- "Pr(A=a3) = 0" denotes impossibility; A cannot be a3
- "Pr(A=a3) = 1" denotes certainty; A Must be a3 and cannot be anything else
- Sum over any distribution of its values = 1
  sum over ai's: $\Sigma \Pr(A=a_i) = 1$
- For every value A=ai
  $\Pr(A=a_i) \geq 0$
- $\Pr(A=a_3 \vee B=b_2) =$
  $\Pr(A=a_3) + \Pr(B=b_2) - \Pr(A=a_3 \wedge B=b_2)$
- $\Pr(A=a_3 \mid B=b_2) = \Pr(A=a_3 \wedge B=b_2) / \Pr(B=b_2)$
  provided $\Pr(B=b_2) \neq 0$
- $\Pr(A=a_3 \wedge B=b_2) = \Pr(A=a_3 \mid B=b_2) * \Pr(B=b_2)$

- Random variables A and B are Independent iff
  $P(A \mid B) = P(A)$
- B provides no information about A
  $P(A \mid B) = P(A \wedge B) / P(B)$

$P(A|B,C) = P(A|C)$

"A is conditionally independent of B given C"

Pr(A|J=T)?

Oops, can't look it up!

Bayes: $\Pr(A|J) = \Pr(J|A) \cdot \Pr(A) / \Pr(J)$

$\Pr(J=T|A=T)$ – from the BN; = 0.9

$\Pr(A=T)$ – marginalize over B and E; ~ 0.0025

$\Pr(J=T)$ – marginalize over A; ~0.052

Turns out $\Pr(A=T|J=T)$ is quite low ~ 0.043 Why?

Quite a few false positives 0.05 and the

Alarm turns out to be unlikely $\Pr(A=T)$ ~ 0.0025

$\Pr(A|\neg b)$?

Marginalizing over E

blending of = $\Pr(A|\neg b,e)$ and $\Pr(A|\neg b,\neg e)$

blend how? By Pr(E)

$0.29 \cdot 0.002 + 0.001 \cdot 0.998 = 0.001578$

Why don't we marginalize over J? or M?

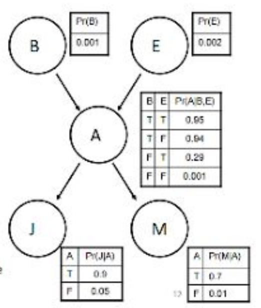
Can CPTs at J or M influence our opinion on A when B=F?

What if we know that John called?

We can ignore J & M when neither is an evidence or query variable; general rules?

Observing John called makes J an evidence variable

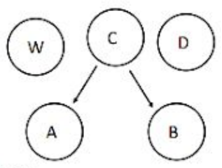
(text fig 14.11)



Suppose A,B,C,D,W are trinary

How many parameters for the joint? $3^5 - 1 = 242$

How many parameters for the BN? 2+2+2+6+6 = 18



Suppose A is binary, B is trinary, C takes on 4 values D takes on 5, W takes on 6?

W 5, D 4, C 3, A 4, B 8 = 24

Define Over-fitting.

– When a classifier has a good performance (low classification error) on training data, but has a bad performance on unseen test data drawn from the same distribution, the classifier is said to be over-fitting.

Explain the process of pruning decision trees.

– The most common way of pruning involves learning a possibly over-fit tree first, till each node has a small number of instances, followed by pruning it where nodes which add very little information are replaced by leaves appropriately. For this, the training data is split, the tree is learned from a part of the data and the other part is later used to test the tree's effectiveness while making the pruning decisions.

How does Pruning address the problem of over-fitting?

– If a decision tree is too deep, the lower split choices get built on the basis of a small number of examples and thus, exhibit low confidence. In general, the hypothesis space becomes too expressive with a large depth. Pruning removes very deep nodes, rectifying these problems and reducing the overall complexity of the classifier, thus reducing the chance of over-fitting.

Give two alternatives to pruning.

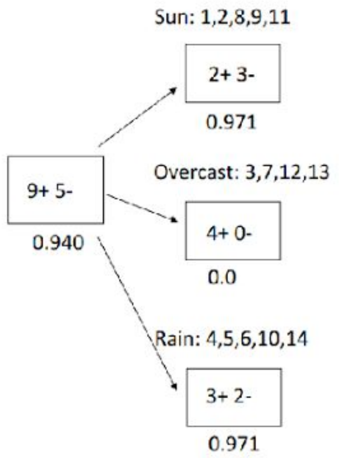
– One of the alternatives is bagging. In this method, a number of decision trees are independently trained by using a different sampled subset of the training data for each, and the test data is labeled by taking the majority of individual labels. The other method is to determine a split-termination criterion for early-stopping (such as best-depth etc). This criterion can be estimated by using a Cross-validation technique.

How to Reduce Overfitting in Decision Trees

- Restrict the expressiveness of H
- Limit the set of tests available
- Limit the depth
  - No deeper than N (say 3 or 12 or 86)
  - But how to choose?
- Limit the minimum number of examples used to select a split
  - Need at least M (is 10 enough? 20?)
  - How to choose? (Adjust for number of tests? Their outcomes?)
  - Want significance; Statistical hypothesis testing can help
- BEST: Learn an overfit tree and prune
  - Partition the labeled examples
  - A: training data – grow the tree
  - B: pruning data – prune the tree from leaves
  - What would be the pruning algorithm?

Outlook Gain = 0.246

1.	S	H	H	W	-
2.	S	H	H	S	-
3.	O	H	H	W	+
4.	R	M	H	W	+
5.	R	C	N	W	+
6.	R	C	N	S	-
7.	O	C	N	S	+
8.	S	M	H	W	-
9.	S	C	N	W	+
10.	R	M	N	W	+
11.	S	M	N	S	+
12.	O	M	H	S	+
13.	O	H	N	W	+
14.	R	M	H	S	-



Information After:

0.971 * 5/14 +

0.0 * 4/14 +

0.971 * 5/14

= 0.694

Information Gain:

0.940 – 0.694

=0.246

$$H(S) = \boxed{\sum_{v \in Labels} -\Pr(v) \cdot \log_2(\Pr(v))}$$

$$\mathbf{H}(S_b) - \sum_i \mathbf{H}(S_{ai}) \frac{|S_{ai}|}{|S_b|}$$

Computational vs. Statistical Learning Theory

- Instead of bounding N, bound the True Error
- CoLT:
  - Given (choose) ε, δ, H
  - We discover how large N must be
- SLT:
  - Given (choose) δ, H, N
  - Bound how far the True Error can be from the empirical error (zero for simple learning)

Chernoff Concentration Bound

(aka Hoeffding – several variants)

- Recall the weak law of large numbers
  $\lim \Pr(|\mu_D - \mu_Z| > \varepsilon) = 0$ as $N \to \infty$
- What about finite N?
- Chernoff bound for a sequence of N Bernoulli events:
  $\boxed{\Pr(\mu_D > \mu_Z + \varepsilon) \leq e^{-2N\varepsilon^2}}$

SLT: Simple Learning

- Recall $\boxed{N \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln|H|\right)}$
- So $\boxed{\varepsilon \geq \frac{1}{N}\left(\ln\frac{1}{\delta} + \ln|H|\right)}$
- For every such ε, w/ prob ≥ 1 - δ, this ε is a bound on the true error from above:
- We can have any equal or larger ε; best is equality.
- Therefore $\boxed{TrueError(h) \leq \frac{1}{N}\left(\ln\frac{1}{\delta} + \ln|H|\right)}$

for every h in H w/ zero empirical error

- This bounds the probability that the sample accuracy of an arbitrary h evaluated on Z is very misleading:

$$\boxed{P(error_D(h) > error_S(h) + \varepsilon) \le e^{-2N\varepsilon^2}}$$

- We can bound the probability that *any* one has a misleading error:

$$\boxed{P((\exists h \in H) error_D(h) > error_S(h) + \varepsilon) \le |H| e^{-2N\varepsilon^2}}$$

- We need this to be bounded by δ:

$$\boxed{P((\exists h \in H) error_D(h) > error_S(h) + \varepsilon) \le |H| e^{-2N\varepsilon^2} \le \delta}$$

- Solving for N:

$$\boxed{N \ge \frac{1}{2\varepsilon^2}\left(\ln|H| + \ln\frac{1}{\delta}\right)}$$

Perceptron Learning

(Widrow-Hoff or delta rule)

$\text{percep}_\mathbf{w}(\mathbf{x})$ assigns 1 for + if $\mathbf{w} \cdot \mathbf{x} > 0$ (vector dot product)
else it assigns 0 for - (a common numeric transform trick)

err = label($\mathbf{x}$) – $\text{percep}_\mathbf{w}(\mathbf{x})$
0: correct -1: false pos 1: false neg

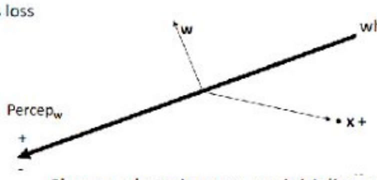
Here, false neg: $\mathbf{w} \cdot \mathbf{x} < 0$ but it should be > 0

loss = distance from boundary = - err $\mathbf{w} \cdot \mathbf{x}$

Want to adjust w_i's to reduce this loss

Loss fcn *gradient* is direction of greatest increase in loss with **w**

Want the opposite: step **w** in direction $-\nabla_w$ loss



Loss function = - err $\mathbf{w} \cdot \mathbf{x}$

Want the opposite: step **w** in direction $-\nabla_w$ loss

What is ∇_w loss for input **x**?

View - err $\mathbf{w} \cdot \mathbf{x}$ as a function of **w**

$\nabla_w(-\text{err } \mathbf{w} \cdot \mathbf{x}) = -\text{err } \mathbf{x}$

So $-\nabla_w(-\text{err } \mathbf{w} \cdot \mathbf{x}) = \text{err } \mathbf{x}$

Update **w** according to:

$\Delta\mathbf{w} = \alpha \text{ err } \mathbf{x}$
where α is a learning rate

Choose a learning rate α; initialize **w** arbitrarily (small works best)

Compute $\Delta\mathbf{w} = \alpha \text{ err } \mathbf{x}$

Add $\Delta\mathbf{w}$ to **w**

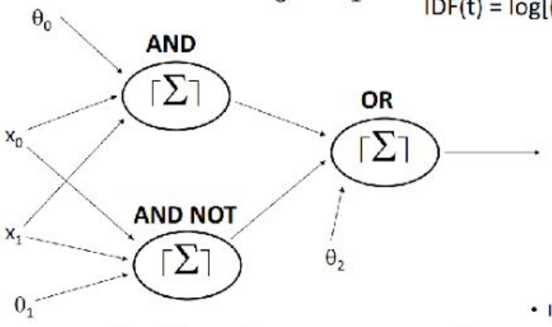
Repeatedly cycle through training examples

TF(t,d) = (raw count of t in d) / (length of d)
Approximates Pr(t | d)

IDF(t) = log[(number of documents) / (documents w/ t)]

ANN XOR

$x_0 \oplus x_1$



Bag of Words

- Consider two utterances:
  - John likes to watch movies. Mary likes movies too.
  - John also likes to watch football games.
- For these "documents" the universe of (all 10) words:
  - { "John", "likes", "to", "watch", "movies", "also", "football", "games", "Mary", "too" }
- Each utterance / document can be represented as a 10-entry vector:
  - [1, 2, 1, 1, 2, 0, 0, 0, 1, 1]
  - [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

K-Means

- Initialization
  - pick K examples at random for the initial cluster seeds
  - randomly assign examples to K sets
- Termination
  - little (or no) re-assignment of examples
  - little (or no) change in the K means

Further Improvements

- Often punctuation is dropped
- Convert everything to all lower case
- Stop words
  - How about the words "the" or "sesquipedality"
  - Some words are too common to be discriminative
  - Others are too unlikely to be seen
  - Stop words: a list of words to be removed from the feature set
- Stemming
  - combine variants of a word
  - earthquake and earthquakes
  - eat and eating
- N-grams: keep statistics on term sequences
  - use pairs of terms (2-gram), triples (3-gram), etc.
  - requires many more training inputs (why?)

Logistic Regression

- Odds ratio: Pr(Y=0 | X) / Pr(Y=1 | X)
- > 1 assign to class Y=0; else Y=1
- Assume the log of the odds ratio is a linear function of the features
- Let P be Pr(Y=0 | X)
- Pr(Y=1 | X) = 1 – Pr(Y=0 | X) = 1 - P
- log(odds) = ln(P / (1 – P)) = logit(P)
- Assume the logit is linear:
  $\ln(P / (1 - P)) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$

b) P(D = T | A = F) = P(D = T, C = T|A = F) + P(D = T, C = F|A = F)
P(D = T, C = T | A = F) = P(D = T | C = T, A = F) * P(C = T | A = F)
= P (D = T | C = T) * (P(C = T, B = T| A = F) + P(C = T, B = F | A = F))

P(C = T, B = T| A = F) = P(C= T| B= T, A = F) * P(B = T|A= F)

= P(C=T | B=T, A=F)*P(B=T)

Similarly

P(C = T, B = F|A = F) = P(C=T|B= F, A = F)*P(B = F)

Therefore, we have

P(D = T|A = F) = g*(e*b+f*(1-b))+h*((1-e)*b+(1-f)*(1-b))

(1 point, 0.5 point if the steps are missing)

c) B is independent of A, P(B=F|A = T) = P(B= F) = 1-b

(1 point, 0.5 point if the student do not state the independency between A and B)

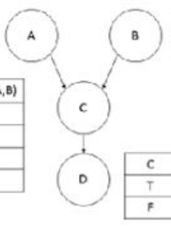
d) P(D = T| A = F) = 0.5*(0.4*0.6+0.7*0.4)+0.4*(0.6*0.6+0.3*0.4) = 0.452

Using counting, we have

P(D = T | A = F) = (840+1008+980+336)/(840+840+1008+1512+980+980+336+504) = 0.452

P(A)
a

P(B)
b



A	B	P(C|A,B)
T	T	c
T	F	d
F	T	e
F	F	f

C	P(D|C)
T	g
F	h

A	B	C	D	Count
T	T	T	T	270
T	T	T	F	270
T	T	F	T	504
T	T	F	F	756
T	F	T	T	300
T	F	T	F	300
T	F	F	T	240
T	F	F	F	360
F	T	T	T	840
F	T	T	F	840
F	T	F	T	1008
F	T	F	F	1512
F	F	T	T	980
F	F	T	F	980
F	F	F	T	336
F	F	F	F	504

a) Based on the counts, please compute the CPT values a through h.

b) Please write $P(D = T|A = F)$ as a function of a through h, please show your steps.

c) Please write $P(B = F|A = T)$ as a function of a through h, please show your steps.

d) Please compute numerically $P(D = T|A = F)$ using your result from part a and b. Please verify your an using the counts table.