Jump Regression Analysis

Peihua Qiu

School of Statistics, University of Minnesota

Keywords

Curve and surface estimation; Local smoothing; Nonparametric regression.

Nonparametric regression analysis provides statistical tools for estimating regression curves or surfaces from noisy data. Conventional nonparametric regression procedures, however, are only appropriate for estimating continuous regression functions. When the underlying regression function has jumps, functions estimated by the conventional procedures are not statistically consistent at the jump positions. Recently, regression analysis for estimating jump regression functions is under rapid development [1], which is briefly introduced here.

1-D Jump Regression Analysis In one-dimensional (1-D) cases, the *jump regression analysis (JRA)* model has the form

$$y_i = f(x_i) + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots, n, \tag{1}$$

where $\{y_i,\ i = 1, 2, \ldots, n\}$ are observations of the response variable y at design points $\{x_i,\ i = 1, 2, \ldots, n\}$, f is an unknown regression function, and $\{\varepsilon_i,\ i = 1, 2, \ldots, n\}$ are random errors. For simplicity, we assume that the design interval is $[0, 1]$. In (1), f is assumed to have the expression

$$f(x) = g(x) + \sum_{j=1}^{p} d_j I(x > s_j), \text{ for } x \in [0, 1], \tag{2}$$

where g is a continuous function in the entire design interval, p is the number of jump points, $\{s_j,\ j = 1, 2, \ldots, p\}$ are the jump positions, and $\{d_j,\ j = 1, 2, \ldots, p\}$ are the corresponding jump magnitudes. If $p = 0$, then f is continuous in the entire design interval. In (2), the function g is called the *continuity part* of f, and the summation $\sum_{j=1}^{p} d_j I(x > s_j)$ is called the *jump part* of f. The major goal of JRA is to estimate g, p, $\{s_j,\ j = 1, 2, \ldots, p\}$ and $\{d_j,\ j = 1, 2, \ldots, p\}$ from the observed data $\{(x_i, y_i),\ i = 1, 2, \ldots, n\}$.

A natural jump detection criterion is

$$M_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} Y_i K_1 \left(\frac{x_i - x}{h_n} \right) - \frac{1}{nh_n} \sum_{i=1}^{n} Y_i K_2 \left(\frac{x_i - x}{h_n} \right), \tag{3}$$

where h_n is a positive bandwidth parameter, K_1 and K_2 are two density kernel functions with supports $[0, 1]$ and $[-1, 0)$, respectively. Obviously, $M_n(x)$ is a difference of two *one-sided* kernel estimators. The first kernel estimator in equation (3) is right-sided; it is a weighted average of the observations in the right-sided neighborhood

$[x, x + h_n]$. Similarly, the second kernel estimator in (3) is left-sided; it is a weighted average of the observations in the left-sided neighborhood $[x - h_n, x)$. Intuitively, $M_n(x)$ would be large if x is a jump point, and small otherwise. So, if we know that there is only one jump point (i.e., $p = 1$) in the design interval $[0, 1]$, then the jump point s_1 can be estimated by the maximizer of $|M_n(x)|$ over $x \in [h_n, 1 - h_n]$, denoted as $\widehat{s}_1$, and d_1 can be estimated by $M_n(\widehat{s}_1)$. In cases when $p > 1$ and p is known, the jump positions $\{s_j, j = 1, 2, \ldots, p\}$ and the jump magnitudes $\{d_j, j = 1, 2, \ldots, p\}$ can be estimated in a similar way. Let s_j^* be the maximizer of $|M_n(x)|$ over the range

$$x \in [h_n, 1 - h_n] \Bigg\backslash \left(\bigcup_{\ell=1}^{j-1} [s_\ell^* - h_n, s_\ell^* + h_n] \right)$$

for $j = 1, 2, \ldots, p$. The order statistics of $\{s_j^*, j = 1, 2, \ldots, p\}$ are denoted by $s_{(1)}^* \leq s_{(2)}^* \leq \ldots \leq s_{(p)}^*$. Then we define $\widehat{s}_j = s_{(j)}^*$ and $\widehat{d}_j = M_n\left(s_{(j)}^*\right)$, for $j = 1, 2, \ldots, p$.

When the number of jumps p is unknown, people often use a threshold value u_n and flag all design points in $\{x_i : |M_n(x_i)| \geq u_n\}$ as candidate jumps. Then, certain deceptive candidate jumps need to be deleted using a modification procedure (cf., [1], Section 3.3.3). An alternative approach is to perform a series of hypothesis tests for $H_0 : p = j$ versus $H_1 : p > j$, for $j = 0, 2, \ldots$, until the first "fail to reject H_0" (cf., [1], Section 3.3.2).

The jump detection criterion $M_n(x)$ in (3) can be regarded as an estimator of the first-order derivative $f'(x)$ of f. It is based on local constant kernel estimation of the one-sided limits $f_-(x)$ and $f_+(x)$. Alternative jump detection criteria, based on other estimators of $f'(x)$ or based on estimators of both the first-order and the second-order derivatives of f, also exist. See [2] for a recent discussion on this topic and on estimation of the continuity part g after jump points being detected.

2-D Jump Regression Analysis In two-dimensional (2-D) cases, the regression model becomes

$$Z_i = f(x_i, y_i) + \varepsilon_i, \; i = 1, 2, \ldots, n, \tag{4}$$

where n is the sample size, $\{(x_i, y_i), i = 1, 2, \ldots, n\}$ are the design points in the design space, f is the 2-D regression function, $\{Z_i, i = 1, 2, \ldots, n\}$ are n observations of the response variable Z, and $\{\varepsilon_i, i = 1, 2, \ldots, n\}$ are random errors. For simplicity, we assume that the design space is the unit square $[0, 1] \times [0, 1]$. In such cases, jump positions of f are curves in the design space, which are called the *jump location curves (JLCs)*. Because jumps are an important structure of f, 2-D JRA is mainly for estimating JLCs and for estimating f with the jumps at the JLCs preserved, which are referred to as *jump detection* and *jump-preserving surface estimation*, respectively, in the literature (cf., [1], Chapters 4 and 5).

Early 2-D jump detection methods assume that the number of JLCs is known; they are usually the generalized versions of their 1-D counterparts, based on estimation of certain first-order directional derivatives of f. In [3], Qiu and Yandell describe the JLCs as *a pointset* in the design space, and suggest estimating the JLCs by another pointset in the same design space. Since points in a pointset need not form curves, the connection among the points of a pointset is much more flexible than the connection among the points on curves, which makes detection of arbitrary JLCs possible. For instance, Qiu and Yandell [3] suggest flagging a design point as a candidate jump point if the estimated gradient magnitude of f at this point is larger than a threshold. In that paper, we also suggest two modification procedures to remove certain deceptive jump candidates. Various other jump detection procedures, based on estimation of the first-order derivatives of f, or the second-order derivatives of f, or both, have been proposed in the literature. See [4] for a recent discussion on this topic.

In the literature, there are two types of jump-preserving surface estimation methods. Methods of the first type usually estimate the surface after jumps are detected [5]. Around the detected jumps, the surface estimator at a given point is often defined by a weighted average of the observations whose design points are located on the same side of the estimated JLC as the given point in a neighborhood of the point. Potential jumps can thus be preserved in the estimated surface. The second type of methods estimates the surface without detecting the jumps explicitly, using the so-called adaptive local smoothing. Adaptive local smoothing procedures obtain certain evidence of jumps from the observed data directly, and adapt to such evidence properly to preserve jumps while removing noise [6].

2-D Jump Regression Analysis and Image Processing Model (4) can be used in cases with arbitrary 2-D design points. In certain applications (e.g., image processing), design points are regularly spaced in the 2-D design space. In such cases, a simpler model would be

$$Z_{ij} = f(x_i, y_j) + \varepsilon_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2, \tag{5}$$

where $\{Z_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots,\ n_2\}$ are observations of the response variable Z observed at design points $\{(x_i, y_j),\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2\}$, and $\{\varepsilon_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2\}$ are random errors.

Model (5) is ideal for describing a monochrome digital image. In the setup of a monochrome digital image, x_i denotes the ith row of pixels, y_j denotes the jth column of pixels, f is the image intensity function, $f(x_i, y_j)$ is the true image intensity level at the (i, j)th pixel, ε_{ij} denotes the noise at the (i, j)th pixel, and Z_{ij} is the observed image intensity level at the (i, j)th pixel. The image intensity function f often has jumps at the outlines of objects. Therefore, 2-D JRA can provide a powerful statistical tool for image processing. In the image processing literature, positions at which f has jumps are called *step edges*, and positions at which the first-order derivatives of f have jumps are called *roof edges* (cf., [1], Chapter 6). Edge detection and edge-preserving image restoration are two major problems in image processing, which are essentially the same problems as jump detection and jump-preserving surface estimation in 2-D JRA. See [7] for a recent discussion about the connections and differences between the two areas.

References

[1] Qiu, P. (2005), *Image Processing and Jump Regression Analysis*, New York: John Wiley & Sons.

[2] Joo, J., and Qiu, P. (2009), “Jump detection in a regression curve and its derivative,” *Technometrics*, **51**, 289-305.

[3] Qiu, P., and Yandell, B. (1997), “Jump detection in regression surfaces,” *Journal of Computational and Graphical Statistics*, **6**, 332–354.

[4] Sun, J., and Qiu, P. (2007), “Jump detection in regression surfaces using both first-order and second-order derivatives,” *Journal of Computational and Graphical Statistics*, **16**, 289–311.

[5] Qiu, P. (1998), “Discontinuous regression surfaces fitting,” *The Annals of Statistics*, **26**, 2218–2245.

[6] Gijbels, I., Lambert, A., and Qiu, P. (2006), “Edge-preserving image denoising and estimation of discontinuous surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 1075–1087.

[7] Qiu, P. (2007), “Jump surface estimation, edge detection, and image restoration,” *Journal of the American Statistical Association*, **102**, 745–756.