Information Theory, Statistics, and Decision Trees

Léon Bottou

COS 424 – 4/6/2010

Summary

1. Basic information theory.

2. Decision trees.

3. Information theory and statistics.

I. Basic Information theory

Why do we care?

Information theory

– Invented by Claude Shannon in 1948

A Mathematical Theory of Communication.
Bell System Technical Journal, October 1948.

– The "quantity of information" measured in "bits".

– The "capacity of a transmission channel".

– Data coding and data compression.

Information gain

– A derived concept.

– Quantify how much information we acquire about a phenomenon.

– A justification for the Kullback-Leibler divergence.

The coding paradigm

Intuition

The quantity of information of a message is the length
of the smallest code that can represent the message.

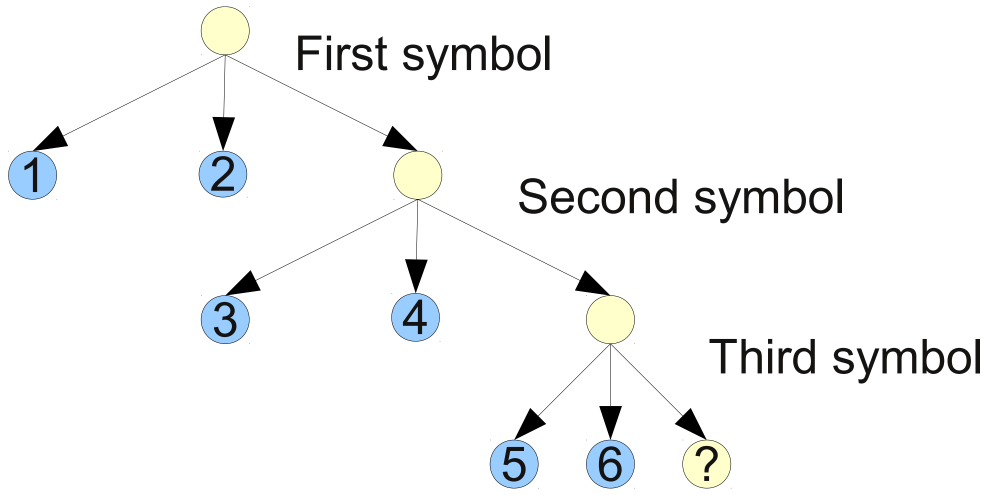
Paradigm

– Assume there are n possible messages $i = 1 \dots n$.
– We want a signal that indicates the occurrence of one of them.
– We can transmit an alphabet of r symbols.
  For instance a wire could carry $r = 2$ electrical levels.
– The code for message i is a sequence of l_i symbols.

Properties

– Codes should be *uniquely decodable*.
– Average code length for a message: $\sum_{x=1}^{n} p_i \, l_i$.

Prefix codes



- Messages 1 and 2 have codes one symbol long ($l_i = 1$).
- Messages 3 and 4 have codes two symbols long ($l_i = 2$).
- Messages 5 and 6,have codes three symbols long ($l_i = 2$).
- There is an unused three symbol code. That's inefficient.

Properties

- Prefix codes are uniquely decodable.
- There are trickier kinds of uniquely decodable codes,
  e.g. $a \mapsto 0, b \mapsto 01, c \mapsto 011$ versus $a \mapsto 0, b \mapsto 10, c \mapsto 110$.

Kraft inequality

Uniquely decodable codes satisfy

$$\sum_{x=1}^{n} \left(\frac{1}{r}\right)^{l_i} \leq 1$$

– All uniquely decodable codes satisfy this inequality.

– If integer code lengths l_i satisfy this inequality,
there exists a prefix code with such code lengths.

Consequences

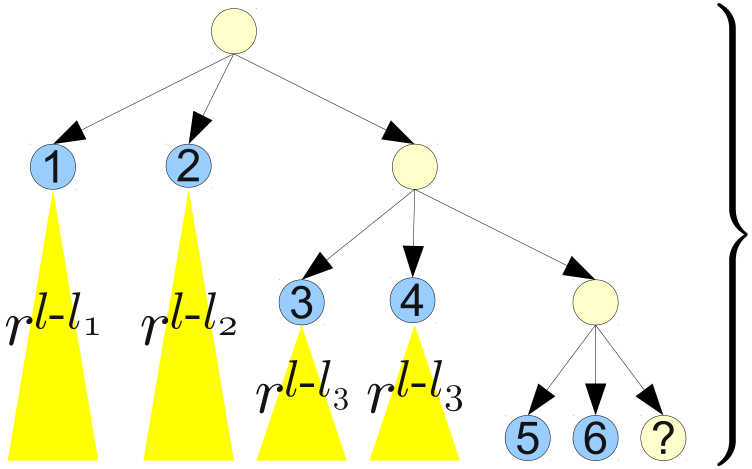
– If some messages have short codes, others must have long codes.

– To minimize the average code length:
  - give short codes to high probability messages.
  - give long codes to low probability messages.

– Equiprobable messages should have similar code lengths.

Kraft inequality for prefix codes

Prefix codes satisfy Kraft inequality



$$\sum_i r^{l-l_i} \leq r^l \iff \sum_i \left(\frac{1}{r}\right)^{-l_i} \leq 1$$

All uniquely decodable codes satisfy Kraft inequality

– Proof must deal with infinite sequences of messages.

Given integer code lengths l_i:

– Build a balanced r-ary tree of depth $l = \max_i l_i$.

– For each message, prune one subtree at depth l_i.

– Kraft inequality ensures that there will be enough branches left to define a code for each message.

Redundant codes

Assume $\sum_i r^{-l_i} < 1$

– There are leftover branches in the tree.

– There are codes that are not used, or

– There are multiple codes for each message.

For best compression, $\sum_i r^{-l_i} = 1$

– This is not always possible with integer code lengths l_i.

– But we can use this to compute a lower bound.

Lower bound for the average code length

Choose code lengths l_i such that

$$\min_{l_1 \dots l_n} \sum_i p_i \, l_i \quad \text{subject to} \quad \sum_i r^{-l_i} = 1, \quad l_i > 0$$

– Define $s_i = r^{-l_i}$, that is, $l_i = -\log_r(s_i)$.
– Maximize $C = \sum p_i \log_r(s_i)$ subject to $\sum_i s_i = 1$
– We get $\frac{\partial C}{\partial s_i} = \frac{p_i}{s_i \, log(r)} = Constant$, that is $s_i \propto p_i$.
– Replacing in the constraint gives $s_i = p_i$.

Therefore

$$l_i = -\log_r(p_i) \quad \text{and} \quad \sum_i p_i \, l_i = -\sum_i p_i \log_r(p_i)$$

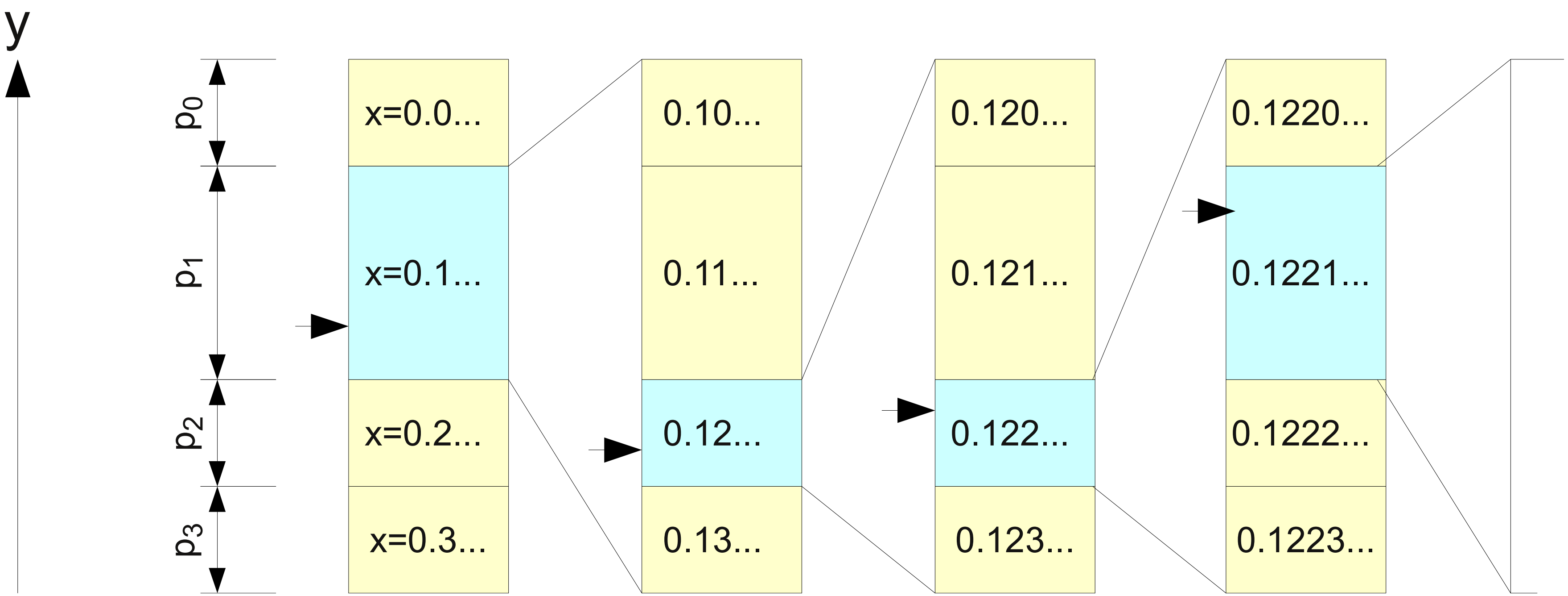
Fractional code lengths

– What does it mean to code a message on 0.5 symbols?

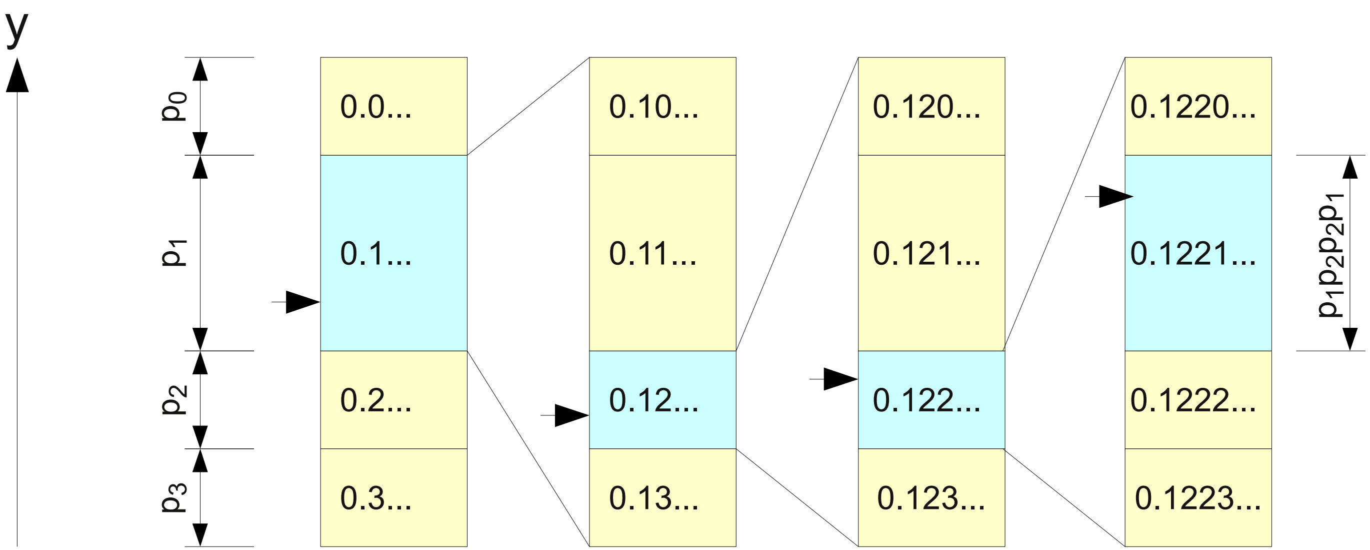
Arithmetic coding

– An infinite sequence of messages $i_1, i_2, \ldots$ can be viewed as a number $x = 0.i_1 i_2 i_3 \ldots$ in base n.

– An infinite sequence of symbols $c_1, c_2, \ldots$ can be viewed as a number $y = 0.c_1 c_2 c_3 \ldots$ in base r.



Arithmetic coding



To encode a sequence of L messages $i_1, \dots, i_L$.

– The code y must belong to an interval of size $\displaystyle\prod_{k=1}^{L} p_{i_k}$.

– It is sufficient to specify $l(i_1 i_2 \dots i_L) = \lceil \displaystyle\sum_{k=1}^{L} \log_r(p_{i_k}) \rceil$ digits of y.

Arithmetic coding

To encode a sequence of L messages $i_1, \dots, i_L$.

– It is sufficient to specify $l(i_1 i_2 \dots i_L) = \left\lceil -\sum_{k=1}^{L} \log_r(p_{i_k}) \right\rceil$ digits of y.

– The average code length per message is

$$\frac{1}{L} \sum_{i_1 i_2 \dots i_L} p_{i_1} \dots p_{i_L} \left\lceil \sum_{k=1}^{L} -\log_r(p_{i_k}) \right\rceil$$

$$\xrightarrow{L\to\infty} \sum_{i_1 i_2 \dots i_L} p_{i_1} \dots p_{i_L} \sum_{k=1}^{L} \frac{\log_r(p_{i_k})}{L}$$

$$= \frac{1}{L} \sum_{k=1}^{L} \sum_{i_1 \dots i_L \setminus i_k} \Big(\prod_{h \neq k} p_{i_h} \Big) \sum_{i_k=1}^{r} p_{i_k} \log p_{i_k} = -\sum_i p_i \log p_i$$

Arithmetic coding reaches the lower bound when $L \to \infty$.

Quantity of information

Optimal code length: $l_i = -\log_r(p_i)$.

Optimal expected code length: $\sum p_i l_i = -\sum p_i \log_r(p_i)$.

Receiving a message x with probability p_x:

– The *acquired information* is $h(x) = -log_2(p_x)$ bits.

– An informative message is a surprising message!

Expecting a message X with distribution $p_1 \dots p_n$:

– The *expected information* is $H(X) = -\sum_{x \in \mathcal{X}} p_x \log_2(p_x)$ bits.

– This is also called *entropy*.

These are two distinct definitions!

Note how we switched to logarithms in base two.

This is a multiplicative factor: $\log_2(p) = \log_r(p) \log_2(r)$.

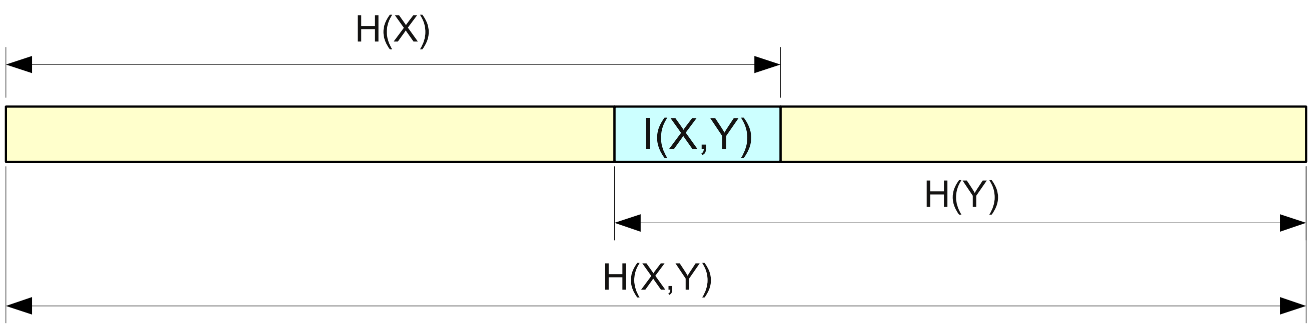
Choosing base 2 defines a unit of information: the bit.

Mutual information

		Hair color					
		Dark	Auburn	Red	Blond	***Marginal***	***Information***
Eyes color	Brown	68	119	26	7	***37.2%***	***1.83***
	Hazel	15	54	14	10	***15.7%***	
	Green	5	29	14	16	***10.8%***	
	Blue	20	84	17	94	***36.3%***	
	Marginal	***18.2%***	***48.3%***	***12.0%***	***21.5%***		
	Information	***1.80***					

		Hair color					
		Dark	Auburn	Red	Blond		
Eyes color	Brown	11.5%	20.1%	4.4%	1.2%		
	Hazel	2.5%	9.1%	2.4%	1.7%		
	Green	0.8%	4.9%	2.4%	2.7%		
	Blue	3.4%	14.2%	2.9%	15.9%		
					Joint information		***3.45***
					Mutual information		***0.18***

– Expected information: $H(X) = -\sum_i P(X=i)\, \log P(X=i)$
– Joint information: $H(X,Y) = \sum_{i,j} \mathbb{P}(X=i, Y=j)\, \log P(X=i, Y=j)$
– Mutual information: $I(X,Y) = H(X) + H(Y) - H(X,Y)$



II. Decision trees

Car mileage

Predict which cars have better mileage than 19mpg.

mpg	cyl	disp	hp	weight	accel	year	name
15.0	8	350.0	165.0	3693	11.5	70	buick skylark 320
18.0	8	318.0	150.0	3436	11.0	70	plymouth satellite
15.0	8	429.0	198.0	4341	10.0	70	ford galaxie 500
14.0	8	454.0	220.0	4354	9.0	70	chevrolet impala
15.0	8	390.0	190.0	3850	8.5	70	amc ambassador dpl
14.0	8	340.0	160.0	3609	8.0	70	plymouth cuda 340
18.0	4	121.0	112.0	2933	14.5	72	volvo 145e
22.0	4	121.0	76.00	2511	18.0	72	volkswagen 411
21.0	4	120.0	87.00	2979	19.5	72	peugeot 504
26.0	4	96.0	69.00	2189	18.0	72	renault 12
22.0	4	122.0	86.00	2310	16.0	72	ford pinto
28.0	4	97.0	92.00	2288	17.0	72	datsun 510
13.0	8	440.0	215.0	4735	11.0	73	chrysler new yorker

...

Questions

Many questions can distinguish cars

– How many cylinders? (3,4,5,8)
– Displacement greater than 200 cu in? (yes, no)
– Displacement greater than x cu in? (yes, no)
– Weight greater than x lbs? (yes, no)
– Model name longer than x characters (yes, no)
– etc. . .

Which question brings the most information about the task?

– Build contingency table.
– Compare mutual informations $I(Question, Mpg > 19)$.

	Possible answers			
	ansA	**ansB**	**ansC**	**ansD**
mpg>19	12	23	65	5
mpg≤19	18	12	4	4

Mutual information

Consider a contingency table, x_{ij}.

– $1 \leq j \leq p$ refers to the question answers X.

– $1 \leq i \leq n$ refers to the target values Y.

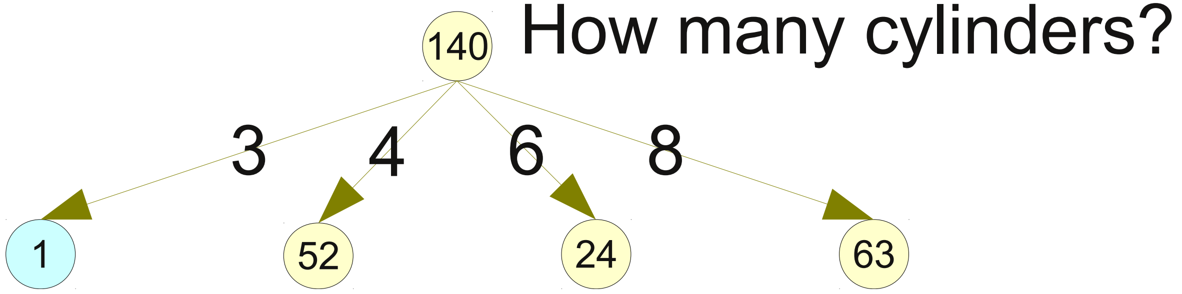
	ansA	**ansB**	**ansC**	**ansD**
mpg>19	12	23	65	5
mpg≤19	18	12	4	4

Let $x_{i\bullet} = \sum_{j=1}^{p} x_{ij}$, $x_{\bullet j} = \sum_{i=1}^{n} x_{ij}$, and $x_{\bullet\bullet} = \sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij}$.

Mutual information:

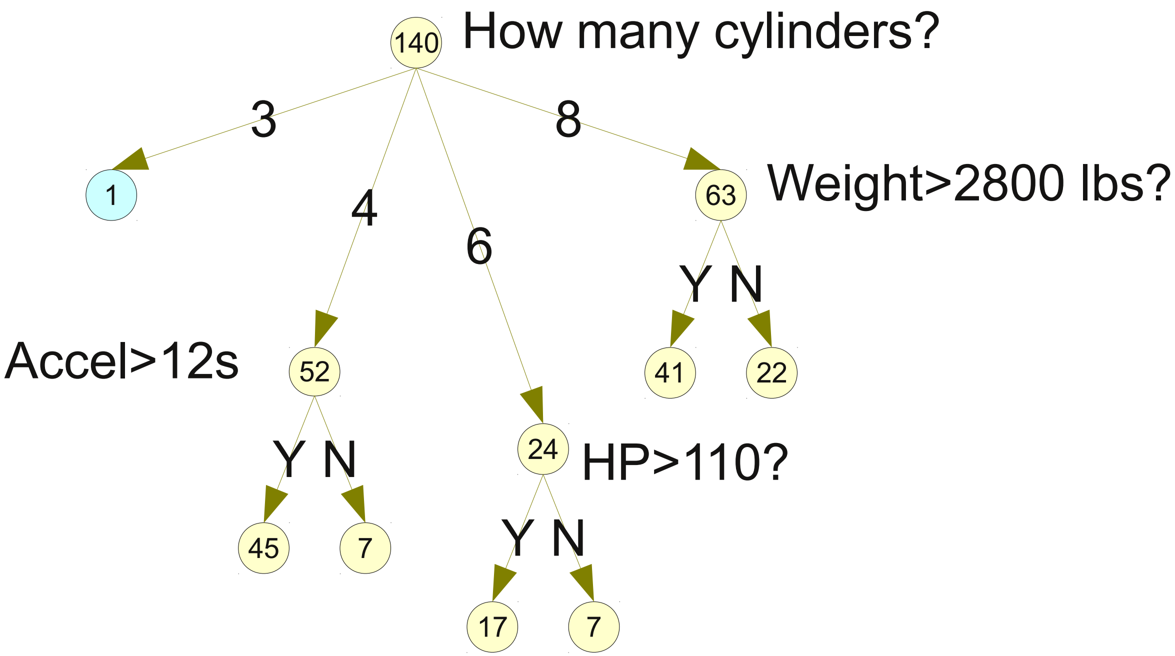
$$
\begin{aligned}
I(X,Y) &= -H(X,Y) + H(X) + H(Y) \\
&= \sum_{ij} \frac{x_{ij}}{x_{\bullet\bullet}} \log \frac{x_{ij}}{x_{\bullet\bullet}} - \sum_{j} \frac{x_{\bullet j}}{x_{\bullet\bullet}} \log \frac{x_{\bullet j}}{x_{\bullet\bullet}} - \sum_{i} \frac{x_{i\bullet}}{x_{\bullet\bullet}} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}}
\end{aligned}
$$

Decision stump



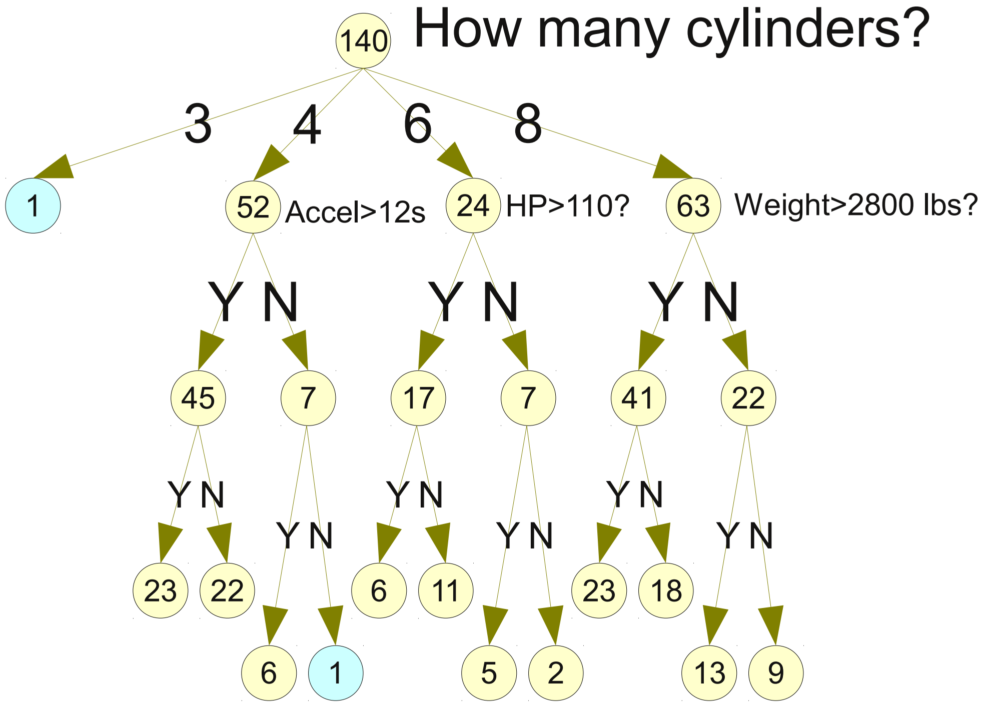
- The question generates a partition of the examples.
- Now we can repeat the process for each node:
  - build the contingency tables.
  - pick the most informative question.

Decision trees



Until all leafs contain a single car.

Decision trees



. . .

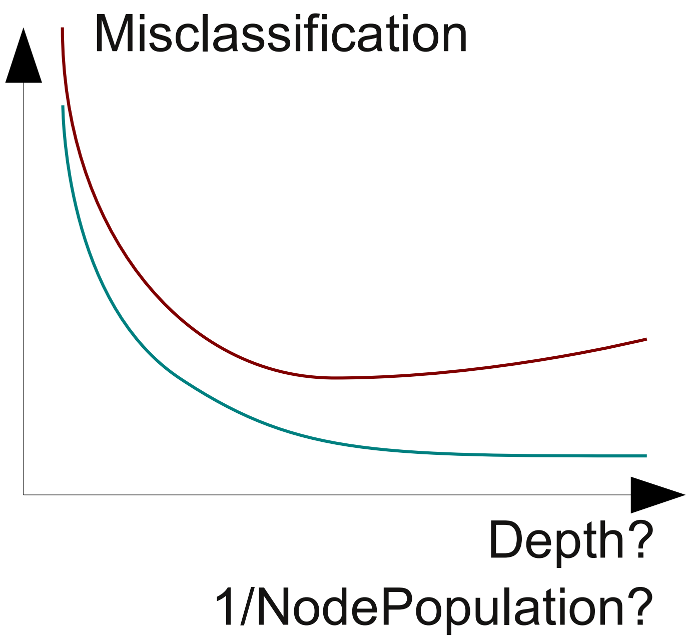
Then label each leaf with class $MPG > 19$ or $MPG \leq 19$.

We can now say if a car does more than 19mpg by asking a few questions.

But that is learning by heart!

Pruning the decision tree

We can label each node with its dominant class $MPG > 19$ or $MPG \leq 19$.



The usual picture.

Should we use a validation set?

Which stopping criterion?
– the node depth?
– the node population?

The χ^2 independence test

We met this test when studying correspondence analysis (lecture 10).

n

		Hair color				
		Dark	Auburn	Red	Blond	***Totals***
Eyes color	Brown	68	119	26	7	***220***
	Hazel	15	54	14	10	***93***
	Green	5	29	14	16	***64***
	Blue	20	84	17	94	***215***
	Totals	***108***	***286***	***71***	***127***	***592***

p

$$x_{i\bullet} = \sum_{j=1}^{p} x_{ij} \qquad x_{\bullet j} = \sum_{i=1}^{n} x_{ij} \qquad x_{\bullet\bullet} = \sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij} \qquad E_{ij} = \frac{x_{i\bullet}\, x_{\bullet j}}{x_{\bullet\bullet}}$$

If the rows and columns variables were independent

$\mathcal{X}^2 = \sum_{ij} \frac{(x_{ij} - E_{ij})^2}{E_{ij}}$ would asymptotically follow a χ^2 distribution with $(n-1)(p-1)$ degrees of freedom.

Pruning a decision tree with the χ^2 test

We want to prune nodes when the contingency table suggests that there is no dependence between the question and the target class.

– Compute $\mathcal{X}^2 = \sum_{ij} \frac{(x_{ij} - E_{ij})^2}{E_{ij}}$ for each node.

– Prune if $1 - F_{\chi^2}(X) > p$.

Parameter p could be picked by cross-validation.
But choosing $p = 0.05$ often works well enough.

Conclusion

Good points

– Decision trees run quickly.
– Decision trees can handle all kinds of input variables.
– Decision trees can be interpreted relatively easily.
– Decision trees can handle lots of irrelevant features.

Bad points

– Decision trees are moderately accurate.
– Small changes in the training set can lead to very different trees.
(were we speaking about interpretability. . .)

Notes

– Other names for decision trees: ID3, C4.5, CART.
– Regression tree when the target is continuous.

III. Information theory and statistics

Revisiting decision trees : likelihoods

The tree as a model of $P(Y|X)$

– Estimate $P(Y|X)$ by the target frequencies in the leaf for X.

– We can compute the likelihood of the data in this model.

Likelihood gain when splitting a node

– Let x_{ij} be the contingency table for a node and a question.

– Splitting the node with a question increases the likelihood:

$$\begin{aligned}
\log L_{after} - \log L_{before} &= \sum_{ij} x_{ij} \log \frac{x_{ij}}{x_{\bullet j}} - \sum_{i} x_{i\bullet} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}} \\
&= \sum_{ij} x_{ij} \log \frac{x_{ij}\, x_{\bullet\bullet}}{x_{\bullet\bullet}\, x_{\bullet j}} - \sum_{i} x_{i\bullet} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}} \\
&= \sum_{ij} x_{ij} \log \frac{x_{ij}}{x_{\bullet\bullet}} - \sum_{j} x_{\bullet j} \log \frac{x_{\bullet j}}{x_{\bullet\bullet}} - \sum_{i} x_{i\bullet} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}}
\end{aligned}$$

Compare with slide 19.

Revisiting decision trees : log loss

The tree as a discriminant function

– Define $f(X) = \log \frac{p_X}{1-p_X}$ where p_X is the frequency of positive examples in the leaf corresponding to X.

$$\log\left(1+e^{-yf(X)}\right) = \begin{cases} \log\left(1 - \frac{1-p_X}{p_X}\right) = -\log(p_X) & \text{if } y=1 \\ \log\left(1 - \frac{p_X}{1-p_X}\right) = -\log(1-p_X) & \text{if } y=-1 \end{cases}$$

Log loss reduction when splitting a node

– Let x_{ij} be the contingency table for a node and a question.

$$\begin{aligned} R_{before} - R_{after} &= -\sum_i x_{i\bullet} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}} + \sum_j \sum_i x_{ij} \log \frac{x_{ij}}{x_{\bullet j}} \\ &= \sum_{ij} x_{ij} \log \frac{x_{ij}}{x_{\bullet\bullet}} - \sum_j x_{\bullet j} \log \frac{x_{\bullet j}}{x_{\bullet\bullet}} - \sum_i x_{i\bullet} \log \frac{x_{i\bullet}}{x_{\bullet\bullet}} \end{aligned}$$

Compare with slides 19 and 28.

Note: regression trees use the mean squared loss.

Kullback Leibler divergence

Definition

– KL divergence between a "true distribution" $P(X)$ and an "estimated distribution" $P_\theta(X)$.

$$D(P\|P_\theta) = \int \log \frac{P(x)}{P_\theta(x)} \, dP(x) = \sum_x P(x) \log \frac{P(x)}{P_\theta(x)}$$

$$= \underbrace{-\sum_x P(x) \log \mathbb{P}_\theta(x)}_{H_{approx}} - \underbrace{-\sum_x P(x) \log P(x)}_{H_{opt}}$$

H_{opt} : Optimal coding length for X.

H_{approx} : Expected code length for X when the code is designed for distribution P_θ instead of the true distribution P.

– The KL divergence measures the excess coding bits when the code is optimized for the estimated distribution instead of the true distribution.

Maximum Likelihood

Minimize KL divergence

$$\min_\theta \; D(P\|P_\theta) \;=\; \int \log \frac{P(x)}{P_\theta(x)}\, dP(x) \iff \max_\theta \int \log P_\theta(x)\, dP(x)$$

Maximize Log Likelihood

$$\max_\theta \frac{1}{n} \sum_{i=1}^{n} \log P_\theta(x_i)$$

The log likelihood estimates $Constant - D(P\|P_\theta)$ using the training set.

– Maximizing the likelihood minimizes an estimate of the excess coding bits obtained by coding the training set.

– One hopes to achieve a good coding performance on future data.

The Vapnik-Chervonenkis theory gives confidence intervals for the deviation

$$\left(\int \log P_{\theta^*}(x)\, dP(x)\right) - \left(\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta^*}(x_i)\right)$$