# Cost Sensitive Classification

---

In the case where  $C(+,+) = C(-,-) = 0$ ,  
a leaf node  $t$  is assigned to the positive class if :

$$\begin{aligned} & p(+|t)C(+, -) > p(-|t)C(-, +) \\ \Rightarrow & p(+|t)C(+, -) > (1 - p(+|t))C(-, +) \\ \Rightarrow & p(+|t) > \frac{C(-, +)}{C(-, +) + C(+, -)}. \end{aligned}$$

If  $C(-, +) < C(+, -)$ , the threshold will be less than 0.5

# Sampling-based Approaches

---

- Modify the distribution of training data so that rare class is well-represented in training set
  - Undersample the majority class
    - ◆ Some useful negative examples may not be chosen for training
  - Oversample the rare class
    - ◆ Oversampling may cause model overfitting
    - ◆ The additional positive examples tend to increase the computation time for model building



# Summary

---

- Classification: Model construction from a set of training data
- Effective and scalable methods
  - Decision tree induction, (Bayes classification methods, linear classifier, ... )
  - No single method has been found to be superior over all others for all data sets
- Evaluation metrics: Accuracy, sensitivity, specificity, precision, recall,  $F$  measure