ASSIGNMENT 2 (part 1)

Mathematics for Machine Learning 811 — 24 January 2022

Problems 1, 2 and 4(a) are intended for pen-and-paper, and your submission should include steps. The rest of the problems can be done with the aid of Python.

What you submit must be your own work, and sources other than the lecture material must be cited. Remember to append a signed plagiarism declaration to your submission.

---

1. Consider the bivariate distribution $p(x, y)$ of two discrete random variables X and Y, given in the table on the right.

Y					
y_1	0.01	0.02	0.03	0.10	0.10
y_2	0.05	0.10	0.05	0.07	0.20
y_3	0.10	0.05	0.03	0.05	0.04
	x_1	x_2	x_3	x_4	x_5
			X		

(a) Find the marginal distributions $p(x)$ and $p(y)$.

(b) Find the conditional distribution $p(x \mid Y = y_1)$.

(c) Suppose the value of x_i is i, and the value of y_i is i. Compute the correlation between X and Y.

2. In a factory there are three machines that make light bulbs. The machines manufacture 20%, 30% and 50% of the total production. From their production, 5%, 4%, and 2% respectively are faulty. I choose a collection of light bulbs at random from the factory's output.

(a) If the collection contains two faulty light bulbs, what is the probability that those two come from the same machine?

(b) If the collection contains three faulty light bulbs, what is the probability that those three come from three different machines?

3. Generate 1,000 samples from the Gaussian distribution $\mathcal{N}\left(\begin{bmatrix}3\\2\end{bmatrix}, \begin{bmatrix}10 & 5\\5 & 5\end{bmatrix}\right)$ by means of

(a) the procedure given in section 6.5.4 of the textbook;

(b) the probability integral transform (see top of page 217 in the textbook).

Hint: the inverse Gaussian cdf can be written in terms of the inverse error function (`scipy.special.erfinv`).

For each method separately, plot the samples, calculate the empirical mean and covariance, and compare with the true mean and covariance.

4. Each of the N rows in `x.dat` is a training sample $\boldsymbol{x}_n \in \mathbb{R}^2$ with corresponding label $y_n \in \{0, 1\}$ in `y.dat`. Your task will be to use gradient descent in order to fit a logistic classifier,

$$\sigma(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b)},$$

to this data. The model parameters are $\boldsymbol{w} \in \mathbb{R}^2$ and $b \in \mathbb{R}$, and a sensible choice for the objective function in this case is the "cross-entropy loss" between true labels y_n and predicted labels $\sigma(\boldsymbol{x}_n)$, averaged over the training set:

$$L(\boldsymbol{w}, b) = \frac{1}{N}\sum_{n=1}^{N}\Big[-y_n \log\big(\sigma(\boldsymbol{x}_n)\big) - (1 - y_n)\log\big(1 - \sigma(\boldsymbol{x}_n)\big)\Big].$$

p.t.o.

(a) Prove that the gradient descent update rules for minimising $L(\boldsymbol{w}, b)$ are as follows:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \gamma \frac{1}{N} \sum_{n=1}^{N} \big(\sigma(\boldsymbol{x}_n) - y_n\big)\boldsymbol{x}_n, \quad b_{i+1} = b_i - \gamma \frac{1}{N} \sum_{n=1}^{N} \big(\sigma(\boldsymbol{x}_n) - y_n\big).$$

(b) Initialise $\boldsymbol{w}_0$ and b_0 with zeros, and implement batch gradient descent. A fixed value for γ between 1 and 5 seems reasonable, but you are free to experiment. To check convergence, plot $L(\boldsymbol{w}_i, b_i)$ as a function of i. Give your final values of $\boldsymbol{w}$ and b.

(c) Plot all $\boldsymbol{x}_n$ points in the x_1–x_2 plane, using two different colours for $y_n = 0$ and $y_n = 1$, as well as the decision boundary of the trained logistic model. Here the decision boundary would consist of all $\boldsymbol{x}$ for which $\sigma(\boldsymbol{x}) = 0.5$, that is, $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b = 0$ (a straight line in the x_1–x_2 plane that tries to separate the two classes).

5. *coming soon...*
