DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 22 2020

Outline

- Naïve Bayes classifier
  - Laplace smoothing
- Feature selection
  - Wrappers, filters, embedded methods
- Decision trees
  - Information gain

Generative vs Discriminative

- Generative model
  - Given X and Y, learns the joint probability $P(X,Y)$
  - Can generate more examples from distribution
  - Examples: LDA, Naïve Bayes, language models (GPT-2)

$$P[X,Y] = \underbrace{P[X=x \mid Y=k]}\,\underbrace{P[Y=k]}_{\text{PRIOR}}$$

- Discriminative model
  - Given X and Y, learns a decision function for classification
  - Examples: logistic regression, kNN

Learning Joint Distributions

Step 1:

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Step 2:

Then, fill in each row with:

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	**0.25**
1	1	1	0.10

Fraction of all records in which A and B are true but C is false

d features, binary $\Rightarrow$ Table 2^d

Naïve Bayes Classifier

Idea: Use the training data to estimate

$$P(X \mid Y) \quad \text{and} \quad P(Y) \; .$$

Then, use Bayes rule to infer $P(Y|X_{\text{new}})$ for new data

---

PRIOR Easy to estimate from data

JOINT PROB

Impractical, but necessary

$$\underbrace{\mathrm{P}[Y = k | X = x]}_{\text{max.}} \overset{\text{BAYES TH}}{=} \frac{\boxed{\mathrm{P}[Y = k]}\boxed{\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d | Y = k]}}{\boxed{\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d]}}$$

Unnecessary, as it turns out

CONSTANT

- Recall that estimating the joint probability distribution $P(X_1, X_2, \ldots, X_d \mid Y)$ is not practical

Naïve Bayes Classifier

Problem: estimating the joint density isn't practical

- Severely overfits, as we saw before

COND INDEPENDENCE

However, if we make the assumption that the attributes are independent given the class label, estimation is easy!

$$\underbrace{P[X_1 = x_1 \cap \ldots \cap X_d = x_d \mid Y = k]}_{\text{JOINT PROB}} = \prod_{j=1}^{d} \underbrace{P[X_j = x_j \mid Y = k]}_{\text{ESTIMATE } d \cdot k \text{ PROB}}$$

$$P[Y = k \mid X = x] = \frac{P[Y = k] \cdot \prod_{j=1}^{d} P[X_j = x_j \mid Y = k]}{P[X = x]}$$

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

LABEL

Sky	**Temp**	**Humid**	**Wind**	**Water**	**Forecast**	**Play?**
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

...

$\mathrm{P}(\neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

...

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$P(\text{play}) = ?$ 3/4

$P(\text{Sky} = \text{sunny} \mid \text{play}) = ?$

$P(\text{Humid} = \text{high} \mid \text{play}) = ?$

...

$P(\neg\text{play}) = ?$ 1/4

$P(\text{Sky} = \text{sunny} \mid \neg\text{play}) = ?$

$P(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$

...

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = ?$ 1

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$ 0

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

<u>Sky</u>	<u>Temp</u>	<u>Humid</u>	<u>Wind</u>	<u>Water</u>	<u>Forecast</u>	<u>Play?</u>
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$ $\quad$ $\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$ $\quad$ $\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 0$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$ 2/3 $\quad$ $\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

... $\quad$ ...

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\text{play}) = 3/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 1$

$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = 2/3$

$\ldots$

$\mathrm{P}(\neg\text{play}) = 1/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = 0$

$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$ 1

$\ldots$

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

EXAMPLE:

$$X \sim \begin{array}{ccc} \text{Red} & \text{Blue} & \text{Green} \\ \end{array} \left(\begin{array}{ccc} 5 & 0 & 7 \end{array} \right)$$

$$P[\text{Red}] = \frac{5}{12}$$

$$P[\text{Blue}] = 0$$

$$P[\text{Green}] = \frac{7}{12}$$

LAPLACE SMOOTHING

$$X \sim \begin{array}{ccc} \text{Red} & \text{Blue} & \text{Green} \\ \end{array} \left(\begin{array}{ccc} 6 & 1 & 8 \end{array} \right)$$

$$P[\text{Red}] = \frac{6}{15}$$

$$P[\text{Blue}] = \frac{1}{15}$$

$$P[\text{Green}] = \frac{8}{15}$$

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

- Fix by using Laplace smoothing:
  - Adds 1 to each count

$$\underbrace{P(X_j = v \mid Y = k)} = \frac{c_v + 1}{\sum_{v' \in \text{values}(X_j)} c_{v'} + \underbrace{|\text{values}(X_j)|}}$$

where

- c_v is the count of training instances with a value of v for attribute j and class label k
- $|\text{values}(X_j)|$ is the number of values X_j can take on

Using the Naïve Bayes Classifier

- Now, we have

$$P[Y = k | X = x] = \frac{P[Y = k]\overbrace{P[X_1 = x_1 \wedge \cdots \wedge X_d = x_d | Y = k]}^{\text{TRAINING}}}{\boxed{P[X_1 = x_1 \wedge \cdots \wedge X_d = x_d]}} \quad C$$

This is constant for a given instance, and so irrelevant to our prediction

GIVEN x in TESTING

$$\log P[Y=k | X=x] = \underbrace{\log P[Y=k] + \sum_{j=1}^{d} \log P[X_j = x_j | Y=k]}\ - \log C$$

PICK k THAT MAX PROB

Naïve Bayes Classifier

TRAIN:

- For each class label k
  1. Estimate prior $\pi_k = P[Y = k]$ from the data
  2. For each value v of attribute X_j
     - Estimate $\mathrm{P}[X_j = v | Y = k]$

TEST

- Classify a new point via:

$$h(\mathbf{x}) = \arg\max_{y_k} \log P(Y = k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite it

Comparison to LDA

- Similarity to LDA
  - Both are generative models
  - They both estimate:

$$P[X = x \text{ and } Y = k] = P[X = x | Y = k] P[Y = k]$$

- Difference from LDA
  - LDA assumes feature densities are normal
  - LDA assumes same variances for all classes
  - Naïve Bayes makes the conditional independence assumption

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, etc.*
- Add terms to Medline abstracts (e.g. "Conscious Sedation" [E03.250])
- Classify business names by industry
- Classify student essays as *A/B/C/D/F*
- Classify email as *Spam/Other*
- Classify email to tech staff as *Mac/Windows/ ...*
- Classify pdf files as *ResearchPaper/Other*
- Determine authorship of documents
- Classify movie reviews as *Favorable/Unfavorable/Neutral*
- Classify technical papers as *Interesting/Uninteresting*
- Classify jokes as *Funny/NotFunny*
- Classify websites of companies by Standard Industrial Classification (SIC) code

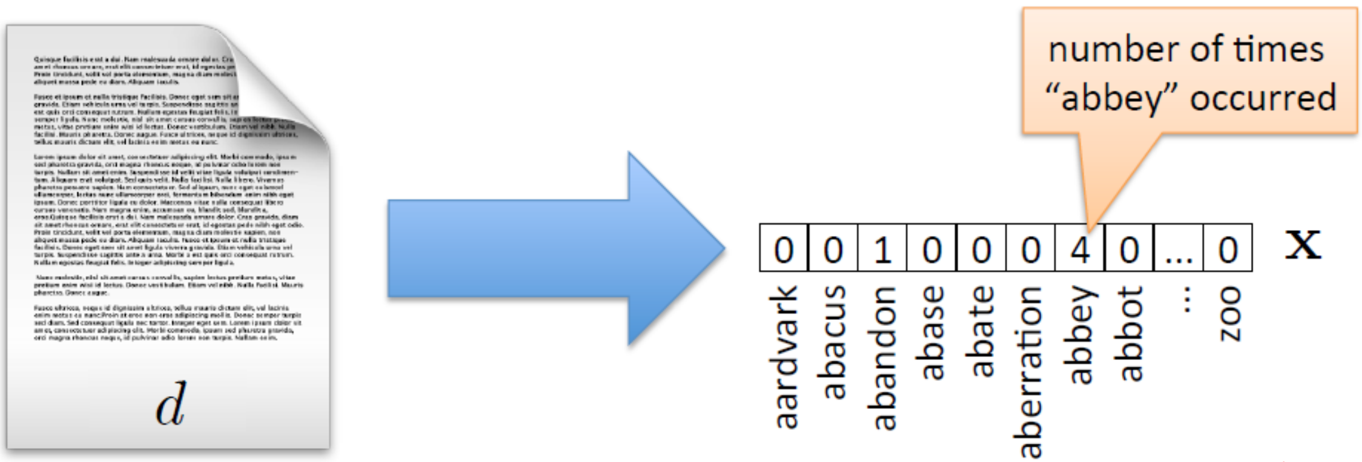
Bag of Words Representation

(BOW)

Represent document d as a vector of word counts $\mathbf{x}$

- x_j represents the count of word j in the document
  - $\mathbf{x}$ is sparse (few non-zero entries)

DOC TEXT



- Naïve Bayes learns the distribution of each word per class
- Naïve Bayes becomes a linear classifier under multi-nomial distribution

Naïve Bayes Summary

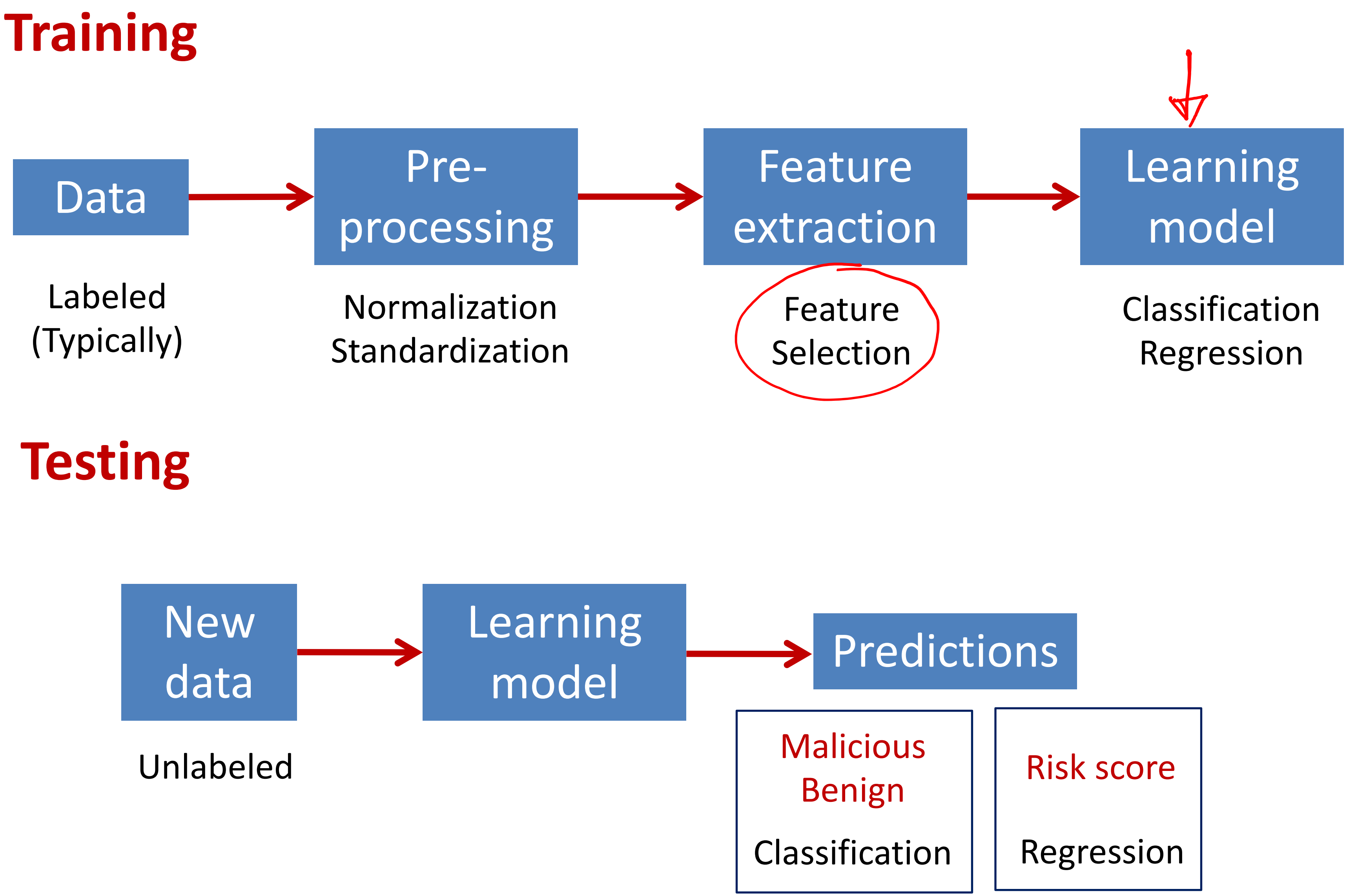
Advantages:

- Fast to train (single scan through data)
- Fast to classify
- Not sensitive to irrelevant features
- Handles real and discrete data
- Handles streaming data well

Disadvantages:

- Assumes independence of features

Supervised Learning Process



Feature selection

- *Feature Selection*
  - Process for choosing an optimal subset of features according to a certain criteria

- Why we need Feature Selection:
  1. To improve performance (in terms of speed, predictive power, simplicity of the model).
  2. To visualize the data for model selection.
  3. To reduce dimensionality and remove noise.

Methods for Feature Selection

- Wrappers

  FORWARD SELECTION

  - Select subset of features that gives best prediction accuracy (using cross-validation)
  - Model-specific
- Filters

  INDEPENDENT DECISION ON EACH FEATURE

  - Compute some statistical metrics (correlation coefficient, information gain)
  - Select features with statistics higher than threshold
- Embedded methods
  - Feature selection done as part of training
  - Example: Regularization (Lasso, L1 regularization)

Filters

Principle: *replace evaluation of model with quick to compute statistics* $J(X_f)$

k	$J(X_k)$
35	0.846
42	0.811
10	0.810
654	0.611
22	0.443
59	0.388
...	...
212	0.09
39	0.05

For each feature X_f

- Compute $J(X_f)$

End

Rank features according to $J(X_f)$

Choose manual cut-off point

Examples of filtering criterion

- The Information Gain with the target variable $J(X_f) = I(X_f; Y)$
- The correlation with the target variable
- Feature importance

Embedded methods: Regularization

Principle: the classifier performs feature selection as part of the learning procedure

Example: the logistic LASSO (Tibshirani, 1996)

$$f(\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x})}} = P(Y = 1|\boldsymbol{x})$$

With Error Function:

$$E = \underbrace{-\sum_{i=1}^{N} \{y_i \log f(\boldsymbol{x}_i) + (1 - y_i) \log(1 - f(\boldsymbol{x}_i))\}}_{\text{Cross-entropy error}} + \underbrace{\lambda \sum_{f=1}^{d} |w_f|}_{\text{Regularizing term}}$$

Pros:
- Performs feature selection as part of learning the procedure

Outline

- Naïve Bayes classifier
  - Laplace smoothing
- Feature selection BOOK CHAPTER
  - Wrappers, filters, embedded methods
- Decision trees
  - Information gain

Sample Dataset

- Columns denote features X_i
- Rows denote labeled instances $< x_i, y_i >$
- Class label denotes whether a tennis game was played

$< x_i, y_i >$

Categorical data

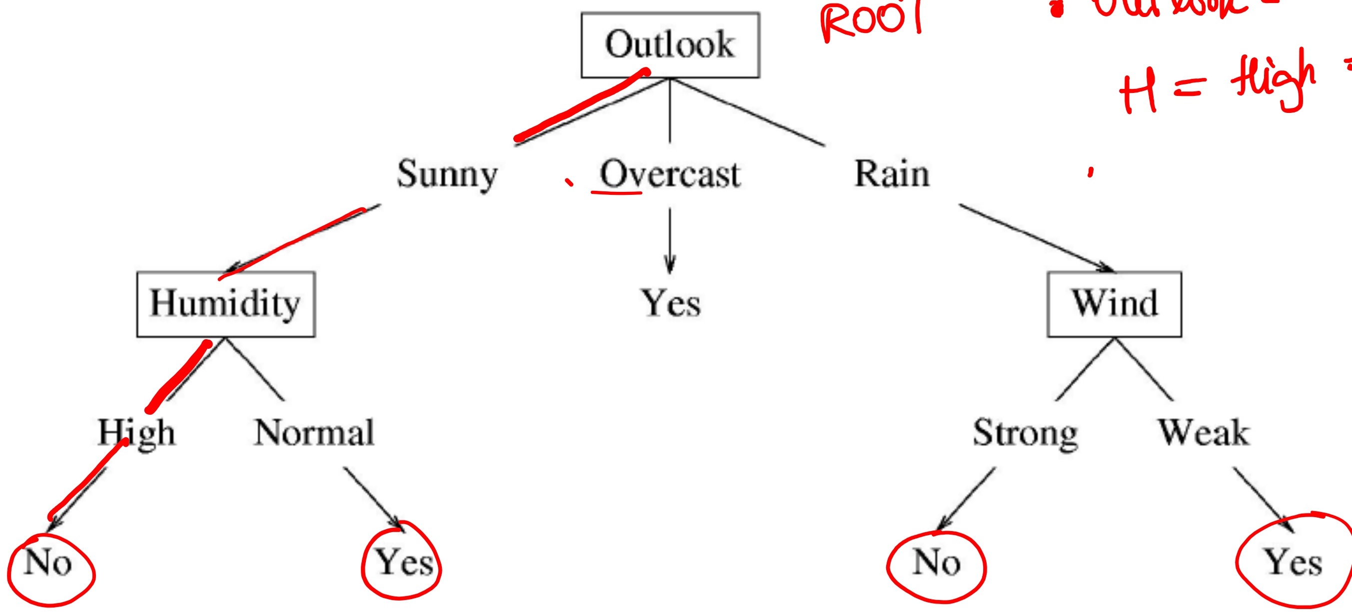
Predictors				Response
Outlook	**Temperature**	**Humidity**	**Wind**	**Class**
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Decision Tree

- A possible decision tree for the data:

ROOT

• Outlook = Sunny ∧ H = High ⇒ NO

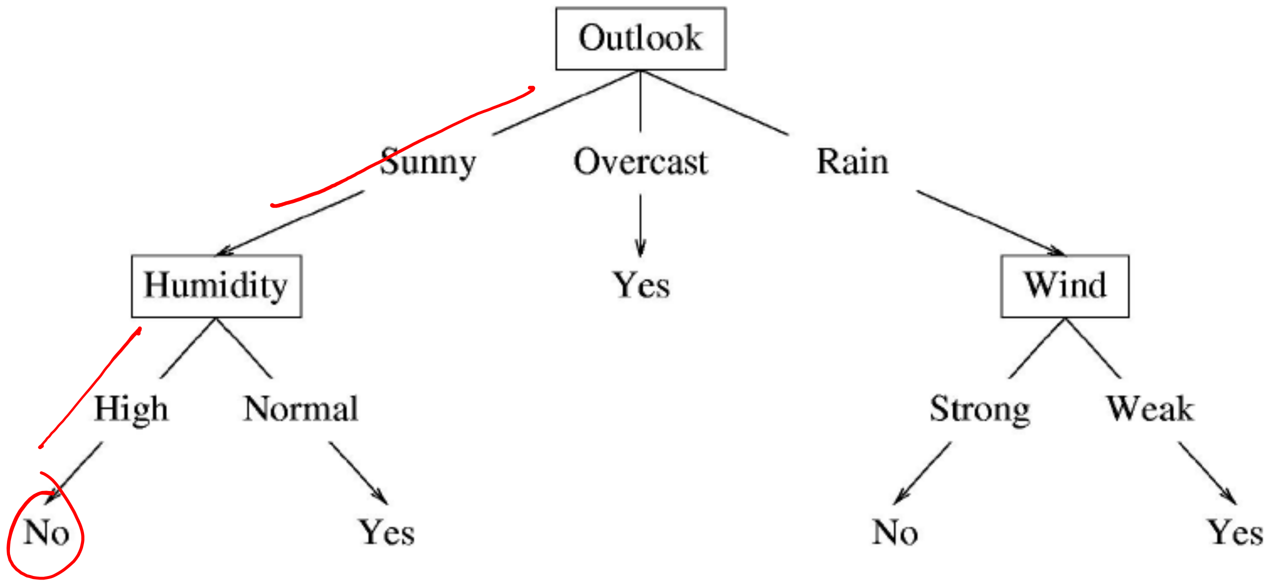


LEAF

- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid x \in \text{leaf})$)

Decision Tree

- A possible decision tree for the data:

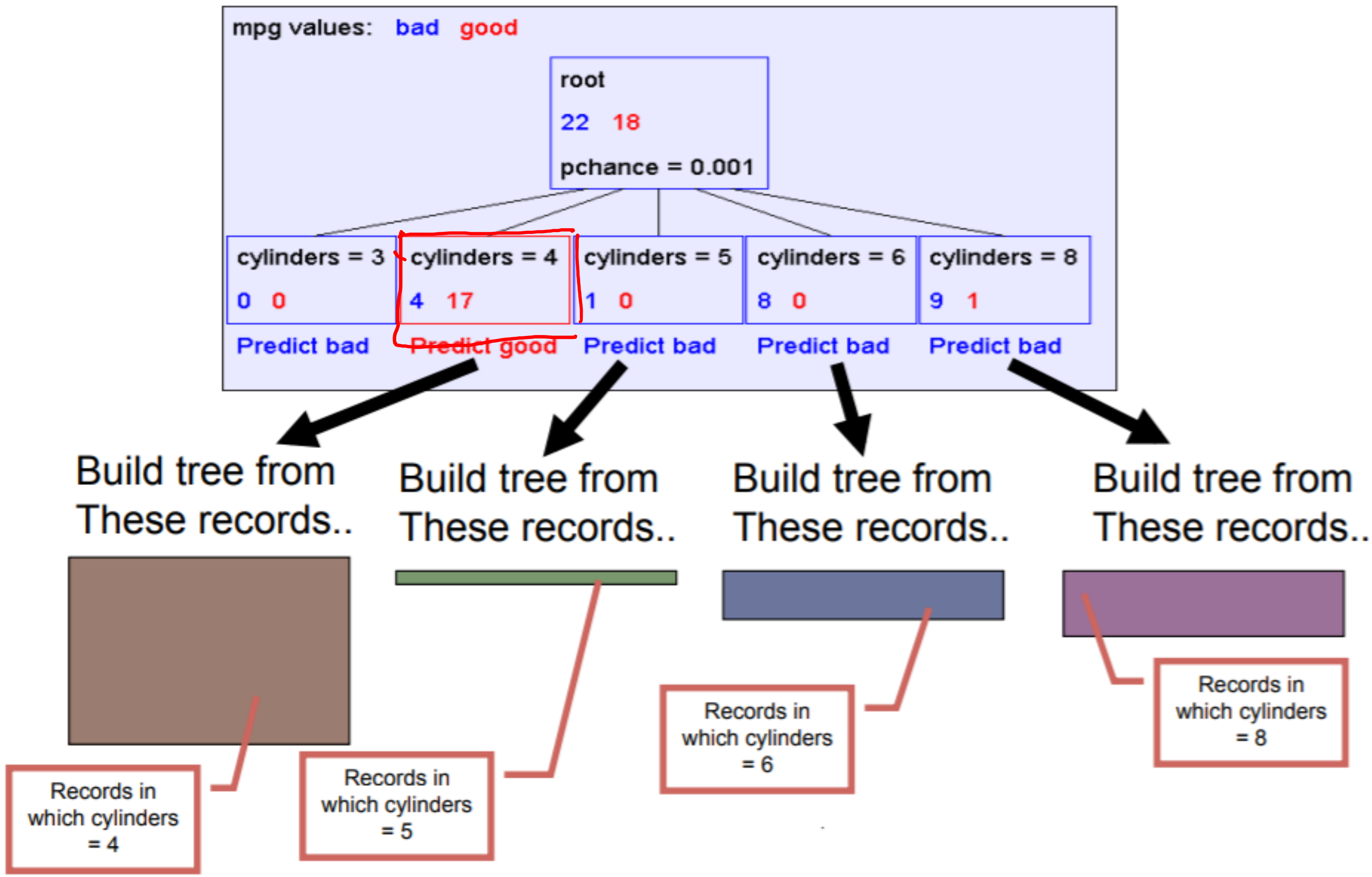


- What prediction would we make for
  <outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

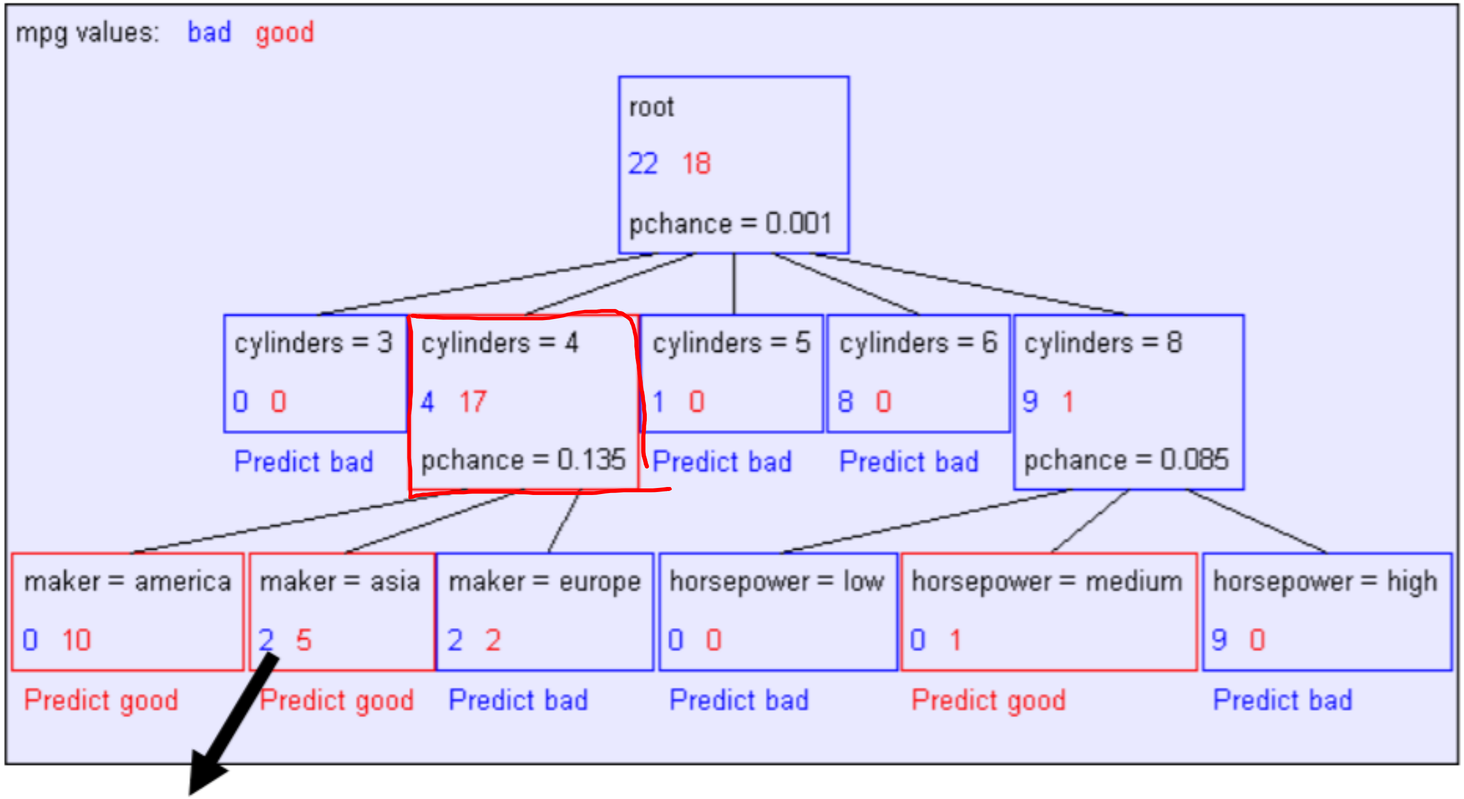
Learning Decision Trees

- Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest '76]
- Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on **next best attribute (feature)** → DIFF ALGORITHMS
  - Recurse

Key Idea: Use Recursion Greedily



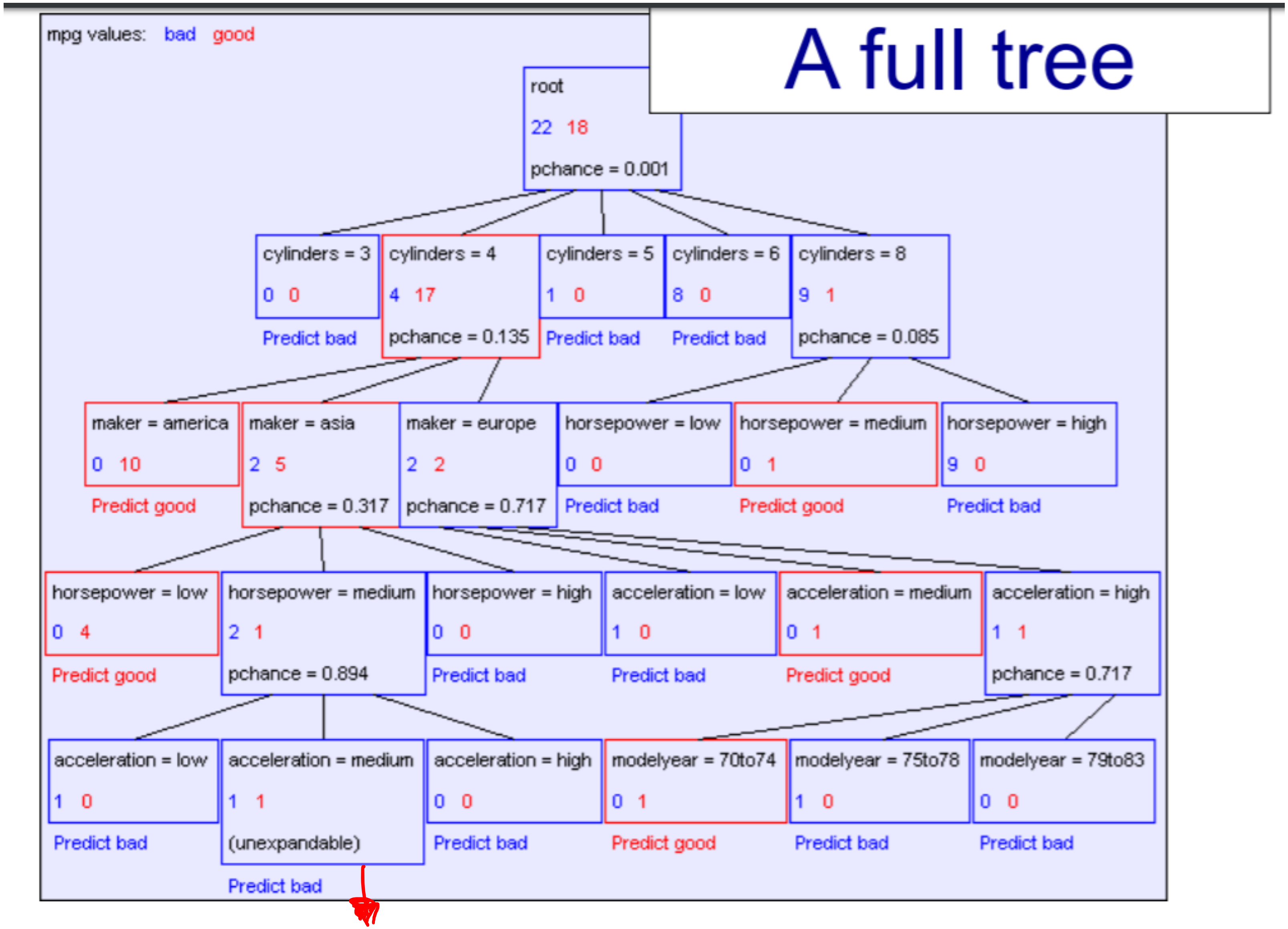
Second Level



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

Full Tree

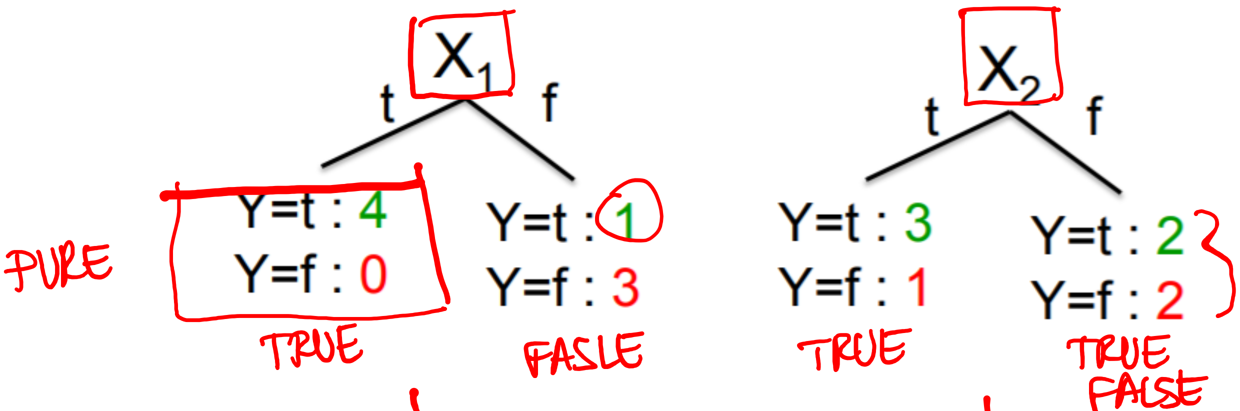

mpg values: bad good
A full tree
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = medium
0 1
Predict good
horsepower = high
9 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
Predict bad
acceleration = low
1 0
Predict bad
acceleration = medium
0 1
Predict good
acceleration = high
1 1
pchance = 0.717
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
acceleration = high
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad

Splitting

Would we prefer to split on X_1 or X_2?

LABEL

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F



Idea: use counts at leaves to define probability distributions, so we can measure uncertaintv!

ERROR: 1

ERROR: 3

Use entropy-based measure (Information Gain)

Entropy

Suppose X can have one of *m* values… $V_1, V_2, \ldots V_m$

$P(X=V_1) = p_1$	$P(X=V_2) = p_2$	….	$P(X=V_m) = p_m$

What's the smallest possible number of bits, on average, per symbol, needed to transmit a stream of symbols drawn from X's distribution? It's

$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m$$

$$= \underbrace{-\sum_{j=1}^{m} p_j \log_2 p_j}$$

H(X) = The entropy of X

- "High Entropy" means X is from a uniform (boring) distribution
- "Low Entropy" means X is from varied (peaks and valleys) distribution

Entropy Examples

$$H(X) = -\sum_{j=1}^{r} p_j \log_2 p_j$$

① $X \sim \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{matrix} \to \text{values} \\ \to \text{prob} \end{matrix}$ UNIFORM

$$H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = -\log_2\frac{1}{2} = 1 \quad \text{MAX}$$

② $Y \sim \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \to$ PEAKED

$H(Y) = 0$ MIN

$X \sim \begin{pmatrix} 1 & 2 \dots & n \\ \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}$

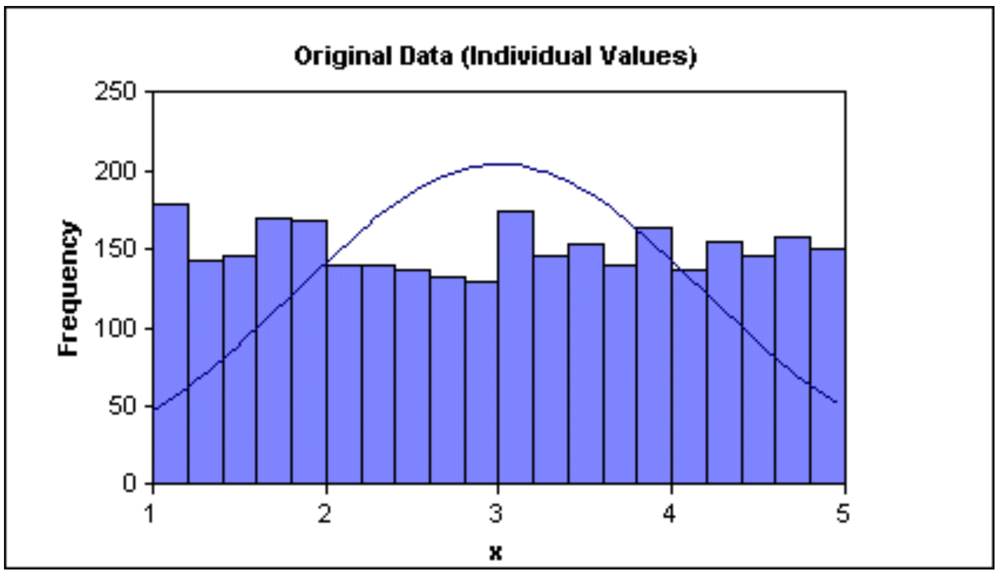
$$H(X) = -\sum \frac{1}{n}\log_2\frac{1}{n}$$
$$= -\log_2\frac{1}{n} = \log_2 n$$

③ $Z \sim \begin{pmatrix} 0 & 1 \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}$

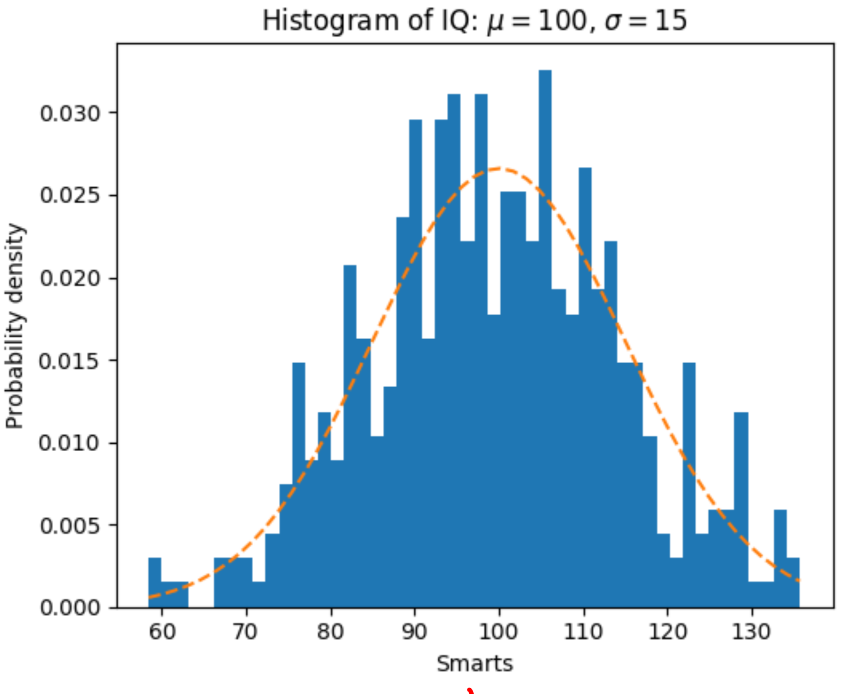
$$H(Z) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.52$$

High/Low Entropy

Which distribution has high entropy?



HIGH



LOW

Conditional Entropy

Suppose I'm trying to predict output Y and I have input X

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Let's assume this reflects the true probabilities

E.G. From this data we estimate

- *P(LikeG = Yes) =* 1/2
- *P(Major = Math & LikeG = No) =* 1/4
- *P(Major = Math) =* 1/2
- *P(LikeG = Yes | Major = History) =* 0

Note:

- *H(X) =* 1.5
- *H(Y) =* 1

$$\begin{pmatrix} M & CS & H \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

$$H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{2}{4}\log_2\frac{1}{4}$$

$$= \frac{1}{2} + 1 = \frac{3}{2}$$

Conditional Entropy

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Specific Conditional Entropy:

$\boxed{H(Y|X=v)}$ = **The entropy of** Y **among only those records in which** X **has value** v

Example:

- $H(Y|X=Math) =$ 1
- $H(Y|X=History) =$ 0
- $H(Y|X=CS) =$ 0

Conditional Entropy

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Conditional Entropy:

$\boxed{H(Y|X)}$ = The average specific conditional entropy of *Y*

= if you choose a record at random what will be the conditional entropy of *Y*, conditioned on that row's value of *X*

= Expected number of bits to transmit *Y* if both sides will know the value of *X*

$$= \underbrace{\Sigma_j Prob(X=v_j) \boxed{H(Y \mid X = v_j)}}$$

Conditional Entropy

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

$H(Y) = 1$

Definition of Conditional Entropy:

$H(Y|X)$ = The average conditional entropy of Y

$= \underbrace{\Sigma_j Prob(X=v_j)\, H(Y \mid X = v_j)}$

Example:

v_j	$Prob(X=v_j)$	$H(Y \mid X = v_j)$
Math	1/2	1
History	1/4	0
CS	1/4	0

$H(Y|X) = \frac{1}{2}$

Information Gain

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Information Gain:

IG(Y|X) = **I must transmit** *Y*. **How many bits on average would it save me if both ends of the line knew** *X***?**

$$\boxed{IG(Y|X) = H(Y) - H(Y \mid X)}$$

Example:

- **H(Y) =** 1
- **H(Y |X) =** $\frac{1}{2}$
- **Thus IG(Y |X) =** $\frac{1}{2}$

$IG(Y|X_1) \ldots\ldots \quad IG(Y|X_d)$

Learning Decision Trees

- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute:

$$\arg\max_i IG(X_i) = \arg\max_i \underbrace{H(Y) - H(Y \mid X_i)}_{IG(Y|X_i)}$$

- Recurse

$\boxed{\text{ID3 algorithm}}$ uses Information Gain
Information Gain reduces uncertainty on Y

Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
- Thanks!