Approximate (Monte Carlo) Inference in Bayes Nets

- Basic idea: Let's repeatedly sample according to the distribution represented by the Bayes Net. If in 400/1000 draws, the variable *X* is *true*, then we estimate that the probability *X* is *true* is 0.4.
- To sample according to Bayes Net, just set the variables one at a time using a total ordering consistent with the partial...

Monte Carlo (continued)

- (Samping continued)… ordering represented by the underlying DAG of the Net. In this way, when we wish to draw the value for *X* we already have the values of its parents, so we can find the probabilities to use from the CPT for *X*.
- This approach is simple to implement using a pseudorandom number generator.

So it seems we're done, right?

- Wrong: what if we take into account evidence (observed values for variables)?
- If the evidence happens to be in the "top" nodes of the network (nodes with no parents), we're still fine. Otherwise...
- No efficient general method exists for sampling according to the new distribution based on the evidence. (There are inefficient ways, e.g., compute full joint.)

Rejection Sampling

- One natural option for sampling with evidence is to use our original sampling approach, and just throw out (*reject*) any setting that does not agree with the evidence. This is *rejection sampling*.
- Problem: if evidence involves many variables, most of our draws will be rejected (few will agree with the evidence).

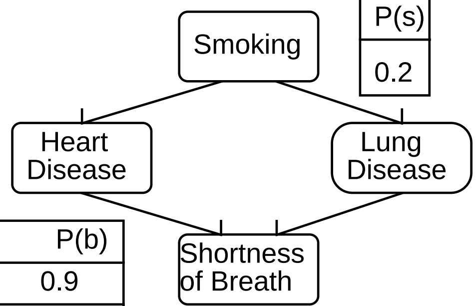
Likelihood Weighting

- Another approach is to set the evidence variables, sample the others with the original Monte Carlo approach, and then correct for improbable combinations by *weighting* each setting by its probability.
- Disadvantage: with many evidence variables, probabilities become vanishingly small. We don't sample the more probable events very thoroughly.

Markov Chain Monte Carlo

- Key idea: give up on *independence* in sampling.
- Generate next setting probabilistically based on current setting (Markov chain).
- Metropolis-Hastings Algorithm for the general case, Gibbs Sampler for Bayes Nets in particular. Key property: *detailed balance* yields *stationary distribution*.

Gibbs Sampling by Example



P(s)
0.2

S	P(h)
T	0.6
F	0.1

S	P(l)
T	0.8
F	0.1

H	L	P(b)
T	T	0.9
T	F	0.8
F	T	0.7
F	F	0.1

Gibbs Sampling Example (Continued)

- Let our query be **P**(*HeartDisease* | *smoking*, *shortnessOfBreath*). That is, we know we've been smoking (*Smoking=True)* and we know we're experiencing shortness of breath (*ShortnessOfBreath=True*), and we wish the know the probability that we have heart disease.
- Might as well keep a tally for *LungDisease* while we're at it.

Other Decisions

- Let's assume we use an off-the-shelf pseudorandom number generator for the range [0..1].
- We can loop through the non-evidence variables in a pre-established order or randomly, uniformly. Let's go with the latter. Tally at each step. (If the former, we could tally at each step or each iteration.)

Other Decisions (Continued)

- One chain or many: let's go with one.
- Length or burn-in: ordinarily 500-1000, but let's go with 2 (don't tally for original setting or setting after first step).
- Initial values: let's say all *True*. Note that *Smoking* and *ShortnessOfBreath* must be initialized to *True*, since this is our evidence. The initial settings for non-evidence variables are arbitrary.

Other Decisions (Continued)

- Use of random numbers in selecting a variable to draw. We have only two non-evidence variables: *HeartDisease* and *LungDisease*. Let's adopt a policy that a random number greater than 0.5 leads us to draw a value for *LungDisease*, and a random number of 0.5 or less leads us to draw for *HeartDisease*.

Other Decisions (Continued)

- Use of random numbers in selecting values of variables. Since all our variables are Boolean, our distributions will be over the values <*True*,False> and will have the form <P(*True*),1-P(*True*)>. If our random number is less than or equal to P(*True*), then we will set the variable to *True*, and otherwise we will set it to *False*.

A Final Supposition for our Example

- Having made all our choices, the only other factor that will affect the activity of the Gibbs Sampling algorithm is the sequence of random numbers that we draw.
- Let's suppose our sequence of random numbers begins 0.154, 0.383, 0.938, 0.813, 0.273, 0.739, 0.049, 0.233, 0.743, 0.932, 0.478, 0.832, …

Round 1

- Our first random number is 0.154, so we will draw a value for *HeartDisease*.
- To draw the value, we must first determine the distribution for *HeartDisease* given its Markov Blanket.
- First, we compute a value for *True*. We multiply P(*heartDisease*|*smoking*) by P(*shortnessOfBreath* | *heartDisease, lungDisease*). Notice we take *LungDisease*

Round 1 (Continued)

- (Continued)… to be *True* because that is its current setting. (We use the current settings of all variables in the Markov Blanket.) This product is (0.6)(0.9) = 0.54.
- Next we repeat the process for *HeartDisease=False*. We multiply the probability that *HeartDisease* is *False* given *smoking* by the probability of *shortnessOfBreath* given *HeartDisease* is

Round 1 (Continued)

- (Continued)… *False* and *LungDisease* is *True*. The resulting product is (0.4)(0.7) = 0.28.
- We now normalize <0.54,0.28> to get the probability distribution <0.66,0.34>. Hence we will set *HeartDisease* to *True* if and only if our random number is at most 0.66. It is 0.383, so *HeartDisease* remains *True*.

Round 2

- Our next random number is 0.938, so we next will draw a value for *LungDisease* given the current settings for the other variables.
- To obtain a value for *LungDisease=True*, we multiply P(*lungDisease* | *smoking*) by P(*shortnessOfBreath* | *heartDisease, lungDisease*). (Recall that *True* is our current setting for *HeartDisease* and *True*

Round 2 (Continued)

- (Continued)… is our candidate setting for *LungDisease*. This product is (0.8)(0.9) = 0.72.
- Similarly, for *LungDisease=False*, we multiply P(*LungDisease=False* | *smoking*) by P(*shortnessOfBreath* | *heartDisease, LungDisease=False*). This product is (0.2)(0.8) = 0.16.

Round 2 (Continued)

- Normalizing <0.72, 0.16> we get the distribution <0.82, 0.18>.
- Our next random number is 0.813, so we (barely) keep *LungDisease* set to *True*.
- This is the first round after our burn-in, so we record the frequencies. We now have counts of 0 for *HeartDisease* and *LungDisease* set to *False*, and counts of 1 for each of these set to *True*.

Round 3

- Our next random number is 0.273, so we draw a value for *HeartDisease* next.
- Because all the variables have the same value as the last time we drew for *HeartDisease*, the distribution is the same: <0.66, 0.34>. Our next random number is 0.739, so we set *HeartDisease* to *False*.

Round 3 (Continued)

- Updating our tallies, we have counts of: 1 for *HeartDisease=False*, 1 for *HeartDisease=True*, 0 for *LungDisease=False*, and 2 for *LungDisease=True.*

Round 4

- The next random number is 0.049. Therefore we draw a value for *HeartDisease* again. Because all other variables are unchanged, and we consider both values of *HeartDisease*, once again the distribution is <0.66, 0.34>. Our next random number is 0.233, so we reset *HeartDisease* to *True*.

Round 4 (Continued)

- Our new counts are as follows: 2 for *HeartDisease=True*, 1 for *HeartDisease=False*, 3 for *LungDisease=True*, and 0 for *LungDisease=False.*

Round 5

- Our next random number is 0.743, so we next draw a value for *LungDisease*.
- The values for all other variables are as they were the first time we drew a value for *LungDisease*, so the distribution remains <0.82,0.18>. Our next random number is 0.932, so we set *LungDisease* to *False*.

Round 5 (Continued)

- Our new tallies are as follows: 3 each for *HeartDisease=True* and *LungDisease=True*, and 1 each for *HeartDisease=False* and *LungDisease=False.*

Round 6

- The next random number is 0.478, so again we sample *HeartDisease.* But since the setting for *LungDisease* has changed, we must recompute the distribution over *HeartDisease.*
- To get a value for *HeartDisease=True*, we multiply P(*heartDisease* | *smoking*) by P(*shortnessOfBreath* | *HeartDisease=True,*

Round 6 (Continued)

- (Continued)… *LungDisease=False*). This results in the product (0.6)(0.8) = 0.48.
- For *HeartDisease=False*, we multiply P(*HeartDisease=False* | *smoking*) by P(*shortnessOfBreath* | *HeartDisease=False, LungDisease=False*). The result is (0.4)(0.1) = 0.04.
- Normalizing these values to obtain a

Round 6 (Continued)

- (Continued)… probability distribution, we get <0.92, 0.08>. Our next random number is 0.832 so we choose *HeartDisease=True.*
- Our tallies now stand at 1 for *HeartDisease=False*, 4 for *HeartDisease=True*, 2 for *LungDisease=False*, and 3 for *LungDisease=True.*

Final Results

- Of course, we have not run the Markov chain nearly long enough to expect an accurate estimate. Nevertheless, let's ask what the answer is to our query at this point.
- We assign a probability of 4/5 or 0.8 to *heartDisease*.
- We also might ask about *lungDisease*, to which we assign 3/5 or 0.6.

How Do These Compare With Exact Results

- Try using variable elimination.
- We find the probability of *heartDisease* given *smoking* and *shortnessOfBreath* is 0.695.
- And we find the probability of *lungDisease* given *smoking* and *shortnessOfBreath* is 0.864.

For Review

- Use variable elimination to compute the answers just given.
- Repeat with the junction tree for this net.
- Use Gibbs Sampling with the same sequence of random numbers, same initial setting, and no burn-in to estimate the probability that a person is a smoker given that you know they have heart disease.

For Review (Continued)

Are the graphs below triangulated? (Use MaximumCardinalityTest.)