Probabilistic Programming Homework 3

Submit your solutions to the TA in the homework submission box in the third floor of the E3-1 building by 2:00pm on 16 May 2018 (Wednesday). If you type up your solutions, you can email them to him (kwonsoo.chae@gmail.com).

Question 1

This question is concerned with operational semantics for probabilistic programs. Throughout the question, we will use the notions of expression, value, redex and evaluation context, and the $\mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow}$ and $\rightsquigarrow$ relations for the likelihood weighted importance sampler that we used in the course. Here is a reminder of these notions:

$$\begin{array}{rl}
\text{Expression} & e ::= c \mid x \mid (\texttt{fn}\ [x_1 \ldots x_n]\ e) \mid (e_0\ e_1\ \ldots\ e_n) \mid (\texttt{if}\ e_0\ e_1\ e_2) \\[1em]
\text{Constant} & c ::= \texttt{true} \mid \texttt{false} \mid \texttt{and} \mid \texttt{not} \mid \leq \mid 1.2 \mid + \mid * \mid \ldots \\
& \quad \mid d \mid \texttt{sample} \mid \texttt{observe} \\[1em]
\text{Distribution Constructor} & d ::= \texttt{normal} \mid \texttt{flip} \mid \texttt{poisson} \mid \ldots \\[1em]
\text{Value} & f ::= c \mid x \mid (\texttt{fn}\ [x_1 \ldots x_n]\ e) \mid (d\ v_1\ \ldots\ v_n) \\[1em]
\text{Redex} & f ::= ((\texttt{fn}\ [x_1 \ldots x_n]\ e)\ v_1 \ldots v_n) \mid (c\ v_1 \ldots v_n) \\
& \quad \mid (\texttt{if}\ \texttt{true}\ e_1\ e_2) \mid (\texttt{if}\ \texttt{false}\ e_1\ e_2) \\[1em]
\text{Evaluation Context} & C ::= [-] \mid (v_0\ v_1 \ldots v_{m-1}\ C\ e_{m+1} \ldots e_{m+n}) \mid (\texttt{if}\ C\ e_1\ e_2)
\end{array}$$

We also recall the $\mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow}$ and $\rightsquigarrow$ relations for the likelihood weighted importance sampler:

$$\frac{}{(\texttt{if}\ \texttt{true}\ e_1\ e_2), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} e_1, w} \qquad \frac{}{(\texttt{if}\ \texttt{false}\ e_1\ e_2), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} e_2, w}$$

$$\frac{}{((\texttt{fn}\ [x_1 \ldots x_n]\ e)\ v_1 \ldots v_n), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} e[x_1 := v_1, \ldots, x_n := v_n], w}$$

$$\frac{c \notin \{\texttt{sample}, \texttt{observe}\} \qquad c' \leftarrow \text{compute\_op}(c, [v_1, \ldots, v_n])}{(c\ v_1 \ldots v_n), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} c', w}$$

$$\frac{c' \leftarrow \text{sample\_dist}(d, [v_1, \ldots, v_n])}{(\texttt{sample}\ (d\ v_1 \ldots v_n)), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} c', w} \qquad \frac{w' \leftarrow \text{score\_dist}(d, [v_1, \ldots, v_n], c)}{(\texttt{observe}\ (d\ v_1 \ldots v_n)\ c), w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} c, w \cdot w'}$$

$$\frac{f, w \mathrel{\rightsquigarrow\!\!\!\!\!\rightsquigarrow} e', w'}{C[f], w \rightsquigarrow C[e'], w'}$$

You can find more information about them in the handwritten notes in the course webpage. Also, the note “Implementing Inference Algorithm for Probabilistic Programs” in the course webpage contains essentially the same information but with slightly different notations and presentation.

(a) Typically, the `let` construct in a programming language is introduced as a macro that gets expanded as follows:

$$(\texttt{let}\ [x\ e]\ e') \equiv ((\texttt{fn}\ [x]\ e')\ e).$$

Suppose that we do not want to treat `let` as macro, but we want to regard it as one of the core constructs in the language. Extend the definitions of redex f, evaluation context C, and the $\mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}}$ relation such that `let` expressions are evaluated directly by the $\mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}}$ relation, instead of first being macro-expanded and then being evaluated. Hint: You need to add one new case to the current definitions of redex f and evaluation context C, and one new rule for $\mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}}$.

(b) Consider the following program:

$$\begin{array}{l}
(\texttt{let}\ [x\ (\texttt{sample}\ (\texttt{uniform-continuous}\ 0.1\ 0.9))] \\
\quad (\texttt{let}\ [y\ (\texttt{observe}\ (\texttt{flip}\ x)\ \texttt{true})] \\
\quad\quad (\texttt{if}\ (>\ x\ 0.5)\ 0\ 1)))
\end{array}$$

Let us use e to denote this program. There are many ways of evaluating $(e, 1)$ to a value-weight pair using the $\rightsquigarrow$ relation extended in Part (a). For instance, the following repeated application of the $\rightsquigarrow$ relation leads to the pair of value 1 and weight 0.2:

$$\left(\begin{array}{l}
(\texttt{let}\ [x\ (\texttt{sample}\ (\texttt{uniform-continuous}\ 0.1\ 0.9))] \\
\quad (\texttt{let}\ [y\ (\texttt{observe}\ (\texttt{flip}\ x)\ \texttt{true})] \\
\quad\quad (\texttt{if}\ (>\ x\ 0.5)\ 0\ 1)))
\end{array},\ 1\right)$$

$$\rightsquigarrow \left(\begin{array}{l}
(\texttt{let}\ [x\ 0.2] \\
\quad (\texttt{let}\ [y\ (\texttt{observe}\ (\texttt{flip}\ x)\ \texttt{true})] \\
\quad\quad (\texttt{if}\ (>\ x\ 0.5)\ 0\ 1)))
\end{array},\ 1\right)$$

$$\rightsquigarrow \left(\begin{array}{l}
(\texttt{let}\ [y\ (\texttt{observe}\ (\texttt{flip}\ 0.2)\ \texttt{true})] \\
\quad (\texttt{if}\ (>\ 0.2\ 0.5)\ 0\ 1))
\end{array},\ 1\right) \rightsquigarrow \left(\begin{array}{l}
(\texttt{let}\ [y\ \texttt{true}] \\
\quad (\texttt{if}\ (>\ 0.2\ 0.5)\ 0\ 1))
\end{array},\ 0.2\right)$$

$$\rightsquigarrow ((\texttt{if}\ (>\ 0.2\ 0.5)\ 0\ 1),\ 0.2) \rightsquigarrow ((\texttt{if}\ \texttt{false}\ 0\ 1),\ 0.2) \rightsquigarrow (1,\ 0.2)$$

(Here I used an unexplained rule of $\rightsquigarrow$ for `let`) Using your rule for `let`, give two other possible execution sequences of the program e. Make sure that the final values of your execution sequences are 0 and 1, respectively.

Question 2

The lightweight Metropolis-Hastings algorithm uses the following acceptance ratio:

$$\alpha((v_{n-1}, w_{n-1}, S_{n-1}), (v', w', S')) = \min\left(1, \frac{w' \cdot |S_{n-1}|}{w_{n-1} \cdot |S'|}\right).$$

Here $(v_{n-1}, w_{n-1}, S_{n-1})$ is the current state consisting of value v_{n-1}, weight w_{n-1}, and sequence of random choices S_{n-1}, and (v', w', S') is a newly proposed state. With the probability $\alpha((v_{n-1}, w_{n-1}, S_{n-1}), (v', w', S'))$, we accept this new state and set the next state (v_n, w_n, S_n) to be (v', w', S'). For detail, look at the note "Implementing Inference Algorithm for Probabilistic Programs" in the course webpage.

Suppose that instead of the correct acceptance ratio above, we used the following variant:

$$\alpha'((v_{n-1}, w_{n-1}, S_{n-1}), (v', w', S')) = \min\left(1, \frac{w'}{w_{n-1}}\right).$$

Explain why this variant computes a wrong estimate for the following program:

```
(if (sample (flip 0.5))
  (or true (sample (flip 0.5)))
  false)
```

This (slightly silly) program does not contain any observe. Thus, its prior and posterior distributions coincide. In fact, the posterior distribution on boolean values assigns 0.5 to `true` and 0.5 to `false`. Hint: One way to answer this question is to focus on what the loop body of the variant of the lightweight MH algorithm does on v_n and to show that the target posterior distribution is not an invariant distribution (or stationary distribution) of the loop body.

Question 3

This question is about stochastic variational inference. Consider finite sets X and Y that are ranged over by x and y, respectively. Assume that we are given a probability distribution $p(x, y)$ on $X \times Y$ in the form of $p(x)$ and $p(y|x)$. The x part of (x, y) is an unobserved latent state, and the y part is an observation. Also, assume that we are given a particular observation y_0, and that we are interested in approximating the posterior distribution $p(x|y_0)$.

In variational inference, we solve this posterior-inference problem as follows. We consider a proposal distribution $q_\theta(x)$ parameterised by $\theta \in \mathbb{R}^m$ such that for fixed x, the function $\theta \mapsto q_\theta(x)$ is a differentiable function from $\mathbb{R}^m$ to the interval $[0, 1]$. Then, we set the following optimisation problem:

$$\text{argmin}_\theta \; \text{KL}\Big[q_\theta(x)\Big|\Big|p(x|y_0)\Big].$$

Finally, we solve this optimisation problem approximately.

Many variational-inference algorithms solve the above optimisation problem approximately by stochastic gradient descent. The core of such an algorithm often lies in its sample-based estimator of the following gradient:

$$\nabla_\theta \text{KL}\Big[q_\theta(x)\Big|\Big|p(x|y_0)\Big]. \tag{1}$$

In the lectures, we covered one such estimator, called score estimator or REINFORCE.

(a) Prove the following equation that forms the basis of the score estimator (and hence the basis of the stochastic variational-inference algorithm based on the score estimator as well).

$$\nabla_\theta \text{KL}\Big[q_\theta(x)\Big|\Big|p(x|y_0)\Big] = \mathbb{E}_{x \sim q_\theta(x)}\left[\Big(\nabla_\theta \log(q_\theta(x))\Big) \cdot \log \frac{q_\theta(x)}{p(x, y_0)}\right]. \tag{2}$$

(b) The equation in (2) gives rise to the following algorithm for estimating the gradient in (1). First, sample $x_1, \ldots, x_N$ from $q_\theta(x)$. Next, estimate the gradient by

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \Big(\nabla_\theta \log(q_\theta(x_i))\Big) \cdot \log \frac{q_\theta(x_i)}{p(x_i, y_0)}$$

Unfortunately, this algorithm is often not very effective, because its estimate has high variance.

One way of reducing variance is to use the estimator based on the following equation instead:

$$\nabla_\theta \mathrm{KL}\Big[q_\theta(x)\Big|\Big|p(x|y_0)\Big] = \mathbb{E}_{x\sim q_\theta(x)}\Big[\Big(\nabla_\theta \log(q_\theta(x))\Big)\cdot\Big(\log\frac{q_\theta(x)}{p(x,y_0)} - B\Big)\Big] \tag{3}$$

for some constant B (called baseline or control variate). Prove the equation (3). Also, change the algorithm such that it uses this equation instead of (2).

(c) Find B that minimises the variance of the estimate for $N = 1$. That is, find B that minimises

$$\mathrm{Variance}\left(\Big(\nabla_\theta \log(q_\theta(x))\Big)\cdot\Big(\log\frac{q_\theta(x)}{p(x,y_0)} - B\Big)\right),$$

which has the same value as

$$\mathbb{E}_{x\sim q_\theta(x)}\left[\left(\Big(\nabla_\theta \log(q_\theta(x))\Big)\cdot\Big(\log\frac{q_\theta(x)}{p(x,y_0)} - B\Big)\right)^2\right] - \left(\nabla_\theta \mathrm{KL}\Big[q_\theta(x)\Big|\Big|p(x|y_0)\Big]\right)^2$$

because of the equation (3).