Homework 2

Note: you may use the .pdf method from any of the classes in scipy.stats. However, beyond that you should calculate all other statistics, including covariance and kl divergence, using your own methods, using only scalar arithmetic and iteration (i.e., do not use the numpy, scipy etc. library functions for covariance, kl divergence etc).

1. Download garage.csv. It contains data similar to that presented on page 40 in the slides for Chapter 3 of Goodfellow, available in this section, (see 03_prob.pdf in the 'Slides' module). Here, one represents true and zero is false.
   a. [3 points] Compute the estimated covariance matrix for all variables except for invoice_no. The columns and rows should both be ordered as "fan", "radiator", "engine", "temp", "starts", "low_oil", "oil_light." You should present this matrix in your pdf writeup.
   b. [3 points] Compute the complete joint probability mass function table as shown in page 41 of the notes for Chapter 3, based on the data in garage.csv. You do NOT have to present this in your writeup, however, it should be clear from reading your code how you did this.
   c. [3 points] Compute the contingency tables for each of "fan", "radiator", "engine", "temp", "starts", "low_oil", "oil_light" based on the dependencies given in page 46 of the notes for Chapter 3. You should present these tables in your pdf writeup of the homework.
   d. [3 points] Compute P[fan = 1 | low_oil = 1, starts = 0], using the table from (b).
   e. [3 points] Compute P[fan = 1 | low_oil = 1, starts = 0], using the tables from (c).

   Include any code you write in an executable file called graphical.py.
2. The scipy.stats library provides a powerful set of probability distribution classes that let you sample from standard pdfs using common methods. See the code sample.py below. However, in machine learning a common problem is that we often work with unknown distributions that we cannot directly sample from. The purpose of this exercise is to practice methods for estimating pdfs via pmfs of empirical distributions. Let U = [0, 0.001, 0.002, ..., 0.999, 1]. We can think of this as a sample of the uniform probability mass function over the interval [0,1]. Let u be the random variable associated with the empirical distribution U. I.e., for any $u \in$ u, P(u

= *u*) = 1/1001. For each of the the following distributions prob ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3)]:

i. Compute Z = prob.pdf(0) + prob.pdf(0.001) + prob.pdf(0.002) + ... + prob.pdf(1) (or use pmf if pdf does not exist for that distribution).
ii. For each *x* ∈ U, define P(x = *x*) = prob.pdf(x = *x*)/Z, where *x* is any value in U. Thus x ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3)] is now a random variable for an empirical estimate of prob.pdf.

Now compute the following. Include any code you write in an executable file called kl.py.

a. [7points] Graph each of the random variables x ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3)] where x is generated as in i-ii above. That is, the horizontal axis should be the possible values of x (i.e., 0, .001, .002, ..., 1) and the y-axis should be the probability of each.
b. [8 points] Create a table where the columns are [bernoulli(.7), norm, laplace, expon, poisson(.3)]. and the rows are:
    1. Z, for each prob ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3)], calculated according to i-ii above.
    2. KL(u || x) for each x ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3)], calculated according to i-ii above.
    3. KL(x || u) for each x ∈ [bernoulli(.7), norm, laplace, expon, poisson(.3), calculated according to i-ii above.

0 % 0 of 2 topics complete

garage
CSV File

sample
PY File