Title

stata.com

npregress intro — Introduction to nonparametric kernel regression

Description Remarks References Also see

Description

Nonparametric regression allows us to model the mean of an outcome given the covariates when we are uncertain about its functional form. Nonparametric kernel-based estimators rely on an optimal bandwidth parameter that trades off bias and variance. Although nonparametric regression is a way to obtain estimates that are robust to functional form misspecification, this robustness comes at a cost; you need many observations and more time to compute the estimates.

`npregress` implements two nonparametric estimators: local linear and local constant. This entry introduces the intuition behind the nonparametric regression estimators implemented in `npregress`. If you are familiar with these methods, you may want to skip to [R] **npregress**.

Remarks

Nonparametric regression is used when we are uncertain about the functional form of the mean of the outcome given the covariates. For example, when we estimate a linear regression, we assume that the functional form for the mean of the outcome is a linear combination of the covariates or a linear combination of the covariates and their interactions. Both parametric (linear) regression and nonparametric regression provide an estimate of the mean for the different values of the covariates. Consider the simulated data in figure 1. The mean of the outcome for all values of `x` is overlayed on these points.

Figure 1

Because the mean of the data in figure 1 is not linear in `x`, using linear regression will not give us a correct picture about the effect of covariate `x` on the outcome. For example, if we use linear regression on these data, we obtain the plot shown in figure 2.

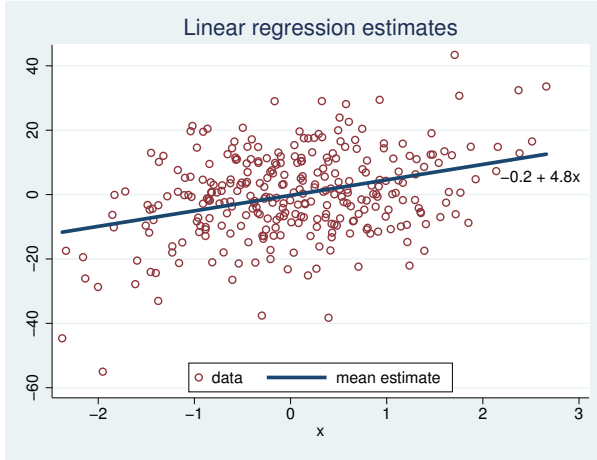

Figure 2

The change in the predicted outcome when `x` changes is positive and constant, yet the true mean is nonlinear. If the assumption about the functional form of the mean is incorrect, the estimates we obtain are inconsistent. If we instead fit the model using `npregress` and graph the estimates, we obtain figure 3.

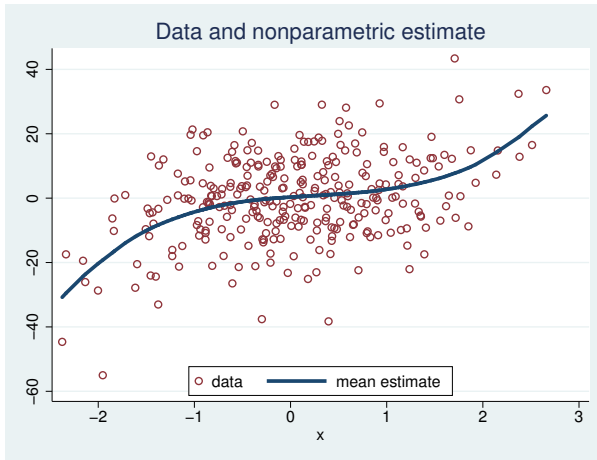

Figure 3

`npregress` gives us the correct relationship between the outcome and the covariates. The nonparametric regression estimates are consistent as long as the true function is sufficiently smooth. If the linear regression assumptions are true, nonparametric regression is still consistent but less efficient.

Although nonparametric regression is a way to obtain estimates that are robust to functional form misspecification, this robustness comes at a cost. You need many observations and more time to compute the estimates. The cost increases with the number of covariates; this is referred to as the curse of dimensionality.

Below, we show how we obtained the estimate of the mean for all values of `x` in our data in figure 3. We build our intuition graphically, using the definition of a mean evaluated at different values of covariate `x`.

Suppose covariate x is discrete. In this case, a consistent estimator of the mean of outcome y given that $\mathtt{x} = a$ is the average of the values of y for which x is equal to a given value a. For instance, the sample average of the yearly income for married individuals is a consistent estimator for the population mean yearly income for married individuals.

Now, consider estimating the mean of y given that $\mathtt{x} = a$ when x is continuous and a is a value observed for x. Because x is continuous, the probability of any observed value being exactly equal to a is 0. Therefore, we cannot compute an average for the values of y for which x is equal to a given value a. We use the average of y for the observations in which x is close to a to estimate the mean of y given that $\mathtt{x} = a$. Specifically, we use the observations for which $|\mathtt{x} - a| < h$, where h is small. The parameter h is called a bandwidth. In nonparametric regression, a bandwidth determines the amount of information we use to estimate the conditional mean at each point a.

For the simulated data in our example, we choose $h = 0.25$ and $a = -0.19$. The vertical lines in figure 4 delimit the values of x around a for which we are computing the mean of y. The light blue square is our estimate of the conditional mean using the observations between the vertical lines.

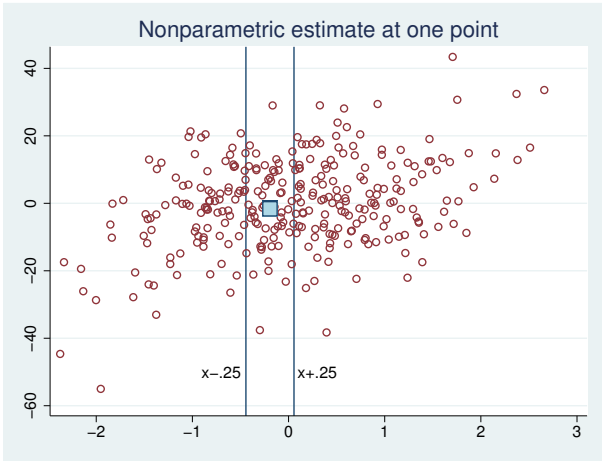

Figure 4

Repeating this estimation when $a = 2.66$ produces figure 5.

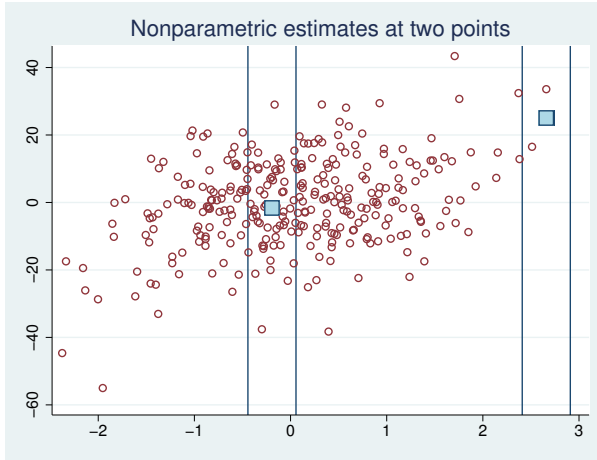

Figure 5

Doing this estimation for each point in our data produces a nonparametric estimate of the mean for a given value of the covariates (see figure 6).

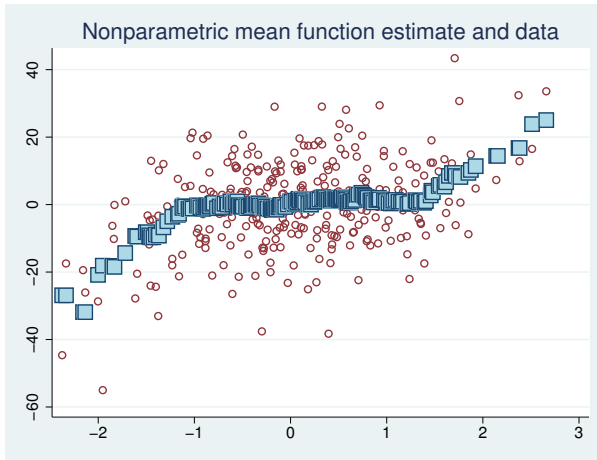

Figure 6

The plotted blue squares in figure 6 form what is known as the conditional mean function. Because these are simulated data, we can compare our estimate with the true conditional mean function, a comparison we show in figure 7.

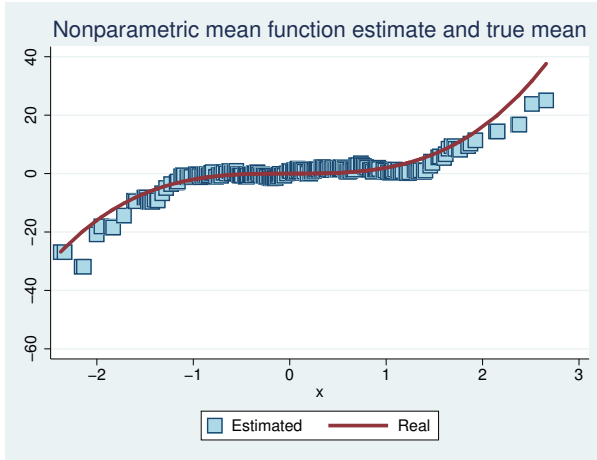

Figure 7

We see that the estimate is a bit less smooth than the true function. The size of the bandwidth h determines the shape and smoothness of the estimated conditional mean function, because the bandwidth defines how many observations around each point are used. For example, if h is arbitrarily large—say, $h = 300$—we get figure 8.

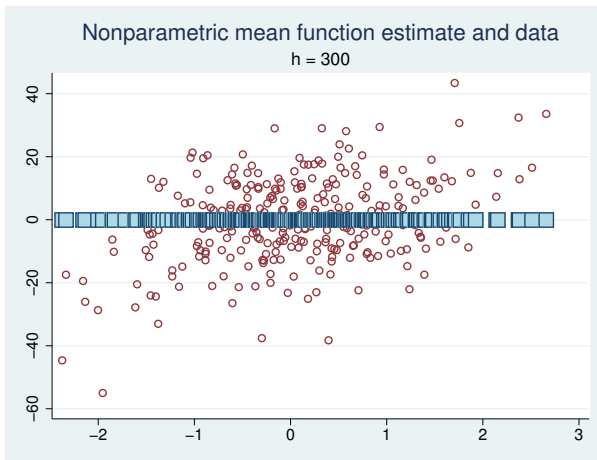

Figure 8

In this case, all observations are used to estimate the conditional mean at each point and the estimate is therefore a constant. On the other hand, a too-small bandwidth produces a jagged function with high variability, as illustrated in figure 9.

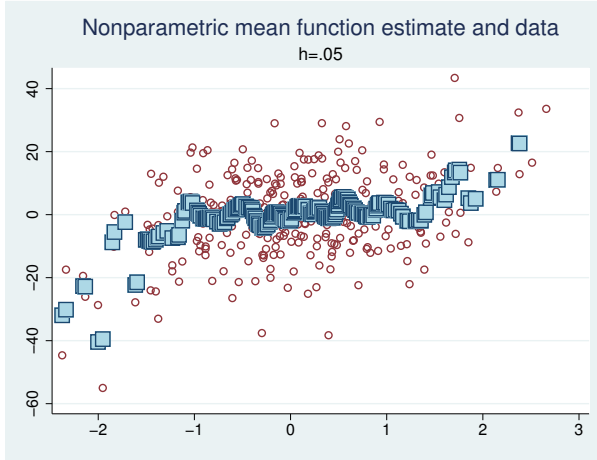

Figure 9

The optimal bandwidth is somewhere in between. A too-large bandwidth includes too many observations, so the estimate is biased but it has a low variance. A too-small bandwidth includes too few observations, so the estimate has little bias but the variance is large. In other words, the optimal bandwidth trades off bias and variance. In the case of `npregress`, the bandwidth is chosen to minimize the cost of this trade-off using either cross-validation, as suggested by Li and Racine (2004), or an improved Akaike information criterion proposed by Hurvich, Simonoff, and Tsai (1998).

How we average the observations around a point is also important. In the examples above, we gave the same weight to each observation for which $|\mathtt{x} - a| < h$. However, we might weight each observation differently. The weights observations receive are determined by functions called kernels. We could have used any of the weights in [R] **kdensity**. For a nice introduction to kernel weighting, see Silverman (1986).

The estimator described above uses only nearby observations and is thus a local estimator. It uses a sample average, which is a regression on a constant, and is thus a locally constant estimator. For these reasons, the estimator described above is known as local-constant regression.

The generalization that uses the prediction from a local-linear regression on covariates is known as local-linear regression. Local-linear regression estimates the derivative of the conditional mean function in addition to the function itself. Understanding how the conditional mean changes when covariates change is sometimes the research question of interest, for example, how income changes for different levels of taxes. Local-linear regression provides an estimate for these changes for continuous and discrete variables.

See Fan and Gijbels (1996) and Li and Racine (2007) for detailed introductions to the estimators implemented in `npregress`.

References

Fan, J., and I. Gijbels. 1996. *Local Polynomial Modelling and Its Applications*. London: Chapman & Hall.

Hurvich, C. M., J. S. Simonoff, and C.-L. Tsai. 1998. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society, Series B* 60: 271–293.

Li, Q., and J. S. Racine. 2004. Cross-validated local linear nonparametric regression. *Statistica Sinica* 14: 485–512.

——. 2007. *Nonparametric Econometrics: Theory and Practice*. Princeton, NJ: Princeton University Press.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Also see

[R] **npregress** — Nonparametric regression

[R] **lpoly** — Kernel-weighted local polynomial smoothing

[R] **kdensity** — Univariate kernel density estimation

[R] **regress** — Linear regression