

WHERE WE ARE

Informatics

- management, manipulation, integration
- emphasis on scale, some emphasis on tools

Analytics

- statistical estimation and prediction
- machine learning, data mining

Visualization

- communication and presentation

WHAT IS MACHINE LEARNING?

“Systems that automatically learn programs from data”

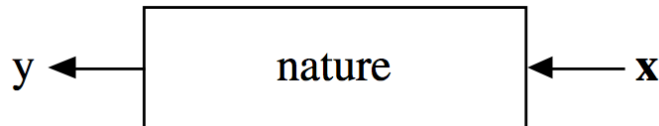
[Domingos 2012]

Teaching a computer about the world

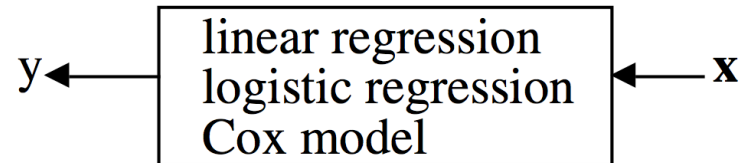
[Mark Dredze]

WHAT'S THE DIFFERENCE BETWEEN STATISTICS AND MACHINE LEARNING?

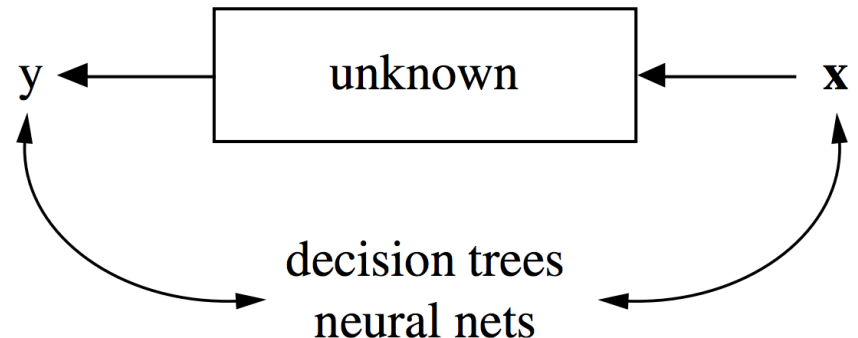
One view:



Emphasis on stochastic models of nature:



Find a function that predicts y from x : no model of nature implied or needed



TOY EXAMPLE

Goal: Predict when we play

hypothesis:
we only play
when its
sunny?

outlook	temperature	humidity	windy	PLAY?
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

No

No

hypothesis: we don't play if its rainy and windy?

[Witten]

TERMINOLOGY

classification

- The learned attribute is categorical (“nominal”)

regression

- The learned attribute is numeric

TERMINOLOGY

Supervised Learning (“Training”)

- We are given examples of inputs and associated outputs
- We learn the relationship between them

Unsupervised Learning (sometimes: “Mining”)

- We are given inputs, but no outputs
 - unlabeled data
- Learn the “latent” labels
- Ex: Clustering, dimension reduction

EXAMPLE: DOCUMENT CLASSIFICATION

“The Falcons trounced the Saints on Sunday”

Sports

“The Mars Rover discovered organic molecules on Sunday”

Science

How do we set this up?

What are the rows and columns of our decision table?

EXAMPLE: CONSTRUCTING THE DOCUMENT MATRIX

d1 : Romeo and Juliet.

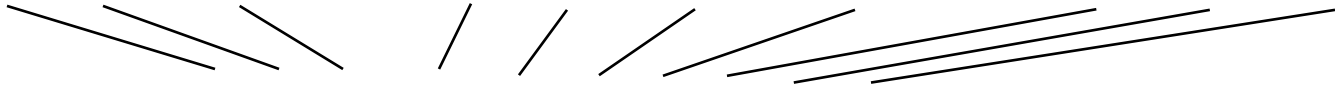
d2 : Juliet: O happy dagger!

d3 : Romeo died by dagger.

d4 : “Live free or die”, that’s the New-Hampshire’s motto.

d5 : Did you know, New-Hampshire is in New-England.

dagger die new-hampshir free happi live new-england motto romeo juliet



d1	[0, 0, 0, 0, 0, 0, 0, 0, 1, 1]
d2	[1, 0, 0, 0, 1, 0, 0, 0, 0, 1]
d3	[1, 1, 0, 0, 0, 0, 0, 0, 1, 0]
d4	[0, 1, 1, 1, 0, 1, 0, 1, 0, 0]
d5	[0, 0, 1, 0, 0, 0, 1, 0, 0, 0]

EXAMPLE: DOCUMENT CLASSIFICATION

Supervised Learning Problem

- A human assigns a topic label to each document in a corpus
- The algorithm learns how to predict the label

Unsupervised Learning Problem

- No labels are given
- Discover groups of similar documents

LEARNING = THREE CORE COMPONENTS

Representation

Evaluation

Optimization

LEARNING = THREE CORE COMPONENTS

Representation

- What exactly is your classifier?
 - A hyperplane that separates the two classes?
 - A decision tree?
 - A neural network?

Evaluation

Optimization

LEARNING = THREE CORE COMPONENTS

Representation

Evaluation

- How do we know if a given classifier is good or bad?
 - # of errors on some test set?
 - Precision and recall?
 - Squared error?
 - Likelihood?

Optimization

LEARNING = THREE CORE COMPONENTS

Representation

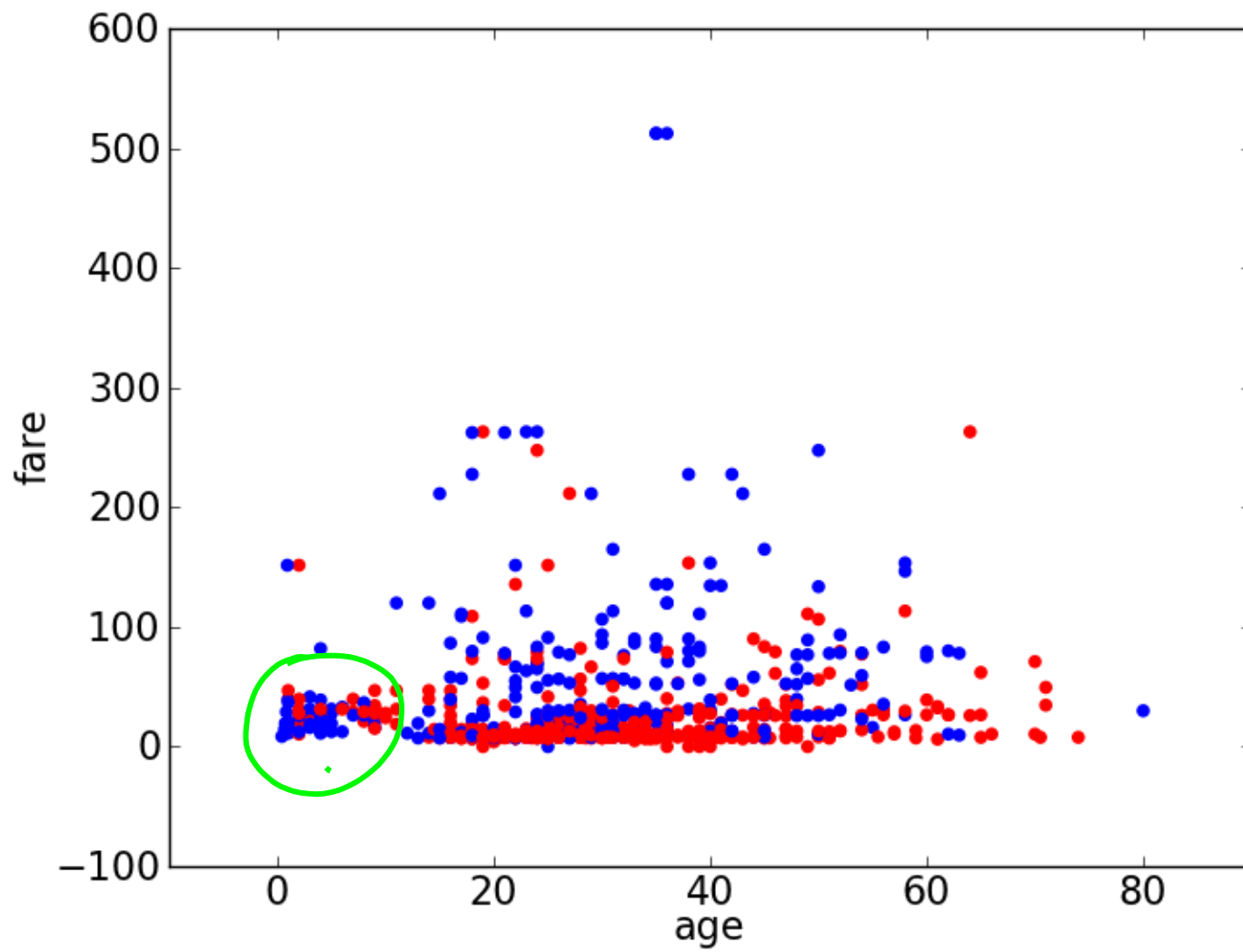
Evaluation

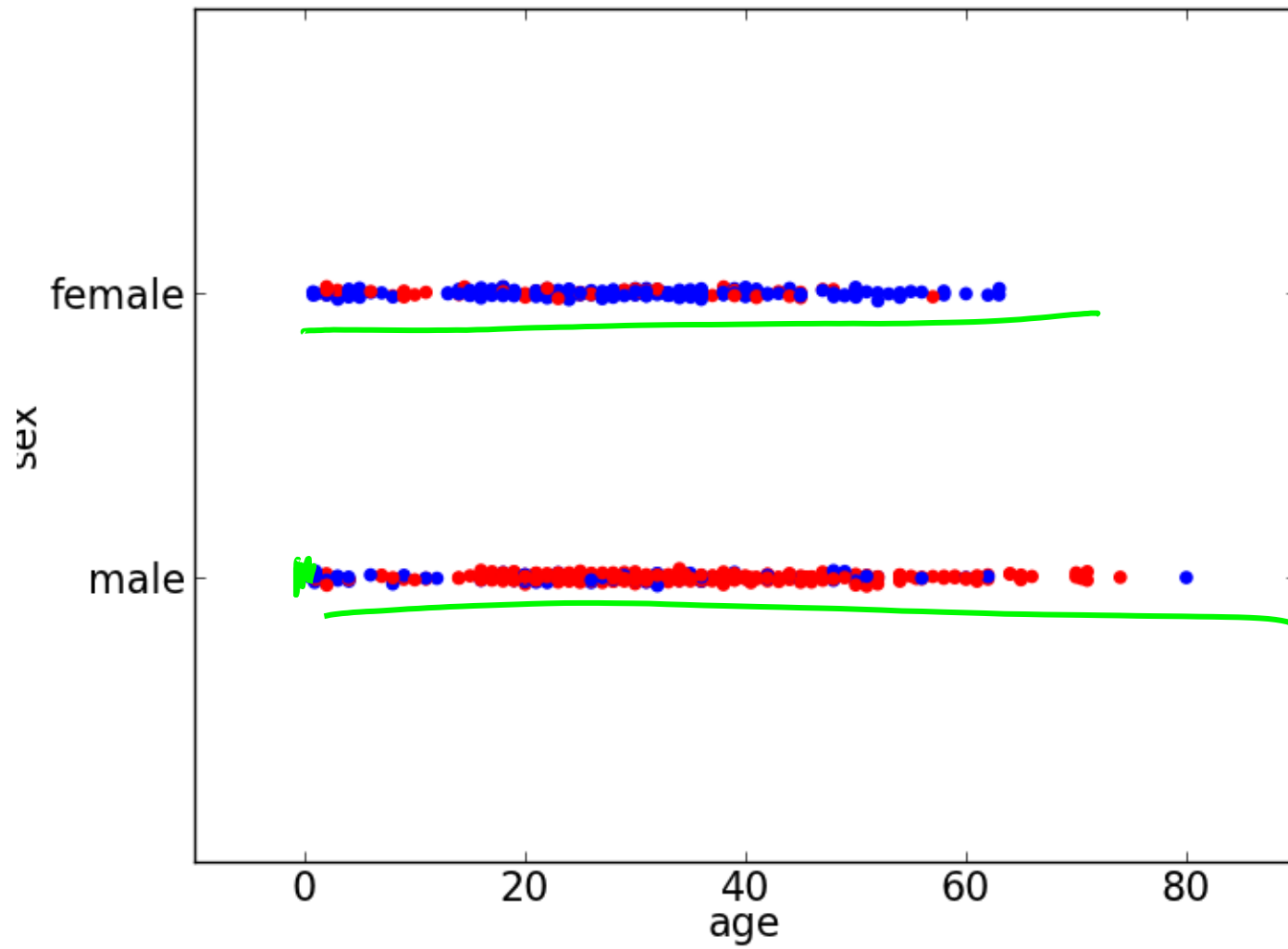
Optimization

- How do you search among all the alternatives?
 - Greedy search?
 - Gradient descent?

TITANIC DATASET

survived	pclass	sex	age	sibsp	parch	fare	cabin	embarked
0	3	male	22	1	0	7.25		S
1	1	female	38	1	0	71.2833	C85	C
1	3	female	26	0	0	7.925		S
1	1	female	35	1	0	53.1	C123	S
0	3	male	35	0	0	8.05		S
0	3	male		0	0	8.4583		Q
0	1	male	54	0	0	51.8625	E46	S
0	3	male	2	3	1	21.075		S
1	3	female	27	0	2	11.1333		S
1	2	female	14	1	0	30.0708		C
1	3	female	4	1	1	16.7	G6	S
1	1	female	58	0	0	26.55	C103	S
0	3	male	20	0	0	8.05		S





A VERY NAÏVE CLASSIFIER

pclass	sex	age	sibsp	parch	fare	cabin	embarked
1	female	35	1	0	53.1	C123	S

Does the new data point x^* **exactly** match a previous point x_i ?

If so, assign it to the same class as x_i

Otherwise, just guess.

This is the “rote” classifier

A MINOR IMPROVEMENT

pclass	sex	age	sibsp	parch	fare	cabin	embarked
1	female	35	1	0	53.1	C123	S

Does the new data point x^* match a set of previous points x_i on some specific attribute?

If so, take a vote to determine class.

Example: If most females survived, then assume every female survives

But there are lots of possible rules like this.
And an attribute can have more than two values.

If most people under 4 years old survive, then assume everyone under 4 survives
If most people with 1 sibling survive, then assume everyone with 1 sibling survives

How do we choose?

IF sex = 'female' THEN survive = yes

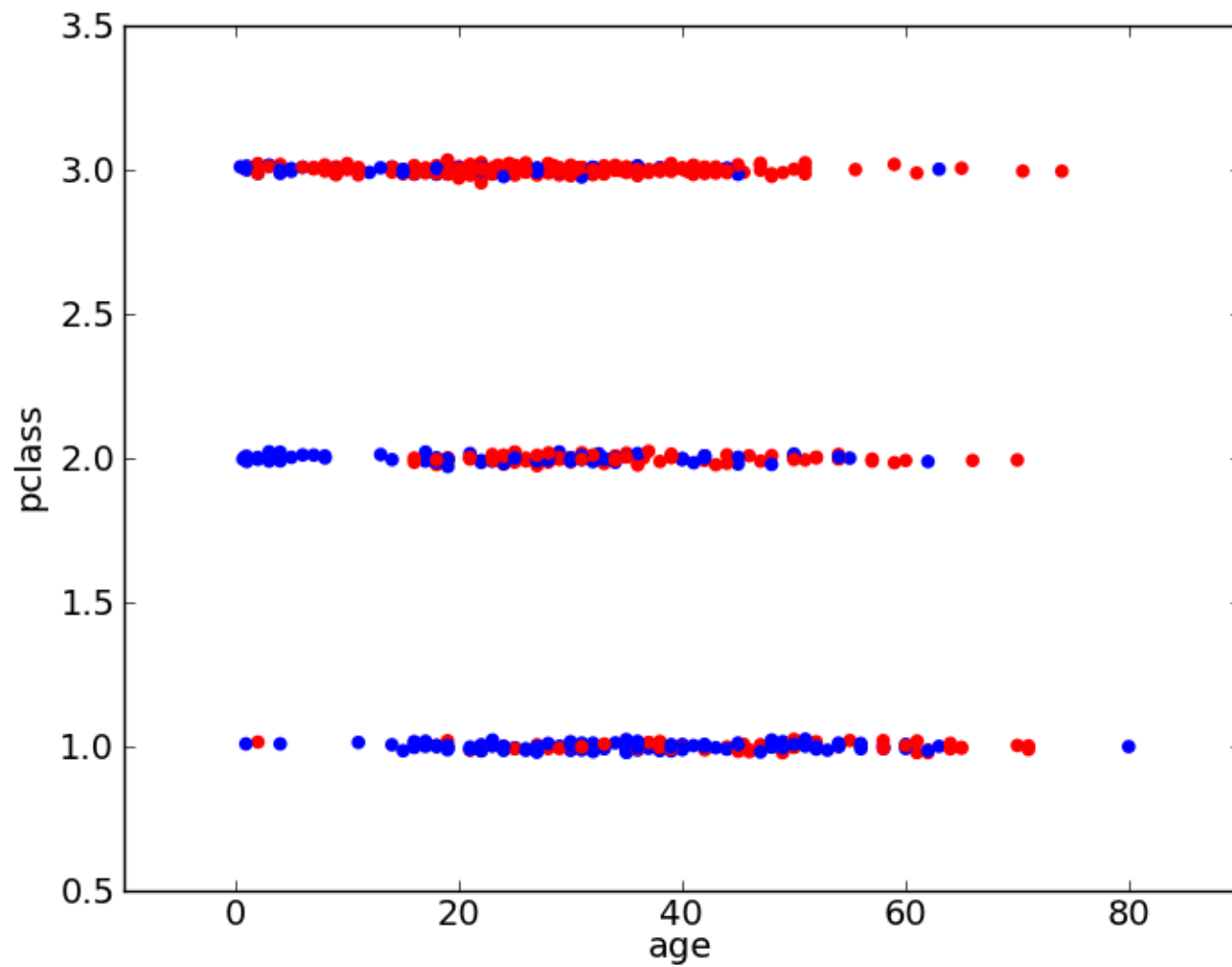
ELSE IF sex = 'male' THEN survive = no

confusion matrix

no	yes		<-- classified as
468	109		no
81	233		yes

$$\frac{468 + 233}{468 + 109 + 81 + 233} = 79\% \text{ correct (and 21\% incorrect)}$$

Not bad!



IF pclass='1' **THEN** survive=yes
ELSE IF pclass='2' **THEN** survive=yes
ELSE IF pclass='3' **THEN** survive=no

confusion matrix

no	yes		<-- classified as
372	119		no
177	223		yes

$$\frac{372 + 223}{372 + 119 + 223 + 177} = 67\% \text{ correct (and 33\% incorrect)}$$

a bit worse...

1-RULE

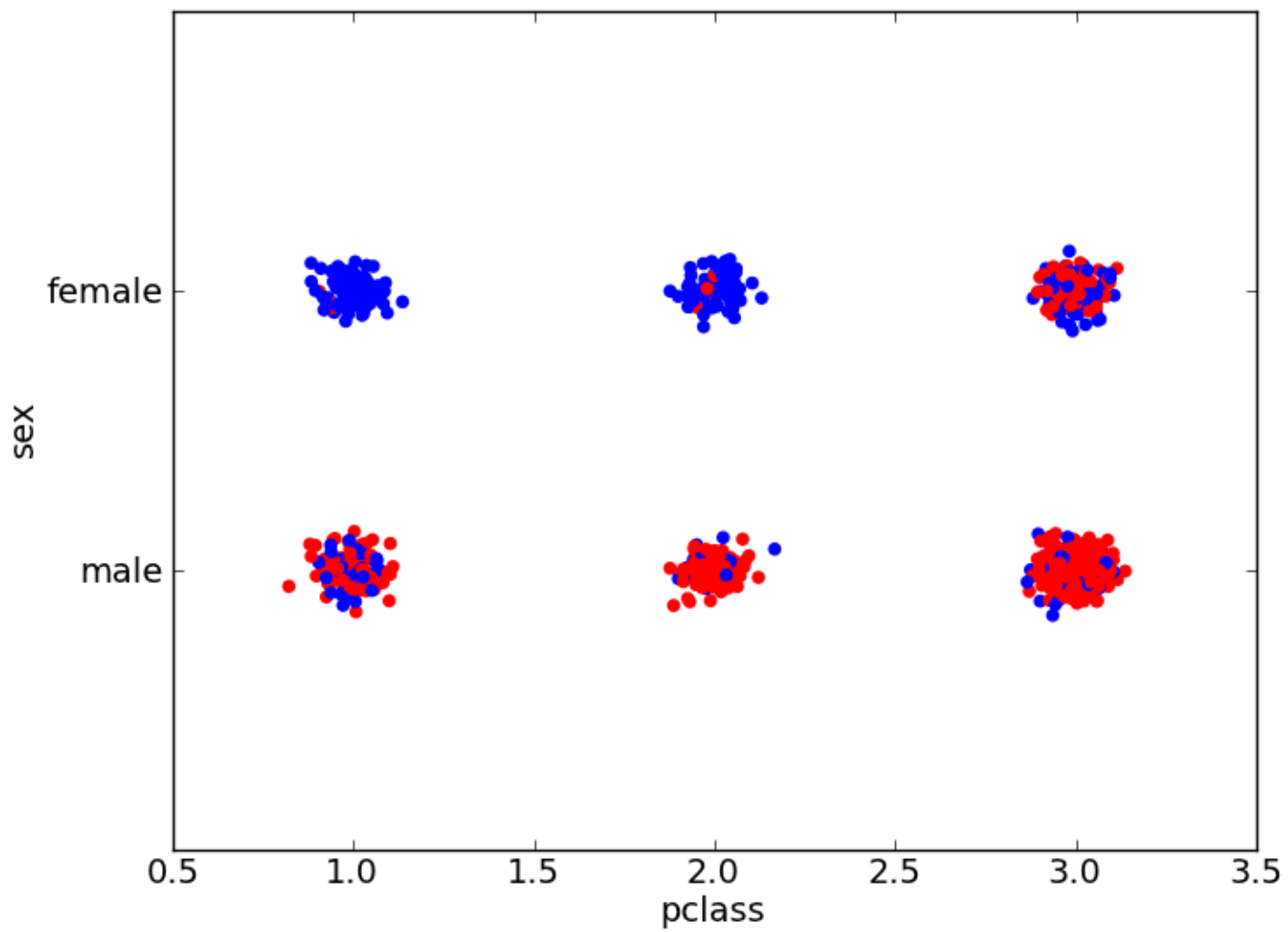
For each attribute A:

For each value V of that attribute, create a rule:

1. count how often each class appears
2. find the most frequent class, c
3. make a rule "if A=V then Class=c"

Calculate the error rate of this rule

Pick the attribute whose rules produce the lowest error rate



HOW FAR CAN WE GO?

```
IF pclass='1' AND sex='female' THEN survive=yes
IF pclass='2' AND sex='female' THEN survive=yes
IF pclass='3' AND sex='female' AND age < 4 THEN survive=yes
IF pclass='3' AND sex='female' AND age >= 4 THEN survive=no
IF pclass='2' AND sex='male' THEN survive=no
IF pclass='3' AND sex='male' THEN survive=no
IF pclass='1' AND sex='male' AND age < 5 THEN survive=yes
...
```


SEQUENTIAL COVERING

Initialize R to the empty set

for each class C {

 while **D** is nonempty {

 Construct one rule r that correctly classifies
 some instances in D that belong to class C and
 does not incorrectly classify any non-C instances

 Add rule r to ruleset R

 Remove from D all instances correctly classified by r

 }

}

return R

SEQUENTIAL COVERING: FINDING NEXT RULE FOR CLASS C

Initialize A as the set of all attributes over D

while r incorrectly classifies some non-C instances of D {

 write r as $\text{ant}(r) \Rightarrow C$

 for each attribute-value pair $(a=v)$,
 where a belongs to A and v is a value of a,
 compute the accuracy of the rule

$\text{ant}(r) \text{ and } (a=v) \Rightarrow C$

 let $(a^*=v^*)$ be the attribute-value pair of
 maximum accuracy over D; in case of a tie,
 choose the pair that covers the greatest
 number of instances of D

 update r by adding $(a^*=v^*)$ to its antecedent:

$r = (\text{ant}(r) \text{ and } (a^*=v^*)) \Rightarrow C$

 remove the attribute a^* from the set A:

$A = A - \{a^*\}$

}

src: Alvarez

STRATEGIES FOR LEARNING EACH RULE

General-to-Specific

- Start with an empty rule
- Add constraints to eliminate negative examples
- Stop when only positives are covered

Specific-to-General (not shown)

- Start with a rule that identifies a single random instance
- Remove constraints in order to cover more positives
- Stop when further generalization results in covering negatives

CONFLICTS

If more than one rule is triggered

- Choose the “most specific” rule
- Use domain knowledge to order rules by priority

RECAP

Representation

- A set of rules: IF...THEN conditions

Evaluation

- coverage: # of data points that satisfy conditions
- accuracy = # of correct predictions / coverage

Optimization

- Build rules by finding conditions that maximize accuracy

One rule is easy to interpret, but a complex set of rules probably isn't

HOW FAR CAN WE GO?

We might consider grouping redundant conditions

IF **pclass**='1' THEN

 IF **sex**='female' THEN **survive**=yes

 IF **sex**='male' AND **age** < 5 THEN **survive**=yes

IF **pclass**='2'

 IF **sex**='female' THEN **survive**=yes

 IF **sex**='male' THEN **survive**=no

IF **pclass**='3'

 IF **sex**='male' THEN **survive**=no

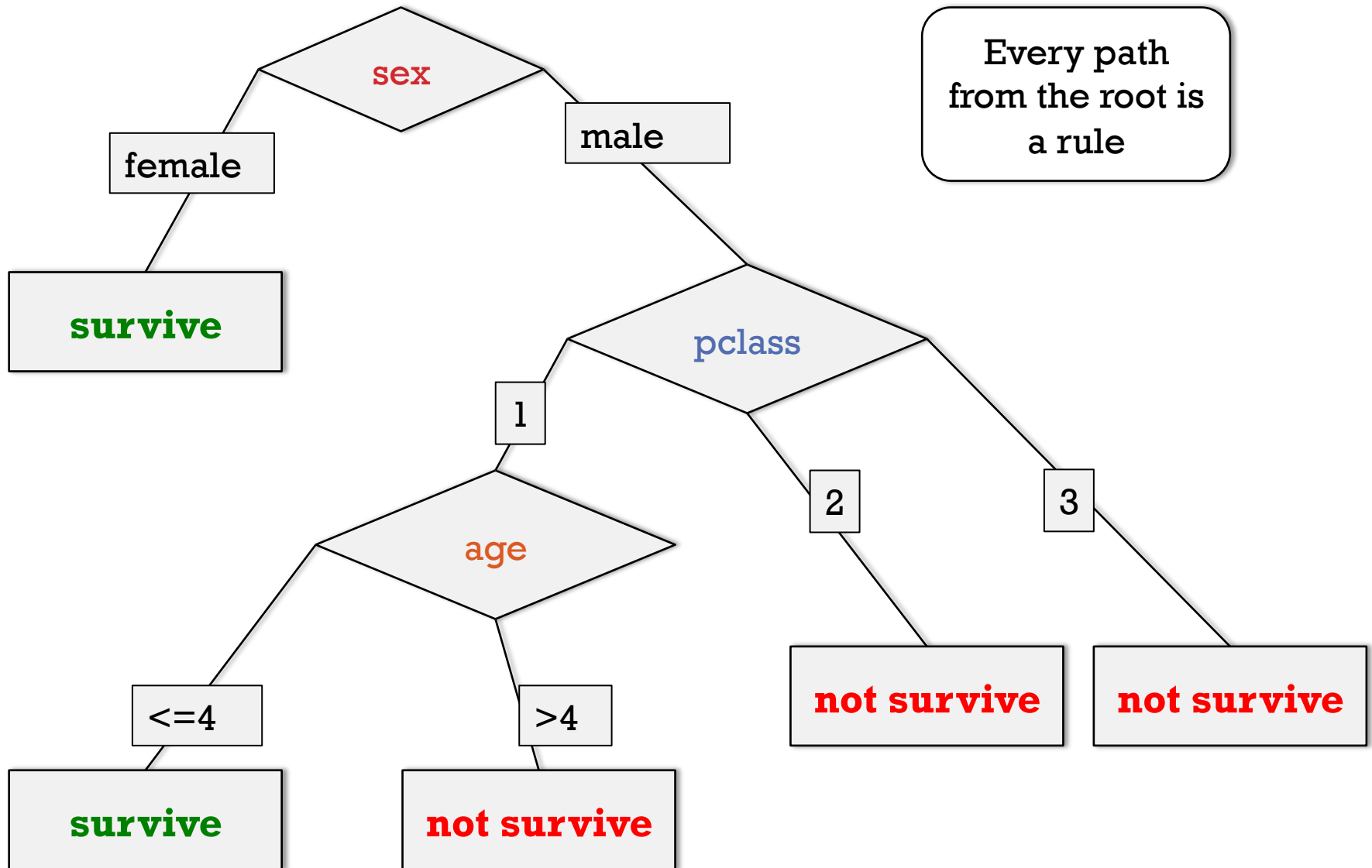
 IF **sex**='female'

 IF **age** < 4 THEN **survive**=yes

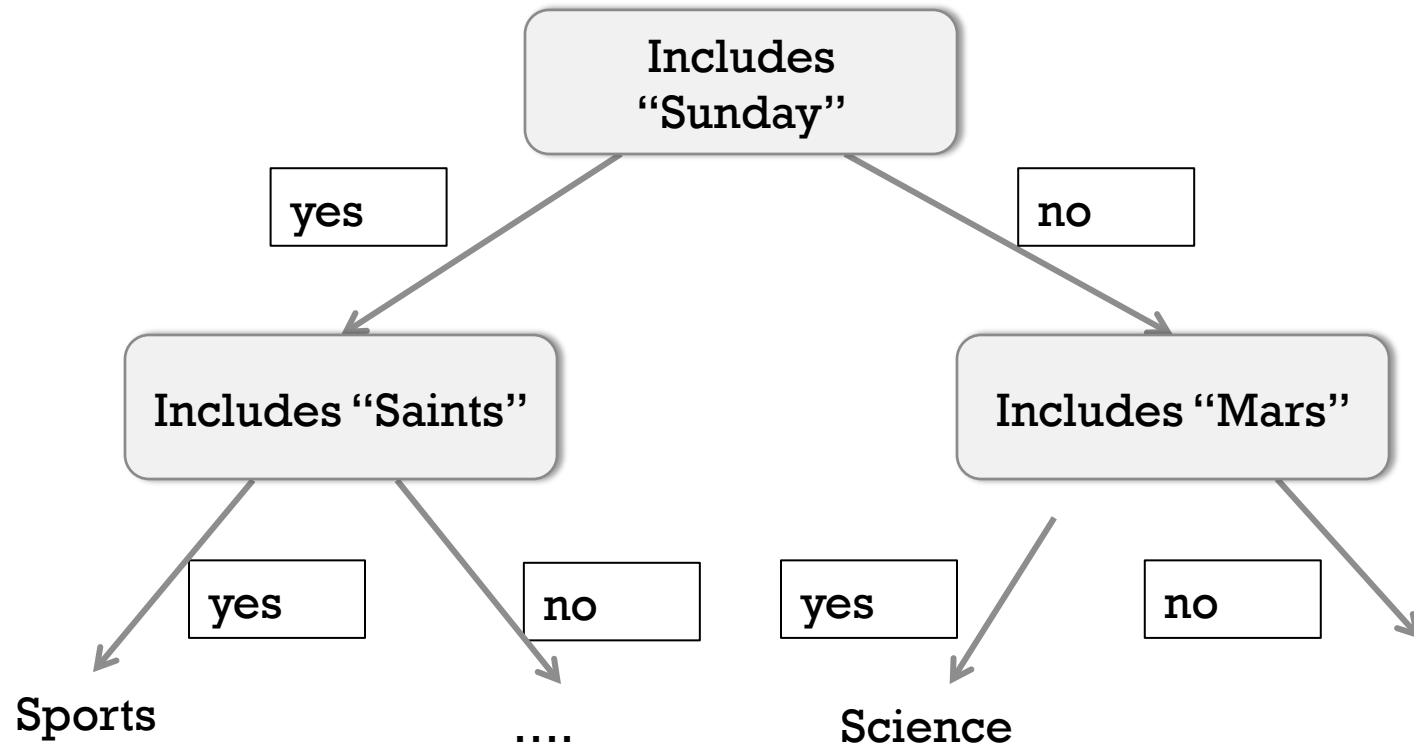
 IF **age** >= 4 THEN **survive**=no

A decision tree

HOW FAR CAN WE GO?



EXAMPLE: DOCUMENT CLASSIFICATION



ASIDE ON ENTROPY

Consider two sequences of coin flips:

HHTHTTHHHHTTHTHTHTTTT....

TTHHTTHTHTTTTTHHHHTHTTT....

How much information do we get after flipping each coin once?

We want some function “Information” that satisfies:

$$\mathbf{Information}_{1\&2}(\mathbf{p}_1\mathbf{p}_2) = \mathbf{Information}_1(\mathbf{p}_1) + \mathbf{Information}_2(\mathbf{p}_2)$$

$$I(X) = \log_2 p_x$$

Expected Information = “Entropy”

$$H(X) = E(I(X)) = \sum_x p_x I(x) = - \sum_x p_x \log_2 p_x$$

EXAMPLE: FLIPPING A COIN

$$\begin{aligned}\text{Entropy} &= - \sum_i p_x \log_2 p_x \\ &= -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) \\ &= 1\end{aligned}$$

EXAMPLE: ROLLING A DIE

$$p_1 = \frac{1}{6}, p_2 = \frac{1}{6}, p_3 = \frac{1}{6}, \dots$$

$$\begin{aligned}\text{Entropy} &= - \sum_i p_i \log_2 p_i \\ &= -6 \times \left(\frac{1}{6} \log_2 \frac{1}{6} \right) \\ &\approx 2.58\end{aligned}$$

EXAMPLE: ROLLING A WEIGHTED DIE

$$p_1 = 0.1, p_2 = 0.1, p_3 = 0.1, \dots \quad p_6 = 0.5$$

$$\begin{aligned}\text{Entropy} &= - \sum_i p_x \log_2 p_x \\ &= -5 \times (0.1 \log_2 0.1) - 0.5 \log_2 0.5 \\ &= 2.16\end{aligned}$$

The weighted die is **more unpredictable** than a fair die

HOW UNPREDICTABLE IS YOUR DATA?

342/891 survivors in titanic training set

$$- \left(\frac{342}{891} \log_2 \frac{342}{891} + \frac{549}{891} \log_2 \frac{549}{891} \right) = 0.96$$

Say there were only 50 survivors

$$- \left(\frac{50}{891} \log_2 \frac{50}{891} + \frac{841}{891} \log_2 \frac{841}{891} \right) = 0.31$$

BACK TO DECISION TREES

Which attribute do we choose at each level?

The one with the highest **information gain**

- The one that reduces the unpredictability the most

Before: 14 records, 9 are “yes”

$$-\left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14}\right) = 0.94$$

If we choose **outlook**:

overcast : 4 records, 4 are “yes”

$$-\left(\frac{4}{4} \log_2 \frac{4}{4}\right) = 0$$

rainy : 5 records, 3 are “yes”

$$-\left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5}\right) = 0.97$$

sunny : 5 records, 2 are “yes”

$$-\left(\frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5}\right) = 0.97$$

Expected new entropy:

$$\frac{4}{14} \times 0.0 + \frac{5}{14} \times 0.97 + \frac{5}{14} \times 0.97$$

$$= 0.69$$

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
rainy	cool	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

Before: 14 records, 9 are “yes”

$$-\left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14}\right) = 0.94$$

If we choose **temperature**:
cool: 4 records, 3 are “yes”

$$= 0.81$$

rainy : 4 records, 2 are “yes”

$$= 1.0$$

sunny : 6 records, 4 are “yes”

$$= 0.92$$

Expected new entropy:

$$0.81(4/14) + 1.0(4/14) + 0.92(6/14)$$

$$= 0.91$$

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
rainy	cool	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

Before: 14 records, 9 are “yes”

$$-\left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14}\right) = 0.94$$

If we choose **humidity**:

normal: 7 records, 6 are “yes”

$$= 0.59$$

high: 7 records, 2 are “yes”

$$= 0.86$$

Expected new entropy:

$$0.59(7/14) + 0.86(7/14)$$

$$= 0.725$$

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
rainy	cool	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

Before: 14 records, 9 are “yes”

$$-\left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14}\right) = 0.94$$

If we choose **windy**:

TRUE: 8 records, 6 are “yes”

$$= 0.81$$

FALSE: 5 records, 3 are “yes”

$$= 0.97$$

Expected new entropy:

$$0.81(8/14) + 0.97(6/14)$$

$$= 0.87$$

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
rainy	cool	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

Before: 14 records, 9 are “yes”

$$-\left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14}\right) = 0.94$$

outlook

$$0.94 - 0.69 = 0.25$$

highest gain

temperature

$$0.94 - 0.91 = 0.03$$

humidity

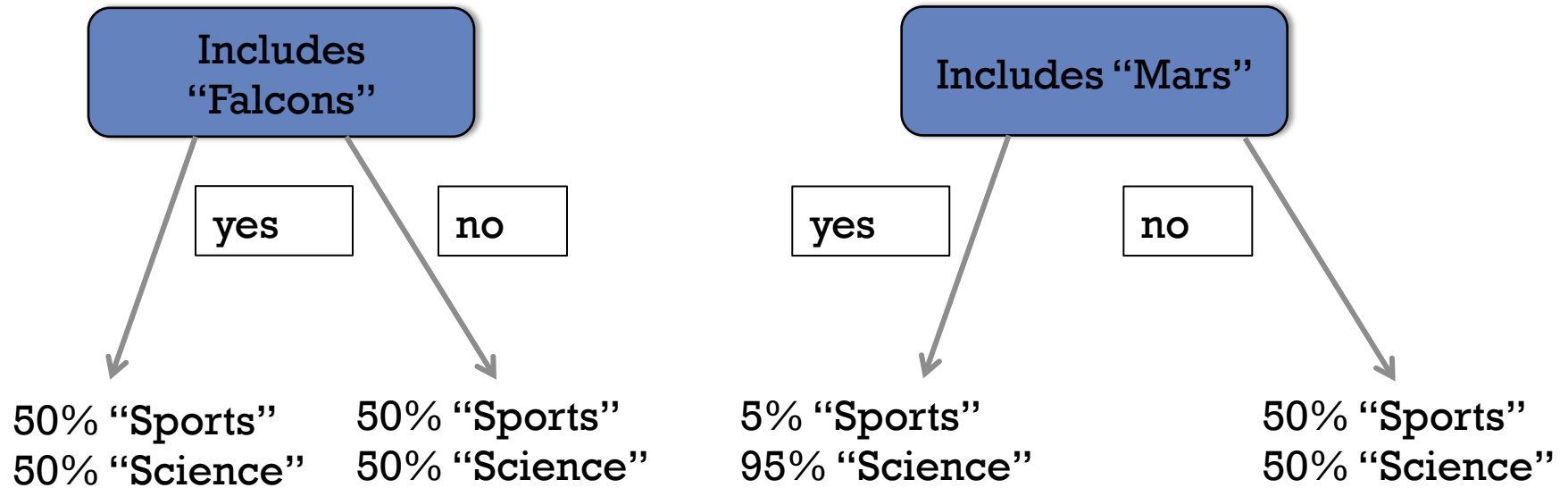
$$0.94 - 0.725 = 0.215$$

windy

$$0.94 - 0.87 = 0.07$$

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
rainy	cool	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

DOCUMENT CLASSIFICATION



BUILDING A DECISION TREE (ID3 ALGORITHM)

Assume attributes are discrete

- Discretize continuous attributes

Choose the attribute with the highest Information Gain

Create branches for each value of attribute

Examples partitioned based on selected attributes

Repeat with remaining attributes

Stopping conditions

- All examples assigned the same label
- No examples left

PROBLEMS

Expensive to train

Prone to overfitting

- Drive to perfection on training data, bad on test data
- Pruning can help: remove or aggregate subtrees that provide little discriminatory power (C45)

C4.5 EXTENSIONS

Continuous Attributes

outlook	temperature	humidity	windy	play
overcast	cool	60	TRUE	yes
overcast	hot	80	FALSE	yes
overcast	hot	63	FALSE	yes
overcast	mild	81	TRUE	yes
rainy	cool	58	TRUE	no
rainy	mild	90	TRUE	no
rainy	cool	54	FALSE	yes
rainy	mild	92	FALSE	yes
rainy	mild	59	FALSE	yes
sunny	hot	90	FALSE	no
sunny	hot	89	TRUE	no
sunny	mild	90	FALSE	no
sunny	cool	60	FALSE	yes
sunny	mild	62	TRUE	yes

Consider every possible binary partition; choose the partition with the highest gain

outlook	temperature	humidity	windy	play				
rainy	mild	54	FALSE	yes	$E(\frac{6}{6}) = 0.0$	$E(\frac{9}{10}) + E(\frac{1}{10}) = 0.47$		
overcast	hot	58	FALSE	yes				
overcast	cool	59	TRUE	yes				
rainy	cool	60	FALSE	yes				
overcast	mild	60	TRUE	yes				
overcast	hot	62	FALSE	yes	$E(\frac{3}{8}) + E(\frac{5}{9}) = 0.95$		$E(\frac{4}{4}) = 0.0$	
rainy	mild	63	TRUE	no				
sunny	cool	80	FALSE	yes				
rainy	mild	81	FALSE	yes				
sunny	mild	89	TRUE	yes				
sunny	hot	90	FALSE	no	$E(\frac{4}{4}) = 0.0$			$E(\frac{4}{4}) = 0.0$
rainy	cool	90	TRUE	no				
sunny	hot	90	TRUE	no				
sunny	mild	92	FALSE	no				

$$Expect = \frac{8}{14}(0.95) + \frac{6}{14}(0) = 0.54$$

$$Expect = \frac{10}{14}(0.47) + \frac{4}{14}(0) = 0.33$$

WHERE WE ARE

Supervised learning and classification problems

- Predict a class label based on other attributes

Rules

- To start, we guessed simple rules that might explain the data
- But the relationships are complex, so we need to automate
- The 1-rule algorithm
- A sequential cover algorithm for sets of rules with complex conditions
- But: Sets of rules are hard to interpret

Decision trees

- Each path from the root is a tree; easy to interpret
- Use entropy to choose best attribute at each node
- Extensions for numeric attributes
- *But: Decision Trees are prone to **overfitting***

OVERFITTING

What if the knowledge and data we have are not sufficient to completely determine the correct classifier? Then we run the risk of just hallucinating a classifier (or parts of it) that is not grounded in reality, and is simply encoding random quirks in the data.

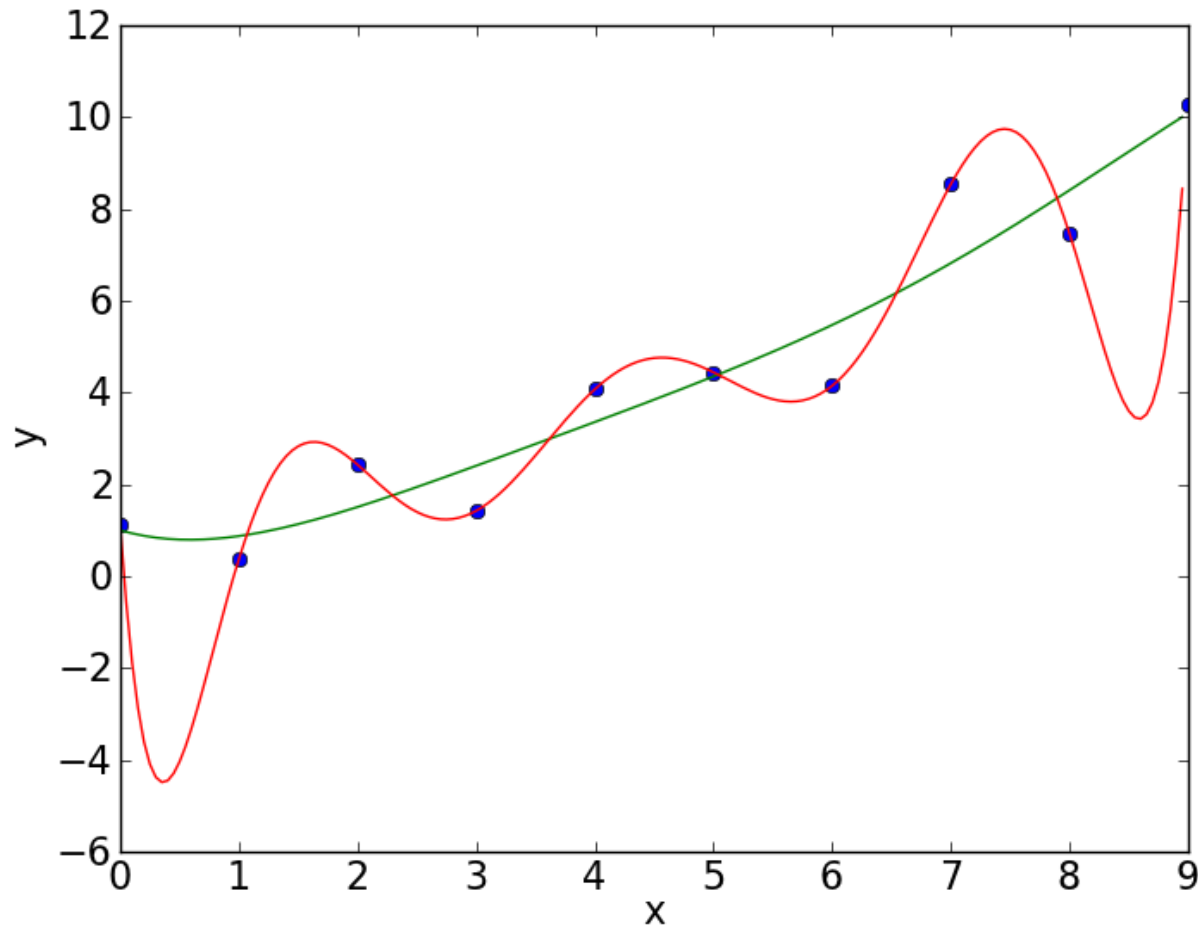
This problem is called *overfitting*, and is the bugbear of machine learning. When your learner outputs a classifier that is 100% accurate on the training data but only 50% accurate on test data, when in fact it could have output one that is 75% accurate on both, it has overfit.

OVERFITTING

Low error on training data and high error on test data

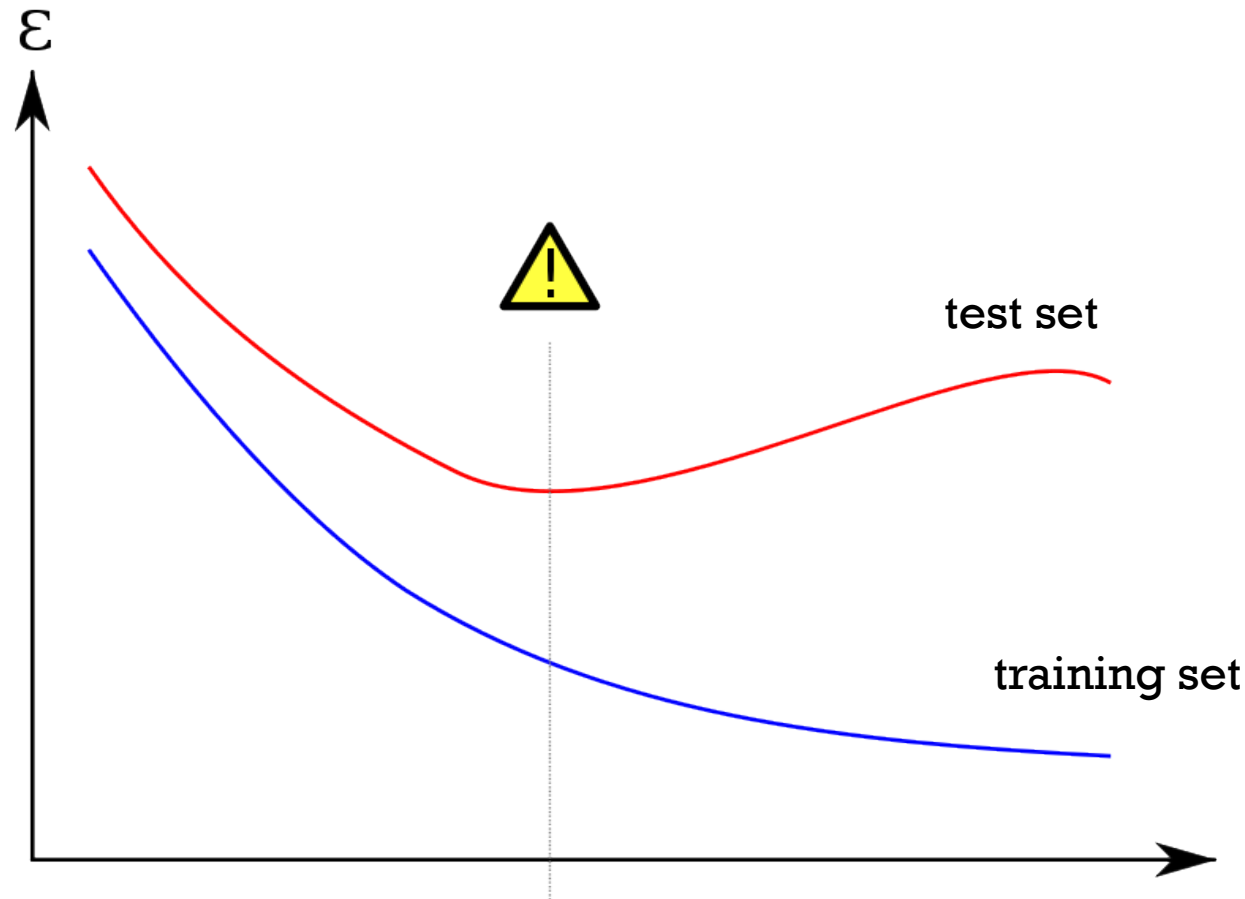
OVERFITTING

Perhaps not the most
useful intuition



OVERFITTING

A better image to
remember



Is the model able to *generalize*? Can it deal with unseen data, or does it overfit the data?
Test on *hold-out data*:

- *split* data to be modeled in training and test set
- *train* the model on training set
- *evaluate* the model on the training set
- *evaluate* the model on the test set
- difference between the fit on training data and test data measures the model's ability to *generalize*

Underfitting:
High Bias
Low Variance

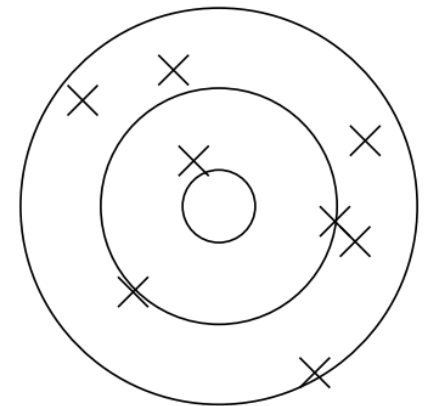
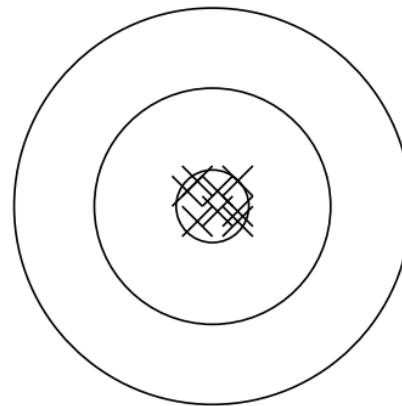
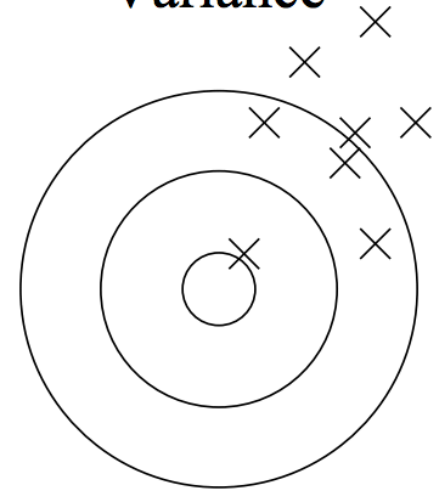
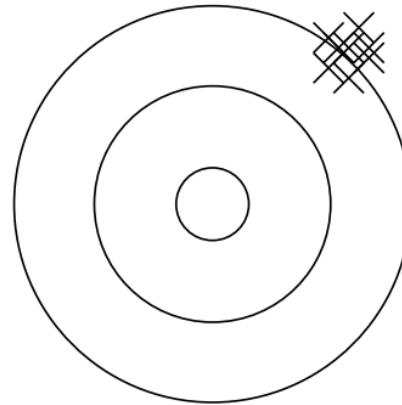
Overfitting:
Low Bias
High Variance

High
Bias

Low
Bias

Low
Variance

High
Variance



EVALUATION

Division into training and test sets

- Fixed
 - Leave out random $N\%$ of the data
- k-fold Cross-Validation
 - Select K folds without replace
- Leave-One-Out Cross Validation
 - Special case
- Related: Bootstrap
 - Generate new training sets by sampling with replacement

LEAVE-ONE-OUT CROSS-VALIDATION (LOOCV)

For each training example (\mathbf{x}_i, y_i)

Train a classifier with all training data except (\mathbf{x}_i, y_i)

Test the classifier's accuracy on (\mathbf{x}_i, y_i)

LOOCV accuracy = average of all n accuracies

ACCURACY

Confusion Matrix

	Predicted +	Predicted -
True +	a	b
True -	c	d

$$Accuracy = \frac{a + d}{a + b + c + d}$$

EVALUATION: ACCURACY ISN'T ALWAYS ENOUGH

How do you interpret 90% accuracy?

- You can't; it depends on the problem

Need a baseline:

- Base Rate
 - Accuracy of trivially predicting the most-frequent class
- Random Rate
 - Accuracy of making a random class assignment
 - Might apply prior knowledge to assign random distribution
- Naïve Rate
 - Accuracy of some simple default or pre-existing model
 - Ex: “All females survived”

ACCURACY

Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	a	b
True Negative	c	d

$$Accuracy = \frac{a + d}{a + b + c + d}$$

$$\frac{\frac{a}{a + b}}{\frac{a + c}{a + b + c + d}}$$

$$lift = \frac{\% positives > threshold}{\% dataset > threshold}$$

ACCURACY

Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	a	b
True Negative	c	d

$$Accuracy = \frac{a + d}{a + b + c + d}$$

$$precision : \frac{a}{a + c}$$

$$recall : \frac{a}{a + b}$$

ROC PLOT

Confusion Matrix

	Predicted Positive	Predicted Negative
True Positive	a	b
True Negative	c	d

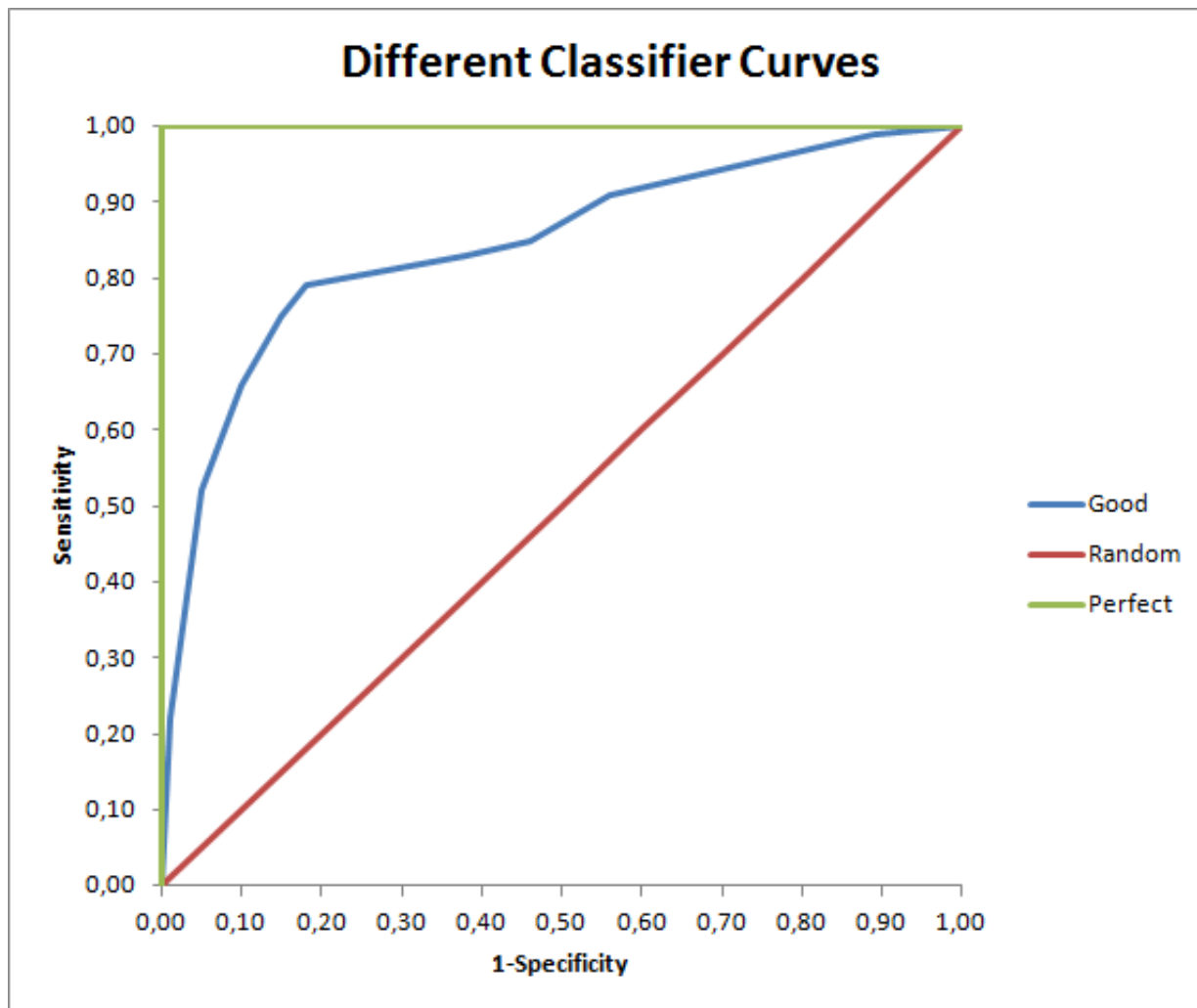
“Receiver Operator Characteristic”

- Historical term from WW2
- Used to measure accuracy of radar operators

$$Accuracy = \frac{a + d}{a + b + c + d}$$

$$sensitivity = \frac{a}{a + b}$$

$$1 - specificity = \frac{1 - d}{c + d}$$



THE BOOTSTRAP

Given a dataset of size N

Draw N samples with replacement to create a new dataset

Repeat ~ 1000 times

You now have ~ 1000 sample datasets

- All drawn from the same population
- You can compute ~ 1000 sample statistics
- You can interpret these as repeated experiments, which is exactly what the frequentist perspective calls for

Very elegant use of computational resources

BOOTSTRAP EXAMPLE

mean

1	2	3	4	5	6	3.5
4	3	4	2	1	6	3.33
2	3	6	1	3	5	3.33
5	1	1	2	3	6	3.00
2	5	2	6	3	4	3.67
4	4	4	2	1	3	3.00
3	4	5	3	2	1	3.00
1	2	3	6	6	1	3.17
5	2	3	1	4	5	3.33

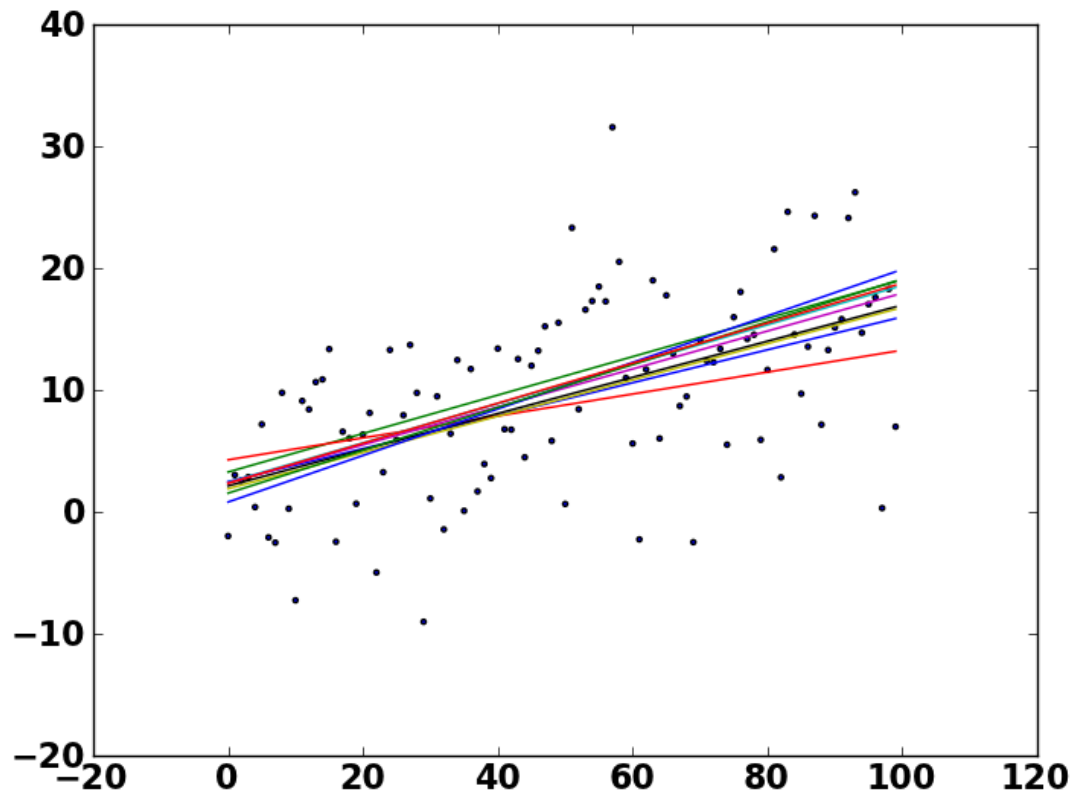
THE BOOTSTRAP

Example:

Generate 1000 samples and
1000 linear regressions

You want a 90% confidence
interval for the slope?

Just take the 5th percentile
and the 95th percentile!



ENSEMBLES: COMBINING CLASSIFIERS

Can a set of weak classifiers be combined to derive a strong classifier? Yes!

Average results from different models

Why?

- Better classification performance than individual classifiers
- More resilience to noise

Why not?

- Time consuming
- Models become difficult to explain

“Wisdom of the (simulated) crowd”

Freund, Schapire 1995

BAGGING

Draw N bootstrap samples

Retrain the model on each sample

Average the results

- Regression: Averaging
- Classification: Majority vote

Works great for overfit models

- Decreases variance without changing bias
- Doesn't help much with underfit/high bias models
 - Insensitive to the training data

BOOSTING

**Instead of selecting data points randomly with the bootstrap,
favor the misclassified points**

Initialize the weights

Repeat:

- Resample with respect to weights

- Retrain the model

- Recompute weights

For each step t

$D_t(i)$ weights: probability of selecting
example i in the sample

x_i, y_i i th example, i th label

h_t trained classifier at step t using sample drawn
according to D_t

$\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$ sum of weights for
misclassified examples

$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$ odds of misclassifying

$D_{t+1}(i) = \beta_t D_t(i)$ adjust weights down for
correctly classified examples

...and normalize to
make sure weights
sum to 1

RANDOM FOREST ALGORITHM

Repeat k times:

- Draw a **bootstrap sample** from the dataset
- Train a decision tree
 - Until the tree is maximum size
 - Choose next leaf node
 - Select m attributes at random from the p available
 - Pick the best attribute/split as usual
- Measure **out-of-bag error**
 - Evaluate against the samples that were not selected in the bootstrap
 - Provides measures of strength (inverse error rate), correlation between trees (which increases the forest error rate), and variable importance

Make a prediction by majority vote among the k trees

RANDOM FORESTS: VARIABLE IMPORTANCE

Key Idea: If you scramble the values of a variable and the accuracy of your tree doesn't change much, then the variable isn't very important

Measure the error increase

Random Forests are more difficult to interpret than single trees; understanding variable importance helps

- Ex: Medical applications can't typically rely on black box solutions

GINI COEFFICIENT

Entropy captured an intuition for “impurity”

- We want to choose attributes that split records into pure classes

The gini coefficient measures inequality

$$\text{Gini}(T) = 1 - \sum_{i=1}^n p_i^2$$

RANDOM FORESTS ON BIG DATA

Easy to parallelize

- Trees are built independently

Handles “small n big p ” problems naturally

- A subset of attributes are selected by importance

SUMMARY: DECISION TREES AND FORESTS

Representation

- Decision Trees
- Sets of decision trees with majority vote

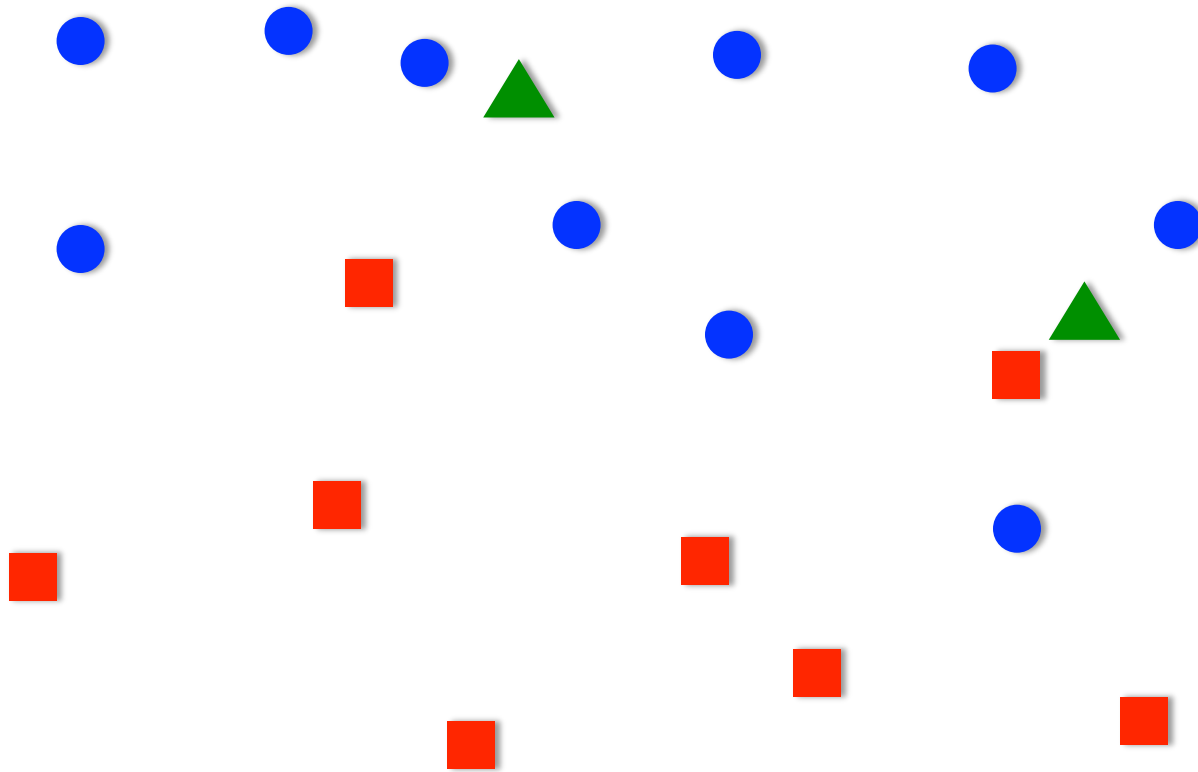
Evaluation

- Accuracy
- Random forests: out-of-bag error

Optimization

- Information Gain or Gini Index to measure impurity and select best attributes

NEAREST NEIGHBOR



NEAREST NEIGHBORS INTUITION

The last document I saw that mentioned “Falcons” and “Saints” was about Sports, so I’ll classify this document as about Sports too

NEAREST NEIGHBOR CHOICES

k nearest neighbors – how do we choose k?

- Benefits of a small k? Benefits of a large k?

Large k = bias towards popular labels

Large k = ignores outliers

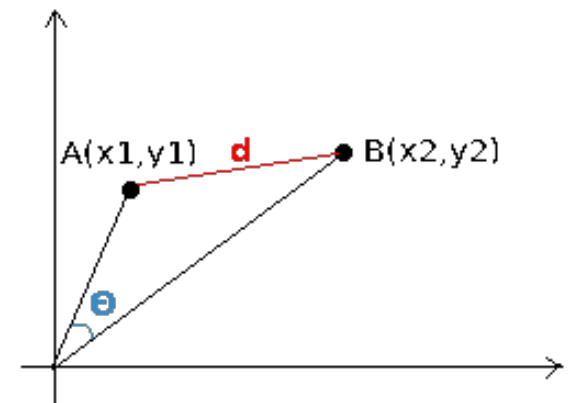
Small k = fast

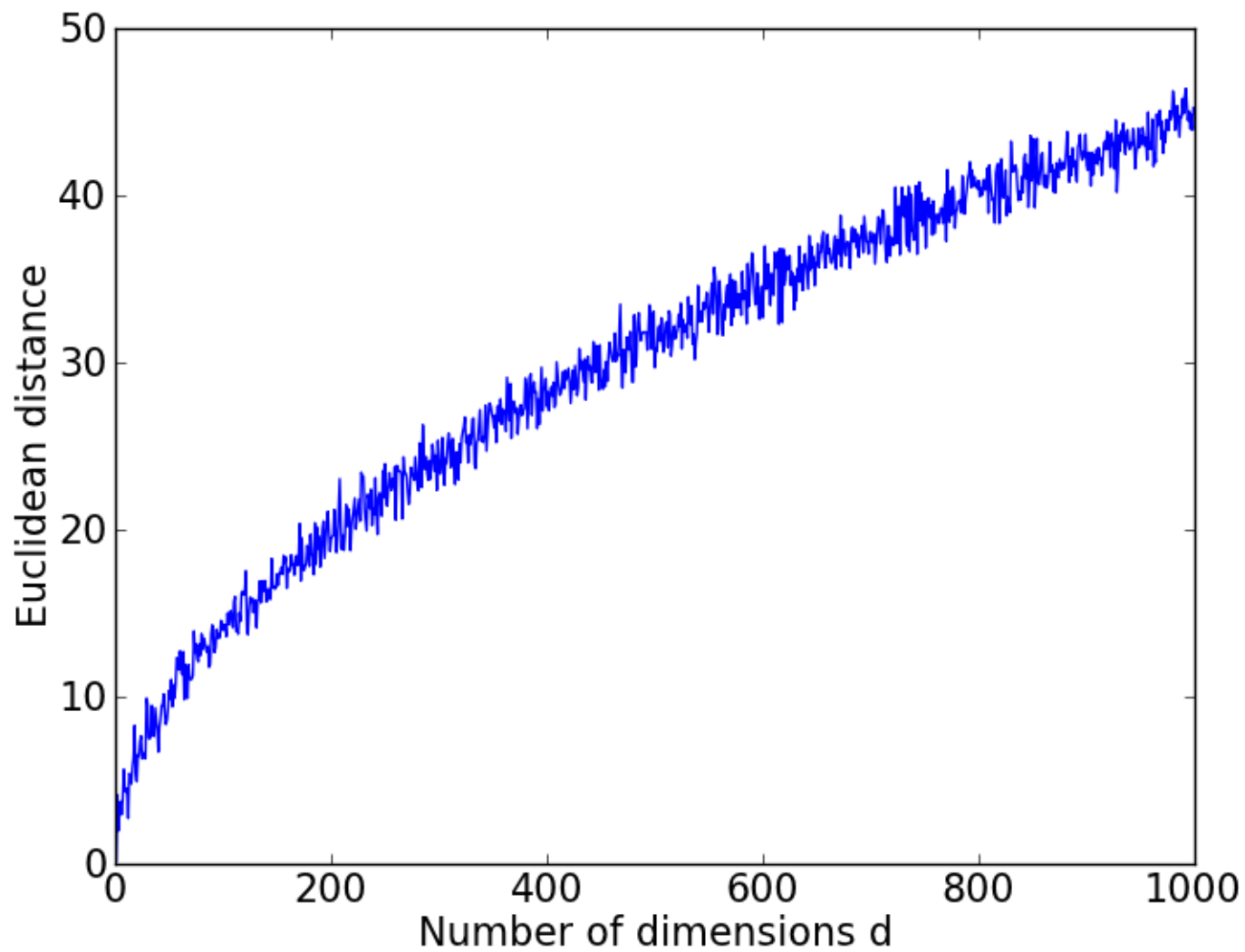
Similarity function

- Euclidean distance? Cosine similarity?

Cosine = favors dominant components

Euclidean = difficult to interpret with sparse data, and high-dimensional data is always sparse



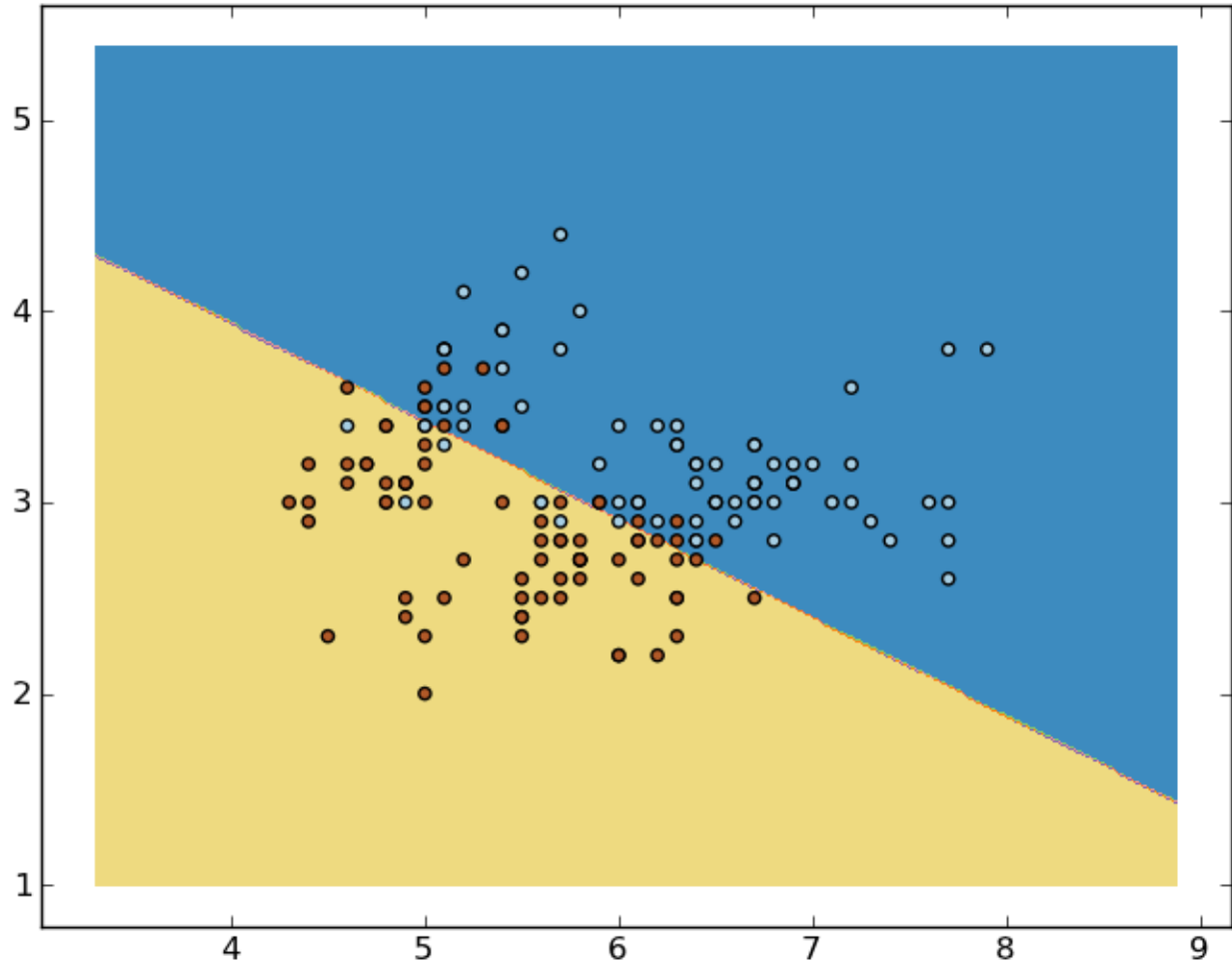


REGRESSION-BASED METHODS

Trees and rules are designed for categorical data

- Numerical data can be discretized, but this introduces another decision

Discriminative models



EVALUATION: MEAN SQUARED ERROR

$$MSE = \frac{1}{n} \sum_{i=1}^n (H_i - H'_i)^2$$

H_i : quantity observed in the data set

H'_i : quantity predicted by model

n : number of instances in the data set

ASIDE ON ERRORS AND NORMS

not a norm

L¹-norm $\sum_i H_i - H'_i$

errors cancel out;
usually not what you want

L²-norm $\sum_i |H_i - H'_i|$

Asserts that 1 error of 7 units is as
bad as 7 errors of 1 unit each

$$\sum_i (H_i - H'_i)^2$$

Asserts that 1 error of 7 units is as
bad as 49 errors of 1 unit each

$$\frac{1}{n} \sum_i^n (H_i - H'_i)^2$$

Average squared error per data point;
useful when comparing methods that
filter the data differently

TERMINOLOGY: NORMS

Norm: any function that assigns a strictly positive number to every non-zero vector

$$L^p\text{-norm} = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

$$\sqrt{\sum_{i=1}^n (\beta_0 + \beta_1 x_i - y_i)^2}$$

SLIDES CAN BE FOUND AT:
TEACHINGDATASCIENCE.ORG