Denisty Estimation in Python

Ben Aldous & Sarah Krstyen

MDCRC 6521 - Final Project

April 27, 2014

Problem

Given a sample, it is sometimes easy to assume the data is independent and identically distributed without actually knowing the distribution from which the data comes. Density estimation techniques attempt to predict the unknown distribution of this data. This is currently a field of high interest in statistics, and much investigational work is yet to be done.

Solution

We created a package, densityestimation 0.1.2, which contains three functions designed to assist in visualizing the unknown density from which a given sample might come. Given a one-dimensional numeric numpy array (or something the function can coerce into such), the functions use the techniques described below to estimate the density and plot the result using matplotlib. (Each technique is a function within the package.)

-hist employs a histogram density estimator. Given a user specified origin o, any real number, and a user specified bandwidth h, any number greater than zero, for each element of a given sample, the function identifies the integer j such that the element falls within the range [o+jh,o+(j+1)h). From this it can be seen that the origin is otherwise known as where the data is centered, and the bandwidth helps determine the intervals surrounding the origin where elements of the given sample fall. The function then predicts the placement of future elements based on the proportion of the sample in each such interval. A plot of the histogram is then returned. Histograms are jagged density estimators, so if users wish to determine the density of their sample based solely on the appearance of a plot, it would be best to use one of the next two functions, which are smooth density estimators.

-kde uses kernel density estimation. This technique attempts to smooth the histogram density estimator by means of a kernel, a function with certain desirable properties. The most commonly used kernels in density estimation are: Gaussian (normal), uniform, triangular, Epanechnikov. These are the four implemented in our function. In general, kernel density estimators are more efficient and more accurate than histogram density estimators. The function allows the user to choose a bandwidth and a kernel, allowing for wide flexibility in application. Once again, this function returns a plot of the kernel density estimate.

-nnde is based on the theory of nearest-neighbor density estimation. Rather than smooth the estimated density via a kernel, it averages distances using the kth neighbor, where k is any real number chosen by the user. In particular, the estimation will be smoother for a larger k. Nearest-neighbor density estimators are good for local density estimation, but it is suggested to use a kernel density estimator when doing global estimation, as to avoid inaccuracy. A plot of the nearest-neighbor density estimate will be returned when using this function.

The math behind the density estimators

-Histogram:

Let the bandwidth, h, be any real number greater than zero. After choosing an origin, the resulting histogram is then

$$\hat{f}(x) = \frac{1}{nh}\sum_{j=1}^{n} \mathbf{I}\{X_j \, is \, in \, the \, same \, bin \, as \, x\}.$$

-Kernel density estimate:

Let the bandwidth, h, be any real number greater than zero. Then consider a "kernel" K

such that $K(x) \geq 0$ and $\int_{-\infty}^{\infty} K(x)dx = 1$. The kernel density estimator is then

$$\hat{f}(x) = \frac{1}{nh} \sum_{j=1}^{n} K\Big(\frac{x - X_j}{h}\Big)$$

for all x.

-Nearest-neighbor density estimate:

Let k be any real number, usually with the property $k << n$. Then define $\rho_1(t) \leq \rho_2(t) \leq \cdots \leq \rho_n(t)$ as the ordered distances t from the sample data observations $X_1, \dots X_n$. The nearest-neighbor density estimator is then

$$\hat{f}(x) = \frac{k-1}{2n\rho_k(x)}.$$

Future Objectives

There are a variety of improvements that could be made to these functions. Some examples include real-time selection of tuning parameters, suggestion of optimal values for these parameters, and expansion to multidimensional data.

Where to find this package

Pypi: https://pypi.python.org/pypi/densityestimation/0.1.2

Github: https://github.com/skrstyen/densityestimation