50.021 -AI

Alex

Week 12: Probabilistic Graphical Models

Probabilistic Graphical models –a way to efficiently capture probability distributions between large numbers of variables

In this class you should see – only for discrete variables

- ways how to sample from a conditional distribution
  - prior sampling
  - rejection sampling
  - likelihood weighting
  - gibbs sampling
  - you will see that each of those methods have problems/failure modes.

  What for is it useful?
- Need simulation for computing expected values of complicated functions over some distribution.

  its unlikely that you can compute the expectation of

$$I = \int_1^2 \frac{e^{-3x} tanh(x + \ln(x))}{\sqrt{x^7 + x^3 + 12} + cos^2(3x)} f(x) dx$$

  with respect to a probability density $f(x)$, but you can simulate the integral in 5 lines of code!

  - draw $x^{(1)}, \ldots, x^{(n)}$, $x^{(s)} \sim f$ for n large
  - $I \approx \frac{1}{n} \sum_{s=1}^{n} \frac{e^{-3x^{(s)}} tanh(x^{(s)} + \ln(x^{(s)}))}{\sqrt{(x^{(s)})^7 + (x^{(s)})^3 + 12} + cos^2(3x^{(s)})}$

  Note: below we will use $x = (x_1, \ldots, x_n)$. The $x^{(s)}$ in this example is not an x_i from below. The $x^{(s)}$ here is a single drawn sample - like the whole $x = (x_1, \ldots, x_n)$ below. $x^{(s)}$ can be a whole vector used to integrate a function that takes values over a vector like.
- For testing of algorithms you may need to generate simulated data which has some properties, e.g. looks like van Gogh image, or finance data where the numbers express some financial conditions.

Recap

We do encode conditional independence in Bayesian networks by making design choices like

$$P(X_i|X_{i-1}, X_{i-2}, \ldots, X_1) = P(X_i|parents(X_i))$$
$$P(X_1 \ldots, X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i))$$

Our goal:

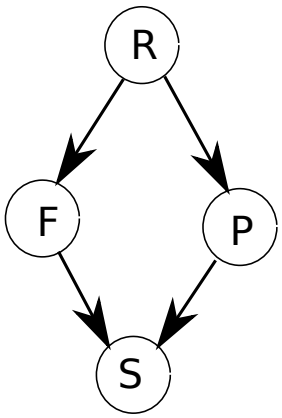
Given evidence in variables $(E_1, \ldots, E_l) \subset (X_1 \ldots, X_n)$, and given Query variables $Q_1, \ldots, Q_k \subset (X_1 \ldots, X_n)$ we want to generate samples $q = (q_1, \ldots q_k)$ from

$$P(Q_1, \ldots, Q_k|E_1, \ldots, E_l)$$

This includes that we have to marginalize out all those variables from $(X_1 \ldots, X_n)$ which are not in either evidence or query variables.

Prior Sampling

Consider the sad example:



R - heavy flood type rain, F - floods, P - power outage, S - me eating sweets bcs i am bored

One way to generate samples is to walk the graph top-down. We want samples (r, f, p, s)

- sample from R, suppose you get a $(r = 1, f =?, p =?, s =?)$. ? denotes not sampled yet.
- now you can either sample from F or P. Suppose you sample P first, then F.
- We have $r = 1$, so we sample for P from the probability $P(P|R = 1)$. We maybe get $(r = 1, f =?, p = 0, s =?)$

- now sample F from $P(F|R=1)$. We get $(r=1, f=1, p=0, s=?)$
- now all parents for the probability $P(S|parents(S))$ have been sampled, so we can sample S
- general rule: can sample from $P(X_i|parents(X_i))$ after one has sampled values for all $parents(X_i)$

an example sample set could be $(r=1, f=0, p=1, s=1), (r=0, f=1, p=1, s=0), (r=0, f=0, p=1, s=0), (r=1, f=0, p=1, s=1)$ (yes, you can get the same sample by above procedure, right?)

The general **Prior Sampling algorithm:**
Find an order that of nodes such that $parents(X_i)$ are sampled before X_i itself

- For $i=1,2,\ldots,n$:
  Sample x_i from $P(X_i|parents(X_i) =$ their sampled values$)$
- return $(x_1,\ldots,x_n)$

Its a very simple idea :) .

From what distribution are the samples generated?

we draw $x_i \sim P(X_i|parents(x_i))$, so $(x_1,\ldots,x_n) \sim \prod_i P(X_i|parents(x_i))$

and this is equal to $P(X_1,\ldots,X_n)$. **So we draw from the joint over all variables.**

Limitations: Suppose you want to compute an expectation with respect to probability $P(R,S|F=1)$. What for do you need samples (r,f,p,s) with $f=0$??

Rejection Sampling

Idea: stop drawing a sample if on the way you encounter a value that is inconsistent with the value observed in an evidence variable E_i. Example: want something about $P(R,S|F=1)$ but get a sample with $f=0$.

The general **Rejection Sampling algorithm:**

1 <a goto jump point>

2 For $i=1,2,\ldots,n$:
  Sample x_i from $P(X_i|parents(X_i) =$ their sampled values$)$
  if x_i is inconsistent with evidence, then stop here, throw away sample, goto 1

3 return $(x_1, \ldots, x_n)$

When this is efficient?

Suppose you are in a very long causal chain $A_1 \to A_2 \to A_3 \to \ldots$ with a 10000 elements, and the rejection condition happens at node 10, then you can often skip to sample 9990 nodes :). You dont gain that much if the rejection condition applies to node 9999 .

When is this inefficient?

Suppose you want $P(R, S|P = 1)$, but observing a sample with $P = 1$, a power outage, is very rare. This is: when you sample, then you see only $P = 1$ maybe only with $P(P = 1) = 0.000001$. On average you need to throw away a million samples before you get a desired sample with $P = 1$.

Rule of thumb: If you condition on many many observed events, then the resulting probability can get very tiny. means: you will throw away a lot.

Example: a causal chain with all $x_i \in \{0, 1\}$ and probabilities are either 0.7/0.3 or 0.5/0.5 depending on the value of X_{i-1}. $P(X_i|X_{i-1}) = X_{i-1} * (X_i * 0.7 + (1 - X_i) * 0.3) + (1 - X_{i-1}) * 0.5$. You know now: the marginal probability to observe 30 X_i having whatever evidence is at most $0.7^{30} = 0.000025$. Keep to reject ratio is $1 : 40k$. You see it is exponentially decreasing in the number of observed evidence variables for such a model.

Rejection sampling is easy but not the ultimate.

From what distribution are the samples generated?

we draw $x_i \sim P(X_i|parents(x_i))$, **except** the case when X_i is an evidence variable E_i, then we choose only samples such that the value is which we want $X_i = E_i = e_i$. Therefore we are looking only at samples with $E_1 = e_1, \ldots, E_l = e_l$

so: $(x_1, \ldots, x_n) \sim \prod_i P(X_i|parents(x_i), E_1 = e_1, \ldots, E_l = e_l)$

and this is equal to $P((X_1, \ldots X_n) \setminus \{E_1, \ldots, E_l\}|E_1 = e_1, E_2 = e_2, \ldots, E_l = e_l)$. **So we draw from the conditional where evidence variables have their values fixed.**

Warning: if you compute the **probability numbers of the BN for these drawn samples** $(x_1, \ldots, x_n)$ using $\prod_i P(X_i|parents(x_i))$ then you **obtain the joint probability with fixed evidence**

$$P((X_1, \ldots X_n) \setminus \{E_1, \ldots, E_l\}, E_1 = e_1, E_2 = e_2, \ldots, E_l = e_l)$$

There is no division of this number by $P(E_1, \dots E_l)$ done anywhere.

Thats a typical error people make when using BNs and rejection sampling.

Likelihood Weighting

Idea: dont draw what you do not observe. Instantiate all evidence variables, sample only the rest. We will maintain a weight w, explain right after that what for the weight is good for.

The general **Likelihood Weighting algorithm:**

1 $w = 1.0$

2 For $i = 1, 2, \dots, n$:
  if $X_i = E_i$ is an evidence variable, then
    set to its observed value:$X_i = e_i$
    update w: $w = w * P(X_i = e_i | parents(X_i))$
  else X_i is not evidence variable, then
    Sample x_i from $P(X_i | parents(X_i))$
  endif

3 return $w, (x_1, \dots, x_n)$

Important: we expect you for the exam to be able to run the algorithm. its ok if you do not understand on the first time seeing the why.

What is the weight w good for?

We sample for non-evidence variables Z_i from $P(Z_i | parents(Z_i))$, so the total probability for all non-evidence variables is

$$\prod_{Z_m \in (X_1, \dots X_n) \setminus \{E_1, \dots, E_l\}} P(Z_m | parents(Z_m))$$

This is **not equal** to

$$P(X_1 \dots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i))$$

What is missing?

$$\prod_{E_m \in \{E_1, \dots, E_l\}} P(E_m | parents(E_m)) \text{ !!!}$$

and this is exactly our returned weight w of a sample.

You can get the joint probability **with fixed evidence !!!** only by multiplying both

$$P((X_1, \dots X_n) \setminus \{E_1, \dots, E_l\}, E_1 = e_1, E_2 = e_2, \dots, E_l = e_l) =$$
$$\prod_{Z_m \in (X_1, \dots X_n) \setminus \{E_1, \dots, E_l\}} P(Z_m | parents(Z_m)) \prod_{E_m \in \{E_1, \dots, E_l\}} P(E_m | parents(E_m))$$
$$= \prod_{i=1}^{n} P(X_i | parents(X_i))$$

important application 1:

you need the weight w when you compute an expectation with respect to the conditional $P((X_1, \dots X_n) \setminus \{E_1, \dots, E_l\} | E_1 = e_1, E_2 = e_2, \dots, E_l = e_l)$.

$$E_{X \sim P((X_1, \dots X_n) \setminus \{E_1, \dots, E_l\} | E_1 = e_1, E_2 = e_2, \dots, E_l = e_l)}[f(X)] \approx \sum_{s=1}^{S} \frac{w(s)}{\sum_t w(t)} f(x_1(s), \dots, x_n(s))$$

Compare to the case when you would draw samples $(x_1, \dots, x_n)$ directly from the conditional distribution, then you would have a weight $\frac{1}{S}$:

$$E_{X \sim P((X_1, \dots X_n) \setminus \{E_1, \dots, E_l\} | E_1 = e_1, E_2 = e_2, \dots, E_l = e_l)}[f(X)] \approx \sum_{s=1}^{S} \frac{1}{S} f(x_1(s), \dots, x_n(s))$$

For sampling with likelihood weighting the weight is $\frac{w(s)}{\sum_t w(t)}$, where $\sum_t w(t)$ is the total weight sum of all drawn samples. Remember: Typically you do not draw one sample $(x_1, \dots x_n)$ but hundreds or thousands of them.

important application 2:

The weight w is a quality indicator of your samples!

$$\prod_{E_m \in \{E_1, \dots, E_l\}} P(E_m | parents(E_m))$$

says: how probable/plausible is it to see my evidence given the other values sampled in the parent nodes.

What do you desire: samples such that these weights are high! You want to have samples $(x_1, \dots, x_n)$ such that w are relatively high, which means that seeing your evidence is very probable.

Possible Problem with likelihood weighting - the short explanation

You may generate samples with very small weight w if you are unlucky. The thing to believe me is: if you generate N samples, then their summed sample weight is a measure for the effective number of samples.

If you generate 100 samples with a method which has very low summed sample weight (compared to other methods), then this means:

$$\sum_{s=1}^{S} \frac{w(s)}{\sum_t w(t)} f(x_1(s), \ldots, x_n(s))$$

will be a bad approximation to the true expected value $E[f]$ that you want to compute

The consequence: if you create consistently many samples with too low weights, then you need much more samples, so that

1. the set of samples represent your data well
2. $\sum_{s=1}^{S} \frac{w(s)}{\sum_t w(t)} f(x_1(s), \ldots, x_n(s))$ can get close to the true expected value!!!

the explanation for those who want to go deeper

Why can it happen to generate lots of samples with very small weight w?

Example:

$A \rightarrow B$. You want to sample with evidence $B = 0$, so all your samples will be $(A, B = 0)$. We define some probabilities:

$$P(A = 1) = 0.9$$
$$P(A = 0) = 0.1$$

$P(B = 0|A = 1) = 0.001$ for $A = 1$ very low plausibility to draw $B = 0$

$P(B = 0|A = 0) = 1$

This is an example case where

- you draw A first.
- you have a high probability to generate samples, namely those with $A = 1$, which have low plausibility to explain your evidence $B = 0$!

This means: if you use sample weighting to draw 100 samples, then on average

- 90 samples will have weight $w = P(B = 0|A = 1) = 0.001$
- 10 samples will have weight $w = P(B = 0|A = 0) = 1$
- $P(B = 0|A = 1)$ here is w - our measure of plausibility of our samples
- total sample weight will be $\sum_{t=1}^{100} w_t = 90 \cdot 0.001 + 10 \cdot 1 = 100 \cdot 0.1009 = 10.09$

- you have generated lots of samples with low plausibility, total sample weight == total plausibility is low

if you would use direct sampling from the conditional probability, then you need $P(A = 1|B = 0)$,$P(A = 0|B = 0)$

$$P(A = 1, B = 0) = P(B = 0|A = 1)P(A = 1) = 0.0009$$
$$P(A = 0, B = 0) = P(B = 0|A = 0)P(A = 0) = 0.1$$
$$p(B = 0) = 0.1009$$

This is $P(A = 1|B = 0)$,$P(A = 0|B = 0)$:

$$P(A = 1|B = 0) \approx 0.01$$
$$P(A = 0|B = 0) \approx 0.99$$

This means: if you use direct sampling from $P(A = 1|B = 0)$,$P(A = 0|B = 0)$ to draw 100 samples, then on average

- 1 sample will have weight $w = P(B = 0|A = 1) = 0.001$
- 99 samples will have weight $w = P(B = 0|A = 0) = 1$
- total sample weight will be $\sum_{t=1}^{100} w_t = \cdot(1 \cdot 0.001 + 99 \cdot 1) = 99.001$
- much higher total sample weight!

(For this simple example, we can compute the expectation of a function $f(A)$ directly.

$$E_{A\sim p(A|B=0)}[f] = f(A = 1)P(A = 1|B = 0) + f(A = 0)P(A = 0|B = 0)$$
$$= f(A = 1)0.01 + f(A = 0)0.99$$

)

Why did this problem happen? We have put in evidence $B = 0$, but **we sample some variables (here: A) before we make use of the evidence**. The variables sampled before, are sampled without knowledge of the evidence. Thus they may sometimes not explain the evidence well – in the sense that $P(E|parents(E))$ – the probability to see the evidence given the already sampled variables – is low. If it is low, then it will result in poor approximations of any expectations computed using the samples, and you may need to use more samples to get a similarly good approximation of expectations compared to other methods.

Gibbs Sampling

Idea: initialize all evidence variables. Randomly initialize all other variables with values. Now you have one sample. Generate one sample from the last existing one by exchanging one component x_i

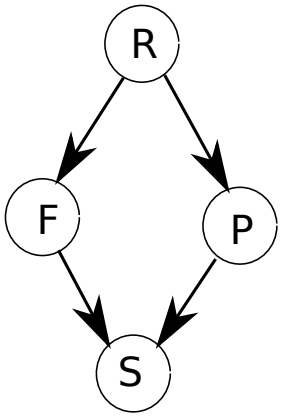
The general **Gibbs Sampling algorithm:**

1 initialize all evidence variables

2 Randomly initialize all other variables with values.

3 Repeat as often as you want to have a new sample:
take your last used sample $x^{(old)} = (x_1, \ldots, x_n)$
Choose a non-evidence variable X_i
draw a new component $\widehat{x_i}$ from $P(X_i|\text{all other variables} = \text{their values from } x)$
new sample $x^{(new)} = x^{(old)}$ where $\widehat{x_i}$ replaced the old value x_i

Nice property: $P(X_i|\text{all other variables} = \text{their values from } x)$ is easy to compute!



Suppose you need to sample from $P(F|R = 1, P = 0, S = 0)$. You can use the BN to directly compute

$$P(F = 0, R = 1, P = 0, S = 0)$$
$$P(F = 1, R = 1, P = 0, S = 0)$$

Then

$$P(F = 0|R = 1, P = 0, S = 0) = \frac{P(F = 0, R = 1, P = 0, S = 0)}{P(F = 0, R = 1, P = 0, S = 0) + P(F = 1, R = 1, P = 0, S = 0)}$$

and

$$P(F = 1|R = 1, P = 0, S = 0) = 1 - P(F = 0|R = 1, P = 0, S = 0)$$

The one-dimensional conditionals are fast to compute when all other variable values are fixed.[1]

[1] for marginals its the opposite, computing a $n - 1$-dim marginal is quick, computing a 1-dim marginal can be costly

Gibbs sampling would look like that:

You want to draw from $P(RFS|P = 0)$, then initialize randomly, for example:

$x^{(1)} = (r = 0, f = 1, p = 0, s = 0)$

Now at every iteration

- select randomly one variable r, f, s
- randomly replace its value drawn according to $P(X_i|\text{all other variables} = \text{their values from } x)$

you may get something like:
$(r = 0, f = 1, p = 0, s = 0)$,
$(r = 0, f = 1, p = 0, s = 1)$,
$(r = 0, f = 0, p = 0, s = 1)$,
$(r = 0, f = 0, p = 0, s = 1)$,
$(r = 0, f = 1, p = 0, s = 1)$,
$(r = 1, f = 1, p = 0, s = 1)$

- every time one component is replaced (it can happen that replacements generates the same as the old value!)

Drawback1: all your samples are correlated. You sample from a markov chain. Central limit theorem does not hold in its general version (but there are versions of CLT for Markov Chains). Consequence: you need more samples to have good integral approximations compared to drawing independent samples as in rejection sampling. Often you need less than likelihood weighting, because evidence is always considered.

Drawback2 / failure mode:

You can get stuck in islands of high probabilities, and not jump over from one island to the next. **Example**:

$$\begin{aligned} P(A = 1, B = 0) &= 10^{-100} \approx 0 \\ P(A = 0, B = 1) &= 10^{-100} \approx 0 \\ P(A = 1, B = 1) &= 0.3 - 2 \cdot 10^{-100} \approx 0.3 \\ P(A = 0, B = 0) &= 0.7 - 2 \cdot 10^{-100} \approx 0.7 \end{aligned}$$

You have two obvious islands: $(A = 1, B = 1)$ and $(A = 0, B = 0)$.

What problem will Gibbs sampling have? Once you are in the island of high probability $(A = 1, B = 1)$, no matter what variable you choose (A or B), with very high probability you will get stuck in $(A = 1, B = 1)$, simply because $P(A = 1, B = 0), P(A = 0, B = 1)$ have almost no probability.

If you are in $(A = 1, B = 1)$ and you want to draw from $P(A|B = 1)$, then you will almost always draw $A = 1$ and stay!!

Same if you are in island $(A = 0, B = 0)$, you will almost never

explore the other island of high probability $(A = 1, B = 1)$.

This means: when you draw samples (a, b), then compute an approximation of an integral using these samples ,

$$I \approx \frac{1}{S} \sum_{s=1}^{S} f(a_s, b_s)$$

then this approximation can be very wrong/imprecise - unless you manage from both modes proportional to the probability mass of both modes (need equal number of samples from both modes) - 30% from one and 70% from the other.

A hacky solution: try multiple restarts with random initializations and pray that you got all modes. You can cluster your obtained samples to see if there are obvious modes with low probability to travel between them.

in consequence:

- prior sampling is nice if you are interesting in the joint distribution
- rejection sampling is okay if your evidence restricts the marginal to a space of high probability, that is when

  $$P(E_1 = e_1, \ldots, E_l = e_l)$$

  is high.
- likelihood weighting is useful when the evidence is observed at the start points of a BN or close to those. If it is close, then you can: look at a small subnetwork involving start points of a BN, compute its marginal (by eliminating the empty heads trick!!), then sample from those marginals, and check approximation quality for these marginals by the weights, and compare to the true distribution from the marginals.
- Gibbs sampling is useful when above 3 are no option and you are willing to check for unexplored regions of the space.
- rejection sampling throws many samples away right from the start
- likelihood weighting never throws away, but may generate many low plausibility samples, so you may need lots of samples.
- gibbs sampling generates correlated samples, so you may need more, but often less than with likelihood weighting -simply because evidence is taken into account when sampling every variable (which does not happend with likelihood weighting unless all evidences are at the start). but you must be careful to explore all modes/ high probability regions of the space.

There is no free lunch in data science, and no solution that solves all!
You trade one advantage for (losing) another.