CSCI 447/547: Assignment 3

April 12, 2018

Due May 4th. Problems marked with an asterisk are only required for grad students. However, undergrads may undertake them at no penalty, and will receive extra credit for successful completion. Turn in responses and plots, as well as code.

1 Bayesian Networks (20/30 pts)

1A 10 pts

Adapted from Bishop Exercise 8.3. Consider three binary variables $a, b, c \in \{0, 1\}$ having the joint distribution given in Table 1. Show by direct evaluation that this distribution has the property that a and b are marginally dependent, so that $P(a, b) \neq P(a)P(b)$, but that they become independent when conditioned on c, so that $P(a, b|c) = P(a|c)p(b|c)$ for both $c = 0$ and $c = 1$.

1B 10 pts

Adapted from Bishop Exercise 8.4. Evaluate the distributions $P(a)$, $P(b|c)$, and $P(c|a)$ corresponding to the joint distribution given in Table 1. Show by direct evaluation that $P(a, b, c) = P(a)P(c|a)P(b|c)$. Draw the corresponding directed graph.

1C (*) 10 pts

Draw the graph associated with

$$P(A, B, C, D, E) = P(A)P(C)P(B|A, C)P(D|C)P(E|D),$$

a	b	c	P(a,b,c)
0	0	0	0.192
0	0	1	0.144
0	1	0	0.048
0	1	1	0.216
1	0	0	0.192
1	0	1	0.064
1	1	0	0.048
1	1	1	0.096

Table 1:

where $A, B, C, D, E \in \{0, 1\}$ with conditional probability tables

$$
\begin{aligned}
P(A = 1) &= 0.3 \\
P(C = 1) &= 0.7 \\
P(B = 1|A, C) &= \begin{cases} 0.3 \text{ if } A = 0, C = 0 \\ 0.7 \text{ if } A = 1, C = 0 \\ 0.1 \text{ if } A = 0, C = 1 \\ 0.9 \text{ if } A = 1, C = 1 \end{cases} \\
P(D = 1|C) &= \begin{cases} 0.4 \text{ if } C = 0 \\ 0.2 \text{ if } C = 1 \end{cases} \\
P(E = 1|D) &= \begin{cases} 0.1 \text{ if } D = 0 \\ 0.1 \text{ if } D = 1 \end{cases} .
\end{aligned}
$$

Compute the probability $P(A = 1|E = 1, C = 1)$. Be sure to think carefully about conditional independence.

2 Markov Models: Gene sequence clustering

2A Construction of a Markov Model (30 pts)

Load the file genes_training.p, which is available in this homework archive. genes_training.p contains 2000 sequences, with each sequence $\mathbf{s}$ consisting of 20 nucleobases $s_i \in \text{Nu}$, $\text{Nu} = \{A, T, G, C\}$. Each of these sequences is generated from one of two separate Markov processes. The label (aka class) of the process that generated the sequence is given in the dataset.

Learn the Markov model for each class given the training data. To do this, for each class compute the prior probability π_c of each nucleobase (the relative frequency of each nucleobase for each class, a vector of length 4) and the matrix of transition probabilities

$$\mathcal{A}_{c,kj} = P(s_{i+1} = \text{Nu}_j | s_i = \text{Nu}_k),$$

which is a 4 by 4 matrix. As a quick sanity check, each row of $\mathcal{A}_c$ should sum to one. Using these priors and transition matrices, write a function that generates a new sequence given the class, i.e. simulates a data point.

2B Sequence classification (30 pts)

Having the prior and transition probabilities gives you the ability to evaluate the likelihood of a sequence for a given class as:

$$P(\mathbf{s}) = P(s_1|\pi_c) \prod_{i=1}^{n-1} P(s_{i+1}|s_i, \mathcal{A}_c),$$

where π_c is the class-conditioned prior probability, and $\mathcal{A}_c$ is the class-conditioned transition matrix. Comparing the computed likeihood for a given sequence between different classes forms the basis

of a classifier in a very similar manner to naive Bayes. The difference this time is that now each random variable depends on the one before it in the sequence, whereas in naive Bayes we assumed that all the random variables were independent.

Load the file genes_test.p, which is similar to genes_training.p. For each sequence, compute the likelihood for both classes and assign a label. Compare this predicted label to the known one, and report the test set accuracy. As a point of comparison, my implementation achieved 98.7% accuracy.

2C (*) Comparision to naive Bayes (10 pts)

Classify the sequences again, but using a naive Bayes model instead of a Markov model. The likelihood in this simpler case is

$$P(\mathbf{s}) = \prod_{i=1}^{n} P(s_i|\pi_c),$$

i.e. the classification is based exclusively on how many times a given nucleobase is seen in the sequence. Note that you already have done the training for this model! The only change from the previous section is the use of a simpler likelihood function. Report the test set accuracy for the naive Bayes model and comment on the difference between this and the Markov model. Why does naive Bayes' do so much worse for this data?