CS267A: Probabilistic Programming and Relational Learning **Spring 2020**

Homework 2: Due Friday Apr. 24, 11:59PM

Instructions: Upload a PDF typeset using LaTeX containing your solutions (be sure to include your name and ID number on the document). No late sub-missions will be accepted. See the syllabus for policies about collaboration and academic honesty.

Problem 1

Topics: Logic

Let α, β, and γ be Boolean formulae, and let MC denote the model count of a Boolean formula. Select whether the following is a true or false statement about model counts and provide a brief justification for your choice.

- $\texttt{MC}(\alpha) + \texttt{MC}(\beta) + \texttt{MC}(\gamma) - \texttt{MC}(\alpha \wedge \beta \wedge \gamma) = \texttt{MC}(\alpha \vee \beta \vee \gamma)$

Problem 2

Topics: Probability rules

Suppose that α and β are independent events, that is $\Pr(\alpha \wedge \beta) = \Pr(\alpha)\Pr(\beta)$. Using basic probability rules, show that $\Pr(\neg\alpha \wedge \neg\beta) = \Pr(\neg\alpha)\Pr(\neg\beta)$.

Problem 3

Topics: Probability rules

Suppose there are 4 random variables A, B, C, D, and we know that A and B are independent given D. Please evaluate whether each of the following statements must be true, and state how you reach your conclusion:

- $\Pr(A, D) = \Pr(B, D)$
- $\Pr(A, B) = \Pr(A) \cdot \Pr(B)$
- $\Pr(A, B, C, D) = \Pr(A|C, D) \cdot \Pr(B|C, D) \cdot \Pr(C|D) \cdot \Pr(D)$
- $\Pr(A, B, C) = \Pr(A|B, C) \cdot \Pr(B|C) \cdot \Pr(C)$

Problem 4

Topics: Bayes rule

Suppose there is a rare and terrible disease which occurs with probability 10^{-6}. Doctors have developed a miracle diagnosis technique, which has the following table describing its accuracy:

(H) Has Disease	(T) Test Positive	$\Pr(T \mid H)$
T	T	$9/10$
T	F	$1/10$
F	T	$1/100$
F	F	$99/100$

Suppose you take the test and it comes back positive.

1. What is the probability that you have the disease?
2. Is this a good test? (I.e., would you be worried if it says you have the disease)

Problem 5

Topics: Independence and Discrete Probability

Consider the following partially-defined joint probability distribution on two random variables x and y:

x	y	$\Pr(x, y)$
0	0	$1/32$
0	1	θ_1
1	0	θ_2
1	1	$21/32$

The variables θ_1 and θ_2 are unknown numerical quantities.

1. What are the constraints on θ_1 and θ_2 so that this table describes a valid probability distribution? (Hint: Look at the axioms of probability).
2. Choose θ_1, θ_2 so that the marginal probability $\Pr(x = 0) = 1/8$.
3. Choose θ_1, θ_2 so that the conditional probability $\Pr(x = 0 \mid y = 1) = 1/21$.
4. Choose θ_1, θ_2 so that x and y are independent.

Problem 6

Programming Exercise: Implement a sampler

Problem 6 continued on next page. . .

Let $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ be a collection of n independent Boolean random variables, with each variable X_i being `true` with probability p_i, and `false` with probability $1 - p_i$. Let Δ be a CNF which uses $\{X_1, X_2, ..., X_n\}$ as its atoms.

Then, we can interpret Δ as a random variable with the following conditional distribution, where x is an assignment to each variable in $\mathbf{X}$:

$$\Pr(\Delta = \texttt{true} | \mathbf{X} = x) = \begin{cases} 1 & \text{if } x \text{ is a satisfying assignment to } \Delta. \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For this problem, you will be computing the marginal probability $\Pr(\Delta = \texttt{true})$, from the joint distribution $\Pr(\Delta, X_1, X_2, ..., X_n)$. Intuitively, this is the probability that Δ is satisfied if we randomly draw assignments to the $\mathbf{X}$ variables according to specified distributions.

Part A: *Probability of Satisfaction*

Let $\mathbf{X} = \{X_1, X_2\}$ be two independent random variables that are each true with probability $\frac{1}{2}$.

- Let $\Delta = X_1 \wedge X_2$. What is $\Pr(\Delta = \texttt{true})$?
- Let $\Delta = X_1 \vee X_2$. What is $\Pr(\Delta = \texttt{true})$?

Part B: *Write a Sampler*

For this question, you will be implementing a method to approximate the probability $P(\Delta = \texttt{true})$. To do this, you will use a *Monte-Carlo* sampler. You will not need to turn in your code; we will only ask for the output of your program.

A Monte-Carlo sampler estimates a probability by randomly drawing many samples. For example, imagine we have a mysterious coin which lands on heads with some unknown probability p. We can flip the coin as many times as we want, but we can't directly observe p. We can estimate p by simply flipping the coin n times, with the estimate getting more accurate as n increases. Imagine we flip the coin 4 times and get {heads, heads, tails, heads}. Then, we can guess that $p = \frac{3}{4}$ based on the proportion of times that we see heads in this finite set.

We want to apply Monte-Carlo sampling to our problem of estimating the probability that Δ is satisfied. Formally, we want you to implement a function which performs the following task:

- *Input*:
  1. A CNF Δ, specified as a list of lists of integers, where a positive integer is a positive literal and a negative integer is a negated literal (just like the CNF from last homework).
     Example: $\Delta = [[1, -2], [-1, 2]]$.
  2. *Probabilities*: A map w from each variable to its probability of being true. You can implement this as a dictionary.
     Example: $w = [(1 \mapsto 0.5), (2 \mapsto 0.9)]$ says variable 1 is true with probability 0.5, and variable 2 is true with probability 0.9.
  3. n, the number of samples to draw.
- *Output*: The approximate probability that $\Pr(\Delta = \texttt{true})$ if we randomly draw n samples according to the distribution specified by w.

We recommend you solve this problem by implementing the following helper methods:

- *Draw Sample*: A method which draws a random assignment to variables according to a specified weight function w.
- *Substitution*: A method which substitutes the value of each sampled variable into Δ, and computes whether or not Δ is satisfied under this assignment.

Part C: *Evaluate*

For each of the following, use $n = 1000$ samples to compute $\Pr(\Delta = \texttt{true})$, given the probabilities for each atom. You should report 3 runs for each question.

1. $(a \vee b \vee \neg c) \wedge (b \vee c \vee d \vee \neg e) \wedge (\neg b \vee \neg d \vee e) \wedge (\neg a \vee \neg b)$ with $\Pr(a) = 0.3, \Pr(b) = 0.6, \Pr(c) = 0.1, \Pr(d) = 0.8, \Pr(e) = 0.4$
2. $(\neg a \vee c \vee d) \wedge (b \vee c \vee \neg d \vee e) \wedge (\neg c \vee d \vee \neg e)$, with $\Pr(a) = 0.2, \Pr(b) = 0.1, \Pr(c) = 0.8, \Pr(d) = 0.3, \Pr(e) = 0.5$

You do not need to turn in your code, just the result of running your code with these parameters.