Probabilistic Foundations of Artificial Intelligence
Problem Set 4

Nov 4, 2016

1. Bayesian networks and Markov chains

Consider the query $P(R|S = t, W = t)$ in the Bayesian network on Slide 19 of `https://las.inf.ethz.ch/courses/pai-f16/slides/pai-06-bayesian-networks-sampling-annotated.pdf` and how Gibbs sampling can answer it.

(i) How many states does the Markov chain have?

(ii) Calculate the transition matrix T containing $P(X_{t+1} = y \mid X_t = x)$ for all x, y.

(iii) What does T^2, the square of the transition matrix, represent?

(iv) What about T^n as $n \to \infty$?

(v) Explain how to do probabilistic inference in Bayesian networks, assuming that T^n is available. Is this a practical way to do inference?

2. Gibbs sampling

In this exercise, you will implement a Gibbs sampling algorithm for performing approximate inference in Bayesian networks. Although using a factor graph is not necessary for Gibbs sampling, we will use the already available factor graph representation from the previous problem set to conveniently acquire the Markov blanket of each variable. That way, all information required to compute the posterior distribution of a variable v given some values for all other variables, is contained in the CPTs of the neighboring factor nodes $\mathcal{N}(v)$.

More concretely, let $\boldsymbol{x}_{-v}$ be the set of all variables except for v, and $\boldsymbol{s}_{-v}$ be the value of those variables at the current iteration. Similarly, let $\boldsymbol{x}_{f\setminus v}$, $\boldsymbol{s}_{f\setminus v}$ be all variables that participate in factor f except for v, and $\boldsymbol{s}_{f\setminus v}$ be the values thereof. Then, to update the value of v you will have to draw from the posterior

$$P(v = d \mid \boldsymbol{x}_{-v} = \boldsymbol{s}_{-v}) = \frac{1}{Z} \prod_{f \in \mathcal{N}(v)} f(v = d, \boldsymbol{x}_{f\setminus v} = \boldsymbol{s}_{f\setminus v}),$$

where Z is a normalization factor. In practice, you will compute the above product (without the $1/Z$ part) for all $d \in \mathrm{dom}(v)$, then normalize to get a proper distribution, and finally draw from that distribution to obtain a new value for v.

You are provided some skeleton Python code in the `.zip` file accompanying this document. Take the following steps for this exercise.

(i) Install the Python dependencies listed in `README.txt`, if your system does not already satisfy them. After that, you should be able to run `demo.py` and produce some plots, albeit wrong ones for now.

(ii) Implement the missing code in `sampling.py` marked with `TODO`. In particular, you have to fill in the part that computes the posterior distribution discussed above, as well as the part that picks a variable and updates the state of the Gibbs sampler.

(iii) If your implementation is correct, you should get (approximately) correct results for the naive Bayes model of the demo file that represents the coin flipping network of exercise 3 in Problem Set 2.

(iv) Now, you can try out your Gibbs sampler on the earthquake network of the previous problem set. Compare your results to those you obtained using belief propagation. There are three parameters you can tune:

- The starting state of the Gibbs sampler. By default, it is created by drawing independent and uniformly random values for each variable.
- The length of the burn-in period, during which the state is updated, but not saved. Therefore, anything sampled during this stage has no effect on the approximate marginals computed afterwards.
- The function used to obtain the approximate marginals that are plotted. By default, this function is a cumulative average, i.e., it computes the approximate marginal distribution of a variable at step i by looking at the average number of occurences of each value of that variable among the samples obtained by the algorithm up to step i. A simple modification would be to only use every k-th sample when computing these averages, since successive samples are heavily correlated.

3. Markov chains and detailed balance

Assume that you are given a Markov chain with state space Ω and transition matrix T, which is defined for all $x, y \in \Omega$ and $t \geq 0$ as $T(x, y) := P(X_{t+1} = y \mid X_t = x)$. Furthermore, let π be the stationary distribution of the chain.

(i) Show that, if for some t the current state X_t is distributed according to the stationary distribution and additionally the chain satisfies the detailed balance equations

$$\pi(x)T(x, y) = \pi(y)T(y, x), \text{ for all } x, y \in \Omega,$$

then the following holds for all $k \geq 0$ and $x_0, \ldots, x_k \in \Omega$:

$$P(X_t = x_0, \ldots, X_{t+k} = x_k) = P(X_t = x_k, \ldots, X_{t+k} = x_0).$$

(This is why a chain that satisfies detailed balance is called *reversible*.)

(ii) Show that, if T is a symmetric matrix, then the chain satisfies detailed balance, and the uniform distribution on Ω is stationary for that chain.