Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 3

From parametric to non-parametric density estimation and classification!

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW3_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW3_[student_id].zip). Please combine all your reports just into a single .pdf file.
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon). Please do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include their pictures in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Do not forget to explain your answers clearly, and provide enough discussions when needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
- ***MATLAB is also allowable.*** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python functions.
- ***Be neat and tidy!*** Your codes must be separated for each question, and for each part. For example, you have to create a separate .py file for part b. of question 3, which must be named 'p3b.py'. (or .ipynb)
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics, or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle, because that is where all assignments as well as course announcements are posted. Homework submissions are only made through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of December 13th**.
- ***Delay policy.*** During the semester, students are given only **10 free late days** which they can use them in their own ways. Afterwards, there will be a 20% penalty for every late day, and no more than four late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings, and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please do not hesitate to contact us through the Telegram group chat or following email addresses: **m.ebadpour@aut.ac.ir** and **atiyeh.moghadam@aut.ac.ir**.

1.Solving a flexible puzzle! (15 Pts.)

Keywords: *density estimation, non-parametric density estimation, Parzen windows, kernel density estimation, k-nearest neighbours.*

In the real world, figuring out the patterns in data is like solving a puzzle with no fixed shape. Traditional methods assume the puzzle pieces fit a certain way, but that's not always true. In **non-parametric density estimation**, we try to understand data without forcing it into a predefined shape.

Figure 1:In the real world, figuring out the patterns in data is like solving a puzzle with no fixed shape.

Non-parametric density estimation doesn't limit us to specific assumptions, making it perfect for real-world data that's messy and doesn't follow a fixed pattern. This flexibility helps us make smarter decisions in uncertain situations, whether it's predicting stock prices, diagnosing illnesses, or assessing environmental risks. In this problem, we are taking a look at two simple approaches to non-parametric density estimation: **Parzen windows** and **k-nearest neighbors.**

You are given a dataset D= {0,1,1,1,3,3,4,4,4,4,4,4,4,5,5,6,6,6}. Using techniques from parametric and non-parametric density estimation, answer the following questions:

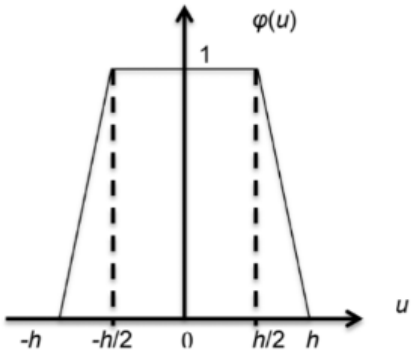
a. Draw a histogram of D with a bin-width of 1 and bins centered at {0,1,2,3,4,5,6}.
b. Write the formula for the kernel density estimate given an arbitrary kernel K.
c. Select a triangle kernel as your window function:

$$K(u) = (1 - |u|) \qquad |u| \leq 1$$

Where u is a function of the distance of sample x_i to the value in question x divided by the bandwidth: $u = \frac{x - x_i}{h}$. Compute the kernel density estimates for the following values of x = {0,1,2,3,4,5,6} bandwidths of 2.

d. Now, what if you assume that, rather, the density is a parametric density: it is a Gaussian. Compute the maximum likelihood estimate of the Gaussian's parameters.
e. Compare the histogram, the triangle-kernel density estimate, and the maximum-likelihood estimated Gaussian. Which best captures the data? What does each miss? Why would you choose one of another if you were forced to?

Now, We perform Parzen Window density estimation, using trapezoidal window functions given (in unnormalized form) in the figure below:



Choose h = 1. Assume that we have the following data:

$$D_{\omega_1} = \{0, 2, 5\}$$

$$D_{\omega_2} = \{4,7\}$$

f. Sketch the Parzen window estimates of the pdfs $p(x|\omega_1)$ and $p(x|\omega_2)$. Please label pertinent values on both axes. Calculate the prior probabilities based on data.

g. Use the estimates you have developed in above to find the decision boundaries and regions for a Bayes minimum-error classifier based on Parzen windows. Only the part of feature space where at least one density is nonzero need to be classified.

2. Non-parametric methods for ~~density estimation~~ classification! (20 Pts.)

Keywords*: parameter estimation, maximum likelihood estimation, Poisson distribution, unbiased estimator, estimator variance, consistent estimator, posterior probability, prior probability*

Density Estimation techniques can be used to perform classification. In fact, density estimation methods provide a more flexible and data-driven approach to classification by estimating the underlying probability density function (PDF) of each class. The PDF represents the probability distribution of the data points within a particular class, allowing for a more nuanced understanding of the class distribution. By leveraging the information captured by the PDF, density estimation methods can effectively classify data points into their respective classes, even in cases where traditional methods may fail. In this problem, we use these techniques for classification.

First, Consider the following set of two-dimensional vectors:

ω_1	ω_2	ω_3
(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-neighbour rule just for categorizing ω_1 and ω_2.
b. Repeat part (a) for categorizing only ω_2 and ω_3 and for a three-category classifier, classifying ω_1, ω_2 and ω_3.

Now, Consider classifiers based on samples from the distributions:

$$p(x|\omega_1) = \begin{cases} 2x & for\ 0 \le x \le 1 \\ 0 & otherwise, \end{cases}$$

And

$$p(x|\omega_2) = \begin{cases} 2 - 2x & for\ 0 \le x \le 1 \\ 0 & otherwise. \end{cases}$$

c. What is the Bayes decision rule and the Bayes classification error?
d. Suppose we randomly select a single point from ω_1 and a single point from ω_2, and create a nearest-neighbor classifier. Suppose too we select a test point from one of the categories (ω_1 for definiteness). find the expected error rate $p_1(e)$.
e. Repeat with two training samples from each category and a single test point in order to find $p_2(e)$. Generalize to find the arbitrary $p_n(e)$.

Up until now, we have utilized nearest neighbours for classification; now, let's employ the Parzen kernel density estimator.

Suppose we have a data set of input vectors $\{x_n\}$ with corresponding target values $t_n \in \{-1,1\}$, and suppose that we model the density of input vectors within each class separately using a Parzen kernel density estimator which is defined as follows:

$$p(x|t) = \frac{1}{N_t} \sum_{n=1}^{N} \frac{1}{Z_k} k(x, x_n) \delta(t, t_n)$$

where $k(x, x')$ is a valid kernel, Z_k is the normalization constant for the kernel and $\delta(t, t_n)$ equals 1 if $t = t_n$ and 0 otherwise.

f. Write down the minimum misclassification-rate decision rule assuming the two classes have equal prior probability.
g. Show that, if the kernel is chosen to be $k(x, x') = x^T x'$, then the classification rule reduces to simply assigning a new input vector to the class having the closest mean.

3. Use This Tool If You Were in a Regulatory Position! (25+5 Pts.)

Keywords: *Density estimation, Non-parametric methods, kernel density estimation, smoothing parameter, bandwidth.*

Every autumn, as we rejoice because of the rain and cold weather, we start to worry about the headaches and heartaches caused by air pollution, especially in Tehran. In the cold months of the year, we almost have air pollution every day. But only God knows in what ways this air pollution is threatening our health.

Figure 2: Over the past few years, Tehran's pollution levels have consistently remained within the 'moderate' range, with notable improvements in PM2.5 readings, highlighting the importance of monitoring fine particulate matter for overall air quality assessment.

In the context of air quality analysis, density estimation can be applied to understand the distribution of a specific air quality parameter, such as pollutant concentrations, over a certain period of time or across different locations. But the important point is how precise this estimation is. The precision required often depends on the intended use of the analysis. For regulatory compliance or health impact assessments, a higher level of precision may be necessary compared to exploratory analyses aimed at understanding general trends.

As we don't have much exact data about Tehran's air pollution, what we do in this problem is try to find a precise density estimator. For this aim, we use some of the distributions in the Marron-Wand distributions benchmark. The formula for these distributions is provided in 'Marron-Wand.txt.'

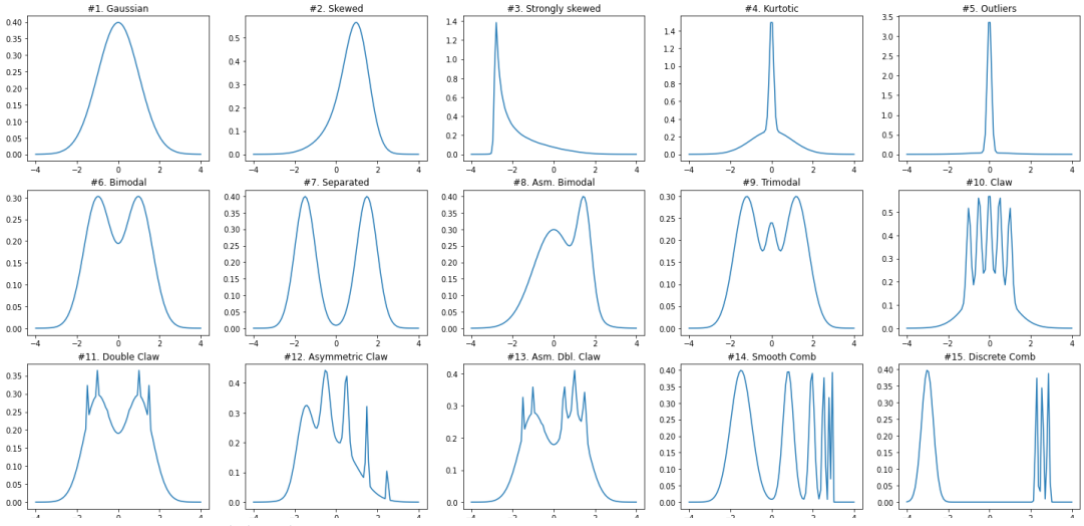


Figure 3: Marron-Wand distributions

a. Plot the PDFs of 'Skewed', 'Kurtotic', 'Asm. Bimodal', 'Trimodal'. Generate N i.i.d samples from the given PDFs, assuming N= {10, 100, 1000}.

b. For a univariate Gaussian kernel, it is often recommended to select $h^* \approx 1.06\hat{\sigma}N^{-\frac{1}{5}}$, where h^* is the optimal choice of bandwidth, N is the number of samples and $\hat{\sigma}$ is the estimate of

the standard deviation of the samples. Calculate the sample standard deviation, $\hat{\sigma}$. For each N, estimate the optimal value for bandwidth, $h^*(N)$.

c. Use kernel density estimation with a Gaussian kernel for each N to estimate the PDFs, considering three different bandwidth values $\{\frac{h^*(N)}{3}, h^*(N), 3h^*(N)\}$.

d. Summarise your results by plotting the two PDFs estimates. For each of the given densities, you need to have 9 plots overall (36 in total). Overlay each plot with the ground truth densities. Comment on the effects of h, N, and the kernel itself on the estimations you obtained.

e. Do the same steps you did in parts (a) to (d) for other Marron-Wand distributions. Is there any other choice of h that can improve the results in these distributions?

4. MLE vs. MAP: Which One Dominates in Image Retrieval Problem? (35 Pts.)

Keywords: *Parameter Estimation, Image Retrieval, Maximum Likelihood Estimation (MLE), Maximum A Posteriori (MAP) Estimation, Image Thresholding, Dirichlet Smoothing*

In this problem, you are to apply **Parameter Estimation** methods in a real-world problem, i.e. **Image Retrieval**, where given a query image and based on features extracted from some query image, the goal is to rank a set of images according to how similar their distribution of features are to the query image.

Consider K distinct types of features that can be measured at each pixel of an image. Assume there are n_1 pixels where a feature of type 1 exists, n_2 pixels where a feature of type 2 exists, and so on. Therefore, each image can be represented by a categorical distribution with K parameters $u_i = n_i / N$, where $N = n_1 + ... + n_K$. These values are essentially MLE estimates, obtained by forming a histogram of feature counts n_i over the whole image and then dividing by the sum of all counts to get a probability mass function that sums to one. Now let's define a scoring function which computes how similar the categorical distribution $[u_1, u_2, ..., u_k]$ acquired for image I is to the categorical distribution $[q_1, q_2, ..., q_k]$ describing the query image Q. In order to define this score, the probability that the features in Q were generated from the categorical distribution representing image I is considered. Assuming that the features in Q are drawn independently from the categorical distribution of I, one can compute this score such that

$$score(Q, I) = cP(Q \mid u_1, ..., u_k) = u_1^{q_1} u_2^{q_2} \dots u_K^{q_k}$$

Here, c is a positive constant and will be ignored. Note that each q_i is a proportional to an integer count n_i, since $q_i = n_i / N$ for some value N. As long as the computed scores are going to be used to rank different images I with respect to the same query image Q, it would be OK to use the q_i values rather than n_i.

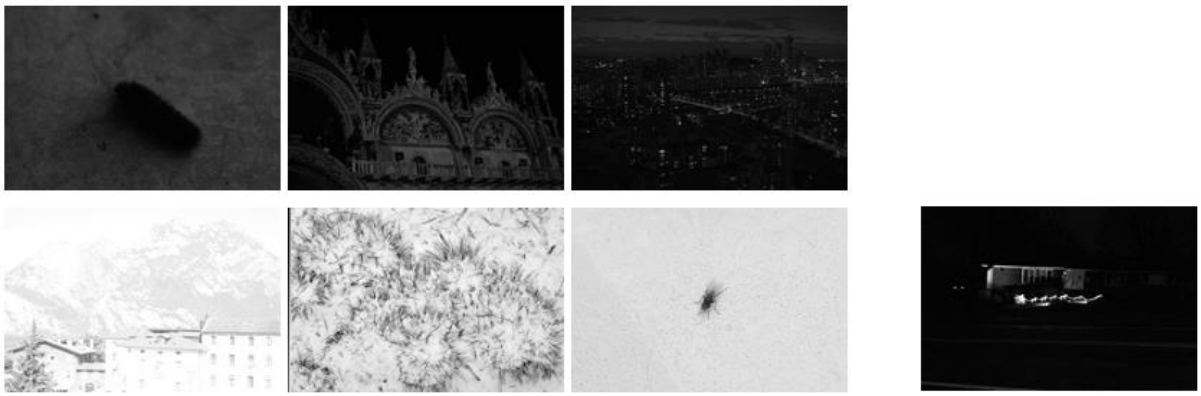


Figure 4 A collection of images given for this problem (a) Gallery images (b) Query image

After defining a score for comparing two images, now it would be possible to compare the query image Q with each of M images $I_1, I_2, ..., I_M$ in the gallery. Clearly, images with the highest scores will be the most similar to the query image Q.

As illustrated in Figure 1, here you will work with a small dataset containing six images $I_1, I_2, ..., I_6$ as the gallery, and a single test image Q to use as the query. Consider four different feature types ($K = 4$) computed by thresholding a pixel's value g. More precisely, each type of feature for a specific pixel can be defined as

$$\text{Feature type} = \begin{cases} 1 & 0 \le g < 64 \\ 2 & 64 \le g < 128 \\ 3 & 128 \le g < 192 \\ 4 & 192 \le g < 256 \end{cases}$$

a. Compute the maximum likelihood estimate of categorical distribution parameters for all images, by finding the comparison scores between Q and each image I. Sort the scores from highest to lowest and determine which images are deemed to be most similar.
b. Based on the image histograms and the categorical parameters computed from them, try to explain why the above scoring/ranking fails so badly.
c. Now consider a Dirichlet prior to impose knowledge about the expected distribution of feature values across the population of all images in the gallery. To apply this method, known as **Dirichlet Smoothing**, start by using MLE to compute the parameters $\{\rho_1, \rho_2, ..., \rho_K\}$ of a categorical distribution of features across all the images in the gallery, in order to get a population distribution. This distribution indicates the overall frequency with which to expect any given feature to appear; a priori. To from a Dirichlet prior to use for MAP estimation, a parameter a_i is applied, such that $a_i = 1 + \eta\rho_i$. Repeat part a. using MAP estimation to determine the categorical distributions representing each image. Play around with different values of η to see how it affects the results. Find the sorted scores, discuss the results and compare them with part a.

Hint: Digital images are in fact matrices of pixels, where each pixel is represented by a numerical value. Therefore, when dealing with grayscale images, you are actually working with a matrix of numbers. As can be seen in Figure 1, the gallery contains three very bright and three very dark images, and the query image is also very dark. Therefore, it is expected to obtain a rank ordering to rank the dark images more highly than the bright ones. The parameter η acts as a variable smoothing parameter. Setting this parameter nearly equal to the number of pixels in the image will cause the prior information be nearly equal importance as the data. If η sets to be very large, the prior will swamp the data, and if η is very low, the data will dominate. $\eta = 0$ would reduce the process to ML estimation.

Good Luck!

Mohsen Ebadpour, Atiyeh *Moghadam, Romina Zakerian*