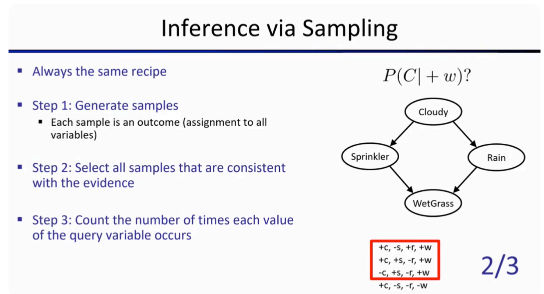
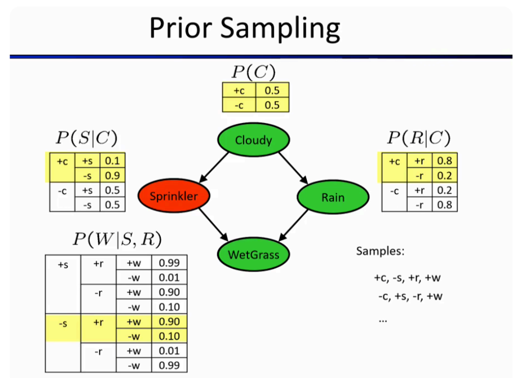
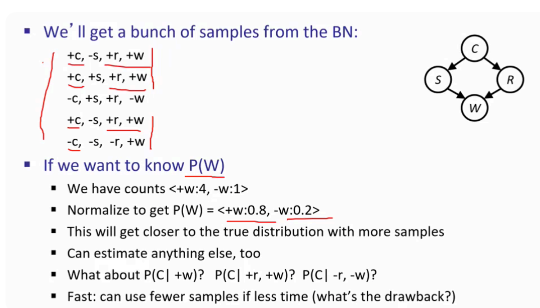
- Sampling is a way to do *approximate* inference (without incurring the [potentially] exponential cost of VE and IBE)
  - like repeated simulation of samples from BN - getting a sample is faster than computing probabilities
    - draw N samples from a sampling distribution S
    - compute an approximate posterior probability (empirical probability)
    - you should be able to show that this converges to the true probability P(Q | E) that you would get using a technique like VE or IBE (enumeration) as your number of samples -> infinity (by LLN)
  - can also use sampling to learn a distribution you don't know (more on this in a later note on ML)



- Sampling from a given distribution
  - Generate a random number in [0, 1) = sample u
  - Convert sample u into an outcome for the given distribution
    - associate each outcome with an interval in [0, 1)
    - e.g. P(red) = 0.6, P(blue) = 0.3, P(green) = 0.1
      - [0, 0.6) = red, [0.6, 0.9) = blue, [0.9, 1) = green
  - After you sample a couple times you can calculate empirical probabilities
  - a sample is an outcome/assignment to all variables
- In complex BN we can sample by using the full joint distr., but we don't want to construct the full joint (the whole point of this)
  - 4 sampling methods will be covered: 1) Prior sampling, 2) Rejection sampling, **3) Likelihood weighting, 4) Gibbs sampling** (bold is what's used in practice most often)

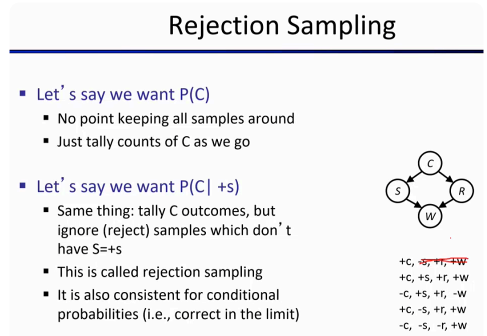


- **Prior sampling** (aka ancestral sampling/forward sampling)
  - topologically order your BN
  - we sample in topological order
    - first we sample from the root variable's CPT (doesn't depend on anything)
    - we get an outcome, proceed to the next variable X
      - all X's parents are guaranteed to have been resolved to some outcome already
      - we sample from the part of the CPT consistent with it's parent's outcomes
    - do this until we get an outcome for each variable:
    - together all outcomes for all variables = one sample of our jt distr.
  - Code:
    - for i = 1...n:
      - sample x_i from P(x_i | parents(x_i))
    - return sample = (x_1, ..., x_n)
  - Proof that this samples according to the joint distr.
    - Let S_PS be the distribution from which we sample
    - S_PS (x_1, ..., x_n) = Product of P(x_i | Parents(x_i)) = P(x_1, ..., x_n)
    - trivially true by our procedure (probability of each sample is product of getting each ancestor sample along the way)
    - Let N_PS(x_1, ..., x_n) be the number of samples with outcome (x_1, ..., x_n)
      - then lim N->infty empirical P(x_1, ..., x_n) = lim N_PS(x_1, ..., x_n) / N = S_PS(x_1, ..., x_n) = P(x_1, ..., x_n)



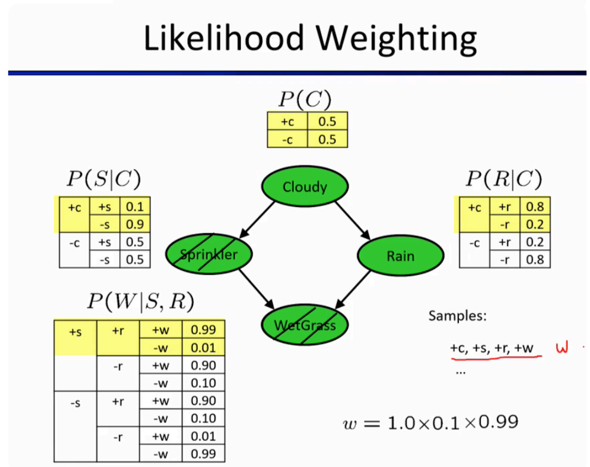
  - With these samples we can calculate any probabilities we want from the BN
    - but this depends on the specific outcome we want even happening at all in any of our samples
    - for unlikely outcomes Prior Sampling requires a large amount of samples before we can answer (e.g. P(C | -r, -w))
    - tradeoff between Speed vs. Accuracy (small samples to get a quick answer, large samples to get a more accurate answer)
- **Rejection sampling** - improvement on PS
  - prior sampling doesn't take into account what our actual query is (i.e. what samples are we actually

interested in?); thus, we can make prior sampling more efficient



- Main idea
  - as soon as we get a partial sample that matches our query we stop sampling any of the other variable outcomes (don't need them)
  - we also stop & throw out samples as soon as we get outcomes inconsistent with the *evidence* of our query
- Proof: same as prior sampling

- **Likelihood weighting** - improvement on RS

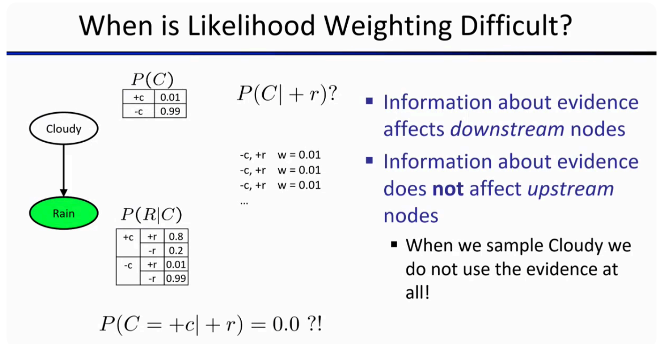


  - Problems with rejection sampling:
    - evidence is not exploited
    - we're still generating random samples, and if evidence is really unlikely to occur, we're going to reject most of our samples --> RS becomes very inefficient if evidence is unlikely to occur
  - instead we should force *ALL* samples to agree with evidence, and sample the rest of the variables (enter Likelihood weighting); but we need to be careful to adjust the probability by the likelihood of the evidence occurring P(evidence | parents(evidence)), i.e. the "weight"
    - so each sample in this case is not worth 1 as with PS and RS; instead its worth its weight
  - e.g. the diagonal lines indicate +evidence variables (S = +s, W = + w)
    - we proceed as with PS in topological order
    - get +c, go to S, don't sample, instead weight by P(+s | + c) = 0.1 (since this is an evidence var.)
    - proceed to R, sample according to P(R | +c), get the sample +r
    - proceed to W, don't sample, instead weight by P(+w | +c, +r) = 0.99 (since W = +w is also an evidence var.)
  - Code:

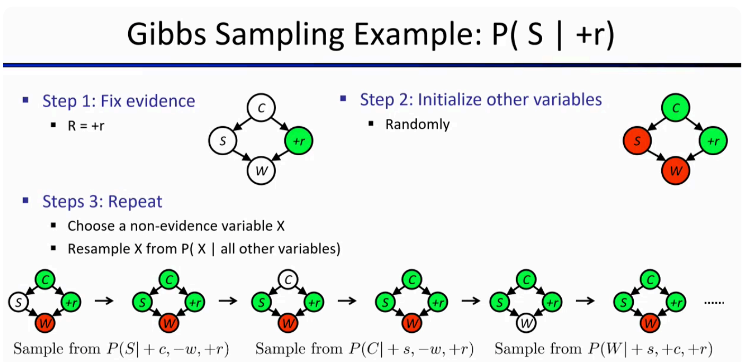
```
▪ IN: evidence instantiation
▪ w = 1.0
▪ for i=1, 2, ..., n
    ▪ if X_i is an evidence variable
        ▪ X_i = observation x_i for X_i
        ▪ Set w = w * P(x_i | Parents(X_i))
    ▪ else
        ▪ Sample x_i from P(X_i | Parents(X_i))
▪ return (x_1, x_2, ..., x_n), w
```

  - Proof that this matches full jt distr.
    - sampling the variables z, with fixed evidence variables e
    - Let S_WS(z, e) = Product of P(z_i | parents(z_i))
      - product of sampling non-evidence variables
    - Let w(z, e) = Product of P(e_i, | parents(e_i))
      - product of sampling evidence variables
    - Together, the weighted sampling distribution is consistent with the full joint:
      - S_WS(z, e) * w(z, e) = Product of P(z_i | parents(z_i)) * Product of P(e_i, | parents(e_i)) = P(z_1, ..., z_m, e_1, ..., e_n)
  - Now all of our samples are going to reflect the evidence
    - however, if our evidence is really unlikely, our weights are going to be small; and a sample with weight 1 is as good as 10 samples with weight 0.1, so we need to generate more samples for smaller weight samples (which will happen if our evidence is really unlikely) = inefficient
    - evidence variables influence choice of downstream variables, but not upstream ones (+s does not affect

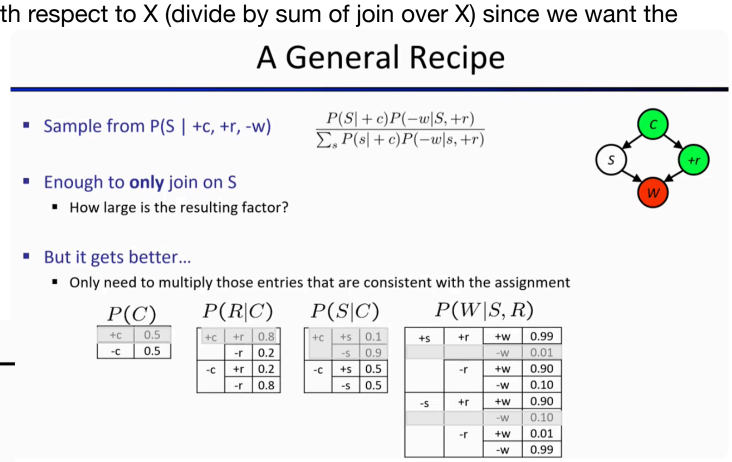
outcome of C in example above)
    - hence we could have *a lot of small weight samples*; unlikely outcomes for evidence given likely upstream variables will still be unlikely and we will need to generate A LOT of samples before we can get an accurate distribution for the query
    - sum of weight = how many "effective" samples were obtained; high weight = good
    - we would like to consider evidence when we sample every variable, enter Gibbs sampling

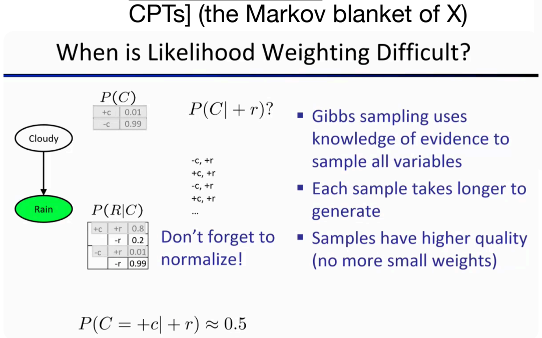


- **Gibbs sampling** - when evidence occurs downstream, this is more efficient (converges to true distr. faster)
  - Procedure:
    - start with a random full assignment to all variables x_1, ..., x_n (fixing evidence to be consistent)
    - sample 1 variable at a time, conditioned on all other variables (don't sample evidence variables)
    - repeat for many many iterations in order to "forget" initial random assignment that didn't match our distr



    - in the limit the resulting sample will come from the correct joint distribution
    - rationale: now both upstream and downstream variables condition on evidence
  - [Resampling of one variable] procedure
    - join on the variable X, renormalize with respect to X (divide by sum of join over X) since we want the distribution P(X | x_i's) the total probability space is over all values of X
      - notice that all other variables are fixed, thus the size of the join is just |X|!
    - the resulting factor depends only on X's <u>parents</u>, its <u>children + children's parents</u> [show up in the CPTs] (the Markov blanket of X)





- Gibbs sampling has some issues when many of the non-evidence variables are heavily correlated (e.g. affected heavily by other non-evidence variables, therefore takes many iterations for values to settle to true distribution given evidence)
  - e.g. Senators voting
    - solution to Gibbs sampling issues: block sampling (resample blocks of variables at a time)
  - Gibbs sampling is a special case of family of general methods for empirical iterative sampling from a

distribution called **Markov chain Monte Carlo (MCMC) methods** [Metropolis Hastings is one of more popular ones, hey EECS126! and Gibbs is actual a flavor of MH]

- BN give you a general purpose way for incorporating evidence into inference procedures to infer probability of certain outcomes given evidence