Deconstructing Data Science

David Bamman, UC Berkeley

Info 290
Lecture 7: Decision trees & random forests

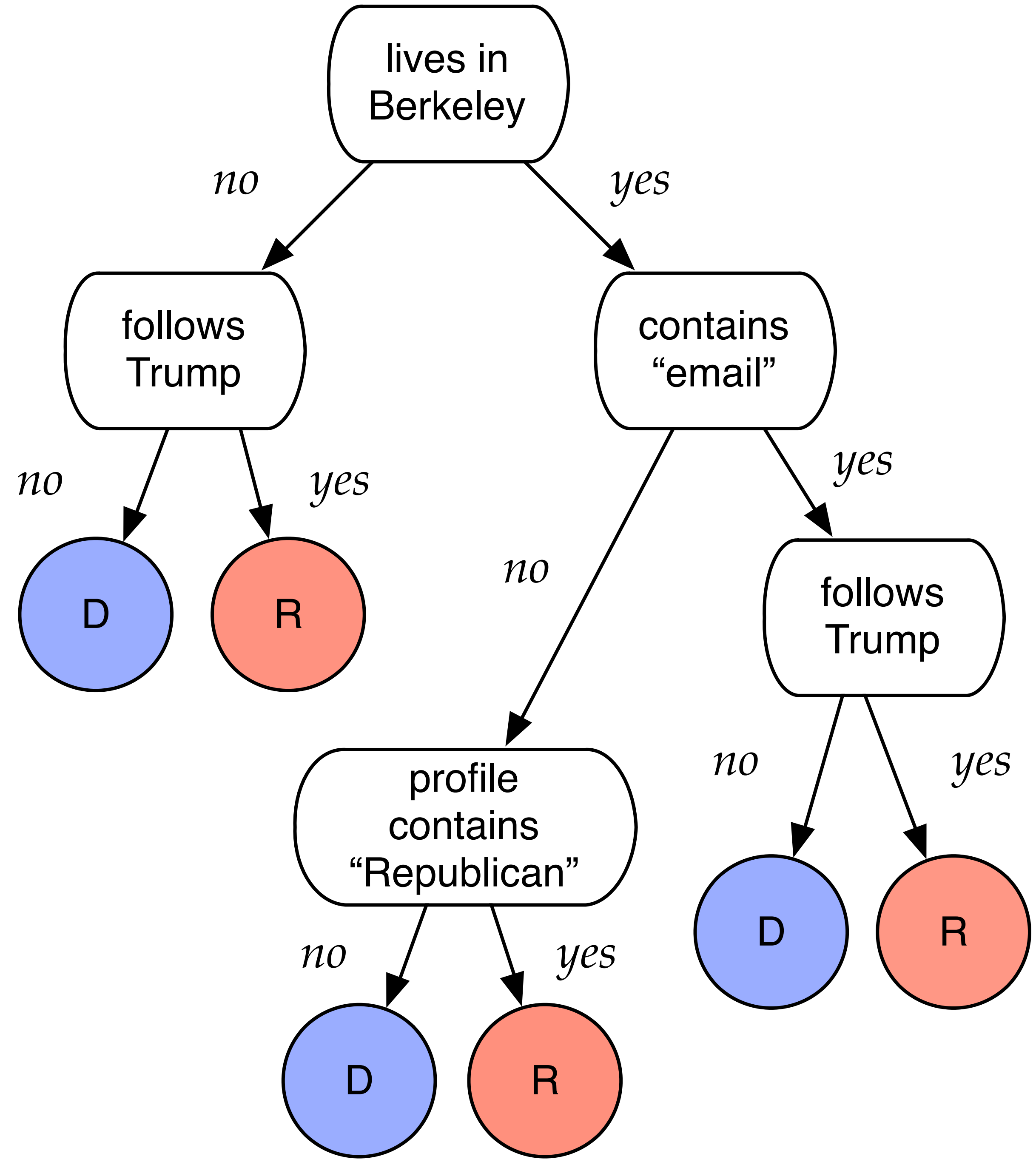
Feb 10, 2016

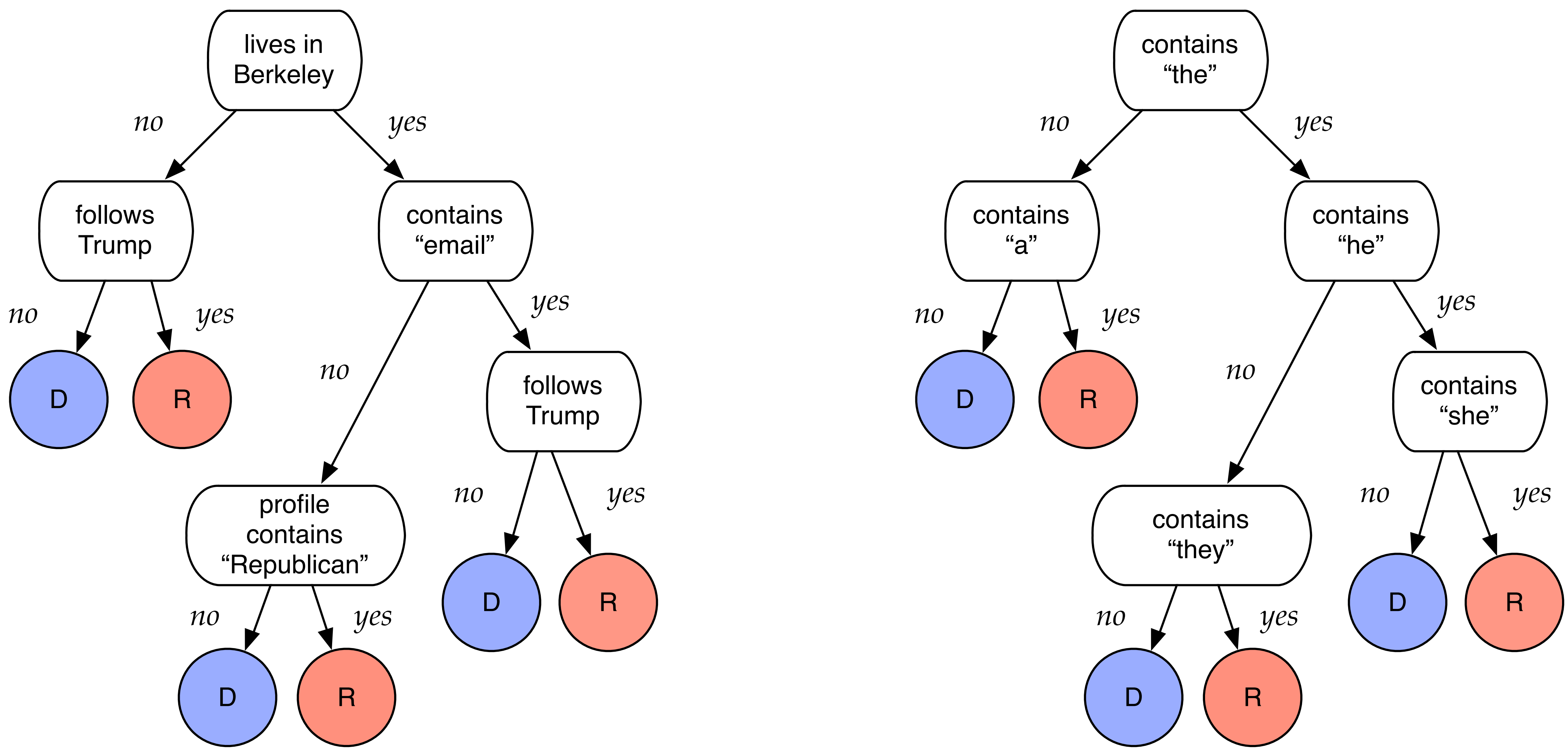

Linear regression
Deep learning
Decision trees
Ordinal regression
Probabilistic graphical models
Random forests
Logistic regression
Networks
Support vector machines
Topic models
Survival models
K-means clustering
Neural networks
Hierarchical clustering
Perceptron


Decision trees
Random forests

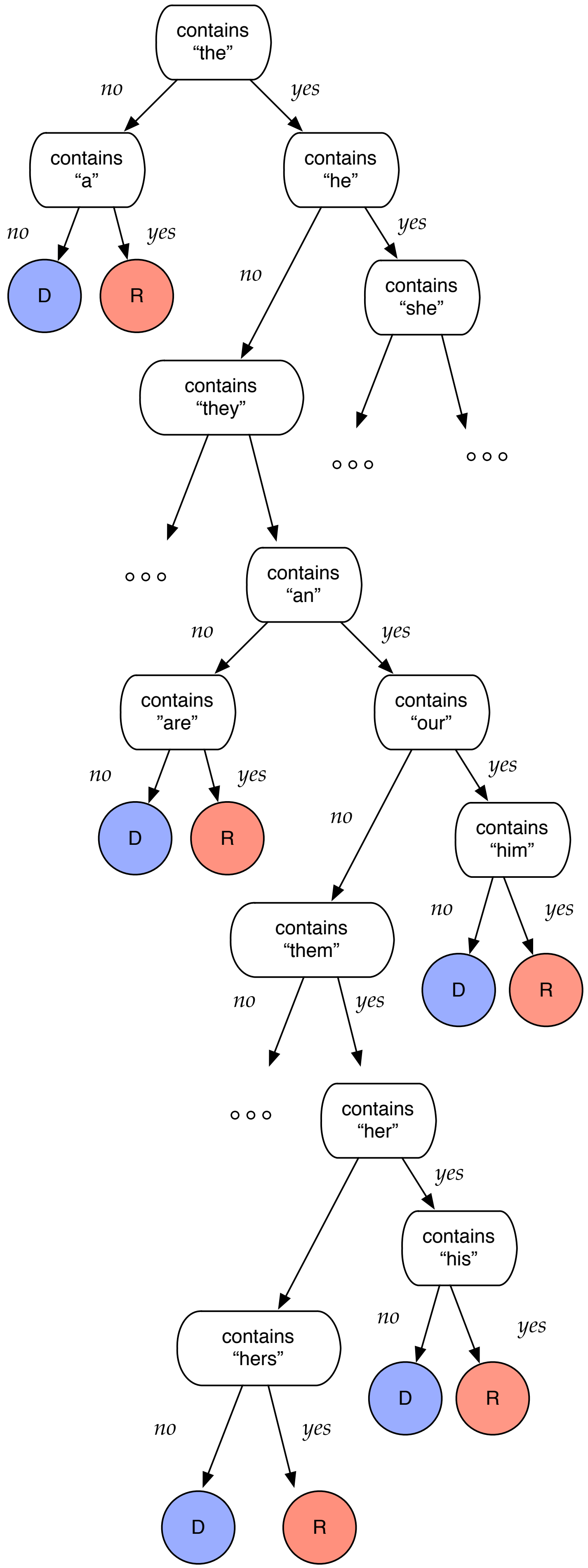
20 questions

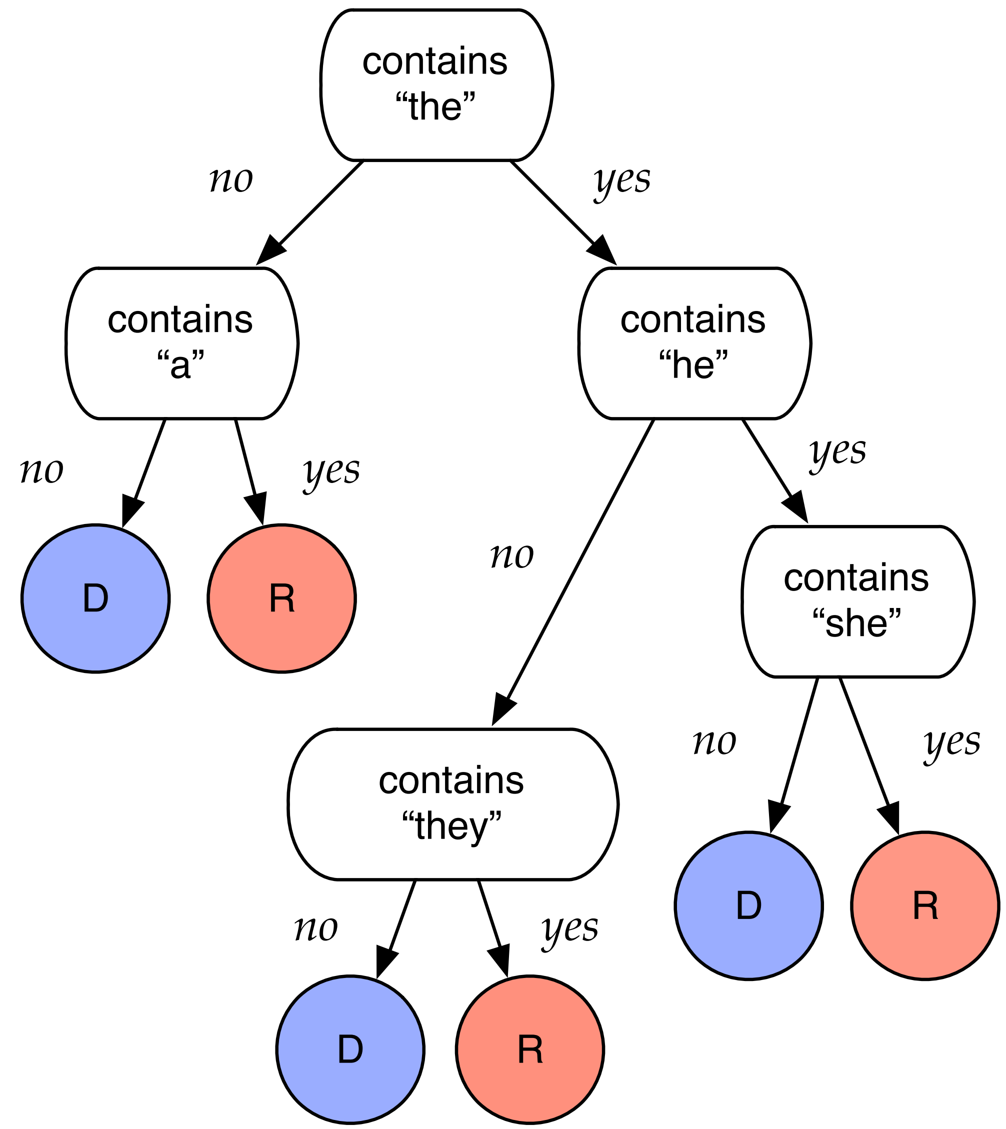
Feature	Value
follow clinton	0
follow trump	0
“benghazi”	0
negative sentiment + “benghazi”	0
“illegal immigrants”	0
“republican” in profile	0
“democrat” in profile	0
self-reported location = Berkeley	1





how do we find the best tree?





how do we find the best tree?

Decision trees

Algorithm 5.1: GrowTree(D, F) – grow a feature tree from training data.

Input : data D; set of features F.

Output : feature tree T with labelled leaves.

if Homogeneous(D) **then return** Label(D) ; // Homogeneous, Label: see text

$S \leftarrow$ BestSplit(D, F) ; // e.g., BestSplit-Class (Algorithm 5.2)

split D into subsets D_i according to the literals in S;

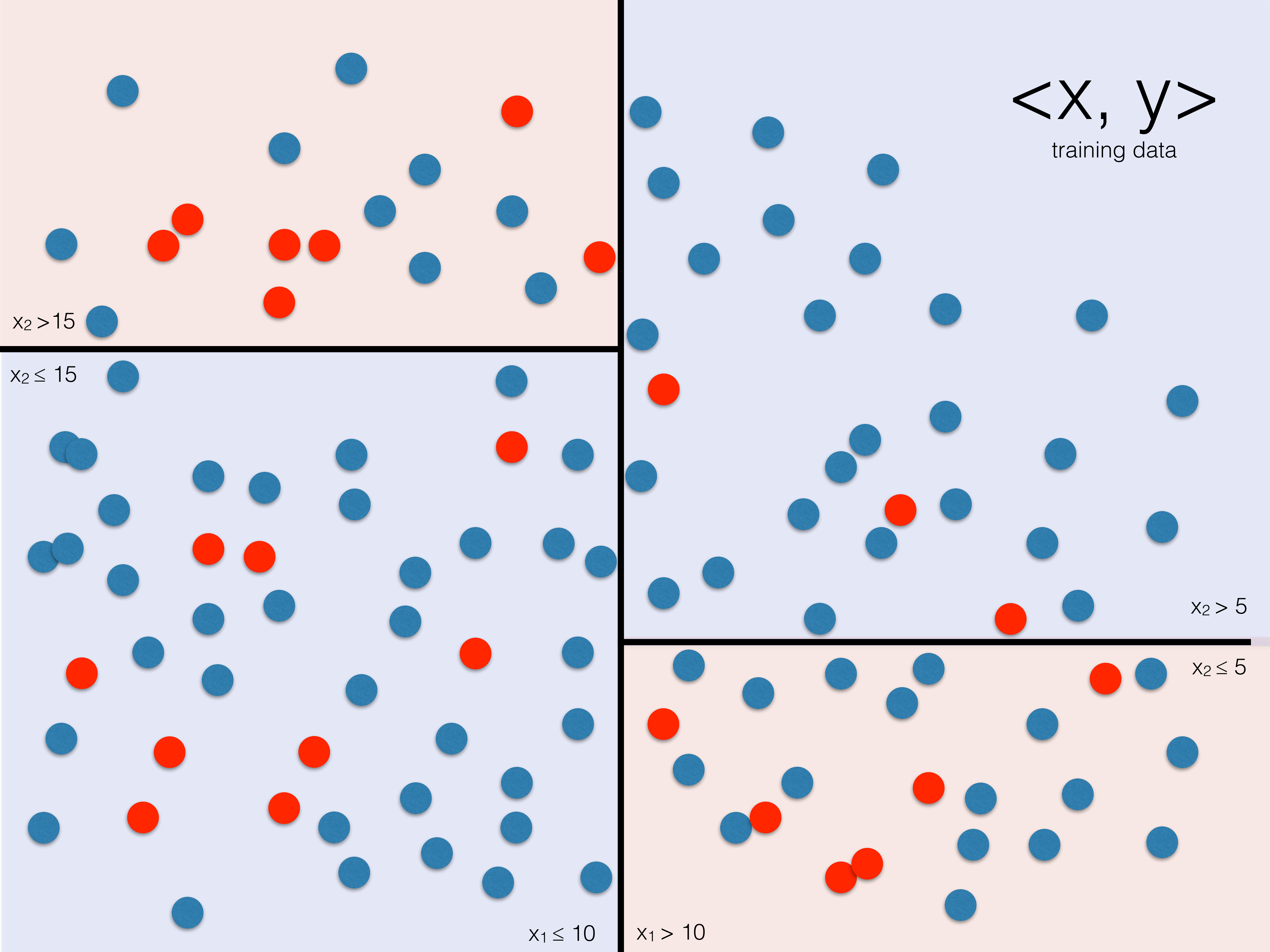
for each i **do**

  if $D_i \neq \emptyset$ **then** $T_i \leftarrow$ GrowTree(D_i, F) **else** T_i is a leaf labelled with Label(D);

end

return a tree whose root is labelled with S and whose children are T_i

from Flach 2014


<x, y>
training data
x2 >15
x2 ≤ 15
x2 > 5
x2 ≤ 5
x1 ≤ 10
x1 > 10

Decision trees

Algorithm 5.1: GrowTree(D, F) – grow a feature tree from training data.

Input : data D; set of features F.

Output : feature tree T with labelled leaves.

if Homogeneous(D) **then return** Label(D) ; // Homogeneous, Label: see text

$S \leftarrow$ BestSplit(D, F) ; // e.g., BestSplit-Class (Algorithm 5.2)

split D into subsets D_i according to the literals in S;

for each i **do**

  if $D_i \neq \emptyset$ **then** $T_i \leftarrow$ GrowTree(D_i, F) **else** T_i is a leaf labelled with Label(D);

end

return a tree whose root is labelled with S and whose children are T_i

from Flach 2014

Decision trees

- Homogeneous(D): the elements in D are homogeneous enough that they can be labeled with a single label
- Label(D): the single most appropriate label for all elements in D

Decision trees

	Homogeneous	Label
Classification	All (or most) of the elements in D share the same label y	y
Regression	The elements in D have low variance	the average of elements in D

Decision trees

Algorithm 5.1: GrowTree(D, F) – grow a feature tree from training data.

Input : data D; set of features F.

Output : feature tree T with labelled leaves.

if Homogeneous(D) **then return** Label(D) ; // Homogeneous, Label: see text

$S \leftarrow$ BestSplit(D, F) ; // e.g., BestSplit-Class (Algorithm 5.2)

split D into subsets D_i according to the literals in S;

for each i **do**

  if $D_i \neq \emptyset$ **then** $T_i \leftarrow$ GrowTree(D_i, F) **else** T_i is a leaf labelled with Label(D);

end

return a tree whose root is labelled with S and whose children are T_i

from Flach 2014

Entropy

Measure of uncertainty in a probability distribution

$$-\sum_{x \in \mathcal{X}} P(x) \log P(x)$$

- a great _______
- the oakland ______

a great …

deal	12196
job	2164
idea	1333
opportunity	855
weekend	585
player	556
extent	439
honor	282
pleasure	267
gift	256
humor	221
tool	184
athlete	173
disservice	108
…	

the oakland …

athletics	185
raiders	185
museum	92
hills	72
tribune	51
police	49
coliseum	41

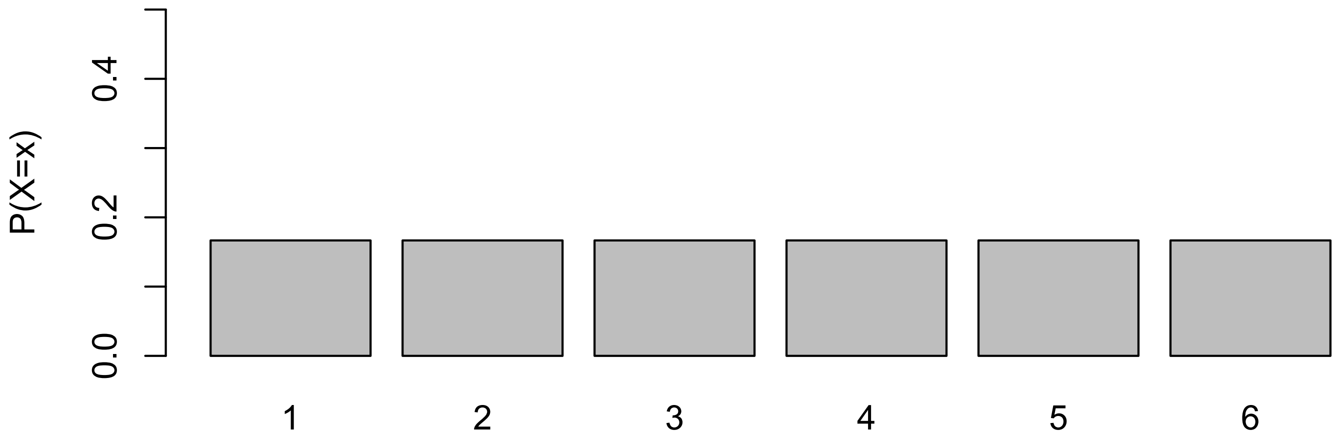
Corpus of Contemporary American English

Entropy

$$-\sum_{x \in \mathcal{X}} P(x) \log P(x)$$

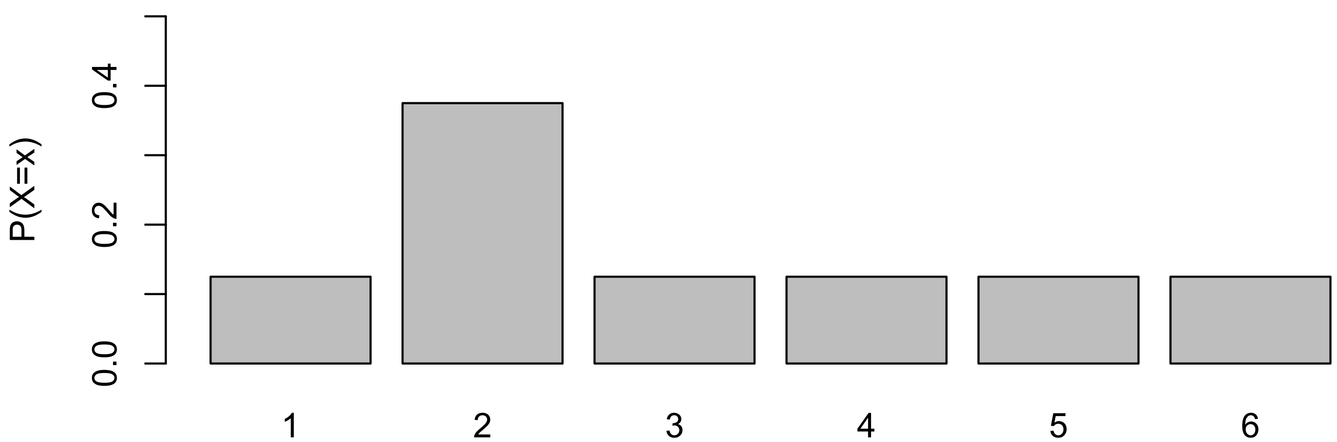
- High entropy means the phenomenon is less predictable
- Entropy of 0 means it is entirely predictable.

Entropy



A uniform distribution has maximum entropy

$$-\sum_{1}^{6} \frac{1}{6} \log \frac{1}{6} = 2.58$$



This entropy is lower because it is more predictable
(if we always guess 2, we would be right 40% of the time)

$$-0.4 \log 0.4 - \sum_{1}^{5} 0.12 \log 0.12 = 2.36$$

Conditional entropy

- Measures your level of surprise about some phenomenon Y if you have information about another phenomenon X

- Y = word, X = preceding bigram (“the oakland ___”)
- Y = label (democrat, republican), X = feature (lives in Berkeley)

Conditional entropy

- Measures you level of surprise about some phenomenon Y if you have information about another phenomenon X

Y = label

X = feature value

$$H(Y \mid X)$$

$$= \sum_{x} P(X = x) H(Y \mid X = x)$$

$$H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x)$$

Information gain

- aka "Mutual Information": the reduction in entropy in Y as a result of knowing information about X

$$H(Y) - H(Y \mid X)$$

$$H(Y) = -\sum_{y \in \mathcal{Y}} p(y) \log p(y)$$

$$H(Y \mid X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x)$$

	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	$\oplus$	$\ominus$	$\ominus$	$\oplus$	$\oplus$	$\ominus$

Which of these features gives you more information about y?

	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	$\oplus$	$\ominus$	$\ominus$	$\oplus$	$\oplus$	$\ominus$

x_1

$x \in \mathcal{X}$	0	1
$y \in \mathcal{Y}$	3 $\oplus$ 0 $\ominus$	0 $\oplus$ 3 $\ominus$

	$x \in \mathcal{X}$	0		1	
x_1					
	$y \in \mathcal{Y}$	$3 \oplus$	$0 \ominus$	$0 \oplus$	$3 \ominus$

$$H(Y \mid X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x)$$

$$P(x=0) = \frac{3}{3+3} = 0.5$$

$$P(x=1) = \frac{3}{3+3} = 0.5$$

$$P(y=+ \mid x=0) = \frac{3}{3+0} = 1$$

$$P(y=- \mid x=0) = \frac{0}{3+0} = 0$$

$$P(y=+ \mid x=1) = \frac{0}{3+0} = 0$$

$$P(y=- \mid x=1) = \frac{3}{3+0} = 1$$

x_1

$x \in \mathcal{X}$	0		1	
$y \in \mathcal{Y}$	3⊕	0⊖	0⊕	3⊖

$$H(Y \mid X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x)$$

$$-\frac{3}{6}(1 \log 1 + 0 \log 0) - \frac{3}{6}(0 \log 0 + 1 \log 1) = 0$$

	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	$\oplus$	$\ominus$	$\ominus$	$\oplus$	$\oplus$	$\ominus$

x_2

$x \in \mathcal{X}$	0	1
$y \in \mathcal{Y}$	1 $\oplus$ 2 $\ominus$	2 $\oplus$ 1 $\ominus$

x_2

$x \in \mathcal{X}$	0		1	
$y \in \mathcal{Y}$	1 $\oplus$	2 $\ominus$	2 $\oplus$	1 $\ominus$

$$P(y = + \mid x = 0) = \frac{1}{1+2} = 0.33$$

$$P(x = 0) = \frac{3}{3+3} = 0.5$$

$$P(y = - \mid x = 0) = \frac{2}{1+2} = 0.67$$

$$P(x = 1) = \frac{3}{3+3} = 0.5$$

$$P(y = + \mid x = 1) = \frac{2}{1+2} = 0.67$$

$$P(y = - \mid x = 1) = \frac{1}{1+2} = 0.33$$

x_2

$x \in \mathcal{X}$	0		1	
$y \in \mathcal{Y}$	1 $\oplus$	2 $\ominus$	2 $\oplus$	1 $\ominus$

$$H(Y \mid X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x)$$

$$-\frac{3}{6}(0.33 \log 0.33 + 0.67 \log 0.67) - \frac{3}{6}(0.67 \log 0.67 + 0.33 \log 0.33) = 0.91$$

Feature	H(Y | X)
follow clinton	0.91
follow trump	0.77
“benghazi”	0.45
negative sentiment + “benghazi”	0.33
“illegal immigrants”	0
“republican” in profile	0.31
“democrat” in profile	0.67
self-reported location = Berkeley	0.80

In decision trees, the feature with the lowest conditional entropy/highest information gain defines the “best split”

$$MI = IG = H(Y) - H(Y \mid X)$$

for a given partition, H(Y) is the same for all features, so we can ignore it when deciding among them

Feature	H(Y | X)
follow clinton	0.91
follow trump	0.77
“benghazi”	0.45
negative sentiment + “benghazi”	0.33
“illegal immigrants”	0
“republican” in profile	0.31
“democrat” in profile	0.67
self-reported location = Berkeley	0.80

How could we use this in other models (e.g., the perceptron)?

Decision trees

Algorithm 5.1: GrowTree(D, F) – grow a feature tree from training data.

Input : data D; set of features F.

Output : feature tree T with labelled leaves.

if Homogeneous(D) **then return** Label(D) ; // Homogeneous, Label: see text

$S \leftarrow$ BestSplit(D, F) ; // e.g., BestSplit-Class (Algorithm 5.2)

split D into subsets D_i according to the literals in S;

for each i **do**

  if $D_i \neq \emptyset$ **then** $T_i \leftarrow$ GrowTree(D_i, F) **else** T_i is a leaf labelled with Label(D);

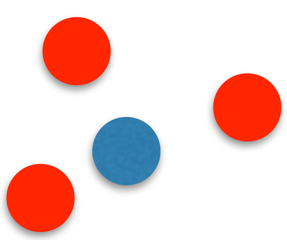
end

return a tree whose root is labelled with S and whose children are T_i

BestSplit identifies the feature with the highest information gain and partitions the data according to values for that feature

Gini impurity

- Measure the “purity” of a partition (how diverse the labels are). If we were to pick an element in D and assign a label in proportion to the label distribution in D, how often would we make a mistake?



Probability of selecting an item with label y at random

$$\sum_{y\in\mathcal{Y}} p_y(1-p_y)$$

The probability of randomly assigning it the wrong label

Gini impurity

$$\sum_{y \in \mathcal{Y}} p_y(1 - p_y)$$

x_1

$x \in \mathcal{X}$	0		1	
$y \in \mathcal{Y}$	$3\oplus$	$0\ominus$	$0\oplus$	$3\ominus$

x_2

$x \in \mathcal{X}$	0		1	
$y \in \mathcal{Y}$	$1\oplus$	$2\ominus$	$2\oplus$	$1\ominus$

$$G(0) = 1 \times (1 - 1) + 0 \times (1 - 0) = 0$$

$$G(0) = 0 \times (1 - 0) + 1 \times (1 - 1) = 0$$

$$G(0) = 0.33 \times (1 - 0.33) + 0.67 \times (1 - 0.67) = 0.44$$

$$G(1) = 0.67 \times (1 - 0.67) + 0.33 \times (1 - 0.33) = 0.44$$

$$G(x_1) = (\frac{3}{3+3})0 + (\frac{3}{3+3})0 = 0$$

$$G(x_2) = (\frac{3}{3+3})0.44 + (\frac{3}{3+3})0.44 = 0.44$$

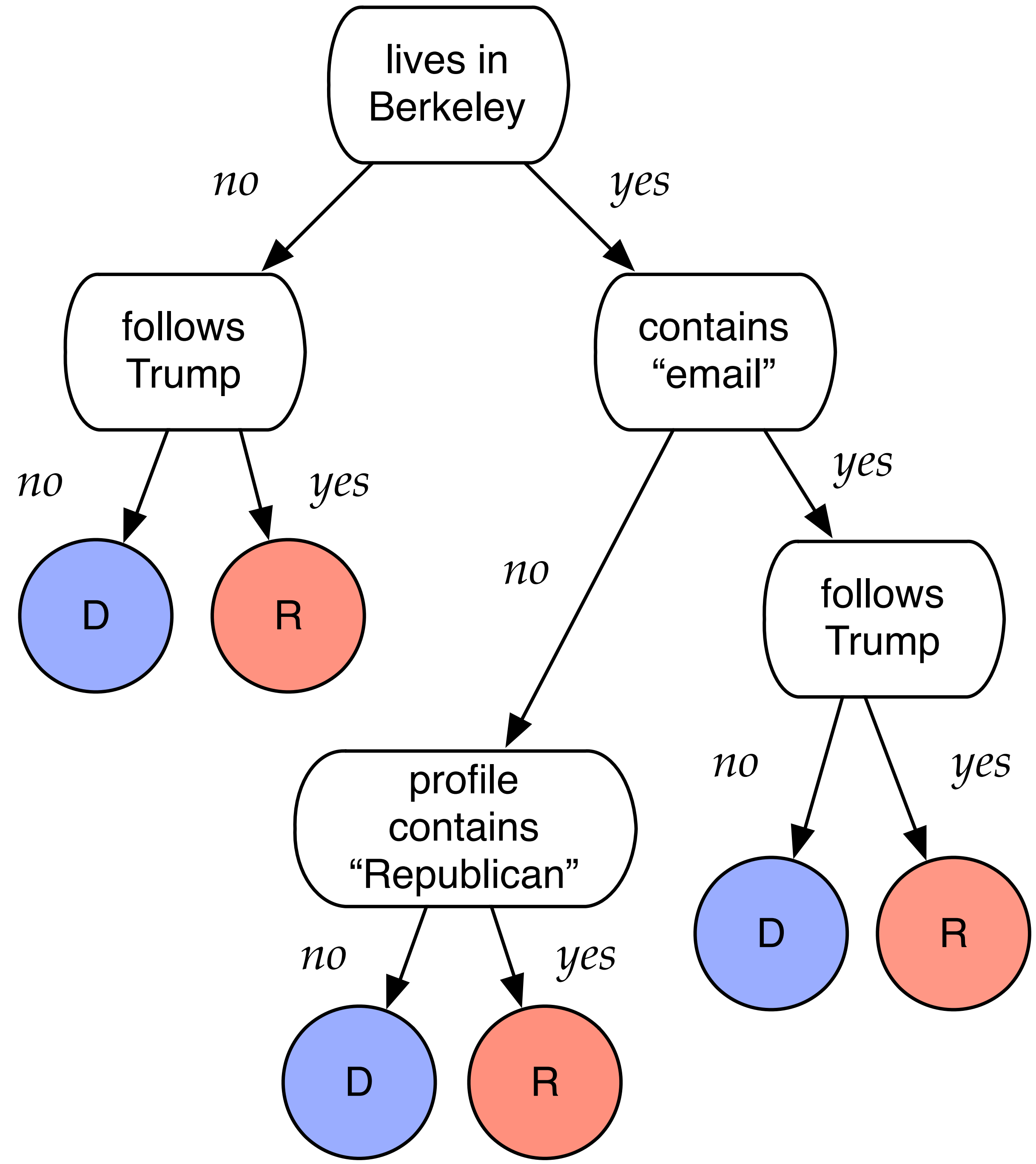
Classification

A mapping h from input data x (drawn from instance space $\mathcal{X}$) to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{X}$ = set of all skyscrapers
$\mathcal{Y}$ = {art deco, neo-gothic, modern}

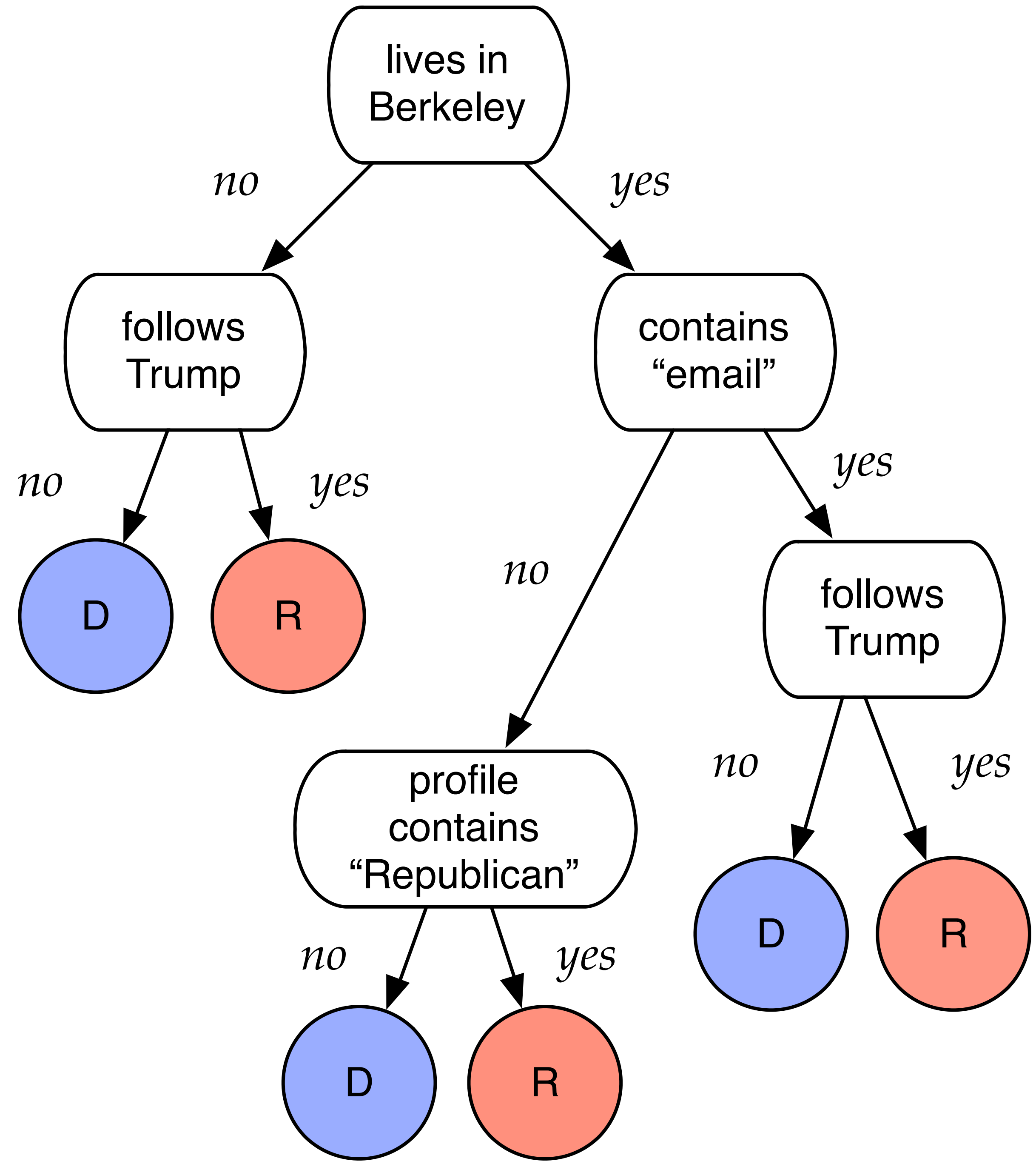
x = the empire state building
y = art deco

Feature	Value
follow clinton	0
follow trump	0
“benghazi”	0
negative sentiment + “benghazi”	0
“illegal immigrants”	0
“republican” in profile	0
“democrat” in profile	0
self-reported location = Berkeley	1



The tree that we’ve learned is the mapping $\hat{h}(x)$

Feature	Value
follow clinton	0
follow trump	0
“benghazi”	0
negative sentiment + “benghazi”	0
“illegal immigrants”	0
“republican” in profile	0
“democrat” in profile	0
self-reported location = Berkeley	1



How is this different from the perceptron?

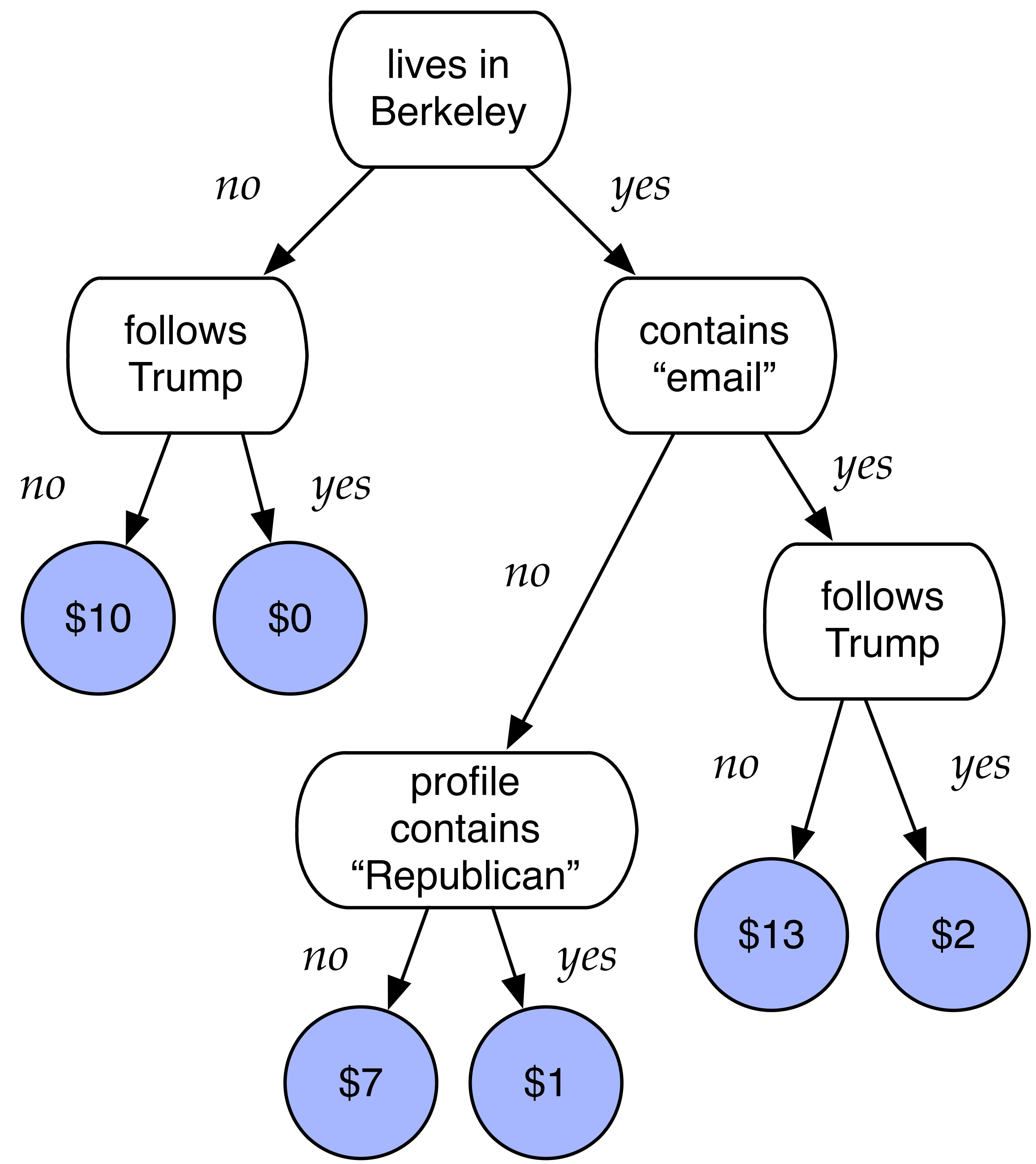
Regression

A mapping from input data x (drawn from instance space $\mathcal{X}$) to a point y in $\mathbb{R}$

($\mathbb{R}$ = the set of real numbers)

x = the empire state building
y = 17444.5625"

Feature	Value
follow clinton	0
follow trump	0
“benghazi”	0
negative sentiment + “benghazi”	0
“illegal immigrants”	0
“republican” in profile	0
“democrat” in profile	0
self-reported location = Berkeley	1



Decision trees

Algorithm 5.1: GrowTree(D, F) – grow a feature tree from training data.

Input : data D; set of features F.

Output : feature tree T with labelled leaves.

if Homogeneous(D) **then return** Label(D) ; // Homogeneous, Label: see text

$S \leftarrow$ BestSplit(D, F) ; // e.g., BestSplit-Class (Algorithm 5.2)

split D into subsets D_i according to the literals in S;

for each i **do**

  if $D_i \neq \emptyset$ **then** $T_i \leftarrow$ GrowTree(D_i, F) **else** T_i is a leaf labelled with Label(D);

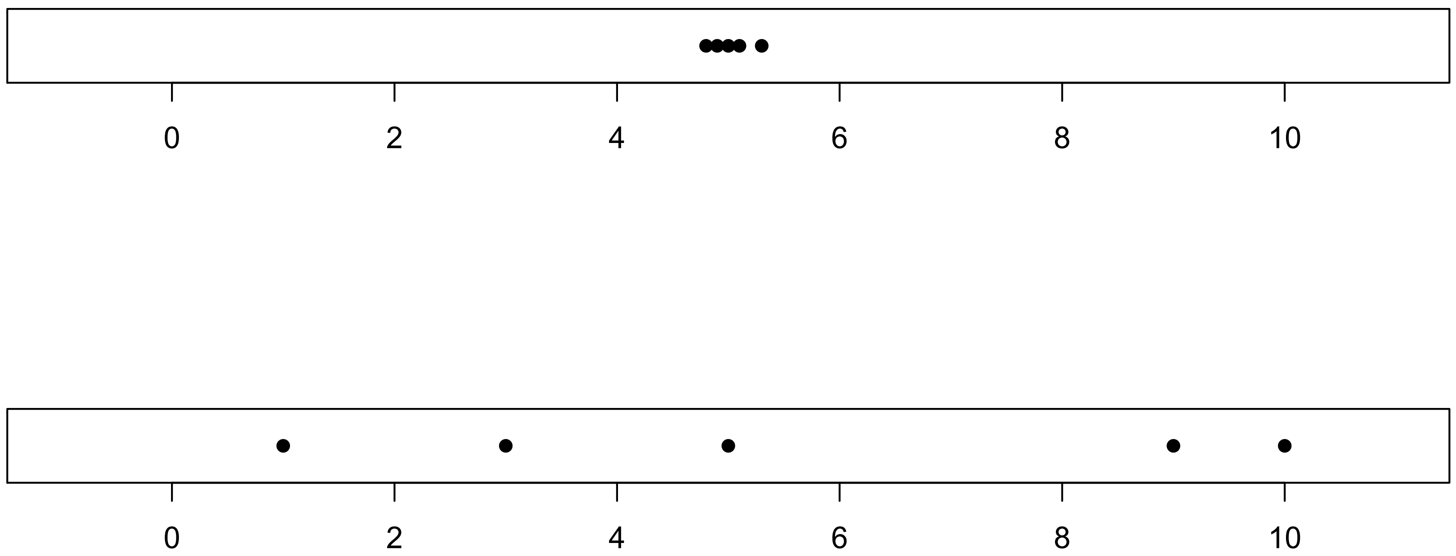
end

return a tree whose root is labelled with S and whose children are T_i

from Flach 2014

Variance

The level of “dispersion” of a set of values, how far they tend to fall from the average



	5	5
	5.1	10
	4.8	3
	5.3	1
	4.9	9
Mean	5.0	5.0
Variance	0.025	10

Variance

The level of "dispersion" of a set of values, how far they tend to fall from the average

$$Var(Y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

	5	5
	5.1	10
	4.8	3
	5.3	1
	4.9	9
Mean	5.0	5.0
Variance	0.025	10

Regression trees

- Rather than using entropy/Gini as a splitting criterion, we'll find the feature that results in the lowest variance of the data after splitting on the feature values.

	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	5.0	1.7	0	10	8	2.2

	$x \in \mathcal{X}$	0	1
x_1	$y \in \mathcal{Y}$	5.0, 10, 8	1.7, 0, 2.2
	Var	6.33	1.33

Average Variance: $$\frac{3}{6}6.33 + \frac{3}{6}1.33 = 3.83$$

	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	5.0	1.7	0	10	8	2.2

	$x \in \mathcal{X}$	0	1
x_2	$y \in \mathcal{Y}$	5.0, 1.7, 0	10, 8, 2.2
	Var	6.46	16.4

Average Variance: $$\frac{3}{6}6.46 + \frac{3}{6}16.4 = 11.43$$

Regression trees

- Rather than using entropy/Gini as a splitting criterion, we'll find the feature that results in the lowest variance of the data after splitting on the feature values.

- Homogeneous(D): the elements in D are homogeneous enough that they can be labeled with a single label. Variance < small threshold.

- Label(D): the single most appropriate label for all elements in D; the average value of y among D

Overfitting

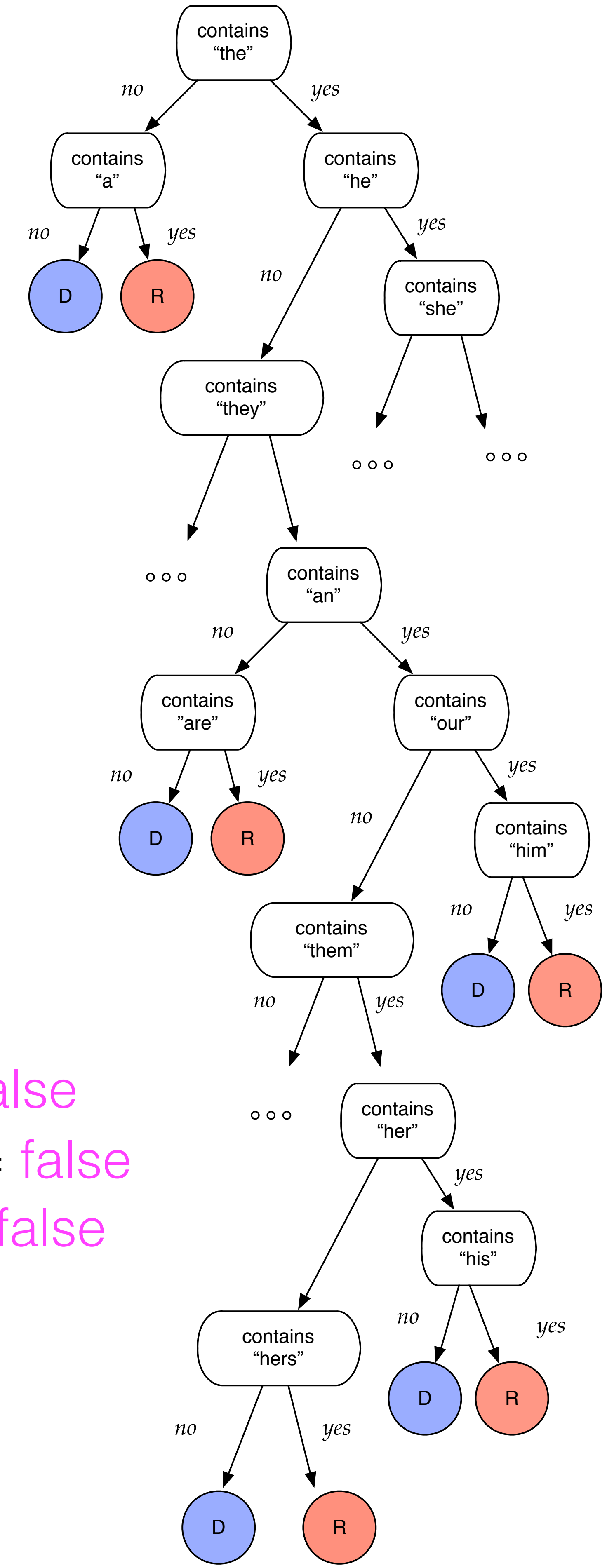
With enough features, you can perfectly memorize the training data, encoding in paths within the tree

follow clinton = false
∧ follow trump = false
∧ "benghazi" = false
∧ "illegal immigrants" = false
∧ "republican" in profile = false
∧ "democrat" in profile = false
∧ self-reported location =
Berkeley = true

→ *Democrat*

follow clinton = true
∧ follow trump = false
∧ "benghazi" = false
∧ "illegal immigrants" = false
∧ "republican" in profile = false
∧ "democrat" in profile = false
∧ self-reported location =
Berkeley = true

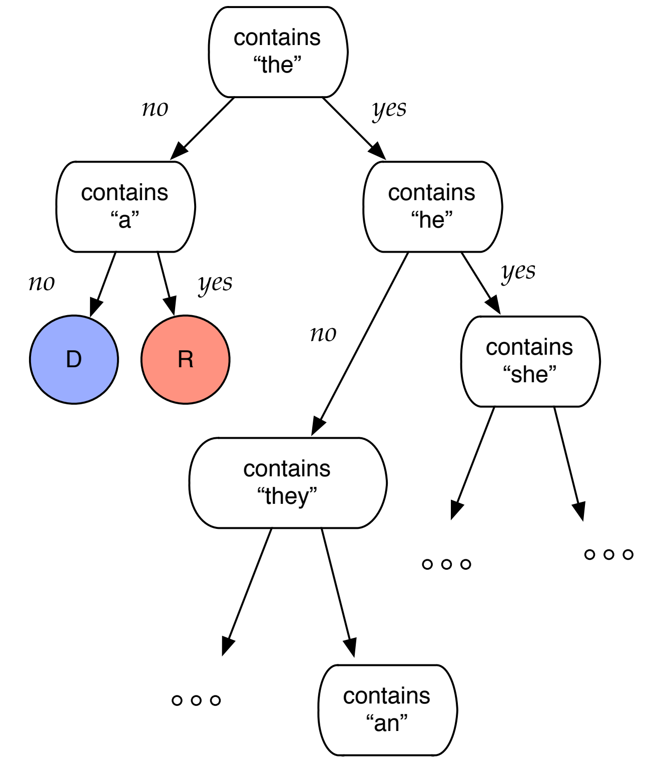
→ *Republican*



Pruning

- One way to prevent overfitting is to grow the tree to an arbitrary depth, and then prune back layers (delete subtrees)

Pruning



- Deeper into the tree = more conjunctions of features; a shallower tree contains only the most important (by IG) features

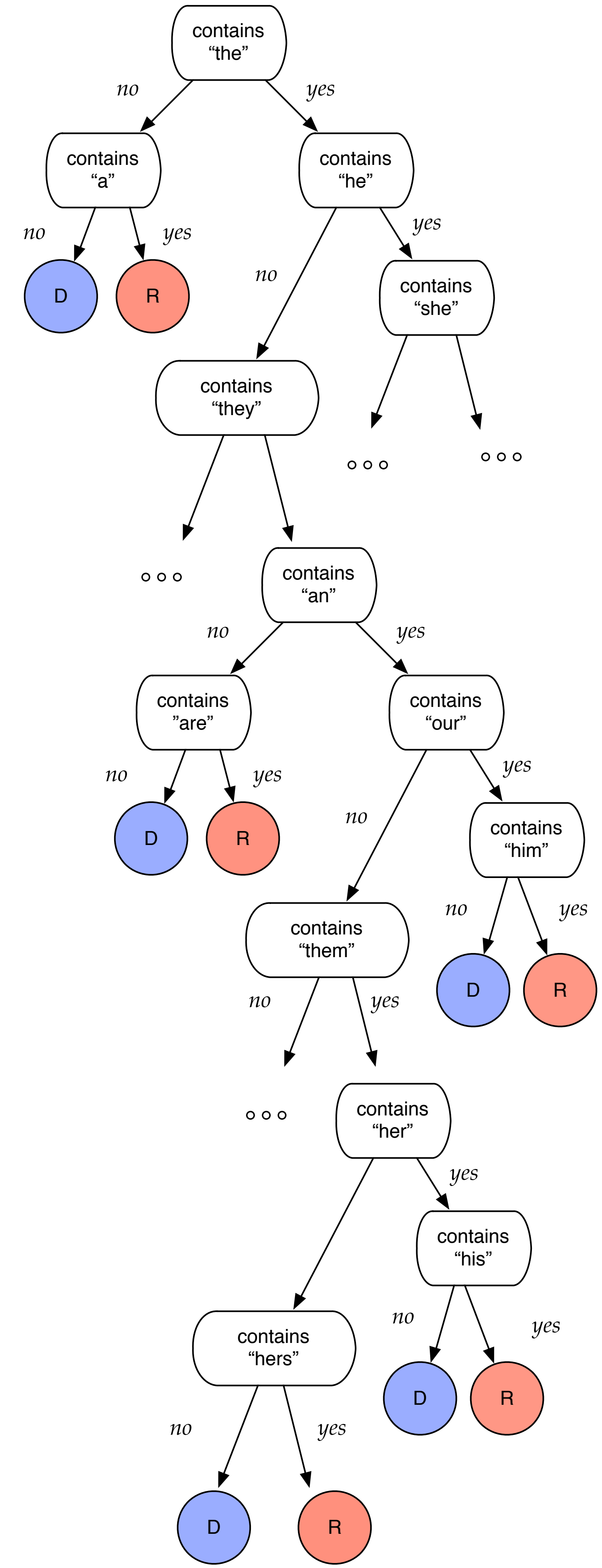
Interpretability

- Decision trees are considered a relatively "interpretable" model, since they can be post-processed in a sequence of decisions

- *If self-reported location = Berkeley and "benghazi" = false, then y = Democrat*

Interpretability

- Manageable for trees of small depth, but not deep trees (each layer = one additional rule)
- Even in small trees, potentially many disjunctions (or for each terminal node)



- Low bias: decision trees can perfectly match the training data (learning a perfect path through the conjunctions of features to recover the true y.
- High variance: because of that, they're very sensitive to whatever data you train on, resulting in very different models on different data

Solution: train many models

- Bootstrap aggregating (bagging) is a method for reducing the variance of a model by averaging the results from multiple models trained on slightly different data.

- Bagging creates multiple versions of your dataset using the bootstrap (sampling data uniformly and with replacement)

Bootstrapped data

original	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
rep 1	x3	x9	x1	x3	x10	x6	x2	x9	x8	x1
rep 2	x7	x9	x1	x1	x4	x9	x10	x7	x5	x6
rep 3	x2	x3	x5	x8	x9	x8	x10	x1	x2	x4
rep 4	x5	x1	x10	x5	x4	x2	x1	x9	x8	x10

Train one decision tree on each replicant and average the predictions (or take the majority vote)

De-correlating further

- Bagging is great, but the variance goes down when the datasets are independent of each other. If there's one strong feature that's a great predictor, then the predictions will be dependent because they all have that feature

- Solution: for each trained decision tree, only use a random subset of features.

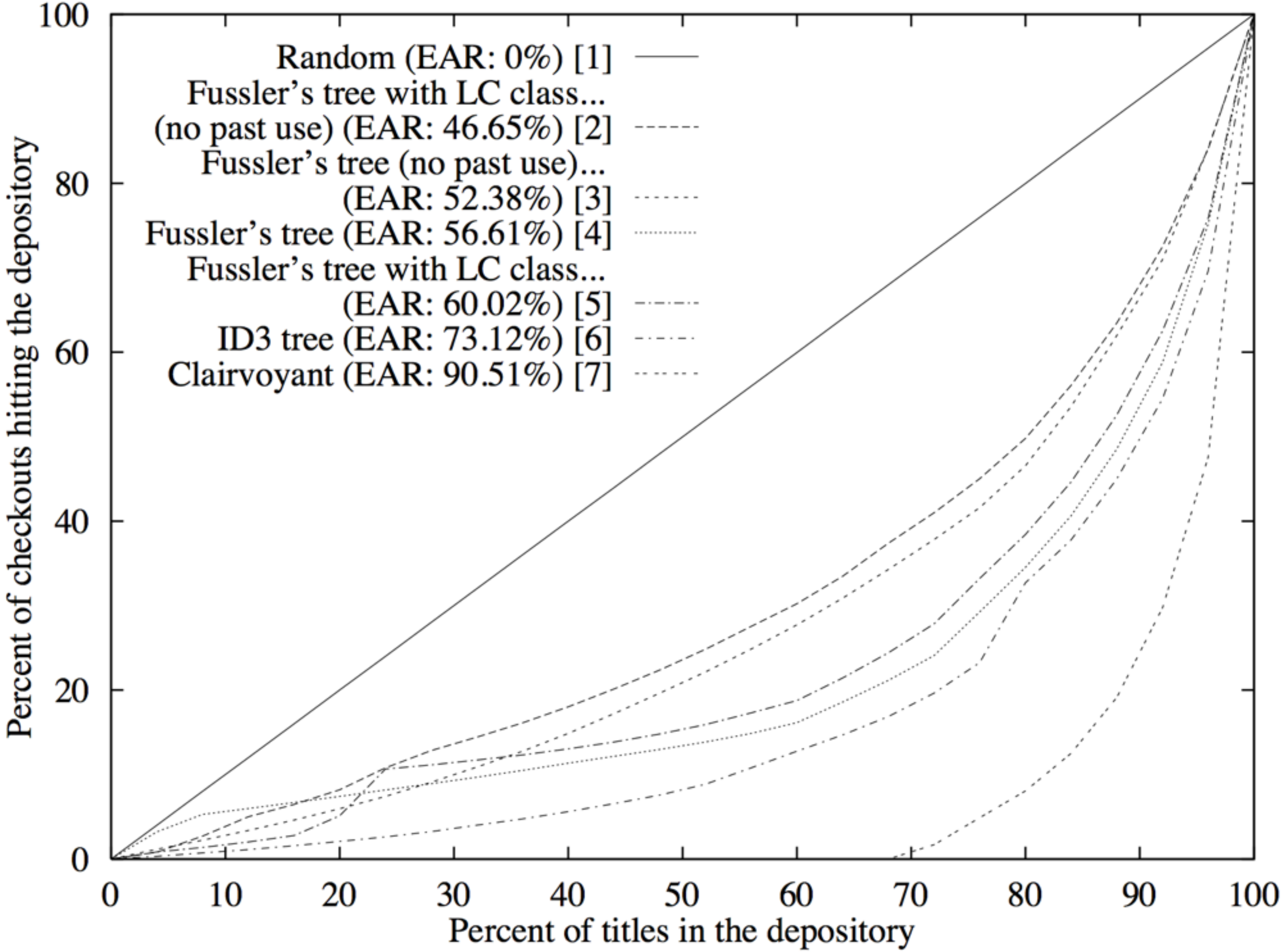
Random forest

Algorithm 11.2: RandomForest(D, T, d) – train an ensemble of tree models from bootstrap samples and random subspaces.

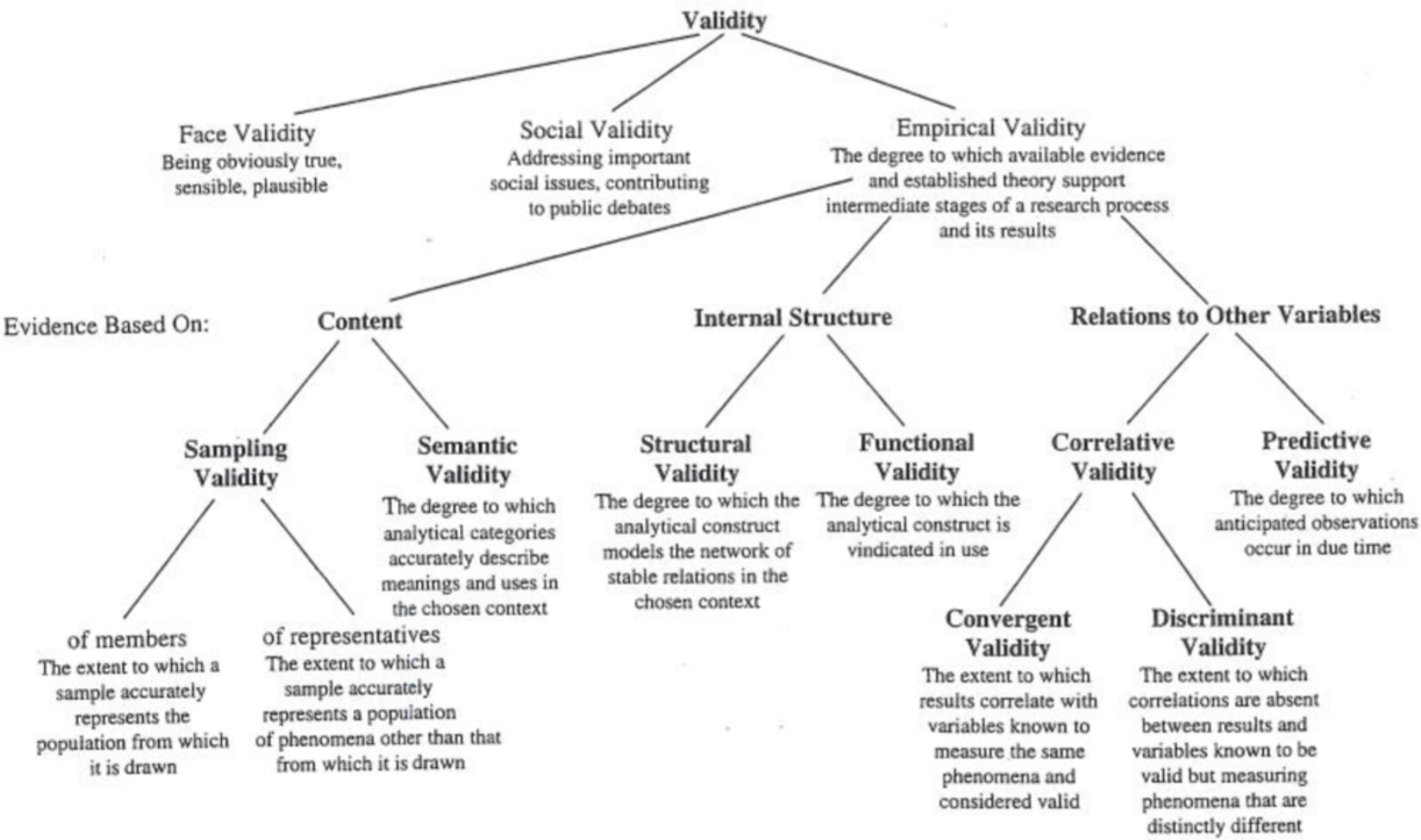
Input : data set D; ensemble size T; subspace dimension d.
Output : ensemble of tree models whose predictions are to be combined by voting or averaging.

```
for t = 1 to T do
    build a bootstrap sample D_t from D by sampling |D| data points with replacement;
    select d features at random and reduce dimensionality of D_t accordingly;
    train a tree model M_t on D_t without pruning;
end
return {M_t | 1 ≤ t ≤ T}
```

Criterion	*Description* *Example values [number of attested values]*
CHECKOUT HISTORY	Number of times the book circulated in the past. 0 times, 1 time, 9 times, 1898 times [90 values]
LAST USE	Number of months since the last use in the past. 0 months, 1 month, 108 months, never used [110 values]
LC CLASS	Alphabetic prefix of the Library of Congress call number. Harvard University keeps some titles under an older classification scheme. Such titles are given an "LC class" by prefixing the Widener prefix with "WID". A, PQ, WID ECON [486 values]
PUBLICATION DATE	Date of publication of the book. 1789, 1900, 1986 [357 values]
LANGUAGE	Language in which book is written. English, Swahili, Achinese [127 values]
COUNTRY	Country in which the book was published, following the Library of Congress specification, in which states of the US and certain other sub-national units are classified as countries. Australia, West Germany, Massachusetts [276 values]



Random (EAR: 0%) [1]
Fussler's tree with LC class...
(no past use) (EAR: 46.65%) [2]
Fussler's tree (no past use)...
(EAR: 52.38%) [3]
Fussler's tree (EAR: 56.61%) [4]
Fussler's tree with LC class...
(EAR: 60.02%) [5]
ID3 tree (EAR: 73.12%) [6]
Clairvoyant (EAR: 90.51%) [7]
Percent of checkouts hitting the depository
Percent of titles in the depository
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80
90
100



Krippendorff (2004)

Project proposal, due 2/19

- Collaborative project (involving 2 or 3 students), where the methods learned in class will be used to draw inferences about the world and critically assess the quality of those results.

- Proposal (2 pages):

    - outline the work you're going to undertake
    - formulate a hypothesis to be examined
    - motivate its rationale as an interesting question worth asking
    - assess its potential to contribute new knowledge by situating it within related literature in the scientific community. (cite 5 relevant sources)
    - who is the team and what are each of your responsibilities (everyone gets the same grade)