Assignment 1

This assignment is due **Wednesday October 7** at **11:59PM**.
Solutions should be turned in through the assignment FTP site in PDF form. The name of the PDF file should be ps1_YourStudentID, e.g., "ps1_9762578.pdf". Your MATLAB code (with comments) should be included in the PDF file too. The FTP site is 140.114.71.2 port 1235. You can use the same ID and password of the course website to login.

---

1. **(20 points)** Matrix Derivatives (Bishop PRML, Appendix C).

   The derivatives of a scalar a with respect to a vector $\mathbf{x}$ and a matrix $\mathbf{X}$ are defined by

   $$\left(\frac{\partial a}{\partial \mathbf{x}}\right)_i = \frac{\partial a}{\partial x_i}$$

   and

   $$\left(\frac{\partial a}{\partial \mathbf{X}}\right)_{ij} = \frac{\partial a}{\partial X_{ij}},$$

   i.e., $\frac{\partial}{\partial \mathbf{x}} a$ is a vector whose ith component is $\frac{\partial}{\partial x_i} a$, and $\frac{\partial}{\partial \mathbf{X}} a$ is a matrix whose (i, j) element is $\frac{\partial}{\partial X_{ij}} a$.

   (1) Show that

   $$\frac{\partial}{\partial \mathbf{x}} \left(\mathbf{x}^T \mathbf{a}\right) = \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{a}^T \mathbf{x}\right) = \mathbf{a}$$

   by writing out the components.

   (2) Show that

   $$\frac{\partial}{\partial \mathbf{X}} \mathrm{Tr}\left(\mathbf{X}^T \mathbf{A} \mathbf{X}\right) = \left(\mathbf{A} + \mathbf{A}^T\right) \mathbf{X}$$

   by writing out the elements.

2. **(10 points)** Kullback-Leibler Divergence between Two Gaussians (PRML (1.113), (2.43), Exercise 2.13).

   Evaluate the Kullback-Leibler divergence between two Gaussians $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$.

3. **(10 points)** Differential entropy (PRML (1.104), (2.43), Exercise 2.15).

   Show that the entropy of the multivariate Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma})$ is given by

$$H[\mathbf{x}] = \frac{1}{2}\ln|\mathbf{\Sigma}| + \frac{D}{2}(1 + \ln(2\pi)) \ ,$$

   where D is the dimensionality of $\mathbf{x}$.

4. **(20 points)** Nonparametric Methods.

   Assume the data are obtained from some unknown probability density $p(\mathbf{x})$ in a D-dimensional Euclidean space. The probability mass with region $\mathcal{R}$ is

$$P = \int_{\mathcal{R}} p(\mathbf{x}) d\mathbf{x} \, .$$

   Each data point has a probability P of falling within $\mathcal{R}$. The total number K of points inside $\mathcal{R}$ is a binomial distribution

$$K \sim \text{Bin}(K|N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K} \, .$$

   Show that $\mathbb{E}[K/N] = P$ and $\text{var}[K/N] = P(1-P)/N$, and therefore, for large N, we have $K \cong NP$.

5. **(10 points)** 1-Nearest-Neighborhood in a High-Dimensional Space.

   Assume that N data points are uniformly distributed in the unit cube $[-1/2, 1/2]^D$. Let R_0 denote the radius of a 1-nearest neighborhood centered at the origin. Show that

$$\text{median}(R_0) = \rho_D^{-1/D} \left(1 - \left(\frac{1}{2} \right)^{\frac{1}{N}} \right)^{1/D} \ ,$$

   where $\rho_D \cdot r^D$ is the volume inside the sphere of radius r in D dimensions.

6. **(30 points)** K-Nearest-Neighbor Classification (MATLAB).

   Please download the MNIST Handwritten Digits data from

   `http://www.cs.toronto.edu/~roweis/data/mnist_all.mat`

After you load the .mat file in MATLAB, you will find 20 matrices containing 8-bit grayscale images of '0' through '9'. Each class has about $6,000$ training examples and $1,000$ test examples. Each image has been transformed into a column vector. For KNN classification, we prefer the data being stored in vector form, but you may 'reshape' it back as a matrix of size 28×28 if you want to see how the original image looks like. For the convenience of computation, you may convert the data from `uint8` into `double`, and divide them by 255 to make all components within $[0, 1]$.

The task is to use the training data to predict the class of the test data based on the K-nearest-neighbor criterion (see the lecture notes). Occasionally a tie among different classes might occur, and the K-nearest-neighbor criterion could not be applied. You may create your own criteria for such situations.

You need to do five-fold cross-validation on the training data to determine the parameter K. You may also try different distance functions, e.g., Manhattan distance, Euclidean distance, or other Minkowski distance with $p < 1$.

Please write a simple MATLAB program to do the task and report the error rates of classification.