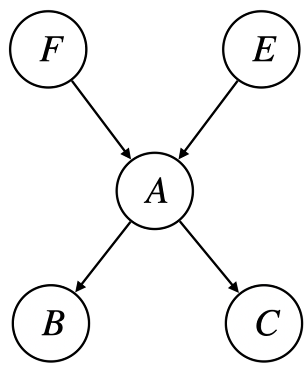
CSC 665: Artificial Intelligence

Homework 5

By turning in this assignment, I agree to abide by SFSU's academic integrity code and declare that all of my solutions are my own work.

1 Inference in Bayesian networks

Consider the alarm Bayesian network we studied in class.



Each of these random variables is binary-valued. F indicates whether your home is on fire, E indicates whether an earthquake is happening, A indicates whether your alarm system has gone off, B indicates whether your neighbor Bob has called to tell you that your alarm has sounded, and C indicates whether your neighbor Carol has called to tell you that your alarm has sounded.

The local conditional probability distributions for each of these random variables are as follows:

$P(F=1)$
0.01

$P(E=1)$
0.02

f	e	$P(A=1 \mid F=f, E=e)$
1	1	0.95
1	0	0.94
0	1	0.29
0	0	0.01

a	$P(B=1 \mid A=a)$
1	0.90
0	0.05

a	$P(C = 1 \mid A = a)$
1	0.70
0	0.01

a. (15 points) Perform exact inference by enumeration to compute $P(F = 1 \mid B = 1)$ and $P(F = 1 \mid C = 1)$. You may do this by hand or you may write a program, but please show your work in either case.

b. (15 points) Write a program to perform rejection sampling to estimate $P(F = 1 \mid B = 1)$ and $P(F = 1 \mid C = 1)$. Perform separate runs of rejection sampling using $N = 10, 100, 1000, 10000$ samples. (Note that N refers to the *total* number of samples on a given run, many of which will be rejected.) Show the results of your sampling runs by producing a plot of the estimated value for each of the two probabilities as a function of N. Include the true value of each probability (computed in part (a)) in the plot as a horizontal line.

c. (15 points) Write a program to perform likelihood weighting to estimate $P(F = 1 \mid B = 1)$ and $P(F = 1 \mid C = 1)$. Perform separate runs of likelihood weighting using $N = 10, 100, 1000, 10000$ samples. Show the results of your sampling runs by producing a plot of the estimated value for each of the two probabilities as a function of N. Include the true value of each probability (computed in part (a)) in the plot as a horizontal line.

d. (10 points *extra credit*) Make an insightful comment about some aspect of the results obtained above.

2 Gradient descent

Suppose we wish to use a linear model for a binary classification in which the label space is $\mathcal{Y} = \{-1, +1\}$. In this case, squared error is no longer an appropriate cost function (think about why). Instead, a common choice of pointwise cost is the *hinge loss*, defined by

$$c(\hat{y}, y) = \max\{0,\, 1 - \hat{y}y\},$$

where $y \in \mathcal{Y}$ is the true label and $\hat{y} = h(x)$ is the prediction produced by our model.

a. (10 points) Sketch a plot of $c(\hat{y}, +1)$ and $c(\hat{y}, -1)$ as a function of $\hat{y}$ on the interval $[-5, 5]$. Based on these plots, briefly argue why the hinge loss is a reasonable cost function for binary classification.

b. (20 points) Suppose that we have a single real-valued feature, i.e. $\mathcal{X} = \mathbb{R}$, and that our hypothesis class is the set of linear models of the form $h(x) = w_0 + w_1 x$. Compute

$$\frac{d}{dw_0} c(h(x), y) \text{ and } \frac{d}{dw_1} c(h(x), y),$$

where c is the hinge loss defined above.

c. (20 points) Consider the following training dataset:

i	x_i	y_i
1	-5	-1
2	1	$+1$
3	-2	-1
4	2	-1
5	4	$+1$
6	7	$+1$

Perform stochastic gradient by hand on this dataset to optimize the hinge loss with respect to the parameters of the linear model. Use the initialization $w_0 = w_1 = 0$. On the ith iteration of SGD, use the single training example (x_i, y_i) to update w_0 and w_1 with a learning rate of $\eta = 0.1$. Perform a total of six iterations of SGD (one for each example), showing the values of w_0 and w_1 after the update on each iteration.

d. (15 points) Given a trained linear model h, we can make class predictions according to the sign of h. That is, given a new input x, we classify x as positive $(+1)$ if $h(x) \geq 0$ and negative (-1) if $h(x) < 0$. Using the final weights from part (c) and this classification rule, what is the misclassification rate of h on the training dataset? What is the best achievable misclassification rate among the set of all possible linear models? If our learned h does not achieve this rate, explain why not.

e. (10 points *extra credit*) Write a program to execute (non-stochastic) gradient descent for the linear model on the dataset above. Each iteration of gradient descent should use the *entire* dataset to update the weight parameters. As usual, let the cost function C for the entire dataset be the sum of the hinge losses c for each training example. Run your program to convergence of the parameter values. Compare the h learned here to the h learned in part (c).

Submission

Submission is done on Canvas. You should submit either two or three files: one containing your solutions to the written problems, one for the coding problem in question 1, and optionally one for the extra credit coding problem in question 2.

- Submit your written solutions in a single PDF file with your name at the top. Make sure to clearly indicate the number and letter of the problem corresponding to each solution. It is okay to hand-write your solutions and then scan them into a PDF, but *only if your handwriting is legible.*
- Submit your coding solutions to question 1 in a file named `bn.py`.
- (Optional) Submit your coding solution to question 2 (e) in a file named `sgd.py`.