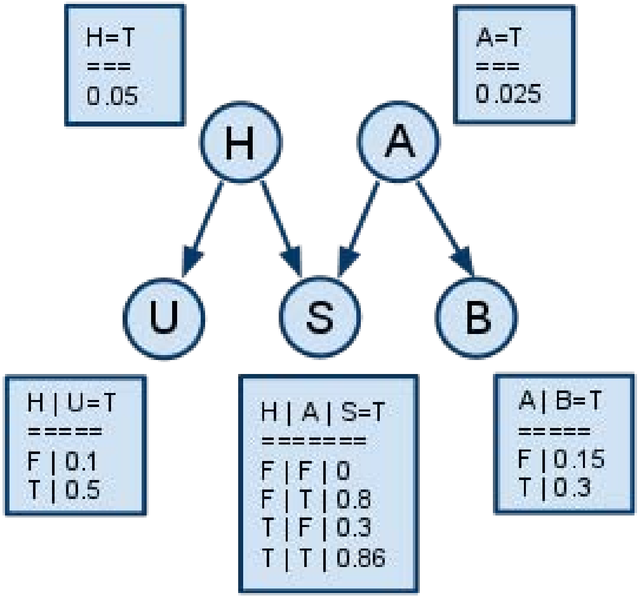
Probability, Bayes Nets, Naive Bayes, Model Selection

Major Ideas:

1. Intro to Bayes nets: what they are and what they represent.
2. How to compute the **joint probability** from the Bayes net.
3. How to compute the **conditional probability** of any set of variables in the net.
    - Marginalization and Exact Inference
    - Bayes Rule (backward inference)
4. Naive Bayes - classification using Bayes Nets
5. Bayesian Model Selection / Structure Search
6. Generative versus Discriminative Models
7. (Optional) D-Separation Rules for determining conditional independence in Bayes Nets
8. (Optional) Noisy OR

Bayes Nets are a compact way to represent the Joint Distribution of a set of Random Variables. The nodes represent Random Variables. Random variables are variables that provides a mapping from values to probabilities.

We have the following Bayes net from recitation. There are five random variables (to simplify we've removed the MIT variable)



-

H = {T, F} There is a **H**ack on Campus
A = {T, F} There is an **A**rt exhibit on Campus
U = {T, F} We see a **U**niformed police man.
S = {T, F} We see a **S**culpture
B = {T, F} We see **B**usiness People in suits on Campus

Next to each node, you have conditional probability tables (**CPT**), these represent the conditional probability of the underlying Random Variable conditioned on its parents.

The CPT for nodes H and A have only one value because (not H) or (not A) is simply one minus the value shown. Variables U, S, B have CPTs that are dependent on their parent variables. Generally, in a CPT, we use the first n-1 columns to denote the settings of the "given" Variables. If the variable is binary, the last column is the True value of probability of the variable for the settings of the given variables. If a variable is multivalued, then When dealing with binary variables we don't show the implicit 1-p column.

For Example S's CPT fully expanded is:

H	**A**	**S**=T	S=F (not shown)
F	F	0	1
F	T	0.8	0.2

T	F	0.3	0.7
T	T	0.86	**0.14**

So using the CPT above. What is the probability of *NOT seeing a sculpture* if we know there is a Hack and there is an Art exhibit? Or what is P(S=F **|** H = T, A = T) Answer: 0.14

A Matter of Parameters

Sometimes questions may ask "*how many parameters*" are in a given Bayes Net? By number of parameters we really mean the number of CPT entries. This is because the Bayes Net is fully specified only when all the parameters are assigned some numerical value. Note that the hidden columns like P(S=F|H,A) does not count in that number, because those entries can be gotten by 1-P (S=T|H,A).

So how many parameters does our "Hack or Art show" network have?
#parameters = #cpt entries in network = 1 + 1 + 2 + 4 + 2 = 10

Joint probability
or computing the probability of a specific world state

Suppose we know there is a hack on campus, and there isn't an art exhibit, we see a uniformed officer, a sculpture, and we don't see any Business men.
The probability of such a world is: P(H=T, A=F, U=T, S=T, B=F)

We can easily compute the Joint probability from a Bayes net!
For any Bayes Net:

$$\boxed{p(V_1...V_n)=\prod_{i=1}^{n}p(V_i|Parents(V_i))}$$

In other words, Bayes Nets is really an encoding of the conditional dependencies of a set of random variables. All variables are independent of other variable given their parents.
For our example:

$$p(H,A,U,S,B)=p(H)p(A)p(U|H)p(S|H,A)p(B|A)$$

So for the specific setting above:

P(H=T, A=F, U=T, S=T, B=F) = P(H=T)P(A=F)P(U=T|H=T)P(S=T|H=T, A=F)P(B=F|A=F)
= 0.05 x (1-0.025) x 0.5 x 0.3 x (1-0.15) = 0.006215625

Marginalization over the Joint

Example: How to compute the probability of P(S)?
We can compute any arbitrary probabilities from joint probabilities by the method of "marginalization" = summing out variables that we don't want.

$$P(S)=\sum_{H,A,U,B}P(H,A,U,S,B)$$
$$P(S)=\sum_{H,A,U,B}P(H)P(A)P(U|H)P(S|H,A)P(B|A)$$
$$P(S)=\sum_{H}\sum_{A}\sum_{U}\sum_{B}P(H)P(A)P(U|H)P(S|H,A)P(B|A)$$

Next move the sums so that a sum is placed only before all the terms that depend on it.
e.g. P(S|H,A) depends on sum A and sum H so those sums occur left of it:

$$P(S)=\sum_{A}P(A)\sum_{H}P(H)P(S|H,A)\sum_{U}P(U|H)\sum_{B}P(B|A)$$

Notice that: $\sum_{B}P(B|A)=1$, and same for $\sum_{U}P(U|H)=1$!

So dropping the B and U terms we get the final summation:

$$P(S)=\sum_A P(A)\sum_H P(H)P(S|H,A)$$

Which works out to:

$$P(S)=P(H)P(A)P(S|H,A)+P(\overline{H})P(A)P(S|\overline{H},A)+P(H)P(\overline{A})P(S|H,\overline{A})$$
$$+P(\overline{H})P(\overline{A})P(S|\overline{H},\overline{A})$$

= 0.0347

Ancestor (Sub)Graph: a subgraph of the Bayes Net where only variables of interest and their ancestors are drawn

Ancestor Graph Shortcut:
Any variable that is not in the "ancestor" graph for the set of variables of interest we can remove from the summation.

Example:
In computing **P(S)**, **S** is the variable of interest, **H, A** are both ancestors of S, but not **U** or **B**. So **P(U|H)**, **P(B|A)** can be safely dropped.
In computing **P(S | B)**, **S, B** are variables of interest, **H, A** are ancestors of S, and B. So **P(U|H)** can be dropped from the summation.

General Method for computing any P(X) from a Bayes Net

1. Write P(X) as a marginalization sum over joint probabilities.
2. Cross out/ignore terms that are not in the ancestor graph of X.
3. Expand the summation and simplify

Example:

$$P(B)=\sum_{A,H,S,U} P(H,A,U,S,B)$$

Ancestor graph of B only include variables B and A.

$$P(B)=\sum_{A,H,S,U} \cancel{P(H)}\cancel{P(A)}\cancel{P(U|H)}P(B|A)P(A)$$

$$P(B)=P(B|A)P(A)+P(B|\overline{A})P(\overline{A})$$

General Method for computing any P(X|Y) from a Bayes Net:

1. Write P(X, Y) as a marginalization sum over joint probabilities (without summing over X, Y).
2. Cross out/ignore terms that are not in the ancestor graph of X and Y.
3. Write P(Y) as a marginalization sum over joint probabilities.
4. Cross out/ignore terms that are not in the ancestor graph of Y.
5. Perform division: P(X,Y)/P(Y), and simplify.

Example: What is P(S|B)?

$$P(S|B)=\frac{P(S,B)}{P(B)}$$

From the definition of conditional Probability

Both of these terms can be computed via marginalization over the Joint.

$$P(S,B)=\sum_{A\in T,F}\sum_{H\in T,F} P(A)P(H)P(S|H,A)P(B|A)$$

$$P(B)=\sum_{A\in T,F} P(B|A)P(A)$$

Both terms above have been simplified using the ancestor graph trick.

So finally dividing one by the other:

$$P(S|B)=\frac{\sum_{A\in T,F}P(A)P(B|A)\sum_{H\in T,F}P(H)P(S|H,A)}{\sum_{A\in T,F}P(B|A)P(A)}$$

All the terms in the equation above can be read off of the Bayes Net CPTs. The rest is just arithmetic (or in Sadoway-ese = stamp collecting).

The methods outlined here is very "algorithmetizable"; you can easily create a computer program that can decompose any probability into a series of CPT lookups. This method is also called "exact inference".

NOTE: There are other ways you can arrive at P(S|B). For instance the following is a possible way:

$$P(S|B)=\sum_{A\in T,F}P(S|A)P(A|B)$$

However, notice that P(S|A) is also a probability that you need to compute (that you can't just read off the network). P(A|B) is not readable from the network but P(B|A) is, but you need to use bayes rule to invert it.

Backward Inference

Sometimes we want to compute probabilities going "against the arrows" in the BN. To do that it might be handy to use **Bayes' Rule**:

$$\boxed{P(X|Y)=\frac{P(Y,X)}{P(Y)}=\frac{P(Y|X)P(X)}{P(Y)}}$$

Here is a variant of Bayes Rule with conditionals:

$$\boxed{P(X|Y,A)=\frac{P(Y|X,A)P(X|A)}{P(Y|A)}}$$

Optional Derivation: This is Bayes rule variant is derived by expanding the joint using the chain rule and then canceling out the P(A) terms.

$$P(X|Y,A)=\frac{P(Y,A,X)}{P(Y,A)}=\frac{P(Y|A,X)P(X|A)P(A)}{P(Y|A)P(A)}=\frac{P(Y|A,X)P(X|A)}{P(Y|A)}$$

Example: Compute **P(H | S)** using Bayes rule.

$$P(H|S)=\frac{P(S|H)P(H)}{P(S)}$$

1. We can read P(H) from the Bayes net. P(S) we've computed earlier.

P(S | H) = $P(S|H,A)P(A)+P(S|H,\overline{A})P(\overline{A})$

= 0.86 * 0.025 + 0.3 * 0.975 = 0.314

2. Plugin in the value of P(S) from above:

P(H | S) = $P(S|H)P(H)/P(S)$

= (0.314 * 0.05) / 0.0347 = **0.45**

Explaining Away

How does the probability of a hack **H** change if we knew only there is a statue P(H|S)

versus if we know there is a statue and also that there is an art exhibit on campus.
In other words, compare: P(H | S) versus P(H | S, A).

P(H | S) we already know as **0.45**

To compute P(H|S,A), we use the conditional variant of Bayes Rule:

P(H | S, A) = P(S A | H) P(H) / P(S, A)
P(H | S, A) = P(S | H, A) P(H | A) / P(S | A) # conditional variant of Bayes Rule
= 0.86 * 0.05 / 0.803 P(H | A) = P(H) (A and H are independent)
= **0.054**

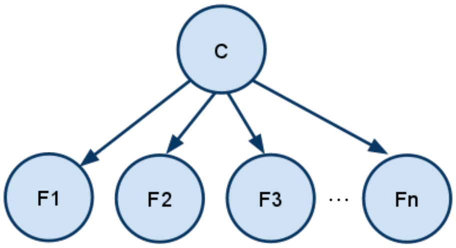
Notice that P(H|S, A) < P(H|S).

We know that a sculpture can be caused by two things, a hack on campus or an art exhibit.
If we know that an art-exhibit is occurring, then the probability of the statue being a hack declines.
Because the knowledge of the art-exhibit cause "**EXPLAINS AWAY**" the hack cause of the statue.

NOTE: this explaining away phenomena occurs when there is a "Noisy OR" configuration in the network.

Naive Bayes

Popular method for doing *classification* in the Bayes Net formalism.
Network structure:



We are given **likelihood** of a class generating some certain set of features P(F_i | c) and we are given the prior class probabilities P(c) (how likely each class is).

Then we use Bayes Rule to make predictions of the class given the features:
i.e. **P(c | features)**

$$P(C|F_1...F_n) = \frac{P(F_1\cdots F_n|C)P(C)}{P(F_1...F_n)}$$

The method is Naive because we assume that features given the class are all independent:

$$P(F_1...F_n|C) = P(F_1|C)P(F_2|C)...P(F_n|C) = \prod_{i=1}^{n} P(F_i|C)$$

$$\boxed{\arg\max_c P(C|F_1...F_n) = \arg\max_c \frac{P(C)\prod_{i=1}^{n} P(F_i|C)}{P(F_1...F_n)} = \arg\max_c P(C)\prod_{i=1}^{n} P(F_i|C)}$$

We are only interested in the **arg max** class. So we can ignore the denominator (the normalization constant P(F_1...F_n), because it is common to all computations.

We estimate P(c) (The class prior) and P(F|C) (feature likelihoods) from data we observe/collect.

In some practice a log is computed instead of a product, because probabilities involved can become very small. Log probabilities are used to prevents double imprecision problems.

$$\arg\max_c P(C)\prod_{i=1}^{n} P(F_i|C) = \arg\max_c \log P(C) + \sum_{i=1}^{n} \log P(F_i|C)$$

Bayesian Sorting Hat for MIT undergrads
We conducted an impromptu survey to determine which dorms group to "sort" incoming freshman. We collected data by surveying existing resident of MIT's 3 major dorm groups. Then we apply Naive Bayes to classify new students based on questionnaires they fill in during orientation.

We surveyed 30 random students, 10 from each of the MIT dorm groups. Each surveyed student is asked to fill out a simple questionnaire with these questions:
0. Which dorm do you live in: { East Campus, West Campus, or FSILG }
1. Are a Pyro - i.e. do you enjoy performing feats with fire (or enjoy inadvertently trigger fire alarms)?
2. Are you a foreign student or do you have a penchant for studying foreign languages?
3. Are you in good shape?

Here are the results. The last column shows how many students were surveyed in each dorm group.

	Pyro	ForeignLang	GoodShape	# surveyed
East Campus	8/10	1/10	3/10	10 10/30
West Campus	3/10	6/10	3/10	10 10/30
FSILG	1/10	3/10	8/10	10 10/30

We next use these counts to make estimates on probabilities:

P(C) (prior probability of being in any campus)
P(F_i | C) the likelihood given a campus.

For example: P(Pyro=True | C=East Campus) = 8 /10 P(Language = True|C= FSILG) = 3/10

Question 1: a new student with the following answers should be sorted into which dorm group?

Pyro = True
ForeignLang = False
GoodShape = False

Translation: Compute argmax P(C | P=T, F=F, G=F)
East Campus:
= P(P=T | C=East) P(L=F|C=East)P(G=F|C=East)P(C=East)
= (8/10) (1-1/10)(1-3/10) (10/30)
= (8*9*7*10)/30000
= 72 * 7 * 10/30k
West Campus:
= P(P=T | C=West) P(L=F|C=West)P(G=F|C=West)P(C=West)
= (3/10) (1-6/10)(1-3/10) (10/30)
= (3*4*7*10)/30000
= 3 * 28 * 10/30k
FSILG:
= P(P=T |C=FSILG) P(L=F|C=FSILG)P(G=F|C=FSILG)P(C=FSILG)
= (1/10) (1-3/10)(1-8/10) (10/30)
= (1*7*2*10)/30000
= 14 * 10 / 30k

The largest value for Pyros is **East Campus**.

Q: What about an all-round student who checks all the boxes true in the incoming questionnaire?

or what is argmax_C P(C | Pyro = True, ForeignLang = False, GoodShape = False)?

East Campus
= P(P=T | C=East)P(L=T|C=East)P(G=T|C=East)P(C=East)
= (8/10) (1/10)(3/10) (10/30)
= (8*1*3*10)/30000
= 8 * 30/30k
West Campus
= P(P=T | C=West)P(L=T|C=West)P(G=T|C=West)P(C=West)
= (3/10) (6/20)(3/20) (20/30)
= (3*6*3*10)/30000
= 18 * 30/30k
FSILG
= P(P=T | C=FSILG)P(L=T|C=FSILG)P(G=T|C=FSILG)P(C=FSILG)
= (1/10) (3/10)(8/10) (10/40)
= (1*3*8*10)/30000
= 8 * 30/30k

The maximum C is **West Campus**.

Real World Applications of Naive Bayes
Naive Bayes is most commonly used in text Classification. Where word are features. The presence or absence of a word can be used to determine the topic of a document.

For instance, detecting whether a message is Spam or Ham:

$$P(S|W_1...W_n) = \arg\max_{S \in T,F} \prod_i^n P(W_i|S)P(S)$$

Here S = T we have spam, S = F we have ham. We train our classifier with documents labeled as spam or ham, and then classify new documents by computing whether S=T or S = F is more likely. P (S) reflects the fraction of normal emails that are either spam or ham.

Bayesian Model Selection / Structure Search
or How to learn the parameters or the structure of models:

$$\boxed{\arg\max_M P(M\,|\,Data) = \arg\max_M \frac{P(Data|M)P(M)}{P(Data)} = \arg\max_M P(Data|M)P(M)}$$

We can use Bayesian probability to find the best Bayes net that explains the collected data.

A model could be a collection of different Bayes nets on the same set of variables. Each model differing in the dependencies they define (i.e. the connections in the net).

P(Data |M) is the likelihood of the data given the model. We can compute this by computing the probability of the observed data set using model M. This probability can be sometimes a joint probability, or the probability over only the observed variables (if some variables are hidden).

P(M) is some prior belief on the likelihood of each model.
We can make P(M) uniform (in which case all models to be equally likely)
Or we can prefer certain "goldilock" models over others. Models that are neither too connected nor too unconnected. For instance, we can make P(M) a Gaussian distribution over the number of connections, with the mean at some k level of connectedness.

Food for thought: With 5 variables, how many models have 0 connections? How many are fully connected? i.e. with 1+2+3+4 = 10 arrows. How many models have k arrows?

To find the best model, we use P(M|data) as the search heuristic. The search algorithm can be a hill climbing, best-first, A* or beam search. At each step we make a slight modification on the current M to make it a new model M' (by changing a dependency link or adding or removing a dependency link). Because searching for a good model involves a search over potential exponential space of possible models this problem is often NP-hard.

Example, Finding The Best Model Parameters
Model selection can also be used to find the best parameters given a model.
In the example given in lecture, model selection was used to determine the most likely coin given some data. We are shown the result of a series of coin flips as: **HHHHT**. We are told that the coin flips are generated by grabbing a random coin from a bag containing 3 types of coins.

Fair coins	P(H|M=fair) = 1/2
Weighted coins	P(H|M=weighted) = 3/4
Heads-only coins	P(H|M=heads-only) = 1.0

Using Model selection: We quickly compute that P(M | data) for all 3 types of coins,
Assuming P(M) = 1/3; uniform for all three types of coins

P(data|M=fair)P(M=fair) = $(1/2)^4$ $(1/2)^1$ (1/3) = 0.01
P(data|M=weighted)P(M=weighted) = $(3/4)^4$ $(1/4)^1$ (1/3)= 0.026
P(data|M=heads-only)P(M=heads-only) = $(1)^4$ $(0)^1$ (1/3) = 0

then argmax_M P(M|data) = weighted

Had we observed a sequence **HHHHH** instead, then

P(data|M=fair)P(M=fair) = $(1/2)^5$ (1/3) = 1/32 = 0.01
P(data|M=weighted)P(M=weighted) = $(3/4)^5$ (1/3)= 0.079
P(data|M=heads-only)P(M=heads-only) = $(1)^5$ (1/3) = 1/3

argmax_M P(M|data) = heads-only

Generative vs. Discriminative Models

Bayesian models (Bayes Nets) are also referred to generative models. Because they model a hypothetical underlying generative process, describing how some observed data might have been produced. Classification using generative models involve finding a model that most closely explains the observations.

Discriminative models aims instead to model the boundary between two classes. SVM is a classic case of a discriminative model. The function we "learn" through the process of training, tell us more about the structure of the separation boundary rather than the underlying processes.

Discriminative Models empirically perform better at classification tasks because they are focused on separatability. Generative models are attractive because it tries to model the process, and hence can better explain the underlying physical processes.

Determining Conditional Independence of Variables in Bayes Nets (Optional)

Bayes Nets encode *conditional dependencies* between random variables but not independencies. To figure out whether two variables X, Y are independent given a third set of variables Z we apply a graphical method called **D-separation**.

The D-separation Algorithm:

1. Find all paths between X and Y nodes in the Bayes Net.
2. Determine if a given path is "blocked".

A path is **blocked** if any component along the path one of the following rules hold true:

chain: i -> m -> j	if m is in the given set Z
chain: i <- m <- j	if m is in the given set Z
fork: i <- m -> j	if m is in the given set Z
collider: i -> m <- j	if m (or descendant of m) is **not** in the given set

3. If all paths from X and Y are blocked then X and Y are independent given set Z.

Examples using the Hack or Art Exhibit Bayes Net:
Q: Are H and A variables independent? (Given nothing)
A: Yes, there is a path from H, A through S. This path triggers the collider rule.
But since Z is empty, i.e. we are given nothing, the path is blocked.
Implication P(A|H) = P(A) and P(H | A) = P(H) by the definition of independence.

Q: Are H and A variables independent given S?
A: No, P(H | S, A) != P(H | A) We actually proved this via computation above.
But using d-separation, the collider path component from H to A through S is no longer blocked! So D-separation tells us that H and A are not independent given S.

Q: Are U and B variables independent? (Given nothing.)
A: The path from U to B passes through H, S, A.
The path component U <- H -> S is a fork, and since H is not given, this path component is unblocked. Similarly the path component S <- A -> B is also unblocked.
But the path component H -> S <- A is blocked via the collider rule, since S is not a given.

Therefore the entire path from U to B is blocked! So **U and B are independent given no information**. This implies P(U|B) = P(U) and P(B|U) = P(B)

Q: Under what circumstances would U and B **not** be independent?
A: When S is known. When S is known, then the path from U to B would be unblocked.
This implies P(U | B, S) != P(U | S)

Q: Is there an even easier way to determine independence relationships?
A: Yes. Here is an alternate way (taught in Machine Learning 6.867) that systematically determines variable conditional independencies graphically.
1. Convert your Bayes net into an undirected graph. (i.e. remove the arrows)
2. Draw the ancestor graph with respect to the variables of interest, X, Y, and given Z
3. "Moralize" the parents in the graph: i.e. draw an undirected edge between any two pairs of parents.
4. Remove all evidence nodes Z.
5. Check if there is exists an path from X to Y.

If there is no path between X and Y, then X, Y are independent given Z. If there is a path then X, Y are not independent given Z.

Noisy OR (Optional)

In order to make CPTs (Conditional Probability Tables) more compact, some causal relationships can be models as an OR of the *negation* of causes.

In our Hack-or-Art-Exhibit example, H and A are causes, and S is the observed result:
To model S's CPT using a Noisy-OR parametrization, we define two probabilities **p** and **q**.

p being the probability that a statue is **does not appear** when there is an Art Show

q being the probability that a statue is **does not appear** when there is a Hack.

Using this noisy OR parameterization we can condensed the original 4 parameter CPT down to 2 parameters.   When the number of causes is k, the number of CPT entries is 2^k but with noisy-OR only k parameters are needed. Hence the number of CPT parameters grows linearly with noisy-OR rather than exponential.

H	**A**	**S**=T	S=F (this column not shown)
F	F	0	1
F	T	1-**p**	**p**
T	F	1-**q**	**q**
T	T	1-**pq**	**pq**

When the CPT of a multi-causal relationship implements a noisy OR configuration, then we can use the **explaining away** intuition to reason about changes in probabilities.   The phenomenom of explaining away can still occur with non-noisy-or configurations but it is not guaranteed to always work.

MIT OpenCourseWare
http://ocw.mit.edu

6.034 Artificial Intelligence
Fall 2010

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.