Classification

UE 141 Spring 2013

Jing Gao
SUNY Buffalo

Classification

features **class labels**

patient	temp.	blood pres.	heart rate	Sick?
	99	110	90	Yes
	100	120	100	Yes
	96	130	65	No

labeled

training

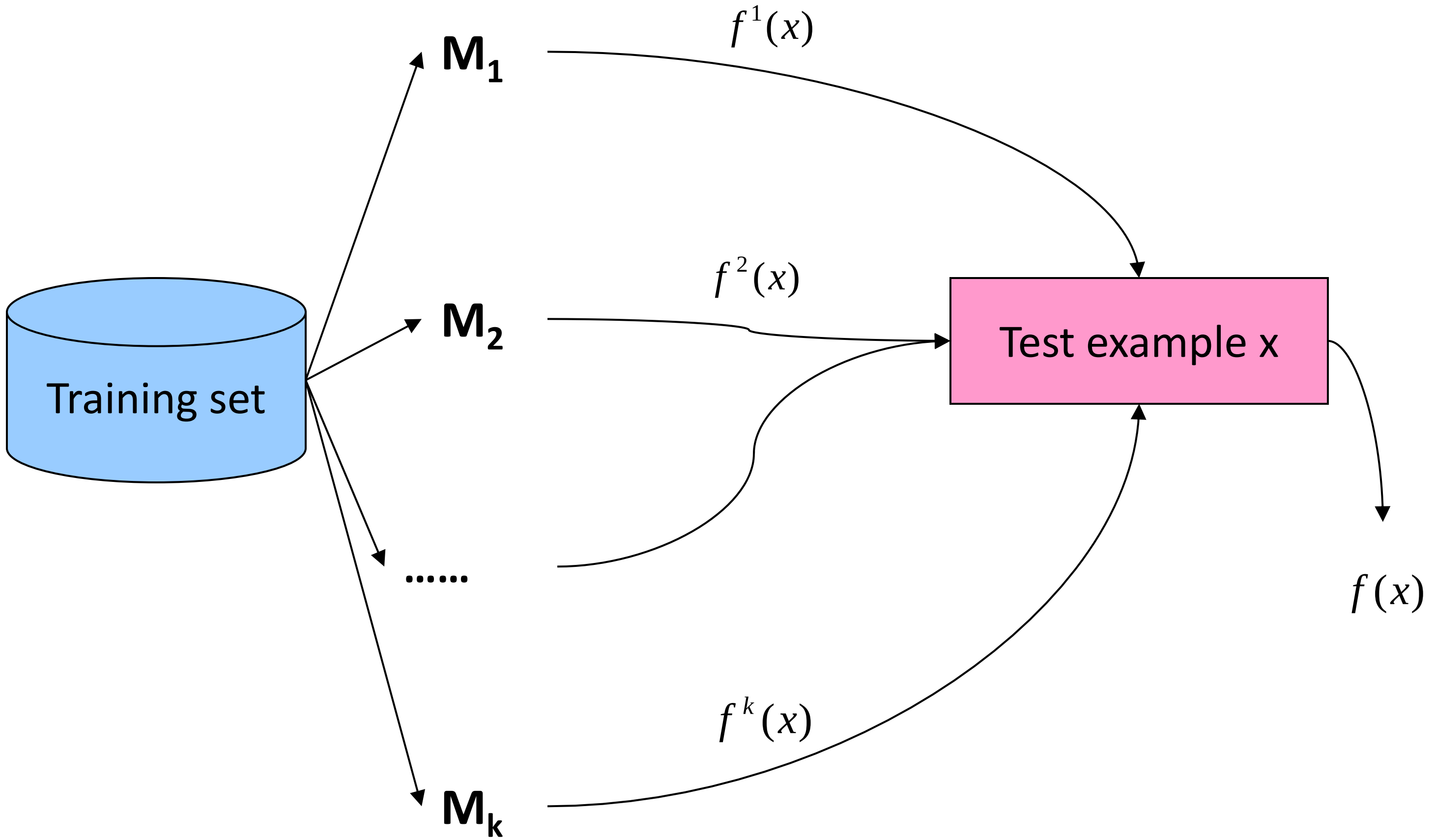
a model: f(x)=y: features → class labels

patient	temp.	blood pres.	heart rate	Sick?
	98	130	80	
	115	110	95	

test

unlabeled

Ensemble Learning



Ensemble Learning

- **Problem**
  - Given a data set $D=\{x_1,x_2,\ldots,x_n\}$ and their corresponding labels $L=\{l_1,l_2,\ldots,l_n\}$
  - An ensemble approach computes:
    - A set of classifiers $\{f_1,f_2,\ldots,f_k\}$, each of which maps data to a class label: $f_j(x)=l$
    - A combination of classifiers f^* based on $\{f_1,f_2,\ldots,f_k\}$

Why Ensemble Works? (1)

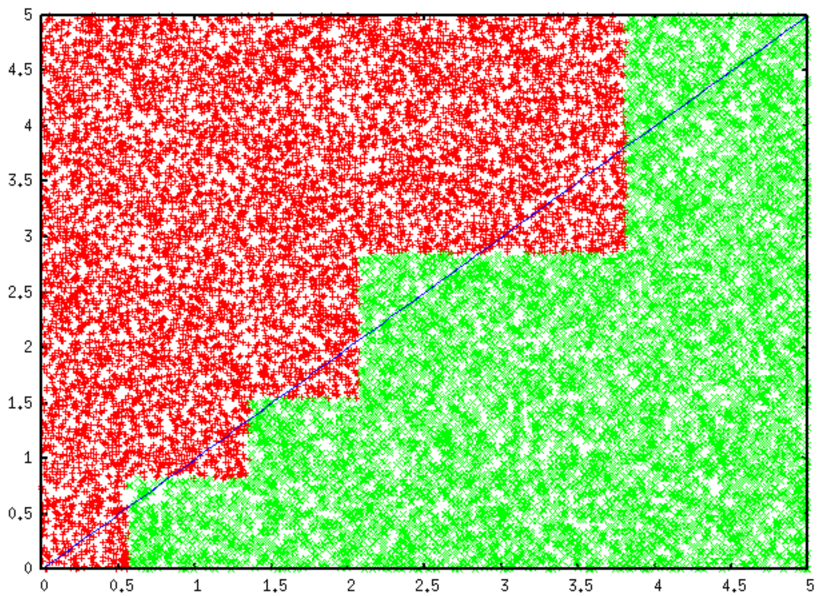
- **Intuition**
  - combining diverse, independent opinions in human decision-making as a protective mechanism (e.g. stock portfolio)
- **Stock investment**
  - Invest all the money on one stock is very risky
  - Distribute your money across multiple stocks is the best way to guarantee stable return

Why Ensemble Works? (2)

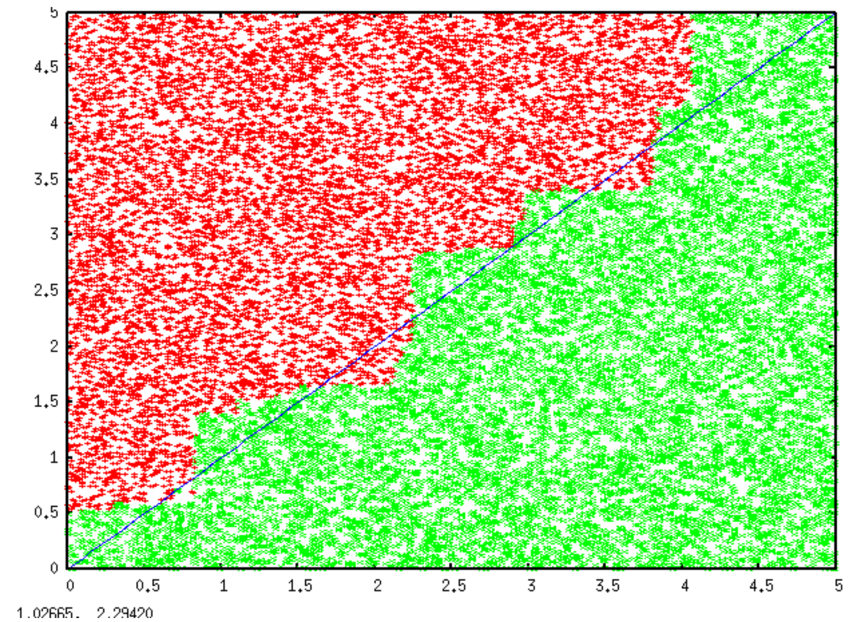
- **Uncorrelated error reduction**
  - Suppose we have 5 completely independent classifiers for majority voting
  - If accuracy is 70% for each
    - 10 (.7^3)(.3^2)+5(.7^4)(.3)+(.7^5)
    - **83.7% majority vote accuracy**
  - 101 such classifiers
    - **99.9% majority vote accuracy**

Why Ensemble Works? (3)

- **Overcome limitations of single hypothesis**
  - The target function may not be implementable with individual classifiers, but may be approximated by model averaging



Decision Tree



Model Averaging

Generating Base Classifiers

- **Sampling training examples**
  - Train k classifiers on k subsets drawn from the training set
- **Using different learning models**
  - Use all the training examples, but apply different learning algorithms
- **Sampling features**
  - Train k classifiers on k subsets of features drawn from the feature space
- **Learning “randomly”**
  - Introduce randomness into learning procedures

Bagging

- **Training set**
  - Sampling with replacement
  - Sample a subset from the training set
- **Ensemble learning**
  - Train a classifier on each sample
  - Use majority voting to determine the class label of ensemble classifier

Bagging

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

Samples and classifiers:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

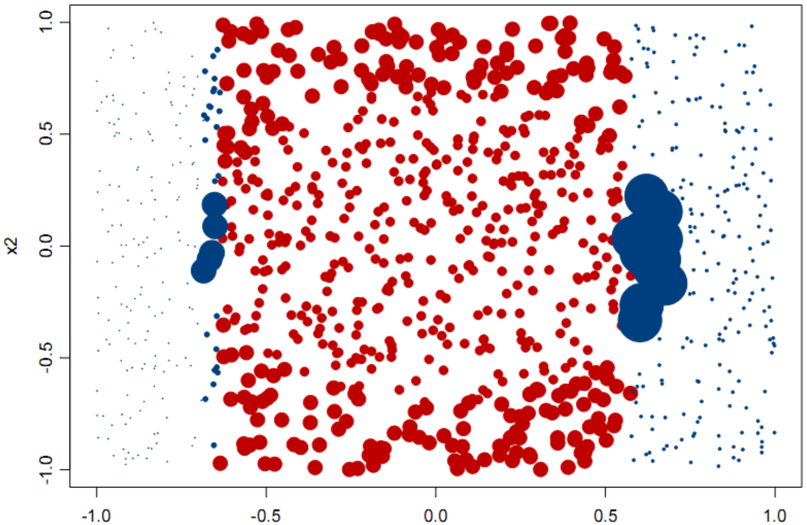
Combine predictions by majority voting

Boosting

- **Principles**
  - Boost a set of weak learners to a strong learner
  - Make records currently misclassified more important

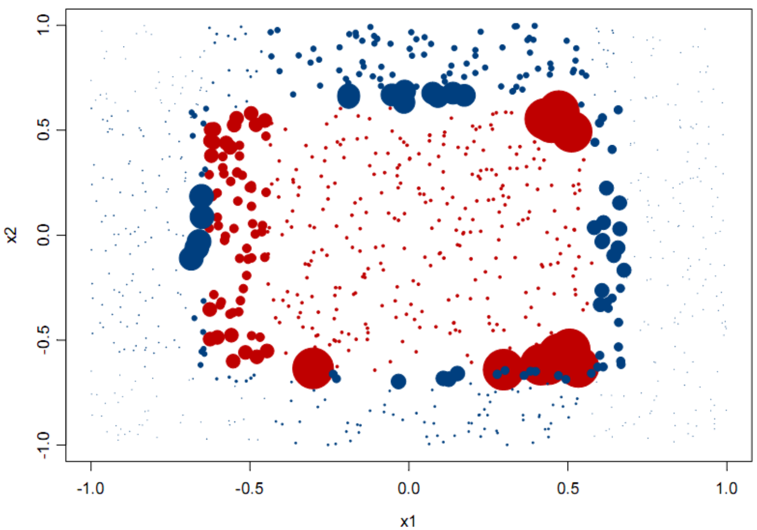
- **Example**
  - Record 4 is hard to classify
  - Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

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

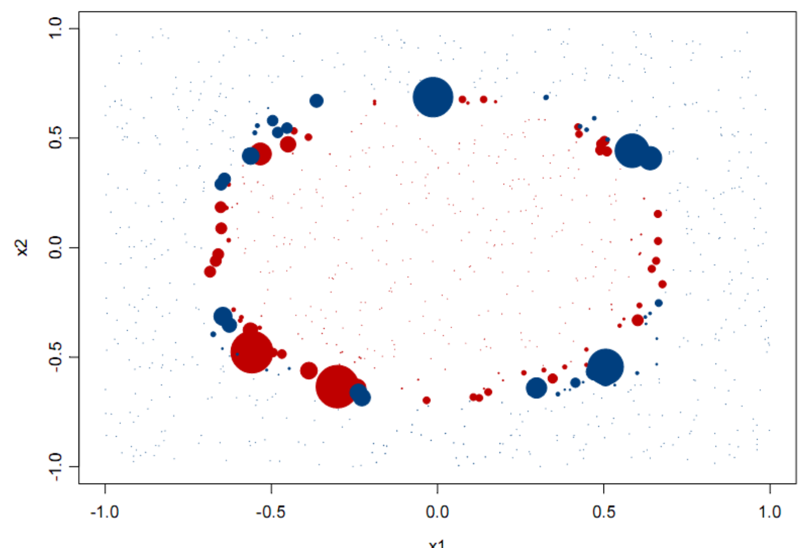


Classifications (colors) and Weights (size) after *1 iteration* of Boosting

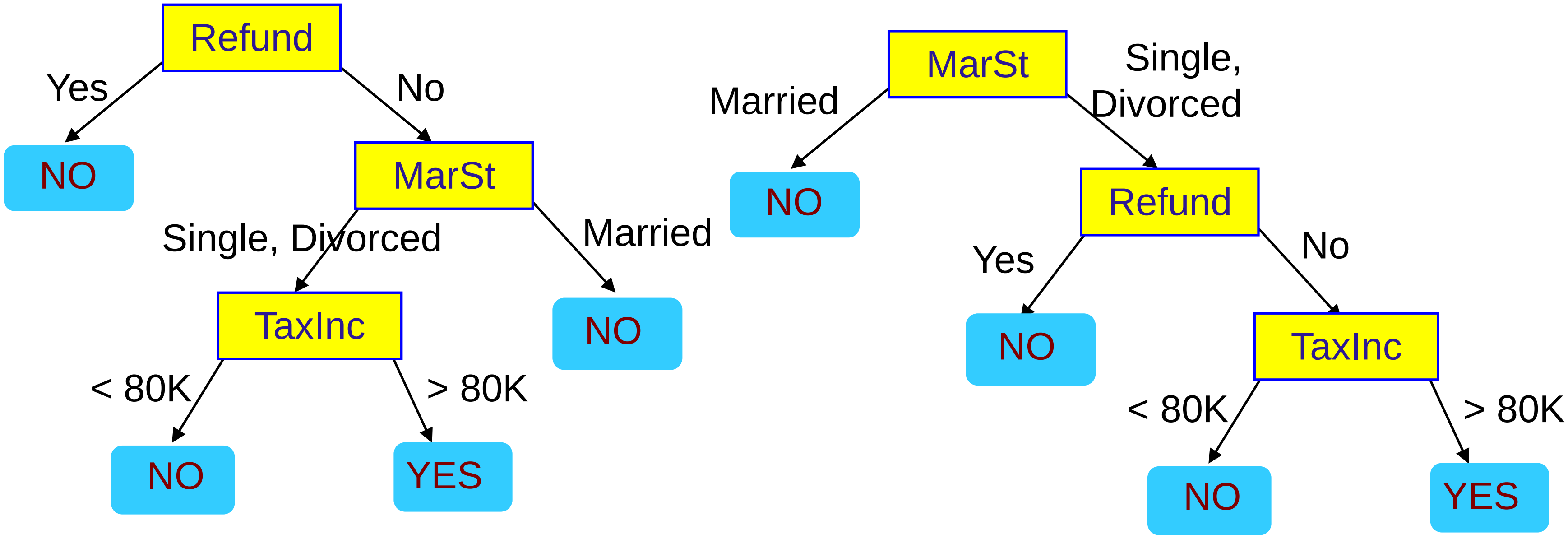
3 iterations



20 iterations



Random Forests



…… A lot more ways to build a decision tree from the data

Instead of selecting one best tree among all the trees, let's combine them!