Statistical Data Analysis 2

Data Analysis Project # 2

E. Szczurek,
K. Gogolewski

1 Gibbs Sampling

Gibbs sampling can be applied when the joint distribution is not known explicitly, but the conditional distribution of each variable is known. The Gibbs sampling algorithm is used to generate an instance from the distribution of each variable in turn, conditional on the current values of the other variables. Gibbs sampling is particularly well-adapted to sampling the posterior distribution of a Bayesian network, since Bayesian networks are typically specified as a collection of conditional distributions, the *Local Probability Distributions* (LPDs).

1.1 The Gibbs Sampler

Let X_i be binary random variables with finite state space $\{T, F\}$ for $true$ and $false$. A Gibbs sampler now runs a Markov chain on $\mathbf{X} = (X_1, ..., X_n)$. For convenience of notation, we denote the set $(X_1, ..., X_{i-1}, X_{i+1}, ..., X_n)$ as $X_{\setminus i}$ and evidence $\mathbf{e} = (e_1, ..., e_m)$. The evidence variables represent the observed values. For the example in Figure 1, $\mathbf{X} = (Rain, Cloudy)$ and $\mathbf{e} = (Sprinkler = \mathrm{T}, WetGrass = \mathrm{T})$.

Then, the following method gives one possible way of creating a Gibbs sampler:

1. Initialise
   (a) Instantiate X_i to one of its possible values $x_i, 1 \leq i \leq n$.
   (b) Let $x^{(0)} = (x_1, ..., x_n)$
2. For $t = 1, 2, ...$
   (a) Pick an index $i, 1 \leq i \leq n$ uniformly at random (other schedules are also possible, e.g. iterate cyclically over all indices).
   (b) Draw $x_i^{(t)}$ from $P(X_i \mid x_{\setminus i}^{(t-1)}, \mathbf{e})$.
   (c) Set $x_{\setminus i}^{(t)} = x_{\setminus i}^{(t-1)}$.

The sampler generates a sequence of samples $x^{(0)}, x^{(1)}, ...$ from the Markov chain over all possible states. The stationary distribution of the Markov chain is the joint distribution $P(X_1, ...X_n \mid e)$. Thus, samples are usually taken from the Markov chain after a certain number of iterations, allowing enough time for the chain to reach the stationary distribution (sometimes referred to as the *burn-in* time). In order to yield independent samples from the distribution $P(X_1, ..., X_n \mid \mathbf{e})$, the set of samples is usually thinned out, e.g., by picking every 100^{th} sample after *burn-in*.

1.2 Sampling in the *Rain Network*

The network shown in Figure 1 is an example of a Bayesian network that models the relationship between the variables *Cloudy* (C), *Rain* (R), *Sprinkler* (S) and *WetGrass* (W). The idea is that if

the grass is wet, either the sprinkler is on or it's raining. Whether it's raining or the sprinkler is on is influenced by whether it's cloudy or not.

Now consider the task of estimating the probability of rain, given that the sprinkler is on and the grass is wet,

$$P(R = \mathsf{T} \mid S = \mathsf{T}, W = \mathsf{T}),$$

through sampling. Since *Sprinkler* and *WetGrass* are observed, the Gibbs sampler needs to draw samples from $P(R, C \mid S = \mathsf{T}, W = \mathsf{T})$. From this probability, the probability above can be derived by marginalization. For the sampler, you will need to compute the two conditional probability distributions of *Rain* and *Cloudy* given all other variables (as mentioned above, *Sprinkler* and *WetGrass* are fixed to T, indicated by green colour in Figure 1). This can be done using the *Local Probability Distributions* (LPDs) and the Markov blanket of the variable of interest.

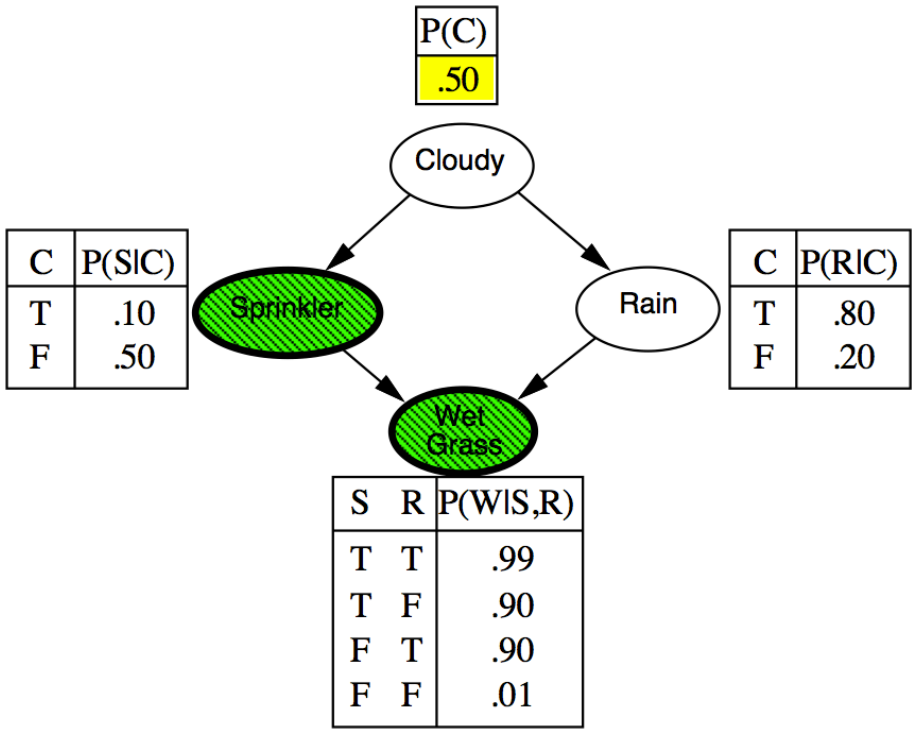

Figure 1: The *Rain Bayesian network*

In particular, perform the following tasks:

1. Derive the formulas for $P(C = \mathsf{T} \mid R = \mathsf{T}, S = \mathsf{T}, W = \mathsf{T})$, $P(C = \mathsf{T} \mid R = \mathsf{F}, S = \mathsf{T}, W = \mathsf{T})$, $P(R = \mathsf{T} \mid C = \mathsf{T}, S = \mathsf{T}, W = \mathsf{T})$ and $P(R = \mathsf{T} \mid C = \mathsf{F}, S = \mathsf{T}, W = \mathsf{T})$ **up to normalization constants in the denominators**, and compute their values by renormalizing the two possible values for each conditional probability distribution. (1 point)

2. Implement the Gibbs sampler sketched above for the Bayesian network in Figure 1 and draw 100 samples from the joint probability distribution $P(R, C \mid S = \mathsf{T}, W = \mathsf{T})$

3. Estimate the marginal probability of rain, given that the sprinkler is on and the grass is wet $P(R = \mathsf{T} \mid S = \mathsf{T}, W = \mathsf{T})$ from the 100 samples. (1 point)

1.3 Convergence Diagnostics

Now, determine whether the Gibbs sampler has reached the stationary distribution. An *adhoc* way is to plot the relative frequencies of $R = \mathsf{T}$ and $C = \mathsf{T}$ up to each iteration t against t, for two independent runs of the sampler. Then, the *burn-in* time can be set to the point where the two sequences converge. In addition, several formal tests are available to test stationarity of the sampler after a given time. Here, we consider the *Gelman and Rubin* multiple sequence diagnostic test [1]. In this method, multiple sequences are simulated and the idea is that after convergence the behaviour of all chains should be approximately the same. In particular, the variance within the chains should be the same as the variance across the chains. The *R* package *coda* provides convergence diagnostics for MCMC samplers, including the *Gelman* test (see convergence_diagnostics.pdf for further information).

In particular, perform the following tasks:

4. Now draw 50,000 samples instead of 100 using the Gibbs sampler.
5. Provide the plot of the relative frequencies of $R = \mathsf{T}$ and $C = \mathsf{T}$ up to each iteration t against t, for two independent runs of the sampler. Suggest a *burn-in* time based on this plot. (1 point)
6. Apply the *Gelman* test and plot potential scale reduction factor changes over the iterations using *gelman.plot()* from the *coda* package. Roughly speaking, this factor measures the ratio of the variances within and between independent runs of the sampler. Thus, for a stationary distribution, this factor should be close to 1.0. Suggest a *burn-in* time based on this plot. (2 points)
7. Investigate the auto-correlation among the samples. We expect adjacent members from a Gibbs sampling sequence to be positively correlated, and we can quantify the amount of this correlation by using the auto-correlation function. The lag-k auto-correlation ρ_k is the correlation between every draw and its kth neighbouring samples. Use the R-function *acf()* to provide plots for both variables *Rain* and *Cloudy*. Suggest an interval for drawing approximately independent samples. (2 points)
8. Re-estimate $P(R = \mathsf{T} \mid S = \mathsf{T}, W = \mathsf{T})$ based on 100 samples obtained after the suggested *burn-in* time and thinning-out. Compare with (3) and comment on your results. (1 point)
9. Compute the probability $P(R = \mathsf{T} \mid S = \mathsf{T}, W = \mathsf{T})$ analytically and compare it to the sampling estimate. In real world applications, sampling is performed, because it is usually not possible to easily compute the probabilities analytically. However, since the Bayesian network in Figure 1 is only a small network with discrete variables, the analytical approach is possible. (2 points)

Your report should consist of the answers to the questions and the concrete results mentioned above (formulas, plots, values, etc.) in a single PDF file, and the R code in a separate file.

Please submit your solutions by Jan 8, 12:00 noon, to k.gogolewski@mimuw.edu.pl Please format the subject of your e-mail as follows: [SAD2]Name_LastName_IndexNo, additionally, please use the same name for the PDF report file.

References

[1] Gelman, A and Rubin, DB, *Inference from iterative simulation using multiple sequences*, Statistical Science, 7, 457-511, 1992.