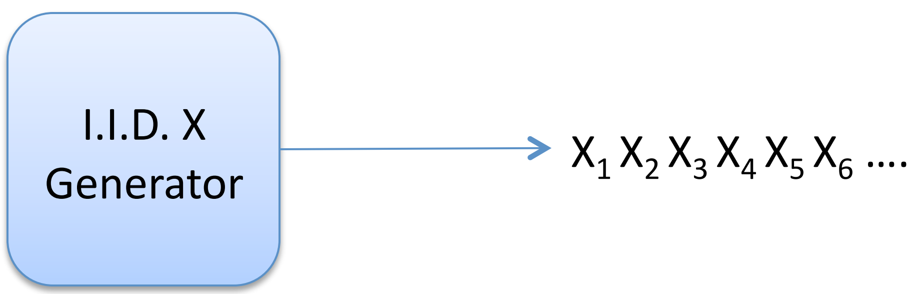
Information Theory and Decision Trees

TexPoint fonts used in EMF.

Entropy

- For a discrete random variable X
- Entropy: H(X) = $-\sum p(x) log_2(p(x))$
- The "average number of bits needed each symbol of X"



Example

- X generates 4 possible symbols {a,b,c,d}
  - P(X = ‘a’) = 0.5
  - P(X = ‘b’) = 0.25
  - P(X = ‘c’) = 0.125
  - P(X = ‘d’) = 0.125

I.I.D. X Generator → **abaabacadabcaadabbaad……**

Naïve Encoding:

a	→	00
b	→	10
c	→	01
d	→	11

Naïve Encoding

- X generates 4 possible symbols {a,b,c,d}
  - P(X = ‘a’) = 0.5
  - P(X = ‘b’) = 0.25
  - P(X = ‘c’) = 0.125
  - P(X = ‘d’) = 0.125

Naïve Encoding:

a	00
b	10
c	01
d	11

Average cost of encoding each symbol?

Length of Encoding symbol ‘a’ * Probability of symbol ‘a’	2 * 0.5	
Length of Encoding symbol ‘b’ * Probability of symbol ‘b’	2 * 0.25	+
…	2 * 0.125	+
…	2 * 0.125	+

= 2 bits

We Can Do Better

- X generates 4 possible symbols {a,b,c,d}
  - P(X = ‘a’) = 0.5
  - P(X = ‘b’) = 0.25
  - P(X = ‘c’) = 0.125
  - P(X = ‘d’) = 0.125

Better Encoding:

a	→	0
b	→	10
c	→	110
d	→	111

Average cost of encoding each symbol?

Length of Encoding symbol ‘a’ * Probability of symbol ‘a’ 1 * 0.5
Length of Encoding symbol ‘b’ * Probability of symbol ‘b’ 2 * 0.25 +
… 3 * 0.125 +
… 3 * 0.125 +

= 1.25 bits

This Encoding is Optimal

We Can Do Better

- X generates 4 possible symbols {a,b,c,d}
  - P(X = ‘a’) = 0.5
  - P(X = ‘b’) = 0.25
  - P(X = ‘c’) = 0.125
  - P(X = ‘d’) = 0.125

Better Encoding:

a	0
b	10
c	110
d	111

Average cost of encoding each symbol?

Length of Encoding symbol ‘a’ * Probability of symbol ‘a’	1 * 0.5	
Length of Encoding symbol ‘b’ * Probability of symbol ‘b’	2 * 0.25	+
…	3 * 0.125	+
…	3 * 0.125	+

= 1.25 bits

But observe that: -Log(0.5) = 1

-Log(0.25) = 2 … etc

Length of Encoding of symbol x = -log(x)

Average Cost of encoding a symbol = $-\sum p(x) log_2(p(x))$ = H(X)

More Generally

- X generates 4 possible symbols {a,b,c,d}
  - P(X = ‘a’) = 0.21
  - P(X = ‘b’) = 0.14
  - P(X = ‘c’) = 0.52
  - P(X = ‘d’) = 0.13

What is the optimal encoding for this?

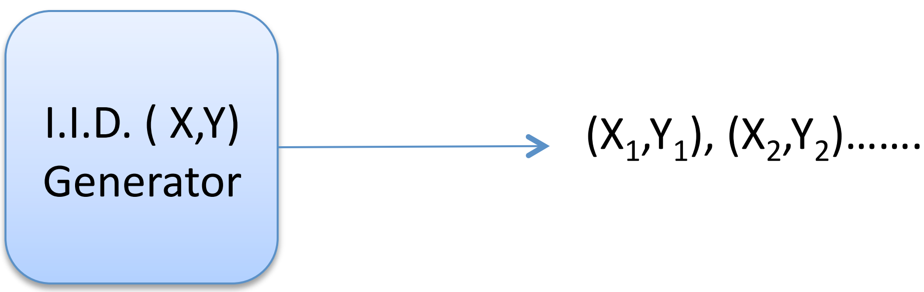
It is possible to attain an average of H(X) bits per symbol, but quite complicated: See **Arithmetic Compression**

Conditional Entropy

- H(X|Y) = $\sum_y p(y) \sum H(X|Y = y)$

  = $\sum_y p(y) \sum_x p(x|y) \log p(x|y)$

Intuition: Average # bits needed to Encode Y, if X is already transmitted



Information Gain

- H(X) = $-\sum p(x)log_2(p(x))$
- H(X|Y) = $\sum_y p(y)\sum H(X|Y=y)$

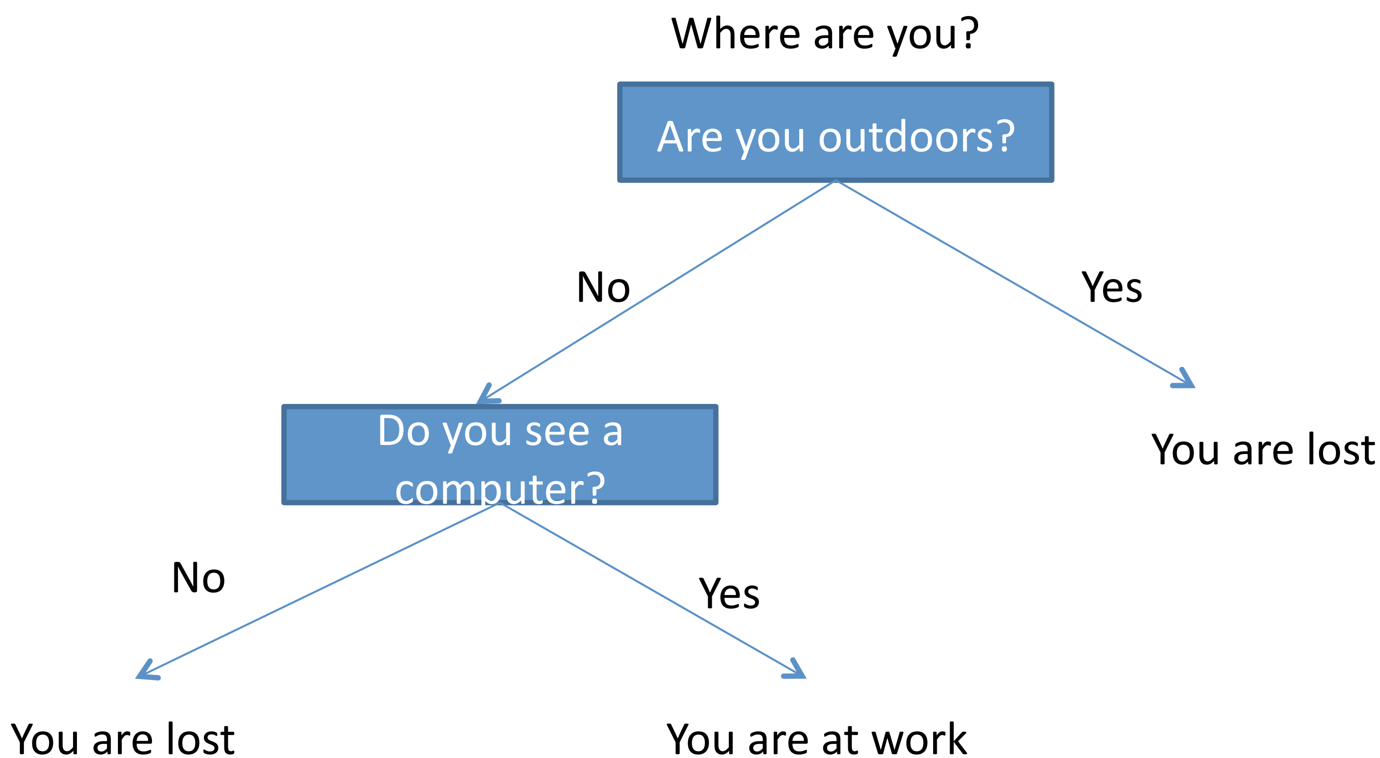
  = $\sum_y p(y)\sum_x p(x|y)\log p(x|y)$

IG(X,Y) = H(X) – H(X|Y)

H(X): # bits to encode X
H(X|Y): # bits to encode X if Y is already transmitted
IG(X,Y): The "number of bits of information" gained about X, by knowing Y

Counter Intuitive Property: IG(X, Y) = IG(Y, X)

Decision Tree



ID3

Heuristic: Pick the variable which provides the most Information Gain about Y

Outdoors	Computer	Lost	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

ID3

Heuristic: Pick the variable which provides the most Information Gain about Y

X1	X2	Y	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

IG(X1,Y) = H(Y) – H(Y| X1)

H(Y) = - (5/10) log(5/10) - 5/10 log(5/10) = 1

H(Y | X1) = P(X1=T) H(Y | X1=T) (4/10) * (1 log(1) – 0 log(0))

P(X1=F) H(Y | X1=F) (6/10) * (-(5/6) log (5/6) - (1/6)log(1/6))

= 0.3900

IG(X1, Y) = 0.6100

ID3

Heuristic: Pick the variable which provides the most Information Gain about Y

X1	X2	Y	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

IG(X2,Y) = H(Y) – H(Y| X2)

H(Y) = - (5/10) log(5/10) - 5/10 log(5/10) = 1

H(Y | X2) = P(X2=T) H(Y | X2=T) (7/10) * (-(2/7) log (2/7) - (5/7)log(5/7))
P(X2=F) H(Y | X2=F) (3/10) * (2/3) log (2/3) - (1/3)log(1/3)

= 0.8797

IG(X2, Y) = 0.1203

ID3

Heuristic: Pick the variable which provides the most Information Gain about Y

Outdoors	Computer	Lost	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

IG(X1, Y) = 0.6100

IG(X2, Y) = 0.1203

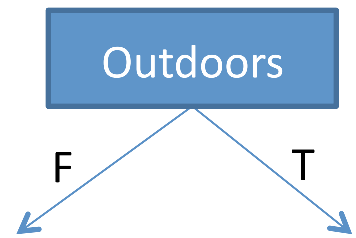
ID3

Heuristic: Pick the variable which provides the most Information Gain about Y

Outdoors	Computer	Lost	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

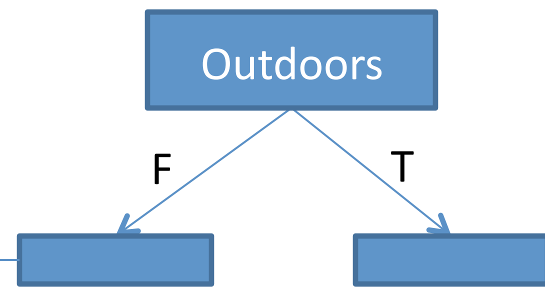
IG(Lost, Outdoors) = 0.6100

IG(Lost, Computer) = 0.1203



ID3

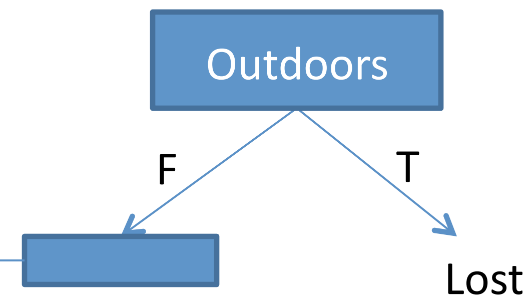
Heuristic: Pick the variable which provides the most Information Gain about Y
Recurse on the branches



Outdoors	Computer	Lost	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

ID3

Heuristic: Pick the variable which provides the most Information Gain about Y
Recurse on the branches



Outdoors	Computer	Lost	Count
T	T	T	2
T	F	T	2
F	T	F	5
F	F	T	1

Chi Square Pruning

- Decision Trees tend to overfit
- Pruning Necessary
- Bottom Up Pruning

Chi Square Pruning

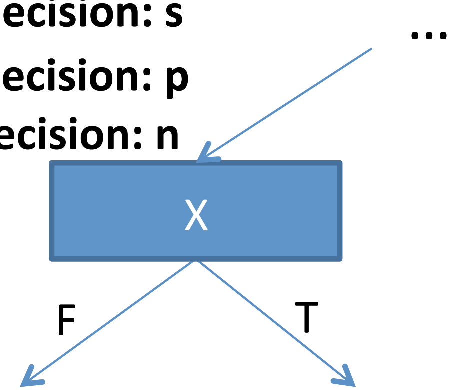
1. Build Complete Tree
2. Consider each “leaf” decision and perform the chi-square test (label vs split variable)

Chi Square Pruning

# of instances entering this decision: s

of + instances entering this decision: p

of - instances entering this decision: n



 # instances here: s_f

of + instances here: p_f

of - instances here: n_f

 # instances here: s_t

of - instances here: p_t

of - instances here: n_t

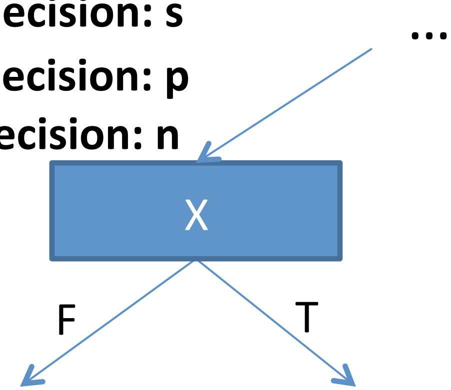
Hypothesis: **X is uncorrelated with the decision**

Chi Square Pruning

# of instances entering this decision: s

of + instances entering this decision: p

of - instances entering this decision: n



 # instances here: s_f

of + instances here: p_f

of - instances here: n_f

 # instances here: s_t

of - instances here: p_t

of - instances here: n_t

Hypothesis: **X is uncorrelated with the decision**

Then p_f should be "close" to (s_f * p/s)

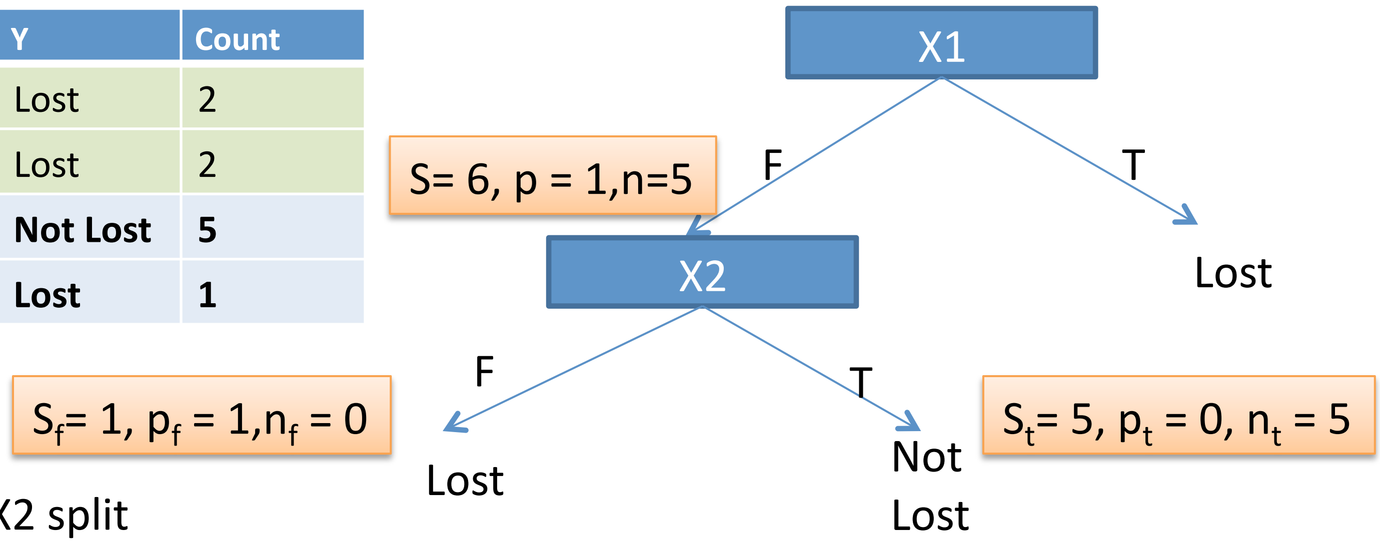
And p_t should be "close" to (s_t * p/s)

Expected counts

Similarly for n_f and n_t

Chi Square Pruning

X1	X2	Y	Count
T	T	Lost	2
T	F	Lost	2
F	**T**	**Not Lost**	**5**
F	**F**	**Lost**	**1**



Consider the X2 split

Y = Lost

Variable Assignment	Real Counts	Expected Counts (S_{x2} * p / S)
X2 = F	1	1/6
X2 = T	0	5/6

Y = Not Lost

Variable Assignment	Real Counts	Expected Counts (S_{x2} * n / S)
X2 = F	0	5/6
X2 = T	5	25/6

Chi Square Pruning

Y = Lost

Variable Assignment	Real Counts	Expected Counts (S_{x2} * p / S)
X2 = F	1	1/6
X2 = T	0	5/6

Y = Not Lost

Variable Assignment	Real Counts	Expected Counts (S_{x2} * n / S)
X2 = F	0	5/6
X2 = T	5	25/6

If uncorrelated, I expect the Real Counts to be close to Expected Counts

Need some kind of measure of "deviation"

$$C = \sum_{X_2} \frac{(\text{Real Count}_{lost} - \text{Expected Count}_{lost})^2}{\text{Expected Count}_{lost}} + \frac{(\text{Real Count}_{notlost} - \text{Expected Count}_{notlost})^2}{\text{Expected Count}_{notlost}}$$

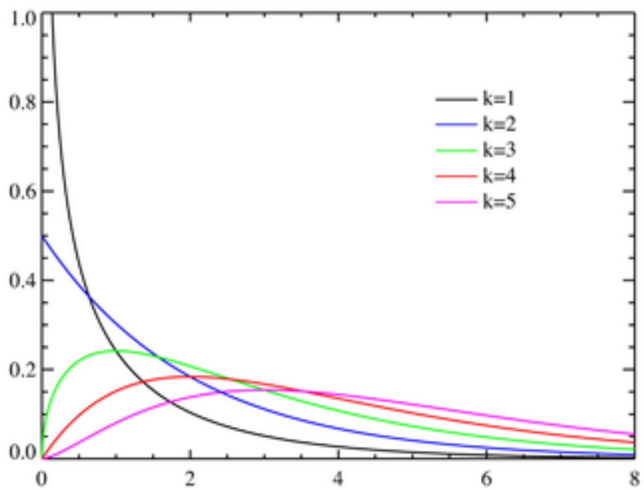
$$C \sim \chi^2((\text{num Y labels} - 1) \times (\text{num X2 labels} - 1))$$

$$C \sim \chi^2(1)$$

Chi Square Pruning

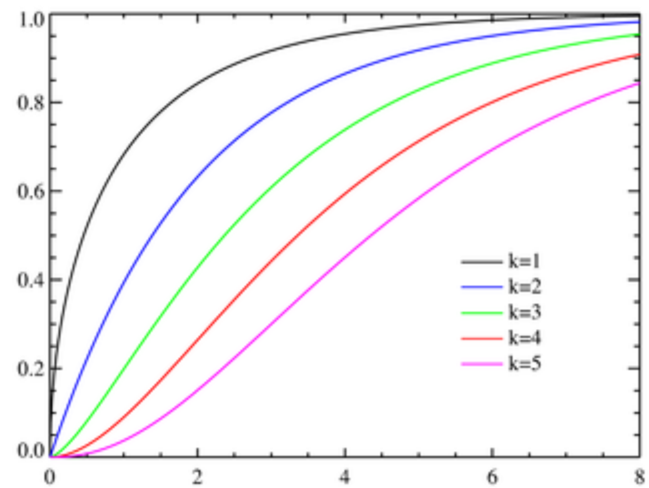
$$c = \sum_{X_2} \frac{(\text{Real Count}_{lost} - \text{Expected Count}_{lost})^2}{\text{Expected Count}_{lost}} + \frac{(\text{Real Count}_{notlost} - \text{Expected Count}_{notlost})^2}{\text{Expected Count}_{notlost}}$$

Intuitively, the smaller C is, the more likely they are uncorrelated.



c = 6

$C \sim \chi^2(1)$



If X2 and Y are uncorrelated,

P(C >= c) is the "probability" that we see such large deviations "by chance".

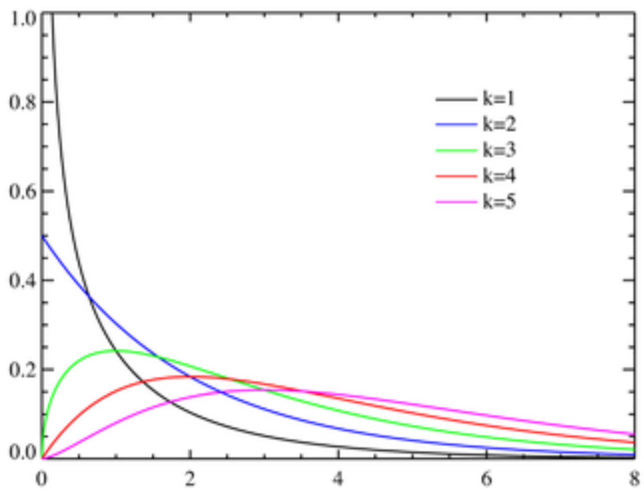
We define "maxPChance" as the "worst chance we are willing to accept"

(Coin Flip Example: we believe coin is unbiased. Then out of 1000 flips,
How many "heads" do you want to see before you stop believing coin is unbiased?)

Chi Square Pruning

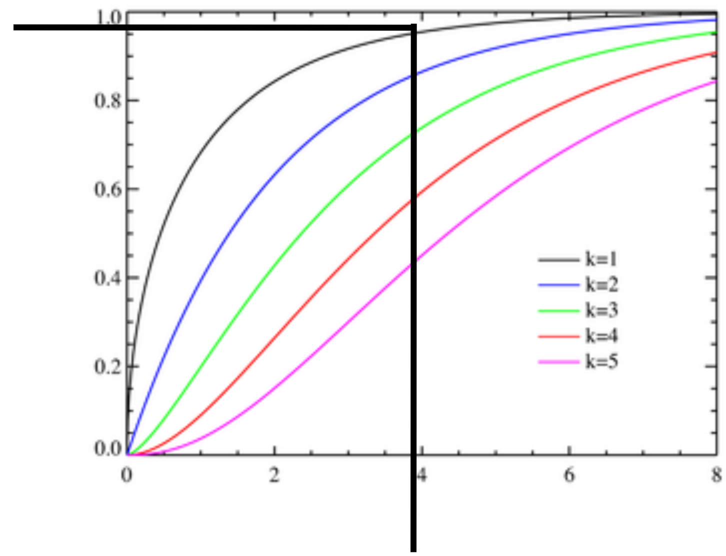
$$c = \sum_{X_2} \frac{(\text{Real Count}_{lost} - \text{Expected Count}_{lost})^2}{\text{Expected Count}_{lost}} + \frac{(\text{Real Count}_{notlost} - \text{Expected Count}_{notlost})^2}{\text{Expected Count}_{notlost}}$$

Intuitively, the smaller C is, the more likely they are uncorrelated.



c = 6

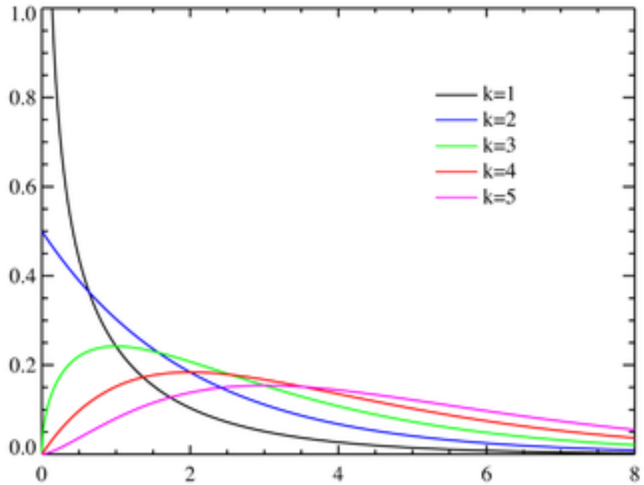
c ~ $\chi^2(1)$



Let maxPchance = 0.05

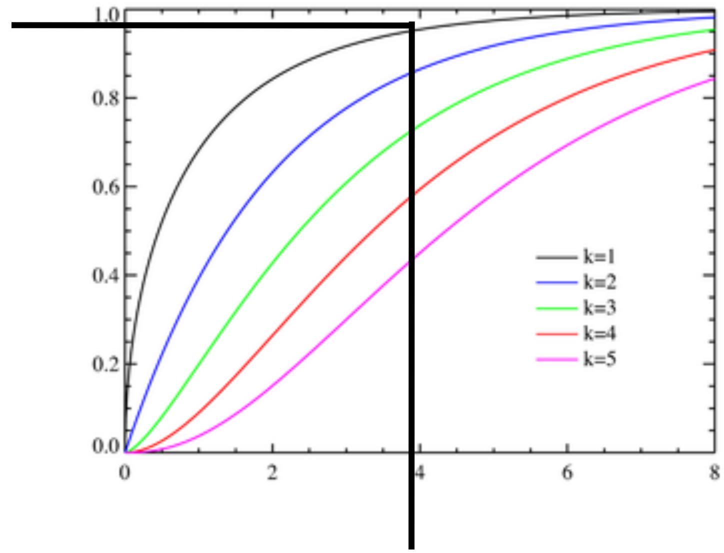
We only stop believing that the splits are "by chance" if the probability of getting a deviation larger than c is < 0.05.

Chi Square Pruning



c = 6

c ~ $\chi^2(1)$



Let maxPchance = 0.05

We only stop believing that the splits are "by chance" if the probability of getting a deviation larger than c is < 0.05.

Look at cdf. P(C <= 3.8415) = 0.95
P(C > 3.8415) = 0.05

If c <= 3.8415 we believe the split is "by chance" and prune the decision
If c > 3.8415 we do not believe the split is "by chance"

Applet

Play With Applet