CSE 151A: Machine learning

Homework 3

Instructions:

- You may discuss problems with your study group, but ultimately all your work (mathematical problems, code, experimental details) must be individual.
- Your solutions must be `typed up` and uploaded to Gradescope by 9.59PM on Thursday October 19. No late homeworks will be accepted under any circumstances, so you are encouraged to upload early.
- A subset of the problems will be graded.

Conceptual and mathematical problems

1. *Gaussian contours.* Roughly sketch the shapes of the following Gaussians $N(\mu, \Sigma)$. You only need to show a representative contour line which is qualitatively accurate (has approximately the right orientation, for instance).

   (a) $\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$

   (b) $\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 1 & -0.75 \\ -0.75 & 1 \end{pmatrix}$

2. *Qualitative appraisal of Gaussian parameters.* A bivariate Gaussian has covariance matrix $\begin{pmatrix} p & q \\ q & r \end{pmatrix}$. Give precise characterizations, in terms of p, q, r, of when the following are true.

   (a) The two variables are negatively correlated.

   (b) The two variables are uncorrelated.

   (c) One variable is a linear function of the other.

   (d) The second variable is a constant (i.e. always takes the same value).

3. Find the unit vector in the same direction as $x = (1, 2, 3)$.

4. Find all unit vectors in $\mathbb{R}^2$ that are orthogonal to $(1, 1)$.

5. The function $f(x) = 2x_1 - x_2 + 6x_3$ can be written as $w \cdot x$ for $x \in \mathbb{R}^3$. What is w?

6. For a certain pair of matrices A, B, the product AB has dimension 10×20. If A has 30 columns, what are the dimensions of A and B?

7. We have n data points $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ and we store them in a matrix X, one point per row.

   (a) What is the dimension of X?

   (b) What is the dimension of XX^T?

   (c) What is the (i, j) entry of XX^T, simply?

8. For $x = (1, 3, 5)$ compute $x^T x$ and xx^T.
9. The quadratic function $f : \mathbb{R}^3 \to \mathbb{R}$ given by

$$f(x) = 3x_1^2 + 2x_1x_2 - 4x_1x_3 + 6x_3^2$$

can be written in the form $x^T Mx$ for some **symmetric** matrix M. What is M?

10. Let $A = \text{diag}(1, 2, 3, 4, 5, 6, 7, 8)$.
    (a) What is $|A|$?
    (b) What is A^{-1}?
11. Vectors $u_1, \ldots, u_d \in \mathbb{R}^d$ all have unit length and are orthogonal to each other. Let U be the $d \times d$ matrix whose rows are the u_i.
    (a) What is UU^T?
    (b) What is U^{-1}?
12. Matrix $A = \begin{pmatrix} 1 & 2 \\ 3 & z \end{pmatrix}$ is singular. What is z?

Programming problems

13. *Classifying MNIST digits using generative modeling.* In class, we have already encountered the MNIST data set of handwritten digits. In this problem, you will build a classifier for this data, by modeling each class as a multivariate (784-dimensional) Gaussian.
    - Download the Jupyter notebook `generative-mnist.ipynb` from the course website. This will help you by loading in the MNIST data set.
      - Look over the notebook to see what it is doing, and then run it, one cell at a time.
      - Make sure you understand the form in which the training and test data are stored.
      - Towards the end of the notebook, there is also a helper function that displays digits.
    - Split the training set into two pieces – a training set of size 50000 (say), and a separate *validation set* of size 10000.
    - Now fit a Gaussian generative model to the training data of 50000 points.
      - Determine the class probabilities: what fraction π_0 of the training points are digit 0, for instance? Call these values $\pi_0, \ldots, \pi_9$.
      - Fit a Gaussian to each digit, by finding the mean and the covariance of the corresponding data points. Let the Gaussian for the jth digit be $P_j = N(\mu_j, \Sigma_j)$. Note that μ_j will be a 784-dimensional vector, and Σ_j will be a 784×784 matrix.

      Using these two pieces of information, you can classify new images x using Bayes' rule: simply pick the digit j for which $\pi_j P_j(x)$ is largest.
    - One last step is needed: it is important to smooth the covariance matrices, and the usual way to do this is to add in cI, where c is some constant and I is the identity matrix. Use the validation set to help you choose the right value of c: that is, choose the value of c for which the resulting classifier makes the fewest mistakes on the validation set.

- There are some important details of *numerical precision* that you will need to attend to. In 784-dimensional space, all probabilities $P_j(x)$ will likely be miniscule, and this can produce all sorts of trouble due to underflow errors. It is better to work with log-probabilities: -1000 is easier to deal with than e^{-1000}. This means that you should classify a point x by picking the j that maximizes $\log \pi_j + \log P_j(x)$. Fortunately, the Python `multivariate_normal` package will directly compute $\log P_j(x)$ for you.

To turn in:

(a) Pseudocode (or code) for your training procedure, making it clear how the validation set was created and used.

(b) What value of c did you get?

(c) What was the error rate on the MNIST test set?

(d) Out of the misclassified test digits, pick five at random and display them. For each instance, list the true label and predicted label.