Kernel Probability Density Function

writeLaTeX

October 19, 2016

Kernel smoothers: create a continuous function to represent the similarities between a neighborhood of a
Bandwith: a scaling factor used to define the width of probability mass around a point.

1 Kernels and Kernel Distributions

1.1 Kernel Smoothing Function

Smoothing functions: represent points via a set of adjusting weights and base points:

$$\hat{f}(x) = \sum_{i=1}^{n} W_i(x) y_i$$

Kernel smoothing function does this in terms of a scaling factor (bandwith), and makes the weights in terms of a continuous function with total density = 1.

$$W_{hi}(x) = \frac{K(\frac{|x_0 - x_i|}{\lambda})}{\sum_{i=1}^{n} K(\frac{x_0 - x_i}{\lambda})}$$

Kernel smoother thus is:

$$\hat{f}(x) = \sum_{i=1}^{n} W_{hi}(x) y_i$$

1.2 Kernel Probability Distribution Function

The kernel distribution function is as follows, with $N(x_0)$ being the neighborhood of total points in the prob distribution, λ being the bandwidth of the function, indicating the scale factor of the distribution.

$$\hat{f}(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i)$$

The Kernel function can be one of many different distributions:

$$Epanechkov : D(t) = \frac{3}{4}(1 - t^2) \text{ such that } |t| < 1$$

$$Gaussian : D(t) = \frac{1}{\sqrt{2\pi}}(e^{-\frac{1}{2}t^2})$$

$$Tri - Cube : D(t) = (1 - t^3)^3 \text{ such that } |t| < 1$$

These are only a couple potential equations of base kernels. Other examples include triangle and box. These distributions are implemented such that:

$$K_\lambda = D(\frac{x_0 - x_i}{\lambda})$$

1.3 Bandwidth

The bandwidth is a scaling factor used to define the width of probability mass around a point. Increasing bandwidth increases bias and decreases variance.

2 Pseudocode of Kernel Probabilities

Algorithm 1 kernelprob: Generate a probability estimate based on a distribution with a kernel base function.

Input: x_0, $X = x_0, x_1, ..., x_n$, bandwidth λ, kernel function K_λ
Output: kernel based probability of $x_0 in X$

```
procedure KERNELPROB(𝒳, λ, 𝒦_λ)
    kernsum = 0;
    for i in range(0, n) do                  ▷ run kernel func on each point
        kernsum += K(|x_i − x_0| / λ)        ▷ get kernel dist between 2 points
    return 1/(nλ) × kernsum
```

3 Pseudocode of Different Kernel Base Distributions

Algorithm 2 epanechkov: Epanechkov distribution used to calculate distance between 2 points.

Input: Value to verify t
Output: Distance d

```
procedure EPANECHKOV(t)
    if |t| < 1 then
        return 3/4 × (1 − t^2)
    else
        return 0
```

Algorithm 3 gaussian: Gaussian distribution used to calculate distance between 2 points.

Input: Value to verify t
Output: Distance d

```
procedure GAUSSIAN(t)
    return 1/√(2π) (e^(−1/2 t^2))
```

Algorithm 4 tricube: Tri-cube distribution used to calculate distance between 2 points.

Input: Value to verify t
Output: Distance d

```
procedure TRICUBE(t)
    if |t| < 1 then
        return (1 − t^3)^3
    else
        return 0
```