Robust Bootstrap Testing for Nonlinear Effect with Kernel Ensemble

Wenying Deng
Department of Biostatistics
Harvard University

Biostat Seminar
October 18, 2018

*joint work with Jeremiah Liu and Brent Coull, with help and suggestions from Dr. Erin Lake.

Contents

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

    - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.

- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.
- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.
- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- Example: Consider a treatment effect study for lung cancer:

Wenying Deng Department of Biostatistics Harvard University

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.
- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- Example: Consider a treatment effect study for lung cancer:
  1. y: Lab measure for lung cancer

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.
- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- Example: Consider a treatment effect study for lung cancer:
  1. y: Lab measure for lung cancer
  2. $\mathbf{x}_1$: Nicotine intake

Problem Setup

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.
- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- Example: Consider a treatment effect study for lung cancer:
  1. y: Lab measure for lung cancer
  2. $\mathbf{x}_1$: Nicotine intake
  3. $\mathbf{x}_2$: Treatment

How to do this?

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.

$\Rightarrow$ Kernel Machine Regression

- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- Example: Consider a treatment effect study for lung cancer:
  1. y: Lab measure for lung cancer
  2. $\mathbf{x}_1$: Nicotine intake
  3. $\mathbf{x}_2$: Treatment

How to do this?

- Data:

$$\{y_i, \mathbf{x}_i\}_{i=1}^n, \quad \text{where} \quad \mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$$

  - Effect of $\mathbf{x}_1$ and $\mathbf{x}_2$ are nonlinear.

- Model:

$$y_i = \mu + f_1(\mathbf{x}_{i1}) + f_2(\mathbf{x}_{i2}) + \epsilon_i$$

- Question: Does $\mathbf{x}_2$ have any impact on y? or

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

$\Rightarrow$ Kernel Machine Regression

$\Downarrow$

Variance Component Test

- Example: Consider a treatment effect study for lung cancer:
  1. y: Lab measure for lung cancer
  2. $\mathbf{x}_1$: Nicotine intake
  3. $\mathbf{x}_2$: Treatment

1 Problem Setup

2 Model & Method

- Estimating f: Kernel Machine Regression (KMR)
- Testing $f = 0$: Variance Component Test

3 Cross-Validated Kernel Ensemble

4 Bootstrap Test

5 Simulation Study

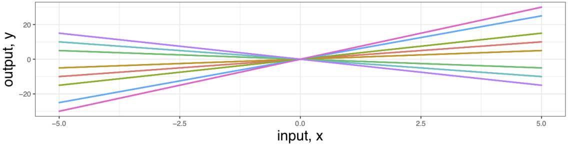
Kernel Machine Regression

- Consider the typical linear model

$$y_i = \mu + x_i^\top \boldsymbol{\beta}$$

- In matrix notation:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{X}\beta$$



Linear Model: $y_i = \mu + x_i\beta$

Kernel Machine Regression

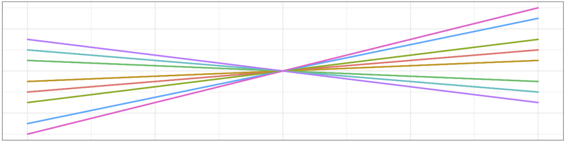
- A more general way to write this:

$$y_i = \mu + f(x_i) \qquad f(x_i) = \sum_{j=1}^{n} k(x_i, x_j)\alpha_j$$

- In matrix notation:

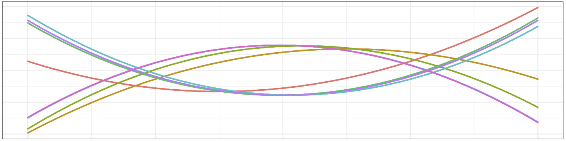
$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}\boldsymbol{\alpha}$$

- Linear regression corresponds to $k(x_i, x_j) = (1 + x_i^\top x_j)$.

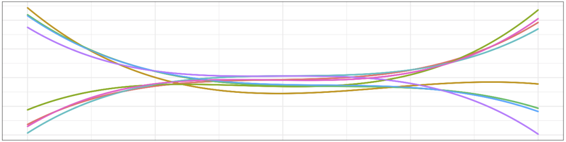


Linear Kernel: $k(x_i, x_j) = (1 + x_i^\top x_j)$

Example: Polynomial Kernels

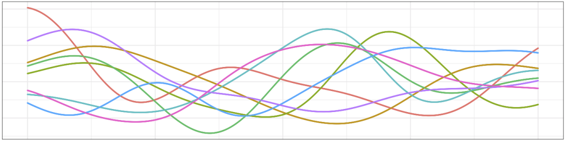


Quadratic Polynomial Kernel: $k(x_i, x_j) = (1 + x_i^\top x_j)^2$

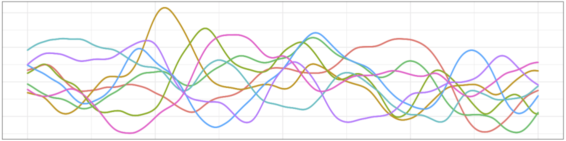


Cubic Polynomial Kernel: $k(x_i, x_j) = (1 + x_i^\top x_j)^3$

Example: (Really) Flexible Kernels



Radial Basis Functions: $k(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{2l^2})$



Matérn 3/2 Kernel: $(1 + \frac{\sqrt{3}||x_i - x_j||}{l})\exp(-\frac{\sqrt{3}||x_i - x_j||}{l})$

KMR as a Linear Mixed Model

As it turns out, KMR can be written as a Linear Mixed Model.
In the case of fitting $f(\mathbf{x})$:

- Kernel Machine Regression:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- Linear Mixed Model:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{h} + \boldsymbol{\epsilon},$$
$$\mathbf{h} \sim \mathrm{N}(\mathbf{0}, \mathbf{K})$$

KMR as a Linear Mixed Model

In the case of fitting $f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$:

- Kernel Machine Regression:

$$\mathbf{y} = \boldsymbol{\mu} + (\mathbf{K}_1 + \mathbf{K}_2)\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- Linear Mixed Model:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{h} + \boldsymbol{\epsilon},$$
$$\mathbf{h} \sim \mathrm{N}(\mathbf{0}, \mathbf{K}_1 + \mathbf{K}_2)$$

Variance Component Test

So if want to test for $H_0 : f_2(\mathbf{x}_2) = 0$:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{h} + \boldsymbol{\epsilon},$$
$$\mathbf{h} \sim \mathrm{N}(\mathbf{0}, \mathbf{K}_1 + \boldsymbol{\tau}\mathbf{K}_2)$$

- Test Hypothesis:

$$H_0 : \boldsymbol{\tau} = 0$$

- Test Statistic:

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

where $\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\boldsymbol{\mu}} - \mathbf{K}_1 \hat{\boldsymbol{\alpha}}$ is null model residual.

- Null Distribution:

$$\hat{P}_0(T) := \text{distribution of } \; T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}} \; \text{ under the null}$$

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: Kernel Machine Regression

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1 \boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But this test doesn't always work in practice...

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: Kernel Machine Regression

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But this test doesn't always work in practice...
  - Why? Easy to misspecify null model $\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha}$.

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: Kernel Machine Regression

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But this test doesn't always work in practice...
  - Why? Easy to misspecify null model $\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha}$.
  - Hard to choose a proper kernel function k.

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: Kernel Machine Regression

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

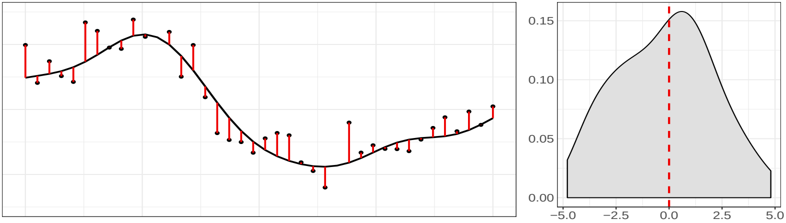
- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But this test doesn't always work in practice...
  - Why? Easy to misspecify null model $\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha}$.
  - Hard to choose a proper kernel function k.
  - How to solve this? **Ensemble learning!**

The Kernel Misspecification Problem

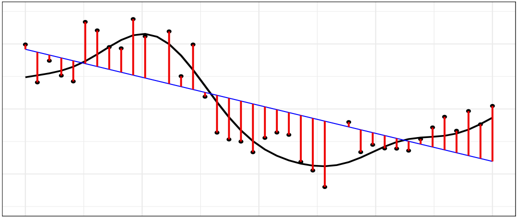
- If just use one kernel, hard to fit data correctly all the time.

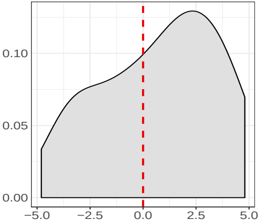


Ideal model fit

Motivation

- Kernel too simple $\Rightarrow$ underfitting $\Rightarrow$ biased residual estimate $\hat{\epsilon}$.

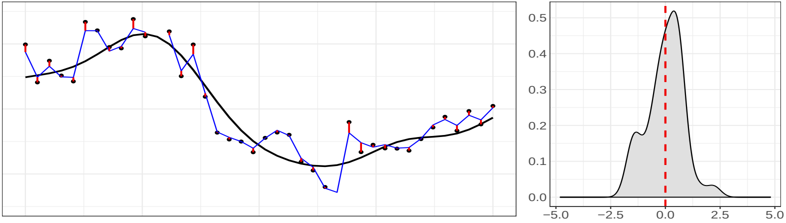




Underfit, biased $\hat{\epsilon}$

Motivation

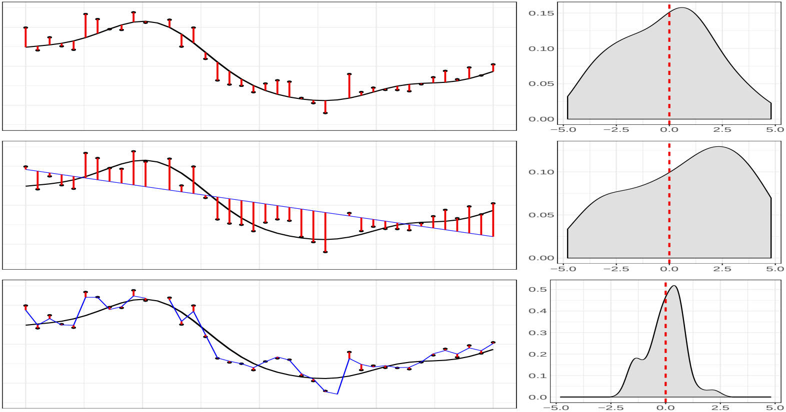
- Kernel too flexible $\Rightarrow$ overfitting $\Rightarrow$ under-estimated residual $\hat{\epsilon}$.



Overfit, under-estimated $\hat{\epsilon}$

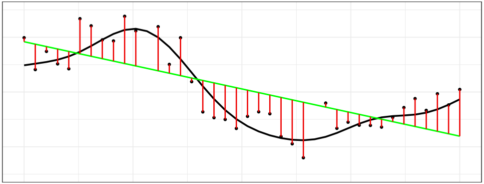
Motivation

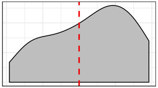
- Incorrect $\hat{\epsilon} \Rightarrow$ incorrect $T \propto \hat{\epsilon}^T \mathbf{K}_2 \hat{\epsilon} \Rightarrow$ incorrect inference!



CVEK: Better $\hat{\epsilon}$ using Ensemble Learning

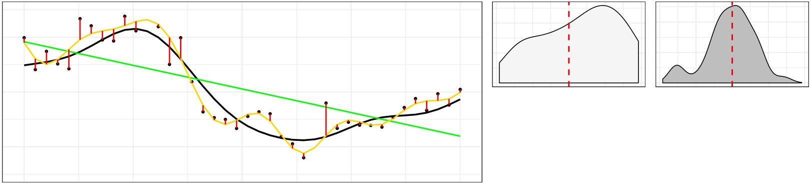
- Idea: Try many possible kernels.





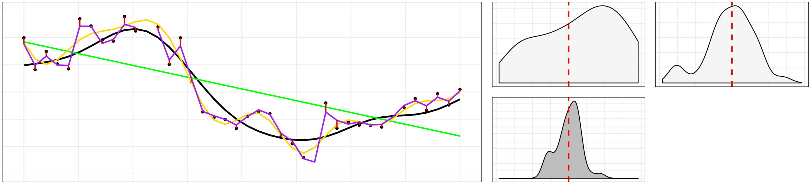
CVEK: Better $\hat{\epsilon}$ using Ensemble Learning

- Idea: Try many possible kernels.



CVEK: Better $\hat{\epsilon}$ using Ensemble Learning

- Idea: Try many possible kernels.

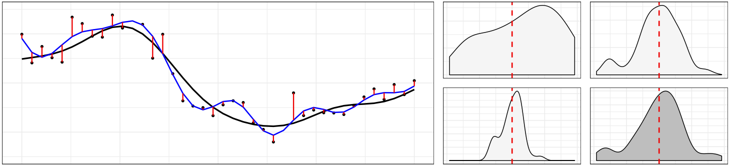


CVEK: Better $\hat{\epsilon}$ using Ensemble Learning

- Combine them to produce proper model fit.
  - based on individual kernel's **CV** performance.
- Hence the name:

Cross-**V**alidated **E**nsemble of **K**ernels

- Better model fit $\Rightarrow$ better residual estimate $\Rightarrow$ better inference!



Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But still doesn't work super well in small sample...

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But still doesn't work super well in small sample...
  - Why? Asymptotic test replies on large-sample approximation.

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything!
  - But still doesn't work super well in small sample...
  - Why? Asymptotic test replies on large-sample approximation.
  - How to solve this? **Bootstrap Test**!

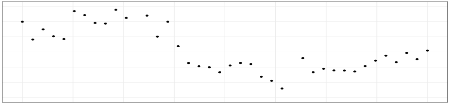
Asymptotic v.s. Bootstrap Test

$$\hat{P}_0(T) := \text{distribution of } \hat{\epsilon}^\top \mathbf{K}_2 \hat{\epsilon} \text{ under the null: } \hat{\epsilon} = \mathbf{y} - \hat{\boldsymbol{\mu}} - \mathbf{K}_1 \hat{\boldsymbol{\alpha}}$$

- Asymptotic
  1. pretend $\hat{\epsilon}$ come from $n \rightarrow \infty$ data
  2. null distribution of $\hat{\epsilon}$: $\hat{P}_{asym}(\hat{\epsilon})$ is derived analytically
  3. null distribution of T: $\hat{P}_0(\hat{\epsilon}^\top \mathbf{K}_2 \hat{\epsilon})$
- Bootstrap
  1. don't assume distribution of $\hat{\epsilon}$, sample directly from empirical distribution
  2. null distribution of $\hat{\epsilon}$: $\hat{P}_{empirical}(\hat{\epsilon})$ is estimated from bootstrap sample
  3. null distribution of T: $\hat{P}_0(\hat{\epsilon}^\top \mathbf{K}_2 \hat{\epsilon})$

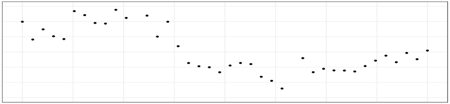
Bootstrap Test: Idea

1. Given training sample

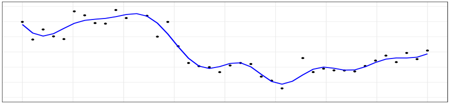


Bootstrap Test: Idea

1. Given training sample

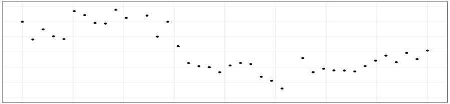


2. Fit null model $\hat{f}_1(\mathbf{x}_1)$ using Kernel Ensemble

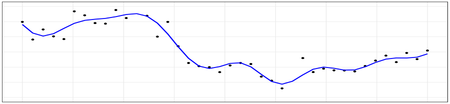


Bootstrap Test: Idea

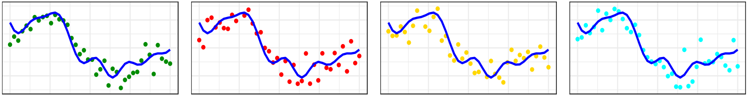
1. Given training sample



2. Fit null model $\hat{f}_1(\mathbf{x}_1)$ using Kernel Ensemble



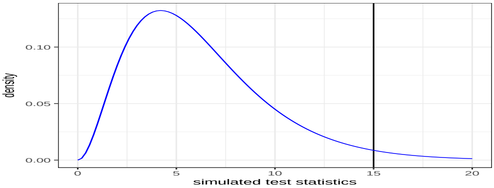
3. Generate many $y^* = \hat{f}_1(\mathbf{x}_1) + \epsilon^*$, then generate $\hat{T}_0$'s based on y^*.



Bootstrap Test: Idea

1. Notice the $\hat{T}_0$'s are generated under null
2. Therefore, bootstrap sample $\{\hat{T}_{0b}\}_{b=1}^{B}$ forms null distribution of T
3. Compute p-value is as simple as:

$$p\text{-value} = \frac{1}{B}\sum_{b=1}^{B} I(T_{obs} < \hat{T}_{0b})$$



Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: **asym** or **boot**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything?
  - Robust null model fit using ensemble learning (CVEK).

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0: \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: **asym** or **boot**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything?
  - Robust null model fit using ensemble learning (CVEK).
  - Robust small sample test using bootstrap.

Summary

- **Data**: $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$
- **Goal**: Test for

$$H_0 : \quad f_2(\mathbf{x}_{i2}) = 0$$

- **Model**: **CVEK**: Kernel Machine Regression + Ensemble Learning

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$

- **Hypothesis Test**: **asym** or **boot**: Variance Component Test

$$T \propto \hat{\boldsymbol{\epsilon}}^\top \mathbf{K}_2 \hat{\boldsymbol{\epsilon}}$$

- Looks like we solved everything?
  - Robust null model fit using ensemble learning (CVEK).
  - Robust small sample test using bootstrap.
  - How does it perform though?

1 Problem Setup

2 Model & Method
- Estimating f: Kernel Machine Regression (KMR)
- Testing $f = 0$: Variance Component Test

3 Cross-Validated Kernel Ensemble

4 Bootstrap Test

5 Simulation Study

Procedure

1. generate data using a specific k:

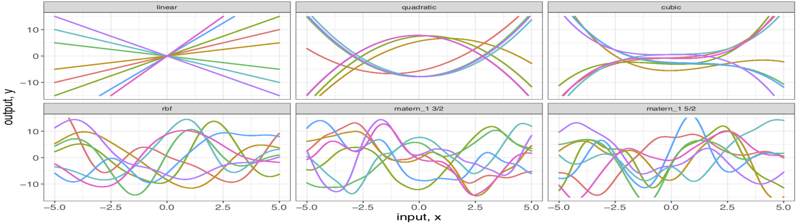
$$\mathbf{y} = \boldsymbol{\mu} + f_1(\mathbf{x}_1) + \delta \cdot f_2(\mathbf{x}_2) + \boldsymbol{\epsilon}.$$

2. fit $\mathbf{y} = \boldsymbol{\mu} + f_1(\mathbf{x}_1) + \boldsymbol{\epsilon}$ with an ensemble of kernels

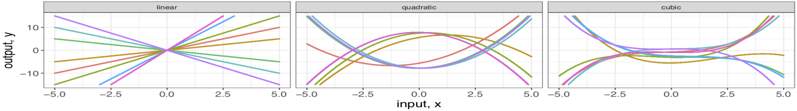
$$\mathbf{k}_{model} = \{k_1, k_2, k_3\}.$$

3. repeat steps 1 and 2 under
   1. different data-generation mechanisms
   2. different types of kernel ensembles $\mathbf{k}_{model}$

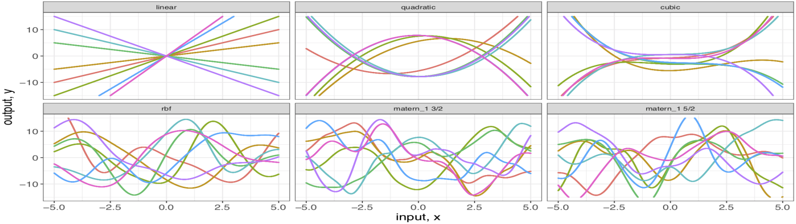
- data I tried:



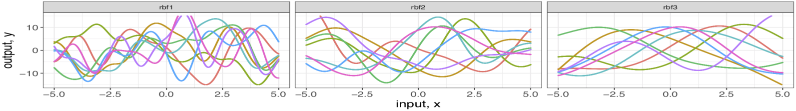
- first set of $\mathbf{k}_{model}$ I tried (Polynomial Kernels):



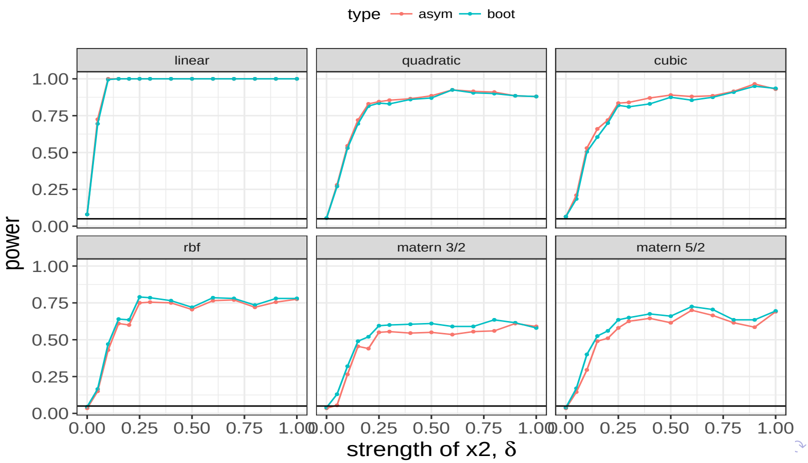
- data I tried:



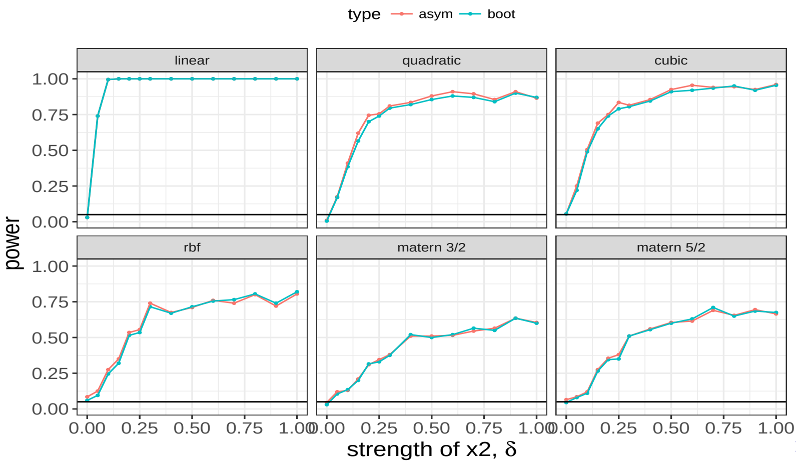
- second set of $\mathbf{k}_{model}$ I tried (RBF Kernels):



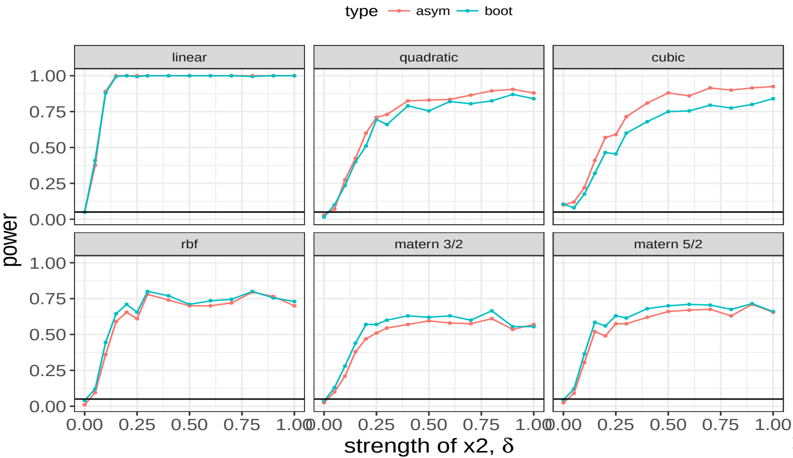
Test Performance, Oracle Ensemble



Test Performance, Polynomial Ensemble



Test Performance, RBF Ensemble



Conclusion

1. better guarantee Type I error.
2. better power under non-polynomial (especially non-smooth) data.
3. R package *CVEK* is coming soon!
   $\{https://github.com/IrisTeng/CVEK\}$

Questions?

Main References

1. Lin X (1997). "Variance component testing in generalised linear models with random effects."
2. Liu JZ, Coull B (2017). "Robust Hypothesis Test for Nonlinear Effect with Gaussian Processes."
3. Maity A, Lin X (2011). "Powerful tests for detecting a gene effect in the presence of possible gene-gene interactions using garrote kernel machines."
4. Liu D, Lin X, Ghosh D (2007). "Semiparametric regression of multidimensional genetic pathway data: least-squares kernel machines and linear mixed models."
5. Hastie T, Tibshirani R, Friedman J (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.
6. Press TM (2006). "Gaussian Processes for Machine Learning."

Appendix

Linear $\Rightarrow$ Nonliear:

$$
\begin{aligned}
\hat{\mathbf{y}} &= \mathbf{X}\hat{\beta} \\
&= \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1}\mathbf{X}^\top \mathbf{y} && \text{(ridge regression)} \\
&= \mathbf{X}\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda I_n)^{-1}\mathbf{y} && \text{(svd)} \\
&= \mathbf{X}\mathbf{X}^\top\hat{\alpha} = \underset{n\times n}{\mathbf{K}}\,\hat{\alpha}
\end{aligned}
$$

1. Obtain parameter estimates from the original data by fitting a null-hypothesis model

$$\hat{\mathbf{y}} = \mathbf{K}_0(\mathbf{K}_0 + \lambda\mathbf{I})^{-1}\mathbf{y} = \mathbf{A}_0\mathbf{y}$$

2. Sample $\mathbf{y}^*$ for each individuals with a random noise, whose variance is also estimated.
3. Compute the test statistic, based on fitting the alternative-hypothesis model to the samples obtained in Step 2.
4. Repeat Steps 2 and 3 for B times, to obtain an approximate distribution of the test statistic.
5. Compute the test statistic for the original data, based on fitting the alternative- hypothesis model.
6. Compute the p-value, by comparing the test statistic in Step 5 to the distribution in Step 4.