Published in final edited form as:

J Comput Graph Stat. 2019 ; 28(2): 466–470. doi:10.1080/10618600.2018.1537927.

Fast Bootstrap Confidence Intervals for Continuous Threshold Linear Regression

Youyi Fong

Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research Center, Department of Biostatistics, University of Washington, Seattle, WA 98109

Abstract

Continuous threshold regression is a common type of nonlinear regression that is attractive to many practitioners for its easy interpretability. More widespread adoption of threshold regression faces two challenges: (i) the computational complexity of fitting threshold regression models and (ii) obtaining correct coverage of confidence intervals under model misspecification. Both challenges result from the non-smooth and non-convex nature of the threshold regression model likelihood function. In this paper we first show that these two issues together make the ideal approach for making model-robust inference in continuous threshold linear regression an impractical one. The need for a faster way of fitting continuous threshold linear models motivated us to develop a fast grid search method. The new method, based on the simple yet powerful dynamic programming principle, improves the performance by several orders of magnitude.

Keywords

segmented regression; change point; model-robust

1. Introduction

Continuous threshold regression models are a type of nonlinear regression model in which the slope of a predictor of interest is allowed to change across an unknown threshold, but no jumps occur at the threshold. This type of model arises naturally from many applied areas. For example, Sprent (1961) reported several examples from biology. More recently, continuous threshold regression models have been applied in environmental studies (Hong et al., 2016) and human immunology (Permar et al., 2015), among many others.

An important appeal of continuous threshold regression models lies in their simplicity. However, with this simplicity comes a price: it has long been recognized that they “can only be a reasonable approximation, adequate for many purposes, but by no means a complete description of what is taking place (Sprent, 1961).” Thus, to draw valid conclusions in data analysis, inferential procedures that are robust against model-misspecification are needed. Recent advances (Hansen, 2017; Fong et al., 2017b) showed that the maximum likelihood estimator (MLE) converges at the regular $\sqrt{n}$ rate even when the model is misspecified and provided asymptotic variance formula. However, there is a major difficulty in applying these asymptotic results to construct analytical confidence intervals: the asymptotic variance of the

MLE depends on the true mean function, which can be estimated nonparametrically, but at a much lower rate than $\sqrt{n}$.

Nonparametric bootstrap methods bypass the need to know the true model in order to make inference about the threshold model parameters. Bootstrap methods are computation-ally intensive, and this is particularly true for threshold models because their log likelihood functions are non-convex and non-smooth. Existing methods for finding the MLE either employ a grid search strategy, which fits a series of submodels with fixed thresholds and chooses one with the maximum likelihood, or approximate the log likelihood function using a smooth transition model (e.g. Pastor-Barriuso et al., 2003) or a first order expansion (Muggeo, 2003). The approximation approach is faster than the grid search approach and generally yields acceptable parameter estimates (Fong et al., 2017a). However, in the next section we show that bootstrap confidence intervals constructed using the smooth approximation approach may substantially under-cover when the sample size is small to medium.

2. Coverage probabilities of analytical and bootstrap confidence intervals

In this section we compare the performance of analytical (Wald) and bootstrap confidence intervals for continuous threshold linear models. Four types of analytical confidence intervals are compared: (i) *model* uses variance estimates based on the asymptotic variance formula assuming the model is correctly specified; (ii) *sandwich* also assumes correct mean model specification, but allows heteroscedastic error distribution; (iii) *robust* uses an asymptotic variance formula that assumes model misspecification, derived in the Supplementary Materials Section A. The variance estimate requires an estimation of the true mean function, m_0 , which we approximate nonparametrically with natural cubic splines. (iv) *robust** is like robust except that we estimate the true mean using knowledge of the data-generating model. Thus, *robust** is not a practical method but rather a reference point for showing the ideal performance of model-robust confidence intervals.

In addition to the four analytical confidence intervals, we compare two symmetric percentile bootstrap confidence intervals, defined as $\hat{\theta}_n \pm q^*$ where q^* is the $(1 - \alpha)$ quantile of $|\hat{\theta}^* - \hat{\theta}_n|$, where $\hat{\theta}_n$ and $\hat{\theta}^*$ are the MLE of the dataset and a bootstrap sample, respectively (Hansen, 2017). As pointed out in Section 13.4 of Efron and Tibshirani (1993), neither the percentile bootstrap confidence intervals nor the pivotal, inverse percentile confidence intervals work well when the sampling distributions are skewed. Skewed sampling distributions under small-to-medium sample sizes appear to be a particularly serious issue for continuous threshold regression models. The symmetrical percentile bootstrap confidence intervals work better than both the percentile and inverse percentile bootstrap methods in practice in our experience. The two symmetrical percentile methods we study differ in the methods used to find the MLE. The *bootstrap* method uses the smooth approximation strategy to search the parameter space, while the *bootstrap** method employs the grid search strategy. The asterisk in the label serves as a reminder that just as the *robust** method is not a practical method, the *bootstrap** method is also not a practical method, unless it can be accelerated, due to its heavy computational burden.

Table 1 compares the performance under model misspecification with data simulated from a quadratic model of x : $Y = -1 + 0.34z - x + 0.3x^2 + \varepsilon$, where ε is a mean 0 normal random variable with standard deviation 0.3. The limit of $\hat{\theta}_n$, θ_0 , can be approximated either by numerical integration or by Monte Carlo experiments with very large n . We take the latter approach and find θ_0 to be $(\beta_z = 0.34, \beta_x = 0.90, \beta_{(x-e)_+} = 1.94, e = 4.7)$. The simulation results show that both the *model*-based and the *sandwich* variance estimator-based confidence intervals undercover. The *model-robust* confidence intervals are constructed by approximating m_0 with a natural cubic spline model of three degrees of freedom for x .

They show substantial improvement over *model* and *sandwich*, but compared to *robust**, which uses a quadratic model for x to estimate m_0 , there is much room for improvement. The performance of smooth approximation-based *bootstrap** is on par with grid search-based *robust**, but the computationally more practical *bootstrap* method has a deteriorated performance that is closer to the *robust* method. We also compare the performance of different methods under correct model specification. The results (Table B.1) show that the *robust* method could be overly conservative, while the bootstrap methods perform well.

Taken together, these results demonstrate that grid search-based bootstrap confidence intervals perform well for continuous threshold linear models under both model misspecification and correct model specification. The question is now how to construct such confidence intervals more efficiently, because performing grid search within a bootstrap procedure takes prohibitively long for most day-to-day uses.

3. Fast bootstrap confidence intervals for continuous threshold linear models

The grid search strategy for finding the MLE of continuous threshold regression models is an intuitive one. By conditioning on threshold parameter values, it reduces the non-smooth, non-convex problem to a series of smooth and convex subproblems. Specifically for continuous threshold linear models, the MLE of each submodel is $\hat{\beta}_e = (X_e^T X_e)^{-1} X_e^T Y$, where Y is a vector of length n , $X_e = [\mathbf{1}, Z, \underline{x}, (\underline{x} - e)_+]$ is the $n \times p$ design matrix of the submodel, $\underline{x}$ is a vector of length n , and e is the threshold value conditioned on. The log likelihood of the fitted submodel is inversely proportional to the residual sum of squares, which can be written as $Y^T (I - H_e) Y = Y^T Y - Y^T H_e Y$ where $H_e = X_e (X_e^T X_e)^{-1} X_e^T$ is the hat matrix. Thus it is sufficient to compare $Y^T H_e Y$ for a set of e , $\{e_1, \dots, e_M\}$, which is often chosen to be the observed x 's with the most extreme values, say 10%, trimmed off for more stable finite sample performance.

While computing $Y^T H_e Y$ does not take much time with modern computing power for many datasets encountered in biomedical studies, doing so for every e in a grid within a bootstrap procedure becomes an issue because it requires $O(n \times B)$ repetitions, where B is the number of bootstrap replicates, typically on the scale of 10^3 . For example, the grid search bootstrap method takes more than 1 minute to run for datasets of sample size 250 and more than 20 minutes to run for datasets of sample size 500, on a state-of-the-art machine

(Table 2). One way to speed up the implementation is to distribute the work to multiple processors. Since each bootstrap sample can be run independently, bootstrap methods are easily parallelizable in principle. However, since most computers have two to four computing cores, parallelization is not enough to sufficiently speed up the method for most practitioners to use continuous threshold linear models on a routine basis.

To find a way to accelerate the computation of $Y^T H_e Y$ for a grid of e 's, we recall a well-known fact in linear regression model diagnostics: to determine changes in estimated regression coefficients with the i^{th} observation deleted, there is no need to re-fit the model with the leave-one-out dataset (e.g. Theorem 10.1 Seber and Lee, 2003). Rather, because of the special relationship between the design matrix of the full dataset, X , and that of the leave-one-out dataset, $X_{(i)}$, some functions of the leave-one-out design matrix can be computed much cheaply. For example, $X_{(i)}^T X_{(i)} = X^T X - \underline{x}_i^T \underline{x}_i$, where $\underline{x}_i$ is the covariate vector of the i^{th} sample. Furthermore, applying a matrix inverse formula, it can be shown that $(X_{(i)}^T X_{(i)})^{-1} = A + (A \underline{x}_i \underline{x}_i^T A) / (1 - h_i)$, where $A = (X^T X)^{-1}$ and h_i is the i^{th} diagonal element of the hat matrix $X(X^T X)^{-1} X^T$.

For our problem, let us consider two neighboring points in a grid of e 's, e_t and e_{t+1} . Denote $\delta = e_{t+1} - e_t > 0$. The design matrices corresponding to e_t and e_{t+1} are $X_{e_t} = [1, Z, \underline{x}, (\underline{x} - e_t)_+]$ and $X_{e_{t+1}} = [1, Z, \underline{x}, (\underline{x} - e_{t+1})_+]$. Suppose we have ordered the samples such that the vector $\underline{x}$ is sorted from small to large. Then X_{e_t} and $X_{e_{t+1}}$ are related by a simple formula:

$$X_{e_t} - X_{e_{t+1}} = [0]_{n \times (p-1)}, [\underline{0}_{n-k} \ \underline{\delta}_k]^T \equiv \Delta_{1,2}$$
, where $[0]_{n \times (p-1)}$ is a n by $p-1$ matrix of 0's, $\underline{0}_{n-k}$ and $\underline{\delta}_k$ are vectors of 0's and δ 's of the specified length, and e_{t+1} is the k^{th} largest observation among the ordered $\underline{x}$ vector. It follows that we can express $X_{e_{t+1}}^T X_{e_{t+1}}$ in terms of $X_{e_t}^T X_{e_t}$:

$$X_{e_{t+1}}^T X_{e_{t+1}} = X_{e_t}^T X_{e_t} - \Delta_{1,2}^T X_{e_t} - X_{e_t}^T \Delta_{1,2} + \Delta_{1,2}^T \Delta_{1,2} \quad (1)$$

This formula is too complicated for $(X_{e_{t+1}}^T X_{e_{t+1}})^{-1}$ to admit a simple recursive relationship with $(X_{e_t}^T X_{e_t})^{-1}$ as in the case of linear regression model diagnostics. But, if we store $X_{e_t}^T X_{e_t}, X_{e_{t+1}}^T X_{e_{t+1}}$ can be computed quickly because the remaining terms in (1) all involve $\Delta_{1,2}$, which simplifies the computation considerably (Section B.2). Similarly, we can compute $X_{e_{t+1}}^T Y$ more quickly if we store $X_{e_t}^T Y$ and use the relationship:

$$X_{e_{t+1}}^T Y = X_{e_t}^T Y - \Delta_{1,2}^T Y, \quad (2)$$

where $\Delta_{1,2}^T Y$ is given in Section B.2. Once we have computed $X_{e_{t+1}}^T X_{e_{t+1}}$ and $X_{e_{t+1}}^T Y$, the target function can be computed by

$$Y^T H_{e_{t+1}} Y = \left(Y^T X_{e_{t+1}} \right) \left(X_{e_{t+1}}^T X_{e_{t+1}} \right)^{-1} \left(X_{e_{t+1}}^T Y \right). \quad (3)$$

These relationships give rise to the following algorithm for computing $\left\{ Y^T H_{e_t} Y \right\}_{t=1}^M$.

Algorithm 1 Fast grid search for continuous threshold linear regression

1. Sort the samples by the order of x_i .
2. Compute and store $X_{e_1}^T X_{e_1}$ and $X_{e_1}^T Y$.
3. Compute and store $Y^T H_{e_1} Y$ according to (3).
4. For t in 1 to $M - 1$:
 - (a) compute $X_{e_{t+1}}^T X_{e_{t+1}}$ and $X_{e_{t+1}}^T Y$ according to (1) and (2),
 - (b) compute and store $Y^T H_{e_{t+1}} Y$ according to (3).

This algorithm is an instance of the dynamic programming technique in computer sciences, which has been key to training complex models such as deep neural network models (Goodfellow et al., 2016, Section 6.5). The essence of dynamic programming is to break down a complex problem into a collection of smaller subproblems and solve the subproblems in sequence, all the while storing the solutions of the subproblems to help reduce computational burden. In our instance, for example, computing $X_{e_1}^T X_{e_1}$ takes $O(np^2)$ number of operations, but computing each subsequent $X_{e_{t+1}}^T X_{e_{t+1}}$ takes $O(p)$ number of operations if we compute the cumulative sums of the covariate vectors in advance and store the results.

We now perform benchmarking experiments to assess the performance of the fast grid search algorithm. We compare the proposed method against the brute force grid search algorithm and the smooth approximation search algorithm (Fong et al., 2017a), and summarize the results in Table 2. These results show that the fast grid search algorithm achieves from 10 to 10^3 fold increase in speed over the brute force grid search algorithm as

the sample size increases from 50 to 500. Compared to the smooth approximation algorithm, the runtime of the fast grid search algorithm starts lower but increases faster, but even at sample size 2000, the fast grid search still outperforms (Figure B.1). Importantly, the fast grid search algorithm takes on average 0.5 seconds or less to fit datasets of sample size 500 or less and generate reliable confidence intervals.

4. Data examples

In this section we present a real data example from HIV-1 prevention studies (Permar et al., 2015) to illustrate the applications of continuous threshold linear models. A second example that uses a classical dataset is also presented in the Supplementary Materials Section B. Our example comprises immune responses measured in HIV-1 infected pregnant women. These women were part of a historical U.S. cohort enrolled prior to the availability of antiretroviral drugs; as such, robust HIV-1 specific antibody responses were observed among these subjects. Two important classes of measurements on antibodies concern their ability to bind HIV-1 specific antigens and their ability to neutralize HIV-1 viruses. Figure 1 shows a scatterplot of two variables. V3_BioV3B measured the binding activities specific to the V3 region of the HIV-1 envelope protein, and NAb_score measured the neutralization activities against a panel of both easy-to-neutralize and difficult-to-neutralize HIV-1 strains. Interestingly, the relationship between these two variables is clearly nonlinear and appears to have a broken-stick pattern. We fit a continuous threshold linear model to the dataset and the results are shown in Figure 1. The fast grid and smooth approximation search strategies yield rather different results. The estimated threshold is 0.47 (95% CI: 0.45, 0.48) by fast grid search and 0.42 (95% CI: 0.41, 0.44) by smooth approximation.

Based on fast grid search, V3_BioV3B increases with NAb_score before the threshold but not after: the estimated slopes are 67 (95% CI: 50, 84) and 2.4 (95% CI: -0.1, 4.9), respectively. One interpretation is that the assay used to measure V3_BioV3B reached saturation earlier than the assays used to measure NAb_score and that the latter collectively had a much greater dynamic range. In Permar et al. (2015), it was shown that both V3_BioV3B and NAb_score were associated with the risk of HIV-1 transmission from infected mothers to infants and that V3_BioV3B was a better predictor than NAb_score. Taken together, these results suggest that it would be very interesting to delve deeper into individual measurements that make up NAb_score to better understand the relationship among antibody binding activities, antibody neutralization activities, and risk of HIV-1 transmission.

5. Conclusions

In this paper we introduced a fast grid search method for fitting continuous threshold linear models, which simultaneously addresses the two challenges that hinder more widespread application of threshold regression: difficulty in model fitting and difficulty in constructing model-robust confidence intervals. The new grid search method achieves several orders of magnitude improvement in computation time and allows ‘real time’ construction of grid search-based bootstrap confidence intervals, which provides proper coverage regardless of

model misspecification. The proposed method is implemented in the R (R Development Core Team, 2008) package *chnppt*.

The concept behind the proposed fast grid search method is a general one. In Section C of the Supplementary Materials we show how the method can be extended to handle weights. Many other extensions are possible, including a) from continuous threshold models to discontinuous threshold models; b) from linear models to generalized linear model; and c) from single-threshold models to multi-thresholds models.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgment

The authors are grateful to the Editor and the AE for advice and comments and to Lindsay N. Carpp for help with editing. This work was supported by the National Institutes of Health (R01-AI122991; UM1-AI068635).

References

- Efron B and Tibshirani RJ (1993), An Introduction to the Bootstrap, Chapman and Hall.
- Fong Y, Huang Y, Gilbert P and Permar S (2017a), “chnppt: threshold regression model estimation and inference,” BMC Bioinformatics, 18, 454–460. [PubMed: 29037149]
- Fong Y, Chong D, Huang Y and Gilbert P (2017b), “Model-robust Inference for Continuous Threshold Regression Models,” Biometrics, 73, 452–462. [PubMed: 27858965]
- Goodfellow I, Bengio Y and Courville A (2016), Deep Learning, Adaptive computation and machine learning, MIT Press, Cambridge, MA.
- Hansen BE (2017), “Regression Kink with an Unknown Threshold,” Journal of Business and Economic Statistics, 35, 228–240.
- Hong J, Wang Y, McDermott S, Cai B, Aelion CM and Lead J (2016), “The use of a physiologically-based extraction test to assess relationships between bioaccessible metals in urban soil and neurodevelopmental conditions,” Environmental Pollution, 212, 9–17. [PubMed: 26840511]
- Muggeo V (2003), “Estimating regression models with unknown break-points,” Statistics in Medicine, 22, 3055–3071. [PubMed: 12973787]
- Pastor-Barriuso R, Guallar E and Coresh J (2003), “Transition models for change-point estimation in logistic regression,” Statistics in Medicine, 22, 1141–1162. [PubMed: 12652559]
- Permar SR, Fong Y, Vandergrift N, Fouda GG, Gilbert P, Parks R et al. (2015), “Maternal HIV-1 envelope-specific antibody responses and reduced risk of perinatal transmission,” Journal of Clinical Investigation, 125, 2702–2706. [PubMed: 26053661]
- R Development Core Team (2008), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria.
- Seber GA and Lee AJ (2003), Linear regression analysis, John Wiley & Sons, New Jersey.
- Sprent P (1961), “Some hypotheses concerning two phase regression lines,” Biometrics, 17, 634–645.

Figure 1:

The HIV-1 immune responses example. Left: results by fast grid search; right: results by smooth approximation search. Top: scatterplots with fitted models (gray lines); bottom: bootstrap distributions of the threshold estimate from 10^3 replicates. The dashed lines correspond to the 95% symmetric bootstrap confidence interval.

Table 1:

Coverage of four analytical and two bootstrap confidence interval methods under model misspecification. Est(%bias): MC mean and percent bias of MLE; range: distance between 2.5% and 97.5% percentiles of the sampling distributions; For each method, MC median width of 95% confidence intervals and estimated coverage (in parantheses) are shown. Results for β_z not shown due to space limit.

n	est(%bias)	range	model	sandwich	robust	robust*	bootstrap	bootstrap*
β_x								
50	0.97(7)	1.23	0.35(42)	0.41(49)	0.77(76)	0.99(84)	0.69(73)	1.02(88)
100	0.93(4)	0.99	0.26(40)	0.35(55)	0.61(78)	0.76(87)	0.56(77)	0.79(89)
250	0.91(1)	0.69	0.17(38)	0.27(60)	0.44(82)	0.55(89)	0.47(84)	0.59(90)
2000	0.90(0)	0.26	0.06(35)	0.12(64)	0.19(87)	0.23(93)	0.24(93)	0.25(94)
$\beta_{(x-e)_+}$								
50	1.82(-1)	1.04	0.54(70)	0.62(76)	0.69(81)	0.74(84)	0.81(88)	0.81(90)
100	1.84(0)	0.80	0.38(68)	0.53(83)	0.57(86)	0.59(87)	0.59(89)	0.60(89)
250	1.85(0)	0.51	0.24(67)	0.40(89)	0.42(90)	0.42(91)	0.42(91)	0.42(91)
2000	1.85(0)	0.18	0.09(67)	0.17(94)	0.17(94)	0.17(94)	0.16(93)	0.17(94)
e								
50	4.80(2)	1.94	0.41(32)	0.42(33)	1.21(81)	1.68(91)	1.09(73)	1.70(92)
100	4.76(1)	1.54	0.29(30)	0.33(34)	0.93(80)	1.27(91)	0.91(77)	1.34(93)
250	4.73(1)	1.07	0.18(28)	0.23(35)	0.66(80)	0.87(93)	0.76(85)	0.97(92)
2000	4.70(0)	0.40	0.07(25)	0.09(35)	0.28(85)	0.36(94)	0.38(94)	0.40(95)

Table 2:

Average run time (second) for continuous threshold linear model estimation using an Intel Xeon E5–2690 CPU clocked at 2.90GHz. 10^3 bootstrap replicates are performed for each dataset. Standard errors estimated from 200 independent datasets are shown in parentheses. The design matrix X_e is of dimension $n \times 4$.

n	grid search	smooth approx	fast grid search
50	0.71 (0.02)	12.09 (1.00)	0.05 (0.01)
100	4.67 (0.10)	12.92 (1.21)	0.09 (0.01)
250	68.72 (2.13)	16.75 (1.71)	0.23 (0.03)
500	1418.61 (67.71)	23.22 (2.26)	0.48 (0.06)
2000		66.95 (6.51)	2.94 (0.32)