Drunkard walk in Bayesian Network

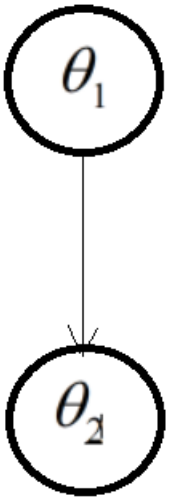
Bhargav Kulkarni

December 16, 2017

1 Problem formulation

This document describes and demonstrates how Gibbs sampling from a Bayesian Network can be used on a problem of random walk. Let us start off with the problem description.

The problem seems to be pretty straightforward and very recognizable if you've worked with MCMC(Monte Carlo Markov Chains) before. Lets assume that a drunk person wants to walk back home, which is on the rightside, from his given starting position. Given the state of the person, he is really uncertain about the direction in which he wants move. So he flips a biased coin that turns up heads with a probability of 0.4 and tails with a probability of 0.6. Whenever the coin flip result is heads, he starts moving right 90 percent of the time and consequently, moves left rest of the time(Remember the drunkard is constrained to move ONLY in these two directions). When the coin turns up tails, he goes right only 20 percent of the time. The problem here is to find the probability that he ends up going home. In other words, we need to compute the marginal density of the person going right. Now, lets construct the bayesian network, whch in this case, is just a chain with two random variables. The first node $theta_1$ is the event of the coin flip. The second node $theta_2$ is the direction in which the drunkard starts moving. The below given diagram illustrates the 2-node bayesian network. Since, we have only 2 nodes, we can model this as a bivariate distribution with joint density $p(\theta_1, \theta_2)$. Lets denote $+\theta_1$ and $-\theta_1$ for heads and tails respectively and $+\theta_2$ and $-\theta_2$ for right and left respectively. We want to be able to find $p(\theta_2)$ without going through the Bayesian computations, only by sampling on the Bayes-Net.



Theta1	P(theta1)
Heads	0.4
Tails	0.6

Theta2	P(theta2|theta1)	
	Heads	Tails
Right	0.9	0.1
Left	0.2	0.8

2 Gibbs Sampling algorithm

Lets consider multivariate distribution $\theta \in (\theta_1, \theta_2, ...\theta_K)$. Then for a j^{th} iteration/sample, we have:

$$\theta_1^{(j)} = p(\theta_1|\theta_2^{(j-1)}, ...\theta_K^{(j-1)})$$

$$\theta_2^{(j)} = p(\theta_2|\theta_1^{(j)}, \theta_3^{(j-1)}, ..., \theta_K^{(j-1)})$$

$$\theta_k^{(j)} = p(\theta_k|\theta_1^{(j)}, \theta_2^{(j)}, ...\theta_{k-1}^{(j)}, \theta_{k+1}^{(j-1)}, ..., \theta_K^{(j-1)})$$

Since, in our problem, we only have two variables, can write our algorithm as follows:

1. Initialize $\theta_1^(0)$ and $\theta_2^(0)$ to random values drawn from uniform distribution $U[0, 1]$.
2. $$\theta_1^{(1)} = p(\theta_1|\theta_2^{(0)})\theta_1^{(1)} = p(\theta_2^1|\theta_1^{(0)})$$

3. Repeat for sufficient number of iterations until convergence.

$p(\theta_2|\theta_1)$ can be directly found out from the conditional distribution given by the bayesian network. And $p(\theta_1|\theta_2)$ is given from Bayes theorem as:

$$p(\theta_1|\theta_2) = \alpha.p(\theta_2|\theta_1).p(\theta_1)$$

where α is a proprtionality constant. Each time, we pick the variable to be sampled randomly . And each time, we compare the uniformly generated random variable, if it is less than the conditional distribution linked with the variable being sampled, then we assign the random variable accordingly and keep incrementing the coount. The total number of counts for which the drawn sample is 'Right' divided by the total number of iterations gives us the probability that the drunk person is going to move Right and reach home ultimately(Hopefully).

The code can be found in the file named Gibbs.py. The total number of iterations can be input by the user. The answer is the marginal density $p(\theta_2)$, which when run for a substantial number of iterations should ideally give 0.48 for 'Right' and 0.48 for 'Left'.

3 References

1. Stuart Russell, Peter Norvig, Google Inc., *Artificial Intelligence: A Modern Approach, 3rd Edition*
2. Peter Abbiel's channel on YouTube Sampling from Bayes' Nets
3. David Barber 2007,2008,2009,2010 *Bayesian Reasoning and Machine Learning*