Knn and Parzen

If we know the forms of the underlying density functions, all we need to calculate are the parameters of the functions. This is called the *parametric* approach to density modeling. However, the common parametric forms rarely fit the densities actually encountered in practice. Specifically, in practice, densities are often multimodal (they have multiple local maximums) and can never be captured with a Gaussian, which is unimodal. Here we focus on *nonparametric* approaches, which require no assumptions about the underlying densities.

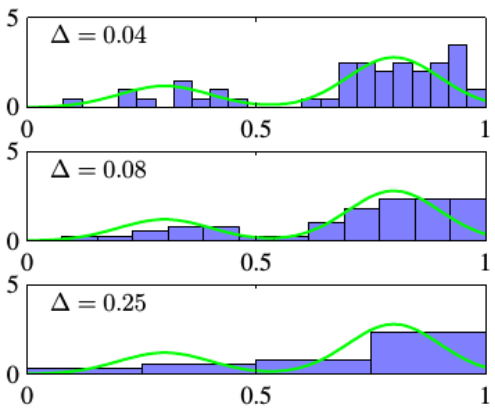
Nonparametric techniques attempt to estimate the underlying density functions from the training data. The idea is: the more data in a region, the larger is the density function. Let's first look at histogram methods and a single continuous variable x. Standard histograms simply partition x into distinct bins of width Δ_i and then count the number n_i of observations of x falling into bin i. In order to turn this count into a normalized probability density, we simply divide by the total number N of observations and by the width Δ_i of the bins to obtain values for each bin given by

$$p_i = \frac{n_i}{N\Delta_i}$$

from which it is easy to see that $\int p(x)\,\mathrm{d}x = 1$.

The following figure shows an example of histogram density estimation. The data is drawn form the distribution, corresponding to the green curve. If Δ is selected to be very small, the resulting model is very spiky. If the Δ is too large, the model is too smooth and fails to capture the bimodal property of the curve.

An illustration of the histogram approach to density estimation, in which a data set of 50 data points is generated from the distribution shown by the green curve. Histogram density estimates, based on (2.241), with a common bin width Δ are shown for various values of Δ.



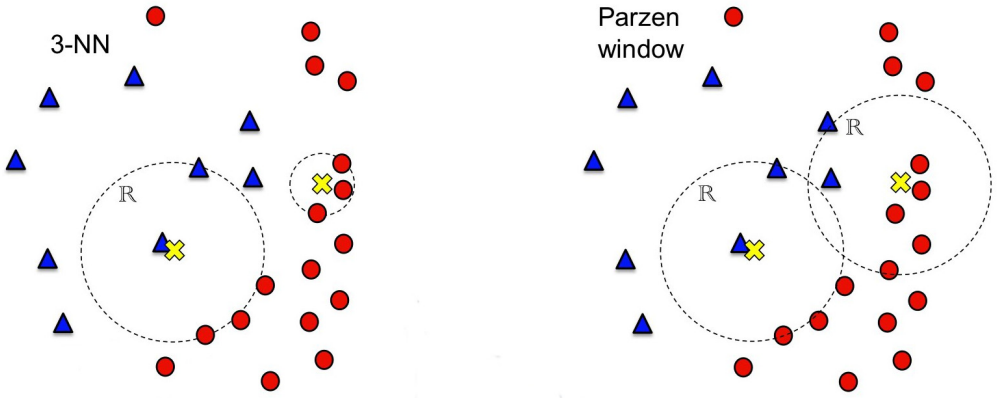
There are two clear advantages of histogram method. First, once the histogram has been calculated, the data can be discarded which is good if the data is very large. (This does not apply to the methods discussed later, knn and parzen). Also, the approach is easily applied if data points are arriving sequentially. In practice, histogram technique is used only for quick visualization of the data. One major drawback is that there are discontinuities at bin edges that have nothing to do with the data.

The histogram approach teaches us two things: First, to estimate the probability density at a particular location, we should consider the data points that lie within some local neighborhood of that point. Note that the concept of locality requires that we assume some form of distance measure. Second, the value of the smoothing parameter should be neither too large nor too small in order to obtain good results.

On this lesson, we will look at kernel density estimation (Parzen windows) and nearest neighbor methods. For these estimates we can derive something similar to the aforementioned equation (the maths, how the equation is derived, can be found in the course book)

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

Either we can fix k_n (number of samples in volume V_n) and determine the value of V_n from the data, which gives rise to the k-nearest-neighbor (knn) technique, or we can fix V_n and determine k_n from the data, giving rise to the kernel approach (Parzen). It can be shown that both the k-nearest-neighbour and the kernel density estimator converge to the true probability density in the limit $n \to \infty$ provided V_n shrinks suitably with n, and k_n grows with n.



Parzen windows:

Basically, the idea of Parzen windows is to choose a fixed value for volume V and determine the corresponding k from the data. As with histogram technique, choosing the appropriate window size (volume) V is difficult. In addition, we need a large number of samples for accurate estimates. However, this estimate is computationally heavy because to classify one point we have to compute a function which potentially depends on all samples. As a classifier, Parzen window estimates have difficulties when the density varies and there is limited amount of data; no single window width is ideal overall. One often used value for V_n is $1/\sqrt{n}$.

A square is a very common window choice for Parzen windowing but in the image below, Duda et al. applied a Gaussian window. The case n=1 (only one data point) tell more about the window function than it tells about the unknown density. By increasing the sample sizes, the results seem a lot better.

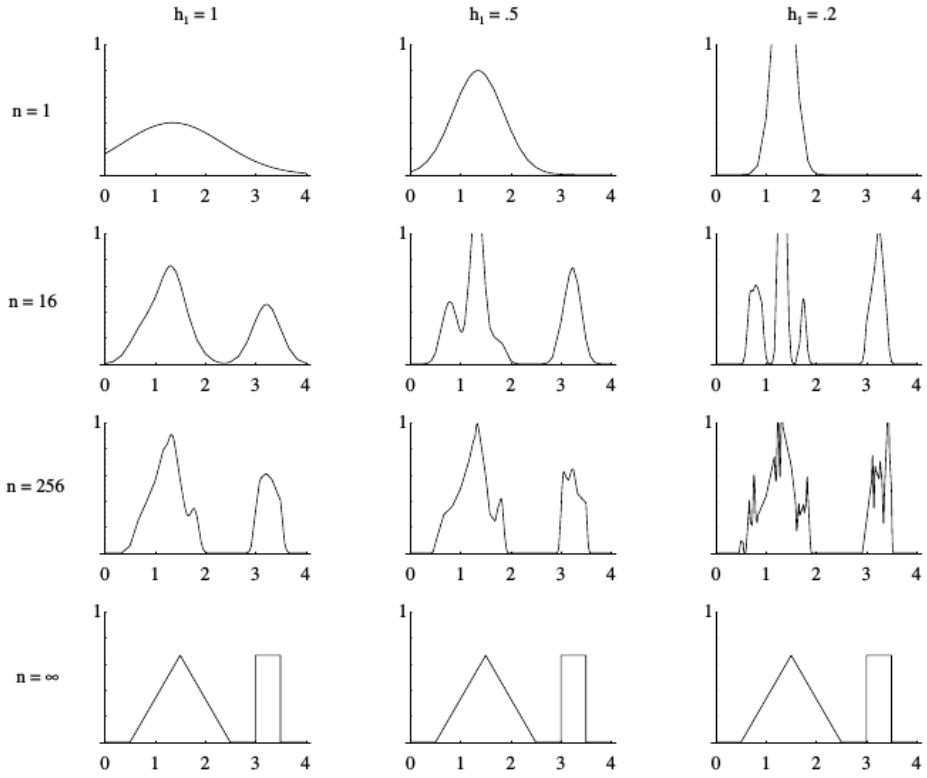


Figure 4.7: Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true generating distribution), regardless of window width h.

Knn:

The idea of knn is the opposite: Choose a fixed value for *k* and determine the corresponding volume *V* from the data. However, straightforward density estimation *p(x)* does not work very well with knn approach. In theory, if infinite number of samples is available, we could construct a series of estimates that converge to the true density using knn estimation. Unfortunately, the number of samples is always limited. Therefore, the resulting density estimate is not really even a density (we will see this shortly at the exercise).

However we shouldn't give up the nearest neighbor approach yet: we can use it straightforwardly for classification. It gives good results if the number of samples is large enough.

In practice, selecting the k should be done carefully. k should be large so that error rate is minimized, because too small k will lead to noisy decision boundaries. On the other hand, k should be small enough so that only nearby samples are included. Too large k will lead to over-smoothed boundaries. A rule of thumb is $k_n=\sqrt{n}$.

The image below shows calculations for similar data as seen with Parzen. This time distributions look very spiky but get better as n increases. Unfortunately, the integral stays large, the fact of which is compensated by the fact that the estimate never plunges to zero just because no samples fall within some arbitrary cell or window.

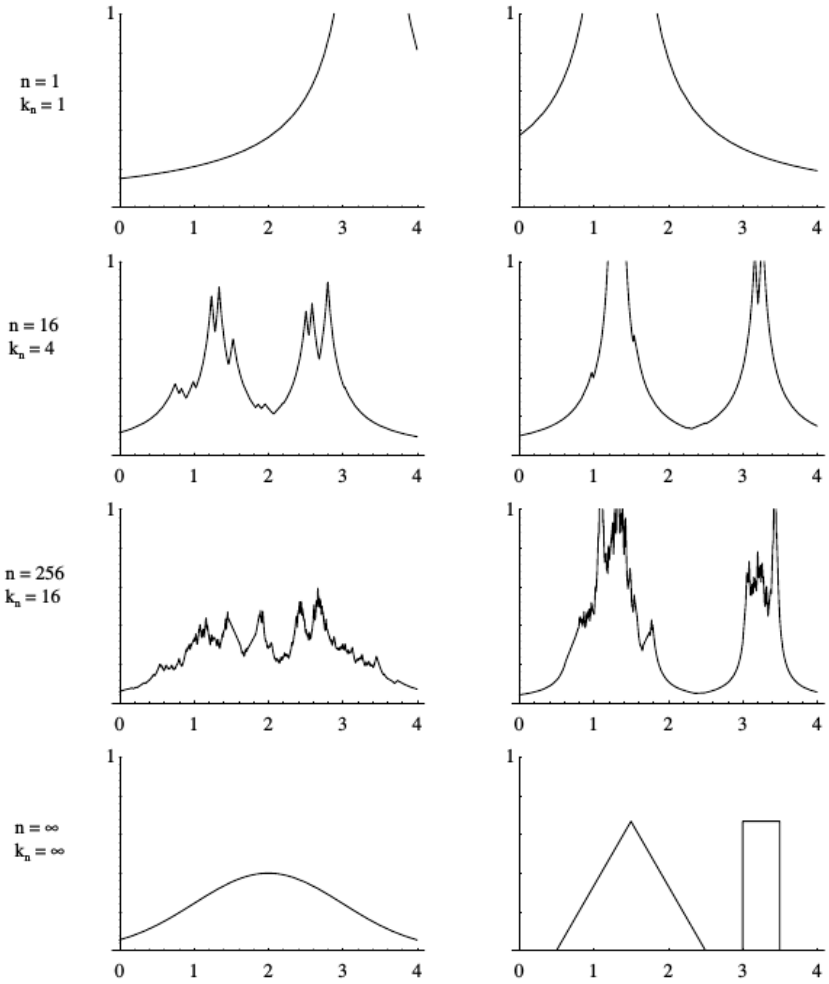
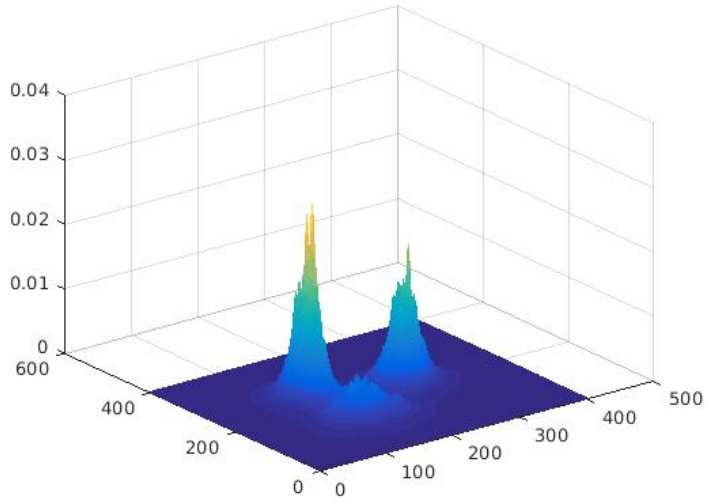


Figure 4.12: Several k-nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite n estimates can be quite "spiky."

Exercise:

In the exercise, we code knn and Parzen estimates. For uniform density estimation we build a grid across the whole data. We then go to each grid point separately and calculate the values that we need for applying the equation for density estimation.

For knn, we begin by going to each grid point and calculating the euclidean distance to each of the data points. After this, we have a vector of distances. We then sort this vector and take the k^{th} distance. This distance, gives us the size of the volume, i.e., we can calculate area of the circle with pi*r^2. Therefore, we have all the information that we need for the equation and we can draw the image below.



For Parzen, we start by defining a small square kernel and moving this square kernel to each of the grid points. We then determine how many of the data points are inside the volume/area/square. Because we used a square kernel, the density estimation looks blockier.