**Assignment #4**

**Nonparametric Estimation**

**How TA evaluates your assignments:**

**Report:** half of your score will be graded proportional to the quality of your report. You should provide a distinct section for each problem, include the desired outputs and explain what you've done. Don't forget to discuss your results as well. It is not necessary to accommodate your source codes in your reports unless you want to refer to them. Compactness, expressiveness and neatness are of high importance.

**Source Code:** create an m-file for any problem and write all your codes there. If a problem consists of several sub-problems, separate them by comments in your code. Finally, name your m-files according to the number of the problems.

As you have to upload you submission electronically, it is of high interest to prepare your reports using Microsoft Office tools or Latex. However, scanned handwritten solutions are also acceptable as long as they are readable, neat and expressive.

**What to hand in:**

You must submit your report (.pdf) and source codes (m-files) for each assignment. Zip all your files into an archive file and use the following template to name it:

**HW4_XXXXX.zip**

where XXXXX must be replaced with your student ID. Your file size must not be bigger than 20MB. Send your files to *mohammadhme@gmail.com* with a subject of PR961_HW4_ XXXXX (replace XXXXX with your student number).

The Due Date for This Assignment is: Azar 19th

## Problems:

1. You are a microcontroller with a sensor that measures the light level, $x_i$, in arbitrary units. Your job is to determine the weather: is today $\omega_1 = "sunny"$, or $\omega_2 = "\text{cloudy}"$? You have been programmed to compute Parzen window estimates of $p(x|\omega_1)$ and $p(x|\omega_2)$ using the rectangular window:

$$\hat{p}(x) = \frac{1}{nv}\sum_{i=1}^{n}\varphi\left(\frac{x-x_1}{h}\right)$$

$$\varphi\left(\frac{x}{h}\right) = \begin{cases} 1 & |x| < \frac{h}{2} \\ 0 & otherwise \end{cases}$$

You have five labeled training days. Three days were sunny, with light levels of $x_1 = 4$, $x_2 = 1$, and $x_3 = 5$ units, respectively. Two days were cloudy, with light levels of $x_4 = 3$ and $x_5 = 2$ units.

a. Plot the Parzen window estimated likelihood $\hat{p}(x|\omega_1)$ as a function of x, using h = 1.

b. Plot the Parzen window estimated likelihood $\hat{p}(x|\omega_2)$ as a function of x, using h = 1.

2. Consider the following set of two-dimensional vectors:

| $\omega_1$ | | $\omega_2$ | | $\omega_3$ | |
|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_1$ | $x_2$ | $x_1$ | $x_2$ |
| 10 | 0 | 5 | 10 | 2 | 8 |
| 0 | -10 | 0 | 5 | -5 | 2 |
| 5 | -2 | 5 | 5 | 10 | -4 |

a. Plot the decision boundary resulting from the nearest-neighbor rule just for categorizing $\omega_1$ and $\omega_2$. Find the sample means $m_1$ and $m_2$ and on the same figure sketch the decision boundary corresponding to classifying x by assigning it to the category of the nearest sample mean.
b. Repeat part (a) for categorizing only $\omega_1$ and $\omega_3$
c. Repeat part (a) for categorizing only $\omega_2$ and $\omega_3$
d. Repeat part (a) for a three-category classifier, classifying $\omega_1$, $\omega_2$ and $\omega_3$

## Computer Projects:

Implement the following projects. Make sure that your codes are well-commented. Code-only submissions will gain at most 33/100 pts. So, do not forget to provide a compact, well-documented and informative submission.

3. Consider Parzen-window estimates and classifiers for points in the table below. Let your window function be a spherical Gaussian, i.e.:

$$\varphi\left(\left(x-x_i\right)/h\right) \propto \exp\left[-\left(x-x_i\right)^t\left(x-x_i\right)/\left(2h^2\right)\right]$$

a. Write a program to classify an arbitrary test point x based on the Parzen window estimates. Train your classifier using the three-dimensional data from your three categories in the table above. Set h = 1 and classify the following three points: $(0.50, 1.0, 0.0)^t$, $(0.31, 1.51, -0.50)^t$ and $(-0.3, 0.44, -0.1)^t$.
b. Repeat with h = 0.1.

| | $\omega_1$ | | | $\omega_2$ | | | $\omega_3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| sample | $x_1$ | $x_2$ | $x_3$ | $x_1$ | $x_2$ | $x_3$ | $x_1$ | $x_2$ | $x_3$ |
| 1 | 0.28 | 1.31 | -6.2 | 0.011 | 1.03 | -0.21 | 1.36 | 2.17 | 0.14 |
| 2 | 0.07 | 0.58 | -0.78 | 1.27 | 1.28 | 0.08 | 1.41 | 1.45 | -0.38 |
| 3 | 1.54 | 2.01 | -1.63 | 0.13 | 3.12 | 0.16 | 1.22 | 0.99 | 0.69 |
| 4 | -0.44 | 1.18 | -4.32 | -0.21 | 1.23 | -0.11 | 2.46 | 2.19 | 1.31 |
| 5 | -0.81 | 0.21 | 5.73 | -2.18 | 1.39 | -0.19 | 0.68 | 0.79 | 0.87 |
| 6 | 1.52 | 3.16 | 2.77 | 0.34 | 1.96 | -0.16 | 2.51 | 3.22 | 1.35 |
| 7 | 2.20 | 2.42 | -0.19 | -1.38 | 0.94 | 0.45 | 0.60 | 2.44 | 0.92 |
| 8 | 0.91 | 1.94 | 6.21 | -0.12 | 0.82 | 0.17 | 0.64 | 0.13 | 0.97 |
| 9 | 0.65 | 1.93 | 4.38 | -1.44 | 2.31 | 0.14 | 0.85 | 0.58 | 0.99 |
| 10 | -0.26 | 0.82 | -0.96 | 0.26 | 1.94 | 0.08 | 0.66 | 0.51 | 0.88 |

4. In this problem, we explore/demonstrate Parzen window density estimators.

**a.** Generate 100 samples from Gaussian distribution with $N(0,1)$

**b.** Use these samples to produce a Parzen window density estimator. For the window, use a Gaussian window with size h, i.e.

$$\varphi(x) = e^{-x^2/h}$$

**c.** For different values of *h*, plot the Parzen window estimate and compare it with the true density N(0, 1). What do you observe?

5. Generate 100 random training points from each of the following two distributions: N(20,5) and N(35,5). Write a program that employs the Parzen window technique with a Gaussian kernel to estimate the density, $\hat{p}(x)$, using all 200 points. Note that this density conforms to a bimodal distribution.

   **a.** Plot the estimated density function for each of the following window widths: h = 0.01, 0.1, 1, 10. [Note: You can estimate the density at discrete values of x in the [0,55] interval with a step-size of 1.]

   **b.** Repeat the above after generating 1000 training points from each of the two distributions.

   **c.** Discuss how the estimated density changes as a function of the window width and the number of training points.