Model-free Prediction and Bootstrap

Dimitris N. Politis

Statistics in the 20th Century

Statistics in the 20th Century

- Once upon a time in the UK

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed
- Shape of $F_\theta(x)$ is known — parameter θ is unknown

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed
- Shape of $F_\theta(x)$ is known — parameter θ is unknown
  e.g. F_θ is $N(\theta, 1)$ or $N(\theta, \sigma^2)$ with σ^2 being a "nuisance"(!)

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed
- Shape of $F_\theta(x)$ is known — parameter θ is unknown
  e.g. F_θ is $N(\theta, 1)$ or $N(\theta, \sigma^2)$ with σ^2 being a "nuisance"(!)
- $F_\theta(x)$ belongs to a parametric family of distributions

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed
- Shape of $F_\theta(x)$ is known — parameter θ is unknown
  e.g. F_θ is $N(\theta, 1)$ or $N(\theta, \sigma^2)$ with σ^2 being a "nuisance"(!)
- $F_\theta(x)$ belongs to a parametric family of distributions
- Goal: Use the data to estimate θ

Statistics in the 20th Century

- Once upon a time in the UK — Fisher, Pearson, Gosset, etc.
- Data: $Y_1, \ldots, Y_n$ i.i.d. from some distribution $F_\theta(x)$
- i.i.d.= independent, identically distributed
- Shape of $F_\theta(x)$ is known — parameter θ is unknown
  e.g. F_θ is $N(\theta, 1)$ or $N(\theta, \sigma^2)$ with σ^2 being a "nuisance"(!)
- $F_\theta(x)$ belongs to a parametric family of distributions
- Goal: Use the data to estimate θ — but also quantify estimation accuracy (standard error, confidence interval, etc.)

R.A. Fisher and Maximum Likelihood Estimation (MLE)

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function
- Under regularity conditions, $\hat{\theta}_{MLE}$ is consistent for θ and ...

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function
- Under regularity conditions, $\hat{\theta}_{MLE}$ is consistent for θ and ...
  ... asymptotically normal, i.e., $\hat{\theta}_{MLE} \sim N(\theta, I(\theta)/n)$ for large n

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function
- Under regularity conditions, $\hat{\theta}_{MLE}$ is consistent for θ and ...
  ... asymptotically normal, i.e., $\hat{\theta}_{MLE} \sim N(\theta, I(\theta)/n)$ for large n
- The Fisher information $I(\theta)$ can be computed from F_θ

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function
- Under regularity conditions, $\hat{\theta}_{MLE}$ is consistent for θ and ...
  ... asymptotically normal, i.e., $\hat{\theta}_{MLE} \sim N(\theta, I(\theta)/n)$ for large n
- The Fisher information $I(\theta)$ can be computed from F_θ
- Can use the asymptotic normal distribution to construct confidence intervals and hypothesis tests for θ

R.A. Fisher and Maximum Likelihood Estimation (MLE)

- $\hat{\theta}_{MLE}$ is the value maximizing the Likelihood function
- Under regularity conditions, $\hat{\theta}_{MLE}$ is consistent for θ and ...
  ... asymptotically normal, i.e., $\hat{\theta}_{MLE} \sim N(\theta, I(\theta)/n)$ for large n
- The Fisher information $I(\theta)$ can be computed from F_θ
- Can use the asymptotic normal distribution to construct confidence intervals and hypothesis tests for θ
- MLE is a complete theory for statistical inference.

What's the catch?

What's the catch?

- 100 years ago, sample sizes were quite small

What's the catch?

- 100 years ago, sample sizes were quite small
- W.S. Gosset (AKA "a student") was working with $n = 9$ at the Guiness Brewery in 1908

What's the catch?

- 100 years ago, sample sizes were quite small
- W.S. Gosset (AKA "a student") was working with $n = 9$ at the Guiness Brewery in 1908
- Asymptotic normality can not be justified

What's the catch?

- 100 years ago, sample sizes were quite small
- W.S. Gosset (AKA "a student") was working with $n = 9$ at the Guiness Brewery in 1908
- Asymptotic normality can not be justified
- Assuming F_θ is $N(\theta, \sigma^2)$, Gosset figured out the exact distribution of the "studentized" sample mean $\frac{\bar{X}-\theta}{\hat{\sigma}}$.

What's the catch?

- 100 years ago, sample sizes were quite small
- W.S. Gosset (AKA "a student") was working with $n = 9$ at the Guiness Brewery in 1908
- Asymptotic normality can not be justified
- Assuming F_θ is $N(\theta, \sigma^2)$, Gosset figured out the exact distribution of the "studentized" sample mean $\frac{\bar{X}-\theta}{\hat{\sigma}}$.
- But how about statistics other than the sample mean $\bar{X}$?

What's the catch–part II

What's the catch–part II

- Why/how can we assume that F_θ belongs to any given parametric family? E.g. why assume F_θ is $N(\theta, \sigma^2)$?

What's the catch–part II

- Why/how can we assume that F_θ belongs to any given parametric family? E.g. why assume F_θ is $N(\theta, \sigma^2)$?
- Answer: for convenience, in view of a small sample

What's the catch–part II

- Why/how can we assume that F_θ belongs to any given parametric family? E.g. why assume F_θ is $N(\theta, \sigma^2)$?
- Answer: for convenience, in view of a small sample
- With a large sample $Y_1, \ldots, Y_n$, the common distribution $F(x)$ can be readily estimated from the data.

What's the catch–part II

- Why/how can we assume that F_θ belongs to any given parametric family? E.g. why assume F_θ is $N(\theta, \sigma^2)$?
- Answer: for convenience, in view of a small sample
- With a large sample $Y_1, \ldots, Y_n$, the common distribution $F(x)$ can be readily estimated from the data.
- $F(x) = P\{Y_i \leq x\}$ can be estimated by $\hat{F}(x) = \frac{\#\{Y_i \leq x\}}{n}$ i.e., the proportion of data points that are $\leq x$.

What's the catch–part II

- Why/how can we assume that F_θ belongs to any given parametric family? E.g. why assume F_θ is $N(\theta, \sigma^2)$?
- Answer: for convenience, in view of a small sample
- With a large sample $Y_1, \ldots, Y_n$, the common distribution $F(x)$ can be readily estimated from the data.
- $F(x) = P\{Y_i \leq x\}$ can be estimated by $\hat{F}(x) = \frac{\#\{Y_i \leq x\}}{n}$ i.e., the proportion of data points that are $\leq x$.
- This is a modern, nonparametric setup.

An example under the nonparametric setup

An example under the nonparametric setup

- $Y_1, \dots, Y_n$ are house sale prices in San Diego in Jan. 2022

An example under the nonparametric setup

- $Y_1, \ldots, Y_n$ are house sale prices in San Diego in Jan. 2022
- The median house price θ can be estimated by the sample median $\hat{\theta}$, i.e., the median of the data points $Y_1, \ldots, Y_n$

An example under the nonparametric setup

- $Y_1, \ldots, Y_n$ are house sale prices in San Diego in Jan. 2022
- The median house price θ can be estimated by the sample median $\hat{\theta}$, i.e., the median of the data points $Y_1, \ldots, Y_n$
- What is the standard error of the sample median $\hat{\theta}$?

An example under the nonparametric setup

- $Y_1, \ldots, Y_n$ are house sale prices in San Diego in Jan. 2022
- The median house price θ can be estimated by the sample median $\hat{\theta}$, i.e., the median of the data points $Y_1, \ldots, Y_n$
- What is the standard error of the sample median $\hat{\theta}$?
- So if $\hat{\theta} = 555K$, how sure are you that this figure —which was based on (say) $n = 300$ points— is close to the true median?

A thought experiment

A thought experiment

- Statistic $\hat{\theta}$ was computed from data $Y_1, \ldots, Y_n$ i.i.d. from F

A thought experiment

- Statistic $\hat{\theta}$ was computed from data $Y_1, \ldots, Y_n$ i.i.d. from F
- If we knew F we could generate more samples, and witness how $\hat{\theta}$ varies across samples.

A thought experiment

- Statistic $\hat{\theta}$ was computed from data $Y_1, \ldots, Y_n$ i.i.d. from F
- If we knew F we could generate more samples, and witness how $\hat{\theta}$ varies across samples.
- Parallel universes:

$$
\begin{array}{l}
\text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(1)} \\
\text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(2)} \\
\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(B)}
\end{array}
$$

A thought experiment

- Statistic $\hat{\theta}$ was computed from data $Y_1, \ldots, Y_n$ i.i.d. from F
- If we knew F we could generate more samples, and witness how $\hat{\theta}$ varies across samples.
- Parallel universes:

$$
\begin{array}{l}
\text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(1)} \\
\text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(2)} \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(B)}
\end{array}
$$

- Approximate the variance of $\hat{\theta}$ by the sample variance of the artificial statistics: $\hat{\theta}^{(1)}, \cdots, \hat{\theta}^{(B)}$.

Resampling and the bootstrap – circa 1980

Resampling and the bootstrap – circa 1980

- This is just a Monte Carlo simulation assuming F is known.

$$
\begin{array}{l}
\text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(1)} \\
\text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(2)} \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(B)}
\end{array}
$$

Resampling and the bootstrap – circa 1980

- This is just a Monte Carlo simulation assuming F is known.

$$\text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(1)}$$
$$\text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(2)}$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(B)}$$

- But F is unknown...

Resampling and the bootstrap – circa 1980

- This is just a Monte Carlo simulation assuming F is known.

$$\begin{array}{ll} \text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(1)} \\ \text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(2)} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } F \text{ and compute } \hat{\theta}^{(B)} \end{array}$$

- But F is unknown... plugging in $\hat{F}$ for F makes this bootstrap

$$\begin{array}{ll} \text{Generate sample } Y_1^{(1)}, \ldots, Y_n^{(1)} \text{ i.i.d. from } \hat{F} \text{ and compute } \hat{\theta}^{(1)} \\ \text{Generate sample } Y_1^{(2)}, \ldots, Y_n^{(2)} \text{ i.i.d. from } \hat{F} \text{ and compute } \hat{\theta}^{(2)} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \text{Generate sample } Y_1^{(B)}, \ldots, Y_n^{(B)} \text{ i.i.d. from } \hat{F} \text{ and compute } \hat{\theta}^{(B)} \end{array}$$

From i.i.d. to non-i.i.d. data

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming...

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming...
  the data are i.i.d. i.e., independent, identically distributed.

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming... the data are i.i.d. i.e., independent, identically distributed.
- i.N.d. = independent, Non-identically distributed data

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming... the data are i.i.d. i.e., independent, identically distributed.
- i.N.d. = independent, Non-identically distributed data
  Regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where the errors ϵ_i are i.i.d.

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming...
  the data are i.i.d. i.e., independent, identically distributed.
- i.N.d. = independent, Non-identically distributed data
  Regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where the errors ϵ_i are i.i.d.
- N.i.d. = Non-independent, identically distributed data

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming... the data are i.i.d. i.e., independent, identically distributed.
- i.N.d. = independent, Non-identically distributed data
  Regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where the errors ϵ_i are i.i.d.
- N.i.d. = Non-independent, identically distributed data
  Stationary Time Series: $Y_i = \beta_0 + \beta_1 Y_{i-1} + \epsilon_i$ with ϵ_i i.i.d.

From i.i.d. to non-i.i.d. data

- Efron's bootstrap works for a variety of statistics assuming... the data are i.i.d. i.e., independent, identically distributed.
- i.N.d. = independent, Non-identically distributed data
  Regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where the errors ϵ_i are i.i.d.
- N.i.d. = Non-independent, identically distributed data
  Stationary Time Series: $Y_i = \beta_0 + \beta_1 Y_{i-1} + \epsilon_i$ with ϵ_i i.i.d.
- Fit Regression and Autoregression models to reduce to i.i.d.

From model-based to model-free – D.P. (2015)

From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.

From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$

From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$
- Find an invertible transformation H_n such that the vector $\underline{\epsilon} = H_n(\underline{Y})$ has i.i.d. components $\epsilon_1, \ldots, \epsilon_n$

From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$
- Find an invertible transformation H_n such that the vector $\underline{\epsilon} = H_n(\underline{Y})$ has i.i.d. components $\epsilon_1, \ldots, \epsilon_n$
- Resample the i.i.d. $\epsilon_1, \ldots, \epsilon_n$, and map back (using the inverse transformation) to obtain bootstrap samples in the Y–domain.

From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$
- Find an invertible transformation H_n such that the vector $\underline{\epsilon} = H_n(\underline{Y})$ has i.i.d. components $\epsilon_1, \ldots, \epsilon_n$
- Resample the i.i.d. $\epsilon_1, \ldots, \epsilon_n$, and map back (using the inverse transformation) to obtain bootstrap samples in the Y–domain.
- Steps: (i) Estimate the common distribution F_ϵ of $\epsilon_1, \ldots, \epsilon_n$

From model-based to model-free – D.P. (2015)

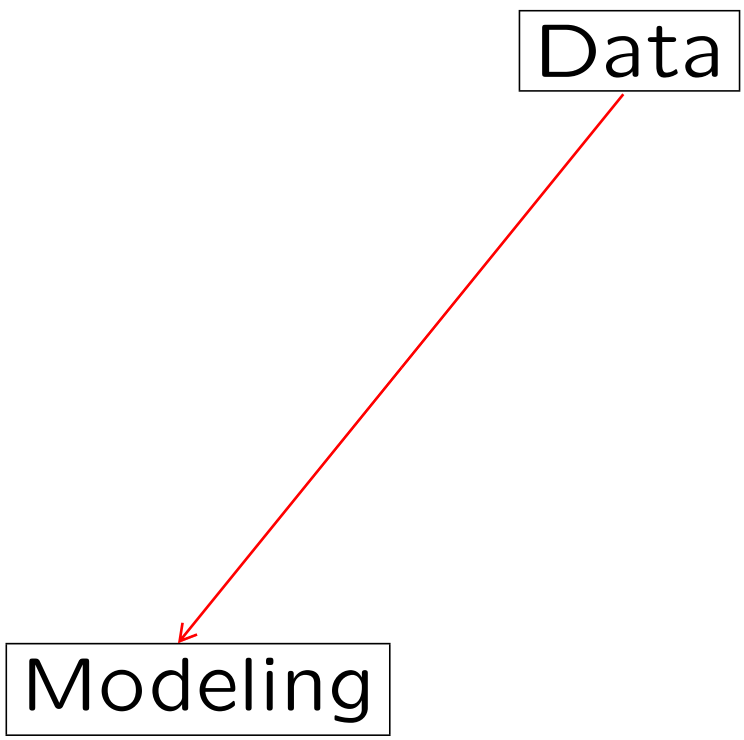
- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$
- Find an invertible transformation H_n such that the vector $\underline{\epsilon} = H_n(\underline{Y})$ has i.i.d. components $\epsilon_1, \ldots, \epsilon_n$
- Resample the i.i.d. $\epsilon_1, \ldots, \epsilon_n$, and map back (using the inverse transformation) to obtain bootstrap samples in the Y–domain.
- Steps: (i) Estimate the common distribution F_ϵ of $\epsilon_1, \ldots, \epsilon_n$
- (ii) Resample from the estimated F_ϵ to create a bootstrap sample $\epsilon_1^*, \ldots, \epsilon_n^*$

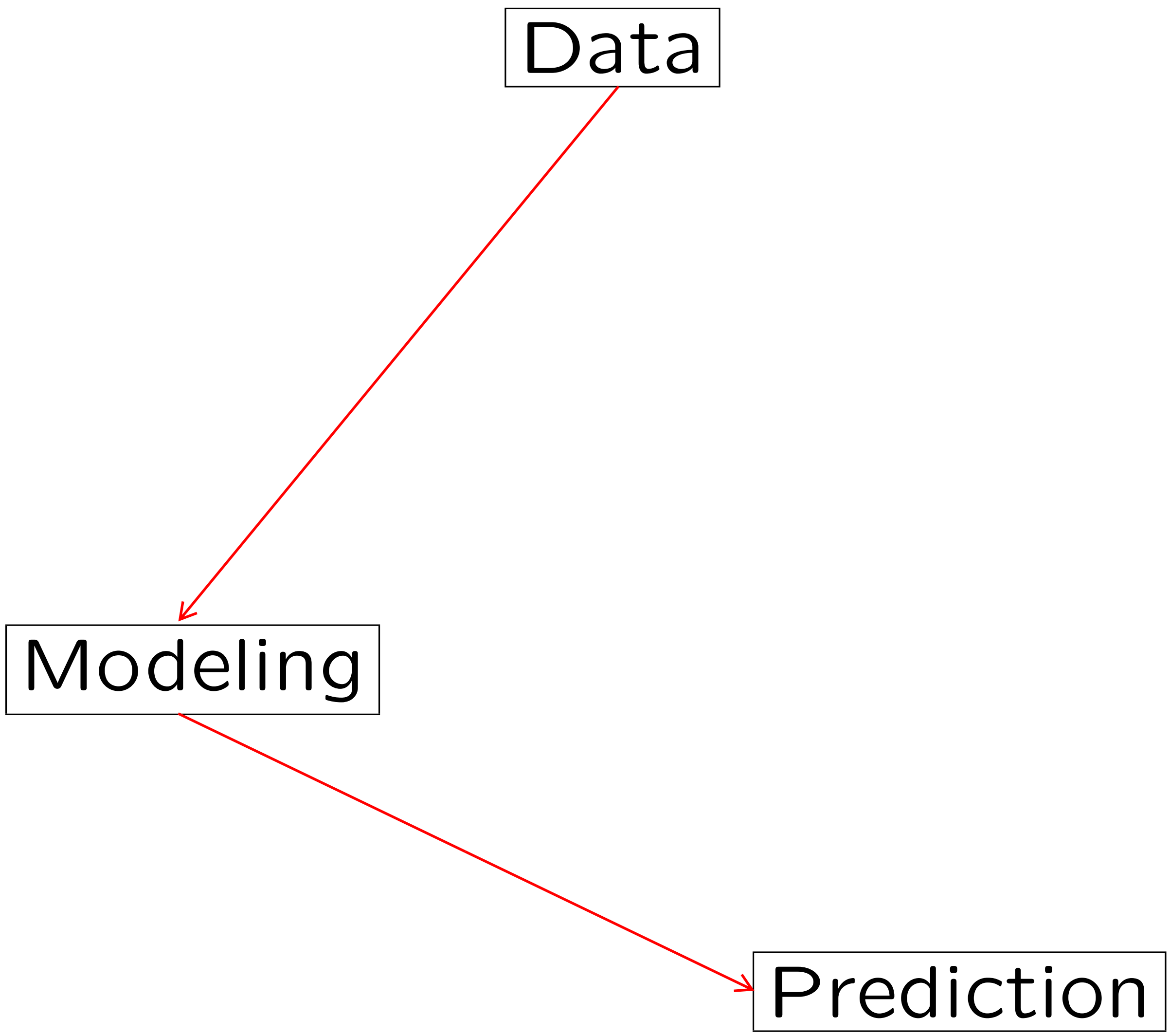
From model-based to model-free – D.P. (2015)

- Data: $Y_1, \ldots, Y_n$ not i.i.d.
- Let $\underline{Y} = (Y_1, \ldots, Y_n)'$
- Find an invertible transformation H_n such that the vector $\underline{\epsilon} = H_n(\underline{Y})$ has i.i.d. components $\epsilon_1, \ldots, \epsilon_n$
- Resample the i.i.d. $\epsilon_1, \ldots, \epsilon_n$, and map back (using the inverse transformation) to obtain bootstrap samples in the Y–domain.
- Steps: (i) Estimate the common distribution F_ϵ of $\epsilon_1, \ldots, \epsilon_n$
- (ii) Resample from the estimated F_ϵ to create a bootstrap sample $\epsilon_1^*, \ldots, \epsilon_n^*$
- (iii) Let $\underline{Y}^* = H_n^{-1}(\underline{\epsilon}^*)$ where $\underline{\epsilon}^* = (\epsilon_1^*, \ldots, \epsilon_n^*)'$

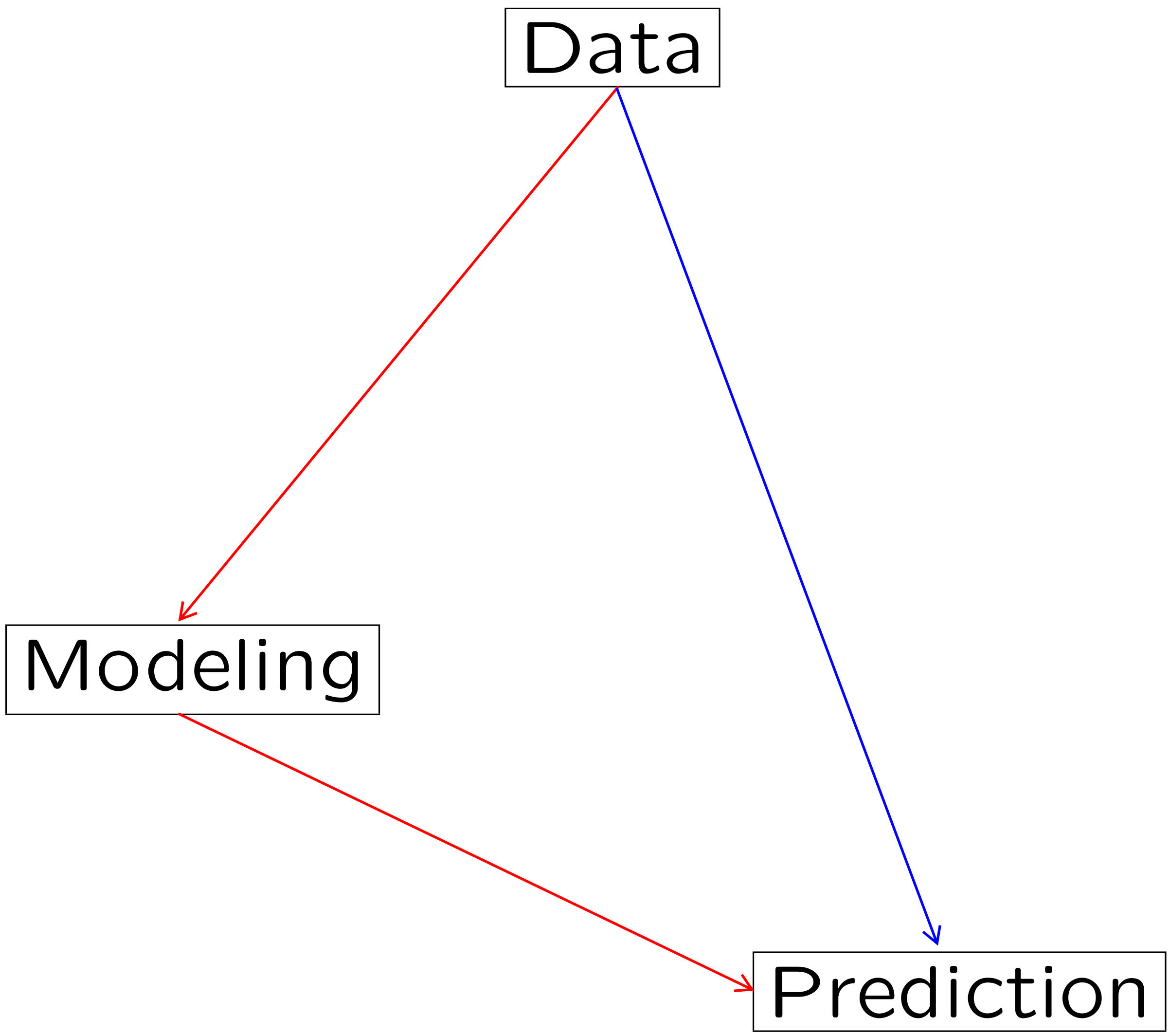
Data

Modeling


Data
Modeling



Data
Modeling
Prediction


Data
Modeling
Prediction

To explain or to predict?

- Models are indispensable for exploring/utilizing relationships between variables: explaining the world.

To explain or to predict?

- Models are indispensable for exploring/utilizing relationships between variables: explaining the world.
- Use of models for prediction can be problematic when:

To explain or to predict?

- Models are indispensable for exploring/utilizing relationships between variables: explaining the world.
- Use of models for prediction can be problematic when:
  - a model is overspecified
  - parameter inference is highly model-specific (and sensitive to model mis-specification)
  - prediction is carried out by plugging in the estimated parameters and treating the model as exactly true.

To explain or to predict?

- Models are indispensable for exploring/utilizing relationships between variables: explaining the world.
- Use of models for prediction can be problematic when:
  - a model is overspecified
  - parameter inference is highly model-specific (and sensitive to model mis-specification)
  - prediction is carried out by plugging in the estimated parameters and treating the model as exactly true.
- "All models are wrong but some are useful"— George Box.

A Toy Example

- Assume regression model: $Y = \beta_0 + \beta_1 X + \beta_2 X^{20} +$ error

A Toy Example

- Assume regression model: $Y = \beta_0 + \beta_1 X + \beta_2 X^{20} +$ error
- If $\hat{\beta}_2$ is barely statistically significant, do you still use it in prediction?

A Toy Example

- Assume regression model: $Y = \beta_0 + \beta_1 X + \beta_2 X^{20} +$ error
- If $\hat{\beta}_2$ is barely statistically significant, do you still use it in prediction?
- If the true value of β_2 is close to zero, and $var(\hat{\beta}_2)$ is large, then it may be advantageous to omit β_2: allow a nonzero Bias but minimize MSE.

A Toy Example

- Assume regression model: $Y = \beta_0 + \beta_1 X + \beta_2 X^{20} +$ error
- If $\hat{\beta}_2$ is barely statistically significant, do you still use it in prediction?
- If the true value of β_2 is close to zero, and $var(\hat{\beta}_2)$ is large, then it may be advantageous to omit β_2: allow a nonzero Bias but minimize MSE.
- A mis-specified model can be optimal for prediction!

Prediction Framework

- a. Point predictors
  b. Interval predictors
  c. Predictive distribution

Prediction Framework

- a. Point predictors
  b. Interval predictors
  c. Predictive distribution
- Abundant Bayesian literature in parametric framework
  —Cox (1975), Geisser (1993), etc.

Prediction Framework

- a. Point predictors
  b. Interval predictors
  c. Predictive distribution
- Abundant Bayesian literature in parametric framework —Cox (1975), Geisser (1993), etc.
- Frequentist/nonparametric literature scarse -- except:

Conformal Prediction in Machine Learning (Vovk, Wasserman, Candes, Chernozhukov, etc.)

I.i.d. set-up

- Let $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. from the (unknown) cdf F_ε

I.i.d. set-up

- Let $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. from the (unknown) cdf F_ε
- GOAL: prediction of future ε_{n+1} based on the data

I.i.d. set-up

- Let $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. from the (unknown) cdf F_ε
- GOAL: prediction of future ε_{n+1} based on the data
- F_ε is the predictive distribution, and its quantiles could be used to form predictive intervals

I.i.d. set-up

- Let $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. from the (unknown) cdf F_ε
- GOAL: prediction of future ε_{n+1} based on the data
- F_ε is the predictive distribution, and its quantiles could be used to form predictive intervals
- The mean and median of F_ε are optimal point predictors under an L_2 and L_1 criterion respectively.

I.i.d. data

- F_ε is unknown but can be estimated by the empirical distribution (edf) $\hat{F}_\varepsilon$.

I.i.d. data

- F_ε is unknown but can be estimated by the empirical distribution (edf) $\hat{F}_\varepsilon$.
- L2 and L1 optimal predictors will be approximated by the mean and median of $\hat{F}_\varepsilon$ respectively. ``Naive" model-free predictive intervals could be based on the quantiles of $\hat{F}_\varepsilon$ but this ignores the variance due to estimation -- need bootstrap!

Non-i.i.d. data

- In general, data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ are not i.i.d.

Non-i.i.d. data

- In general, data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ are not i.i.d.
- So the predictive distribution of Y_{n+1} given the data will depend on $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ which is a matrix of observable, explanatory (predictor) variables.

Non-i.i.d. data

- In general, data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ are not i.i.d.
- So the predictive distribution of Y_{n+1} given the data will depend on $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ which is a matrix of observable, explanatory (predictor) variables.
- **Key Examples:** Regression and Time series

Models

- **Regression:** $Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. (0,1)

Models

- **Regression:** $Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. (0,1)
- **Time series:**
  $Y_t = \mu(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t) + \sigma(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t)\ \varepsilon_t$

Models

- **Regression:** $Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. (0,1)
- **Time series:**
  $Y_t = \mu(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t) + \sigma(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t)\ \varepsilon_t$
- The above are flexible, nonparametric models.

Models

- **Regression:** $Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. (0,1)
- **Time series:**
  $Y_t = \mu(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t) + \sigma(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t)\ \varepsilon_t$
- The above are flexible, nonparametric models.
- Given one of the above models, optimal model-based predictors of a future Y-value can be constructed.

Models

- **Regression:** $Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. (0,1)
- **Time series:**
  $Y_t = \mu(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t) + \sigma(Y_{t-1}, \cdots, Y_{t-p}; \underline{x}_t)\ \varepsilon_t$
- The above are flexible, nonparametric models.
- Given one of the above models, optimal model-based predictors of a future Y-value can be constructed.
- Nevertheless, the prediction problem can be carried out in a fully model-free setting, offering—at the very least—robustness against model mis-specification.

Transformation vs. modeling

- DATA: $\underline{Y}_n = (Y_1, \ldots, Y_n)'$

Transformation vs. modeling

- DATA: $\underline{Y}_n = (Y_1, \ldots, Y_n)'$
- GOAL: predict future value Y_{n+1} given the data

Transformation vs. modeling

- DATA: $\underline{Y}_n = (Y_1, \dots, Y_n)'$
- GOAL: predict future value Y_{n+1} given the data

- Find invertible transformation H_m so that (for all m) the

  vector $\underline{\epsilon}_m = H_m(\underline{Y}_m)$ has i.i.d. components ϵ_k where

  $\underline{\epsilon}_m = (\epsilon_1, \dots, \epsilon_m)'$

Transformation vs. modeling

- DATA: $\underline{Y}_n = (Y_1, \ldots, Y_n)'$
- GOAL: predict future value Y_{n+1} given the data

- Find invertible transformation H_m so that (for all m) the

  vector $\underline{\epsilon}_m = H_m(\underline{Y}_m)$ has i.i.d. components ϵ_k where

  $\underline{\epsilon}_m = (\epsilon_1, \ldots, \epsilon_m)'$

$$\underline{Y} \xrightarrow{H_m} \underline{\epsilon}$$

$$\underline{Y} \xleftarrow{H_m^{-1}} \underline{\epsilon}$$

Transformation

$$(i) \quad (Y_1, \ldots, Y_m) \xrightarrow{H_m} (\epsilon_1, \ldots, \epsilon_m)$$

$$(ii) \quad (Y_1, \ldots, Y_m) \xleftarrow{H_m^{-1}} (\epsilon_1, \ldots, \epsilon_m)$$

- (i) implies that $\epsilon_1, \ldots, \epsilon_n$ are known given the data $Y_1, \ldots, Y_n$

Transformation

$$(i) \quad (Y_1, \ldots, Y_m) \xrightarrow{H_m} (\epsilon_1, \ldots, \epsilon_m)$$

$$(ii) \quad (Y_1, \ldots, Y_m) \xleftarrow{H_m^{-1}} (\epsilon_1, \ldots, \epsilon_m)$$

- (i) implies that $\epsilon_1, \ldots, \epsilon_n$ are known given the data $Y_1, \ldots, Y_n$
- (ii) implies that Y_{n+1} is a function of $\epsilon_1, \ldots, \epsilon_n$, and ϵ_{n+1}

Transformation

$$(i) \quad (Y_1, \ldots, Y_m) \xrightarrow{H_m} (\epsilon_1, \ldots, \epsilon_m)$$

$$(ii) \quad (Y_1, \ldots, Y_m) \xleftarrow{H_m^{-1}} (\epsilon_1, \ldots, \epsilon_m)$$

- (i) implies that $\epsilon_1, \ldots, \epsilon_n$ are known given the data $Y_1, \ldots, Y_n$
- (ii) implies that Y_{n+1} is a function of $\epsilon_1, \ldots, \epsilon_n$, and ϵ_{n+1}
- So, given the data $\underline{Y}_n$, Y_{n+1} is a function of ϵ_{n+1} only, i.e.,

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε
- The mean and median of $\tilde{h}(\epsilon)$ where $\epsilon \sim F_\varepsilon$ are optimal point predictors of Y_{n+1} under L_2 or L_1 criterion

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε
- The mean and median of $\tilde{h}(\epsilon)$ where $\epsilon \sim F_\varepsilon$ are optimal point predictors of Y_{n+1} under L_2 or L_1 criterion
- The whole predictive distribution of Y_{n+1} is the distribution of $\tilde{h}(\epsilon)$ when $\epsilon \sim F_\varepsilon$

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε
- The mean and median of $\tilde{h}(\epsilon)$ where $\epsilon \sim F_\varepsilon$ are optimal point predictors of Y_{n+1} under L_2 or L_1 criterion
- The whole predictive distribution of Y_{n+1} is the distribution of $\tilde{h}(\epsilon)$ when $\epsilon \sim F_\varepsilon$
- To predict Y_{n+1}^2, replace $\tilde{h}$ by $\tilde{h}^2$;
  to predict $g(Y_{n+1})$, replace $\tilde{h}$ by $g \circ \tilde{h}$.

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε
- The mean and median of $\tilde{h}(\epsilon)$ where $\epsilon \sim F_\varepsilon$ are optimal point predictors of Y_{n+1} under L_2 or L_1 criterion
- The whole predictive distribution of Y_{n+1} is the distribution of $\tilde{h}(\epsilon)$ when $\epsilon \sim F_\varepsilon$
- To predict Y_{n+1}^2, replace $\tilde{h}$ by $\tilde{h}^2$;
  to predict $g(Y_{n+1})$, replace $\tilde{h}$ by $g \circ \tilde{h}$.
- The unknown F_ε can be estimated by $\hat{F}_\varepsilon$, the edf of $\epsilon_1, \ldots, \epsilon_n$.

Model-free prediction principle

$$Y_{n+1} = \tilde{h}(\epsilon_{n+1})$$

- Suppose $\epsilon_1, \ldots, \epsilon_n \sim$ cdf F_ε
- The mean and median of $\tilde{h}(\epsilon)$ where $\epsilon \sim F_\varepsilon$ are optimal point predictors of Y_{n+1} under L_2 or L_1 criterion
- The whole predictive distribution of Y_{n+1} is the distribution of $\tilde{h}(\epsilon)$ when $\epsilon \sim F_\varepsilon$
- To predict Y_{n+1}^2, replace $\tilde{h}$ by $\tilde{h}^2$;
  to predict $g(Y_{n+1})$, replace $\tilde{h}$ by $g \circ \tilde{h}$.
- The unknown F_ε can be estimated by $\hat{F}_\varepsilon$, the edf of $\epsilon_1, \ldots, \epsilon_n$.
- But the predictive distribution needs bootstrapping—also because $\tilde{h}$ is estimated from the data.

Nonparametric Regression

$$\textit{MODEL } (\star): \quad Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$$

- x_t univariate and deterministic

Nonparametric Regression

$$\textit{MODEL } (\star): \quad Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$$

- x_t univariate and deterministic
- Y_t data available for $t = 1, \ldots, n$.

Nonparametric Regression

$$\textit{MODEL } (\star): \quad Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$$

- x_t univariate and deterministic
- Y_t data available for $t = 1, \ldots, n$.
- $\varepsilon_t \sim$ i.i.d. (0,1) from (unknown) cdf F

Nonparametric Regression

$$\textit{MODEL } (\star): \quad Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$$

- x_t univariate and deterministic
- Y_t data available for $t = 1, \ldots, n$.
- $\varepsilon_t \sim$ i.i.d. (0,1) from (unknown) cdf F
- the functions $\mu(\cdot)$ and $\sigma(\cdot)$ unknown but smooth

Nonparametric Regression

Note: $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$.

- Let m_x, s_x be smoothing estimators of $\mu(x), \sigma(x)$.

Nonparametric Regression

Note: $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$.

- Let m_x, s_x be smoothing estimators of $\mu(x), \sigma(x)$.
- Examples: kernel smoothers, local linear fitting, wavelets, etc.

Nonparametric Regression

Note: $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$.

- Let m_x, s_x be smoothing estimators of $\mu(x), \sigma(x)$.
- Examples: kernel smoothers, local linear fitting, wavelets, etc.
- E.g. Nadaraya-Watson estimator $m_x = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x-x_i}{h}\right)$

Nonparametric Regression

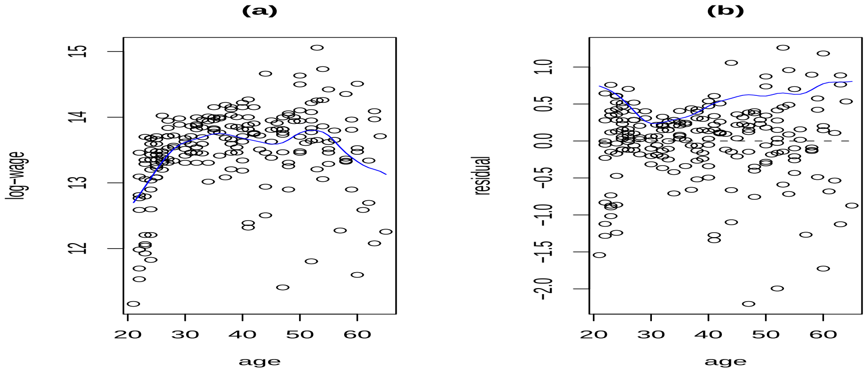
Note: $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$.

- Let m_x, s_x be smoothing estimators of $\mu(x), \sigma(x)$.
- Examples: kernel smoothers, local linear fitting, wavelets, etc.
- E.g. Nadaraya-Watson estimator $m_x = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x-x_i}{h}\right)$
- here $K(x)$ is the kernel, h the bandwidth, and $\tilde{K}\left(\frac{x-x_i}{h}\right) = K\left(\frac{x-x_i}{h}\right) / \sum_{k=1}^{n} K\left(\frac{x-x_k}{h}\right)$.

Nonparametric Regression

Note: $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$.

- Let m_x, s_x be smoothing estimators of $\mu(x), \sigma(x)$.
- Examples: kernel smoothers, local linear fitting, wavelets, etc.
- E.g. Nadaraya-Watson estimator $m_x = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x-x_i}{h}\right)$
- here $K(x)$ is the kernel, h the bandwidth, and $\tilde{K}\left(\frac{x-x_i}{h}\right) = K\left(\frac{x-x_i}{h}\right) / \sum_{k=1}^{n} K\left(\frac{x-x_k}{h}\right)$.
- Similarly, $s_x^2 = M_x - m_x^2$ where $M_x = \sum_{i=1}^{n} Y_i^2 \tilde{K}\left(\frac{x-x_i}{q}\right)$



(a) Log-wage vs. age data with fitted kernel smoother m_x.
(b) Unstudentized residuals $Y - m_x$ with superimposed s_x.

- 1971 Canadian Census data cps71 from `np` package of R; wage vs. age dataset of 205 male individuals with common education.
- Kernel smoother problematic at the left boundary; local linear is better (Fan and Gijbels, 1996) or reflection (Hall and Wehrly, 1991).

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$

Residuals

- (⋆): $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$
- predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$
- predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$
- $m_x^{(t)}$ and $s_{x_t}^{(t)}$ are the estimators m and s computed from the delete-Y_t dataset: $\{(Y_i, x_i)$, for all $i \neq t\}$.

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$
- predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$
- $m_x^{(t)}$ and $s_{x_t}^{(t)}$ are the estimators m and s computed from the delete-Y_t dataset: $\{(Y_i, x_i)$, for all $i \neq t\}$.
- $\tilde{e}_t$ is the (standardized) error in trying to predict Y_t from the delete-Y_t dataset.

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$
- predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$
- $m_x^{(t)}$ and $s_{x_t}^{(t)}$ are the estimators m and s computed from the delete-Y_t dataset: $\{(Y_i, x_i), \text{ for all } i \neq t\}$.
- $\tilde{e}_t$ is the (standardized) error in trying to predict Y_t from the delete-Y_t dataset.
- Selection of bandwidth parameters h and q is often done by cross-validation, i.e., pick h, q to minimize PRESS$=\sum_{t=1}^{n} \tilde{e}_t^2$.

Residuals

- $(\star)$: $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$
- fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$
- predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$
- $m_x^{(t)}$ and $s_{x_t}^{(t)}$ are the estimators m and s computed from the delete-Y_t dataset: $\{(Y_i, x_i)$, for all $i \neq t\}$.
- $\tilde{e}_t$ is the (standardized) error in trying to predict Y_t from the delete-Y_t dataset.
- Selection of bandwidth parameters h and q is often done by cross-validation, i.e., pick h, q to minimize PRESS$=\sum_{t=1}^{n} \tilde{e}_t^2$.
- BETTER: L_1 cross-validation: pick h, q to minimize $\sum_{t=1}^{n} |\tilde{e}_t|$.

Model-based (MB) point predictors

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- GOAL: Predict a future response Y_f associated with point x_f.

Model-based (MB) point predictors

($\star$) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- GOAL: Predict a future response Y_{f} associated with point x_{f}.
- L_2–optimal predictor of Y_{f} is $E(Y_{\mathrm{f}}|x_{\mathrm{f}})$, i.e., $\mu(x_{\mathrm{f}})$

Model-based (MB) point predictors

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- GOAL: Predict a future response Y_f associated with point x_f.
- L_2–optimal predictor of Y_f is $E(Y_\mathrm{f}|x_\mathrm{f})$, i.e., $\mu(x_\mathrm{f})$ which is approximated by m_{x_f}.

Model-based (MB) point predictors

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- GOAL: Predict a future response Y_f associated with point x_f.
- L_2–optimal predictor of Y_f is $E(Y_\mathrm{f}|x_\mathrm{f})$, i.e., $\mu(x_\mathrm{f})$ which is approximated by m_{x_f}.
- L_1–optimal predictor of Y_f is the conditional median, i.e., $\mu(x_\mathrm{f}) + \sigma(x_\mathrm{f}) \cdot median(F)$

Model-based (MB) point predictors

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- GOAL: Predict a future response Y_f associated with point x_f.
- L_2–optimal predictor of Y_f is $E(Y_\mathrm{f}|x_\mathrm{f})$, i.e., $\mu(x_\mathrm{f})$
  which is approximated by m_{x_f}.
- L_1–optimal predictor of Y_f is the conditional median, i.e.,
  $\mu(x_\mathrm{f}) + \sigma(x_\mathrm{f}) \cdot median(F)$
  which is approximated by $m_{x_\mathrm{f}} + s_{x_\mathrm{f}} \cdot median(\hat{F}_e)$
  where $\hat{F}_e$ is the edf of the (fitted) residuals $e_1, \ldots, e_n$

Model-based (MB) point predictors 2

($\star$) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- DATASET cps71: salaries are logarithmically transformed, i.e., Y_t= log-salary.

Model-based (MB) point predictors 2

($\star$) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- DATASET cps71: salaries are logarithmically transformed, i.e., Y_t= log-salary.
- To predict salary at age x_{f} we need to predict $g(Y_{\mathrm{f}})$ where $g(x) = \exp(x)$.

Model-based (MB) point predictors 2

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- DATASET cps71: salaries are logarithmically transformed, i.e., Y_t= log-salary.
- To predict salary at age x_{f} we need to predict $g(Y_{\mathrm{f}})$ where $g(x) = \exp(x)$.
- MB L_2–optimal predictor of $g(Y_{\mathrm{f}})$ is $E(g(Y_{\mathrm{f}})|x_{\mathrm{f}})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right)$.

Model-based (MB) point predictors 2

($\star$) $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- DATASET cps71: salaries are logarithmically transformed, i.e., Y_t= log-salary.
- To predict salary at age x_{f} we need to predict $g(Y_{\mathrm{f}})$ where $g(x) = \exp(x)$.
- MB L_2–optimal predictor of $g(Y_{\mathrm{f}})$ is $E(g(Y_{\mathrm{f}})|x_{\mathrm{f}})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right)$.
- Naive predictor $g(m_{x_{\mathrm{f}}})$ is suboptimal when g is nonlinear.

Model-based (MB) point predictors 2

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- DATASET cps71: salaries are logarithmically transformed, i.e., Y_t= log-salary.
- To predict salary at age x_f we need to predict $g(Y_\text{f})$ where $g(x) = \exp(x)$.
- MB L_2–optimal predictor of $g(Y_\text{f})$ is $E(g(Y_\text{f})|x_\text{f})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_\text{f}} + \sigma_{x_\text{f}} e_i\right)$.
- Naive predictor $g(m_{x_\text{f}})$ is suboptimal when g is nonlinear.
- MB L_1–optimal predictor of $g(Y_\text{f})$ estimated by the sample median of the set $\{g\left(m_{x_\text{f}} + \sigma_{x_\text{f}} e_i\right),\ i = 1, ..., n\}$; naive plug-in ok iff g is monotone!

Which residuals to use?

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- MB L_2–optimal predictor of $g(Y_{\mathrm{f}})$ is $E(g(Y_{\mathrm{f}})|x_{\mathrm{f}})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right)$.
- MB L_1–optimal predictor of $g(Y_{\mathrm{f}})$ estimated by the sample median of the set $\{g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right),\ i = 1, ..., n\}$.

Which residuals to use?

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- MB L_2–optimal predictor of $g(Y_\mathrm{f})$ is $E(g(Y_\mathrm{f})|x_\mathrm{f})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_\mathrm{f}} + \sigma_{x_\mathrm{f}} e_i\right)$.
- MB L_1–optimal predictor of $g(Y_\mathrm{f})$ estimated by the sample median of the set $\{g\left(m_{x_\mathrm{f}} + \sigma_{x_\mathrm{f}} e_i\right),\ i = 1, ..., n\}$.
- Traditionally, the above are calculated using the fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$.

Which residuals to use?

($\star$) $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- MB L_2–optimal predictor of $g(Y_{\mathrm{f}})$ is $E(g(Y_{\mathrm{f}})|x_{\mathrm{f}})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right)$.
- MB L_1–optimal predictor of $g(Y_{\mathrm{f}})$ estimated by the sample median of the set $\{g\left(m_{x_{\mathrm{f}}} + \sigma_{x_{\mathrm{f}}} e_i\right),\ i = 1, ..., n\}$.
- Traditionally, the above are calculated using the fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$.
- MF Prediction Principle suggests the transformation $\underline{Y} \mapsto \underline{\tilde{e}}$.
- $\underline{\tilde{e}}$ is vector of predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$.

Which residuals to use?

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- MB L_2–optimal predictor of $g(Y_{\rm f})$ is $E(g(Y_{\rm f})|x_{\rm f})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_{\rm f}} + \sigma_{x_{\rm f}} e_i\right)$.
- MB L_1–optimal predictor of $g(Y_{\rm f})$ estimated by the sample median of the set $\{g\left(m_{x_{\rm f}} + \sigma_{x_{\rm f}} e_i\right),\ i = 1, ..., n\}$.
- Traditionally, the above are calculated using the fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$.
- MF Prediction Principle suggests the transformation $\underline{Y} \mapsto \underline{\tilde{e}}$.
- $\underline{\tilde{e}}$ is vector of predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$.
- e_t and $\tilde{e}_t$ are centered at zero but different scale: $|e_t| < |\tilde{e}_t|$.

Which residuals to use?

(⋆) $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- MB L_2–optimal predictor of $g(Y_\mathrm{f})$ is $E(g(Y_\mathrm{f})|x_\mathrm{f})$ estimated by $n^{-1}\sum_{i=1}^{n} g\left(m_{x_\mathrm{f}} + \sigma_{x_\mathrm{f}} e_i\right)$.
- MB L_1–optimal predictor of $g(Y_\mathrm{f})$ estimated by the sample median of the set $\{g\left(m_{x_\mathrm{f}} + \sigma_{x_\mathrm{f}} e_i\right),\ i = 1, ..., n\}$.
- Traditionally, the above are calculated using the fitted residuals: $e_t = (Y_t - m_{x_t})/s_{x_t}$.
- MF Prediction Principle suggests the transformation $\underline{Y} \mapsto \underline{\tilde{e}}$.
- $\underline{\tilde{e}}$ is vector of predictive residuals: $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$.
- e_t and $\tilde{e}_t$ are centered at zero but different scale: $|e_t| < |\tilde{e}_t|$.
- Makes little difference for point predictors but huge difference for prediction intervals: MF/MB alleviates undercoverage.

Model-based bootstrap for predictive distribution of $g(Y_{\mathrm{f}})$

Prediction root: $g(Y_{\mathrm{f}}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.

Model-based bootstrap for predictive distribution of $g(Y_{\mathrm{f}})$

Prediction root: $g(Y_{\mathrm{f}}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, ...n$.

Model-based bootstrap for predictive distribution of $g(Y_\mathrm{f})$

Prediction root: $g(Y_\mathrm{f}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, ...n$.
- Calculate a bootstrap pseudo-response $Y_\mathrm{f}^\star = m_{x_\mathrm{f}} + s_{x_\mathrm{f}} r$ where r is drawn randomly from $(r_1, ..., r_n)$.

Model-based bootstrap for predictive distribution of $g(Y_\mathrm{f})$

Prediction root: $g(Y_\mathrm{f}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, ...n$.
- Calculate a bootstrap pseudo-response $Y_\mathrm{f}^\star = m_{x_\mathrm{f}} + s_{x_\mathrm{f}} r$ where r is drawn randomly from $(r_1, ..., r_n)$.
- Based on the pseudo-data $Y_1^\star, ..., Y_n^\star$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by $m_x^\star$ and $s_x^\star$.

Model-based bootstrap for predictive distribution of $g(Y_{\mathrm{f}})$

Prediction root: $g(Y_{\mathrm{f}}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, \ldots, r_n$ to create pseudo-residuals $r_1^\star, \ldots, r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, \ldots n$.
- Calculate a bootstrap pseudo-response $Y_{\mathrm{f}}^\star = m_{x_{\mathrm{f}}} + s_{x_{\mathrm{f}}} r$ where r is drawn randomly from $(r_1, \ldots, r_n)$.
- Based on the pseudo-data $Y_1^\star, \ldots, Y_n^\star$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by $m_x^\star$ and $s_x^\star$.
- Calculate bootstrap root: $g(Y_{\mathrm{f}}^\star) - \Pi(g, m_x^*, s_x^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.

Model-based bootstrap for predictive distribution of $g(Y_{\mathrm{f}})$

Prediction root: $g(Y_{\mathrm{f}}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, ...n$.
- Calculate a bootstrap pseudo-response $Y_{\mathrm{f}}^\star = m_{x_{\mathrm{f}}} + s_{x_{\mathrm{f}}} r$ where r is drawn randomly from $(r_1, ..., r_n)$.
- Based on the pseudo-data $Y_1^\star, ..., Y_n^\star$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by $m_x^\star$ and $s_x^\star$.
- Calculate bootstrap root: $g(Y_{\mathrm{f}}^\star) - \Pi(g, m_x^*, s_x^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.
- Repeat the above B times, and collect the B bootstrap roots in an empirical distribution with α—quantile denoted $q(\alpha)$.

Model-based bootstrap for predictive distribution of $g(Y_\mathrm{f})$

Prediction root: $g(Y_\mathrm{f}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, ..., r_n$ to create pseudo-residuals $r_1^\star, ..., r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, ...n$.
- Calculate a bootstrap pseudo-response $Y_\mathrm{f}^\star = m_{x_\mathrm{f}} + s_{x_\mathrm{f}} r$ where r is drawn randomly from $(r_1, ..., r_n)$.
- Based on the pseudo-data $Y_1^\star, ..., Y_n^\star$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by $m_x^\star$ and $s_x^\star$.
- Calculate bootstrap root: $g(Y_\mathrm{f}^\star) - \Pi(g, m_x^*, s_x^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.
- Repeat the above B times, and collect the B bootstrap roots in an empirical distribution with α—quantile denoted $q(\alpha)$.
- Our estimate of the predictive distribution of $g(Y_\mathrm{f})$ is the empirical df of bootstrap roots shifted to the right by Π.

Model-based bootstrap for predictive distribution of $g(Y_{\mathrm{f}})$

Prediction root: $g(Y_{\mathrm{f}}) - \Pi$ where Π is the point predictor.

- Bootstrap the (fitted or predictive) residuals $r_1, \ldots, r_n$ to create pseudo-residuals $r_1^\star, \ldots, r_n^\star$ whose edf is denoted by $\hat{F}_n^\star$.
- Create pseudo-data $Y_i^\star = m_{x_i} + s_{x_i} r_i^\star$, for $i = 1, \ldots n$.
- Calculate a bootstrap pseudo-response $Y_{\mathrm{f}}^\star = m_{x_{\mathrm{f}}} + s_{x_{\mathrm{f}}} r$ where r is drawn randomly from $(r_1, \ldots, r_n)$.
- Based on the pseudo-data $Y_1^\star, \ldots, Y_n^\star$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by $m_x^\star$ and $s_x^\star$.
- Calculate bootstrap root: $g(Y_{\mathrm{f}}^\star) - \Pi(g, m_x^*, s_x^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.
- Repeat the above B times, and collect the B bootstrap roots in an empirical distribution with α—quantile denoted $q(\alpha)$.
- Our estimate of the predictive distribution of $g(Y_{\mathrm{f}})$ is the empirical df of bootstrap roots shifted to the right by Π.
- Then, a $(1-\alpha)100\%$ equal-tailed predictive interval for $g(Y_{\mathrm{f}})$ is given by: $[\Pi + q(\alpha/2), \Pi + q(1-\alpha/2)]$.

Model-free prediction in regression

Previous discussion hinged on model: $(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ from cdf F.

- What happens if model $(\star)$ does not hold true?

Model-free prediction in regression

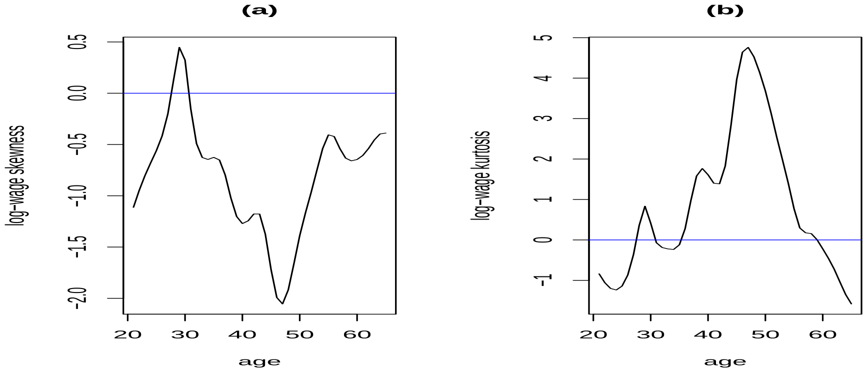
Previous discussion hinged on model: $(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ from cdf F.

- What happens if model $(\star)$ does not hold true?
- E.g., the skewness and/or kurtosis of Y_t may depend on x_t.

Model-free prediction in regression

Previous discussion hinged on model: $(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ from cdf F.

- What happens if model $(\star)$ does not hold true?
- E.g., the skewness and/or kurtosis of Y_t may depend on x_t.
- cps71 data: skewness/kurtosis of salary depend on age.



(a) Log-wage SKEWNESS vs. age.
(b) Log-wage KURTOSIS vs. age.

- Both skewness and kurtosis are nonconstant!

General background-

- Could try skewness reducing transformations—but log already does that.

General background-

- Could try skewness reducing transformations—but log already does that.
- Could try ACE, AVAS, etc.

General background-

- Could try skewness reducing transformations—but log already does that.
- Could try ACE, AVAS, etc.
- There is a simpler, more general solution!

General background

- The Y_ts are still independent but not identically distributed.

General background

- The Y_ts are still independent but not identically distributed.
- We will denote the conditional distribution of Y_{f} given x_{f} by $D_x(y) = P\{Y_{\mathrm{f}} \leq y | x_{\mathrm{f}} = x\}$

General background

- The Y_ts are still independent but not identically distributed.
- We will denote the conditional distribution of Y_{f} given x_{f} by $D_x(y) = P\{Y_{\mathrm{f}} \leq y | x_{\mathrm{f}} = x\}$
- Assume the quantity $D_x(y)$ is continuous in both x and y.

General background

- The Y_ts are still independent but not identically distributed.
- We will denote the conditional distribution of Y_f given x_f by $D_x(y) = P\{Y_\mathrm{f} \leq y | x_\mathrm{f} = x\}$
- Assume the quantity $D_x(y)$ is continuous in both x and y.
- With a categorical response, standard methods like Generalized Linear Models can be invoked, e.g. logistic regression, Poisson regression, etc.

General background

- The Y_ts are still independent but not identically distributed.
- We will denote the conditional distribution of Y_f given x_f by $D_x(y) = P\{Y_\mathrm{f} \leq y | x_\mathrm{f} = x\}$
- Assume the quantity $D_x(y)$ is continuous in both x and y.
- With a categorical response, standard methods like Generalized Linear Models can be invoked, e.g. logistic regression, Poisson regression, etc.
- Since $D_x(\cdot)$ depends in a smooth way on x, we can estimate $D_x(y)$ by the 'local' empirical $N_{x,h}^{-1} \sum_{t:|x_t - x| < h/2} \mathbf{1}\{Y_t \leq y\}$ where $\mathbf{1}\{\cdot\}$ is indicator, and $N_{x,h}$ is the number of summands, i.e., $N_{x,h} = \# \{t : |x_t - x| < h/2\}$.

Constructing the transformation

- More general estimator $\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\}\tilde{K}\left(\frac{x-x_i}{h}\right)$.

Constructing the transformation

- More general estimator $\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\} \tilde{K}\left(\frac{x-x_i}{h}\right)$.
- $\hat{D}_x(y)$ is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_t \leq y\}, \; t = 1, \ldots, n$.

Constructing the transformation

- More general estimator $\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\}\tilde{K}\left(\frac{x-x_i}{h}\right)$.
- $\hat{D}_x(y)$ is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$.
- Can use local linear smoother of $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$ but ensure the result is a proper c.d.f.–see e.g. Hansen (2004).

Constructing the transformation

- More general estimator $\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\}\tilde{K}\left(\frac{x-x_i}{h}\right)$.
- $\hat{D}_x(y)$ is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$.
- Can use local linear smoother of $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$ but ensure the result is a proper c.d.f.–see e.g. Hansen (2004).
- Estimator $\hat{D}_x(y)$ enjoys many good properties including asymptotic consistency; see e.g. Li and Racine (2007).

Constructing the transformation

- More general estimator $\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\}\tilde{K}\left(\frac{x-x_i}{h}\right)$.
- $\hat{D}_x(y)$ is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$.
- Can use local linear smoother of $\mathbf{1}\{Y_t \leq y\},\ t = 1, \ldots, n$ but ensure the result is a proper c.d.f.–see e.g. Hansen (2004).
- Estimator $\hat{D}_x(y)$ enjoys many good properties including asymptotic consistency; see e.g. Li and Racine (2007).
- But $\hat{D}_x(y)$ is discontinuous in y, and therefore unacceptable!
- Could use linear interpolation or smooth it by kernel methods, i.e., $\tilde{D}_x(y) = \sum_{i=1}^{n} \Lambda\left(\frac{y-Y_i}{h_0}\right)\tilde{K}\left(\frac{x-x_i}{h}\right)$ where $h_0 \sim h^2$.

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.
- To see why, note that if we let $\eta_i = D_{x_i}(Y_i)$ for $i = 1, \ldots, n$ our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform(0,1).

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.
- To see why, note that if we let $\eta_i = D_{x_i}(Y_i)$ for $i = 1, \ldots, n$ our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform(0,1).
- $D_x(\cdot)$ not known but we have estimator $\tilde{D}_x(\cdot)$ as its proxy.

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.
- To see why, note that if we let $\eta_i = D_{x_i}(Y_i)$ for $i = 1, \ldots, n$ our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform(0,1).
- $D_x(\cdot)$ not known but we have estimator $\tilde{D}_x(\cdot)$ as its proxy.
- Therefore, our proposed transformation for the MF prediction principle is $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.
- To see why, note that if we let $\eta_i = D_{x_i}(Y_i)$ for $i = 1, \ldots, n$ our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform(0,1).
- $D_x(\cdot)$ not known but we have estimator $\tilde{D}_x(\cdot)$ as its proxy.
- Therefore, our proposed transformation for the MF prediction principle is $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- $\tilde{D}_x(\cdot)$ is consistent, so $u_1, \ldots, u_n$ are approximately i.i.d.

Constructing the transformation–

- Since the Y_ts are continuous r.v.'s, the probability integral transform is the key idea to transform them to 'i.i.d.–ness'.
- To see why, note that if we let $\eta_i = D_{x_i}(Y_i)$ for $i = 1, \ldots, n$ our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform(0,1).
- $D_x(\cdot)$ not known but we have estimator $\tilde{D}_x(\cdot)$ as its proxy.
- Therefore, our proposed transformation for the MF prediction principle is $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- $\tilde{D}_x(\cdot)$ is consistent, so $u_1, \ldots, u_n$ are approximately i.i.d.
- The probability integral transform was used in the past for building better density estimators—Ruppert and Cline (1994).

Model-free optimal predictors

- Transformation: $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.

Model-free optimal predictors

- Transformation: $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- Inverse transformation $\tilde{D}_x^{-1}$ is well-defined since $\tilde{D}_x(\cdot)$ is strictly increasing.

Model-free optimal predictors

- Transformation: $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- Inverse transformation $\tilde{D}_x^{-1}$ is well-defined since $\tilde{D}_x(\cdot)$ is strictly increasing.
- Let $u_\mathrm{f} = D_{x_\mathrm{f}}(Y_\mathrm{f})$ and $Y_\mathrm{f} = D_{x_\mathrm{f}}^{-1}(u_\mathrm{f})$.

Model-free optimal predictors

- Transformation: $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- Inverse transformation $\tilde{D}_x^{-1}$ is well-defined since $\tilde{D}_x(\cdot)$ is strictly increasing.
- Let $u_\mathrm{f} = D_{x_\mathrm{f}}(Y_\mathrm{f})$ and $Y_\mathrm{f} = D_{x_\mathrm{f}}^{-1}(u_\mathrm{f})$.
- $\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)$ has (approximately) the same distribution as Y_f (conditionally on x_f) for any i.

Model-free optimal predictors

- Transformation: $u_i = \tilde{D}_{x_i}(Y_i)$ for $i = 1, \ldots, n$.
- Inverse transformation $\tilde{D}_x^{-1}$ is well-defined since $\tilde{D}_x(\cdot)$ is strictly increasing.
- Let $u_\mathrm{f} = D_{x_\mathrm{f}}(Y_\mathrm{f})$ and $Y_\mathrm{f} = D_{x_\mathrm{f}}^{-1}(u_\mathrm{f})$.
- $\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)$ has (approximately) the same distribution as Y_f (conditionally on x_f) for any i.
- So $\{\tilde{D}_{x_\mathrm{f}}^{-1}(u_i),\ i = 1, ..., n\}$ is a set of bona fide potential responses that can be used as proxies for Y_f.

- These n valid potential responses $\{\tilde{D}_{x_{\mathrm{f}}}^{-1}(u_i),\ i = 1, \ldots, n\}$ gathered together give an approximate empirical distribution for Y_{f} from which our predictors will be derived.

- These n valid potential responses $\{\tilde{D}_{x_\mathrm{f}}^{-1}(u_i),\ i = 1, ..., n\}$ gathered together give an approximate empirical distribution for Y_f from which our predictors will be derived.
- The L_2—optimal predictor of $g(Y_\mathrm{f})$ will be the expected value of $g(Y_\mathrm{f})$ that is approximated by $n^{-1} \sum_{i=1}^{n} g\left(\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)\right)$.

- These n valid potential responses $\{\tilde{D}_{x_\mathrm{f}}^{-1}(u_i),\ i = 1, ..., n\}$ gathered together give an approximate empirical distribution for Y_f from which our predictors will be derived.
- The L_2—optimal predictor of $g(Y_\mathrm{f})$ will be the expected value of $g(Y_\mathrm{f})$ that is approximated by $n^{-1} \sum_{i=1}^{n} g\left(\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)\right)$.
- The L_1—optimal predictor of $g(Y_\mathrm{f})$ will be approximated by the sample median of the set $\{g\left(\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)\right),\ i = 1, ..., n\}$.

Model-free optimal point predictors

	Model-free method
L_2—predictor of $Y_{\rm f}$	$n^{-1}\sum_{i=1}^{n} \tilde{D}_{x_{\rm f}}^{-1}(u_i)$
L_1—predictor of $Y_{\rm f}$	median$\{\tilde{D}_{x_{\rm f}}^{-1}(u_i)\}$
L_2—predictor of $g(Y_{\rm f})$	$n^{-1}\sum_{i=1}^{n} g\left(\tilde{D}_{x_{\rm f}}^{-1}(u_i)\right)$
L_1—predictor of $g(Y_{\rm f})$	median$\{g\left(\tilde{D}_{x_{\rm f}}^{-1}(u_i)\right)\}$

TABLE. Model-free (MF) and Limit Model-free (LMF) predictors.
Basic MF: $u_i = \tilde{D}_{x_i}(Y_i)$
Limit MF: $u_i \sim$i.i.d. Uniform$(0,1)$.

Model-free model-fitting

- The MF predictors (mean or median) can be used to give the equivalent of a model fit.

Model-free model-fitting

- The MF predictors (mean or median) can be used to give the equivalent of a model fit.
- Focus on the L_2—optimal case with $g(x) = x$.

Model-free model-fitting

- The MF predictors (mean or median) can be used to give the equivalent of a model fit.
- Focus on the L_2—optimal case with $g(x) = x$.
- Calculating the MF predictor $\Pi(x_\mathrm{f}) = n^{-1} \sum_{i=1}^{n} g\left(\tilde{D}_{x_\mathrm{f}}^{-1}(u_i)\right)$ for many different x_f values—say on a grid—, the equivalent of a nonparametric smoother of a regression function is constructed—Model-Free Model-Fitting.

M.o.a.T.

- MF relieves the practitioner from the need to find the optimal transformation for additivity and variance stabilization such as Box/Cox, ACE, AVAS, etc.—see Figures 3 and 4.

M.o.a.T.

- MF relieves the practitioner from the need to find the optimal transformation for additivity and variance stabilization such as Box/Cox, ACE, AVAS, etc.—see Figures 3 and 4.
- No need for log-transformation of salaries!

M.o.a.T.

- MF relieves the practitioner from the need to find the optimal transformation for additivity and variance stabilization such as Box/Cox, ACE, AVAS, etc.—see Figures 3 and 4.
- No need for log-transformation of salaries!
- MF is totally automatic!!

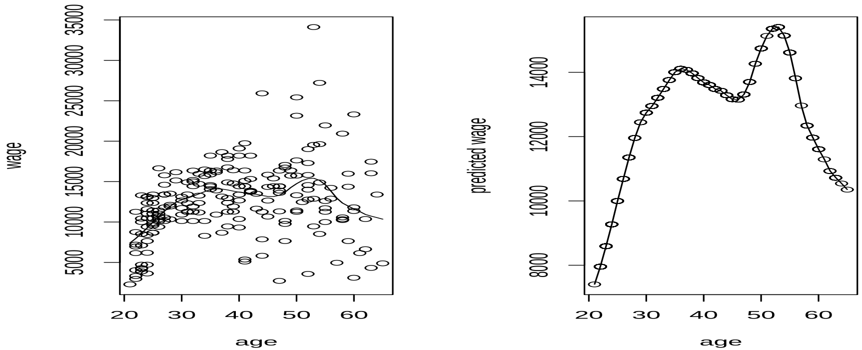


FIGURE 3: (a) Wage vs. age scatterplot. (b) Circles indicate the salary predictor $n^{-1}\sum_{i=1}^{n} g\left(\tilde{D}_{x_f}^{-1}(u_i)\right)$ calculated from log-wage data with $g(x)$ exponential. For both figures, the superimposed solid line represents the MF salary predictor calculated from the raw data (without log).

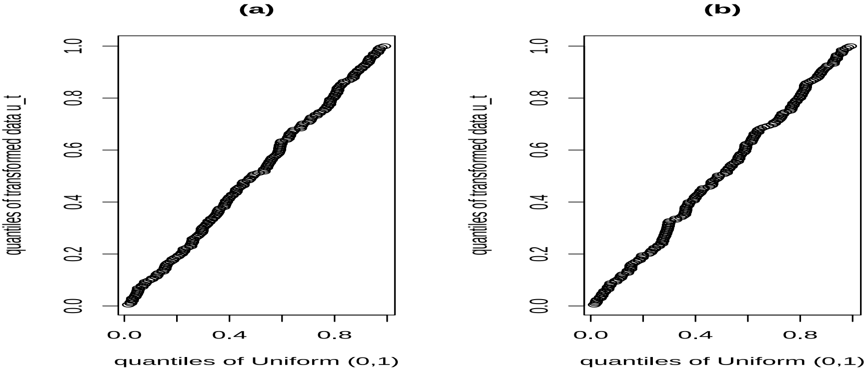


FIGURE 4: Q-Q plots of the u_i vs. the quantiles of Uniform (0,1).
(a) The u_i's are obtained from the log-wage vs. age dataset of Figure 1 using bandwidth 5.5; (b) The u_i's are obtained from the raw (untransformed) dataset of Figure 3 using bandwidth 7.3.

MF predictive distributions

- For MF we can always take $g(x) = x$; no need for other preliminary transformations.

MF predictive distributions

- For MF we can always take $g(x) = x$; no need for other preliminary transformations.
- Let $g(Y_{\mathrm{f}}) - \Pi$ be the prediction root where Π is either the L_2– or L_1–optimal predictor, i.e., $\Pi = n^{-1} \sum_{i=1}^{n} g\left(\tilde{D}_{x_{\mathrm{f}}}^{-1}(u_i)\right)$ or $\Pi = \text{median}\left\{g\left(\tilde{D}_{x_{\mathrm{f}}}^{-1}(u_i)\right)\right\}$.

MF predictive distributions

- For MF we can always take $g(x) = x$; no need for other preliminary transformations.
- Let $g(Y_{\mathrm{f}}) - \Pi$ be the prediction root where Π is either the L_2– or L_1–optimal predictor, i.e., $\Pi = n^{-1} \sum_{i=1}^{n} g\left(\tilde{D}_{x_{\mathrm{f}}}^{-1}(u_i)\right)$ or $\Pi = \text{median}\ \{g\left(\tilde{D}_{x_{\mathrm{f}}}^{-1}(u_i)\right)\}$.
- Based on the Y–data, estimate the conditional distribution $D_x(\cdot)$ by $\tilde{D}_x(\cdot)$, and let $u_i = \tilde{D}_{x_i}(Y_i)$ to obtain the transformed data $u_1, \ldots, u_n$ that are approximately i.i.d.

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, \ldots, u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, \ldots, u_n$); alternatively, let $u_1^*, \ldots, u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, \ldots, u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, \ldots, u_n$); alternatively, let $u_1^*, \ldots, u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, \ldots n$.

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, ..., u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, ..., u_n$); alternatively, let $u_1^*, ..., u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, ...n$.
- Generate a bootstrap pseudo-response $Y_{\mathrm{f}}^* = \tilde{D}_{x_{\mathrm{f}}}^{-1}(u)$ with u drawn randomly from set $(u_1, ..., u_n)$—or from Uniform(0,1).

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, ..., u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, ..., u_n$); alternatively, let $u_1^*, ..., u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, ...n$.
- Generate a bootstrap pseudo-response $Y_{\mathrm{f}}^* = \tilde{D}_{x_{\mathrm{f}}}^{-1}(u)$ with u drawn randomly from set $(u_1, ..., u_n)$—or from Uniform(0,1).
- Based on the pseudo-data $Y_t^\star$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimator by $\tilde{D}_x^*(\cdot)$.

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, ..., u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, ..., u_n$); alternatively, let $u_1^*, ..., u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, ...n$.
- Generate a bootstrap pseudo-response $Y_{\mathrm{f}}^* = \tilde{D}_{x_{\mathrm{f}}}^{-1}(u)$ with u drawn randomly from set $(u_1, ..., u_n)$—or from Uniform(0,1).
- Based on the pseudo-data $Y_t^{\star}$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimator by $\tilde{D}_x^*(\cdot)$.
- Calculate the bootstrap root $g(Y_{\mathrm{f}}^*) - \Pi^*$ where $\Pi^* = n^{-1} \sum_{i=1}^n g\left(\tilde{D}_{x_{\mathrm{f}}}^{*\,-1}(u_i^*)\right)$ or Π^* =median $\{g\left(\tilde{D}_{x_{\mathrm{f}}}^{*\,-1}(u_i^*)\right)\}$

MF bootstrap predictive distribution of $g(Y_{\mathrm{f}})$

- Let $u_1^*, ..., u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, ..., u_n$); alternatively, let $u_1^*, ..., u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, ...n$.
- Generate a bootstrap pseudo-response $Y_{\mathrm{f}}^* = \tilde{D}_{x_{\mathrm{f}}}^{-1}(u)$ with u drawn randomly from set $(u_1, ..., u_n)$—or from Uniform(0,1).
- Based on the pseudo-data $Y_t^\star$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimator by $\tilde{D}_x^*(\cdot)$.
- Calculate the bootstrap root $g(Y_{\mathrm{f}}^*) - \Pi^*$ where $\Pi^* = n^{-1} \sum_{i=1}^n g\left(\tilde{D}_{x_{\mathrm{f}}}^{*-1}(u_i^*)\right)$ or $\Pi^* =$median $\{g\left(\tilde{D}_{x_{\mathrm{f}}}^{*-1}(u_i^*)\right)\}$
- Repeat the above steps B times, and collect the B bootstrap roots in the form of an e.d.f. with α—quantile denoted $q(\alpha)$.

MF bootstrap predictive distribution of $g(Y_{\rm f})$

- Let $u_1^*, ..., u_n^* \sim$i.i.d.$\hat{F}_n$ (the e.d.f. of $u_1, ..., u_n$); alternatively, let $u_1^*, ..., u_n^* \sim$i.i.d. Uniform(0,1)—LMF version.
- Use the inverse transformation $\tilde{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., $Y_t^* = \tilde{D}_{x_t}^{-1}(u_t^*)$ for $t = 1, ...n$.
- Generate a bootstrap pseudo-response $Y_{\rm f}^* = \tilde{D}_{x_{\rm f}}^{-1}(u)$ with u drawn randomly from set $(u_1, ..., u_n)$—or from Uniform(0,1).
- Based on the pseudo-data $Y_t^\star$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimator by $\tilde{D}_x^*(\cdot)$.
- Calculate the bootstrap root $g(Y_{\rm f}^*) - \Pi^*$ where $\Pi^* = n^{-1}\sum_{i=1}^n g\left(\tilde{D}_{x_{\rm f}}^{*^{-1}}(u_i^*)\right)$ or $\Pi^* =$median $\{g\left(\tilde{D}_{x_{\rm f}}^{*^{-1}}(u_i^*)\right)\}$
- Repeat the above steps B times, and collect the B bootstrap roots in the form of an e.d.f. with α—quantile denoted $q(\alpha)$.
- Predictive distribution of $g(Y_{\rm f})$ is the above edf shifted to the right by Π, and MF/LMF $(1-\alpha)100\%$ equal-tailed, prediction interval for $g(Y_{\rm f})$ is $[\Pi + q(\alpha/2), \Pi + q(1-\alpha/2)]$.

Simulation: regression under model $(\star)$

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\ \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- Design points $x_1, \ldots, x_n$ for $n = 100$ equi-spaced on $(0, 2\pi)$

Simulation: regression under model $(\star)$

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- Design points $x_1, \ldots, x_n$ for $n = 100$ equi-spaced on $(0, 2\pi)$
- $\mu(x) = \sin(x)$, $\sigma(x) = 1/2$ and errors N(0,1) or Laplace.

Simulation: regression under model $(\star)$

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0,1)$ with cdf F.

- Design points $x_1, \ldots, x_n$ for $n = 100$ equi-spaced on $(0, 2\pi)$
- $\mu(x) = \sin(x)$, $\sigma(x) = 1/2$ and errors N(0,1) or Laplace.
- Prediction points: $x_{\mathrm{f}} = \pi$; $\mu(x)$ has high slope but zero curvature—easy case for estimation.

Simulation: regression under model $(\star)$

$(\star)$ $Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t$ with $\varepsilon_t \sim$ i.i.d. $(0, 1)$ with cdf F.

- Design points $x_1, \ldots, x_n$ for $n = 100$ equi-spaced on $(0, 2\pi)$
- $\mu(x) = \sin(x)$, $\sigma(x) = 1/2$ and errors N(0,1) or Laplace.
- Prediction points: $x_\mathrm{f} = \pi$; $\mu(x)$ has high slope but zero curvature—easy case for estimation.
- $x_\mathrm{f} = \pi/2$ and $x_\mathrm{f} = 3\pi/2$; $\mu(x)$ has zero slope but high curvature—peak and valley so large bias of m_x.

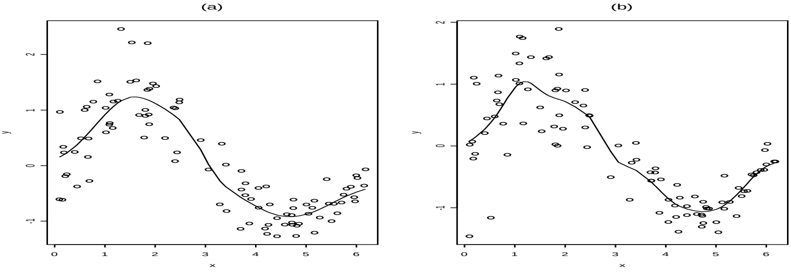


FIGURE 6: Typical scatterplots with superimposed kernel smoothers; (a) Normal data; (b) Laplace data.

Simulation: regression without model ($\star$)

Instead: $Y = \mu(x) + \sigma(x)\ \varepsilon_x$ with $\varepsilon_x = \frac{c_x Z + (1-c_x) W}{\sqrt{c_x^2 + (1-c_x)^2}}$

Simulation: regression without model ($\star$)

Instead: $Y = \mu(x) + \sigma(x)\ \varepsilon_x$ with $\varepsilon_x = \frac{c_x Z + (1-c_x) W}{\sqrt{c_x^2 + (1-c_x)^2}}$

- $Z \sim N(0,1)$ independent of W that is also (0,1) but has exponential shape, i.e., $\frac{1}{2}\chi_2^2 - 1$.

Simulation: regression without model ($\star$)

Instead: $Y = \mu(x) + \sigma(x)\ \varepsilon_x$ with $\varepsilon_x = \frac{c_x Z + (1-c_x) W}{\sqrt{c_x^2 + (1-c_x)^2}}$

- $Z \sim N(0,1)$ independent of W that is also (0,1) but has exponential shape, i.e., $\frac{1}{2}\chi_2^2 - 1$.
- ε_x independent but not i.i.d.: $c_x = x/(2\pi)$ for $x \in [0, 2\pi]$

Simulation: regression without model ($\star$)

Instead: $Y = \mu(x) + \sigma(x)\ \varepsilon_x$ with $\varepsilon_x = \frac{c_x Z + (1-c_x) W}{\sqrt{c_x^2 + (1-c_x)^2}}$

- $Z \sim N(0,1)$ independent of W that is also (0,1) but has exponential shape, i.e., $\frac{1}{2}\chi_2^2 - 1$.
- ε_x independent but not i.i.d.: $c_x = x/(2\pi)$ for $x \in [0, 2\pi]$
- Large x: ε_x is close to Normal.
  Small x: ε_x is skewed/kurtotic.

x_f/π	0.15	0.3	0.5	0.75	1	1.25	1.5
Norm	0.878	0.886	0.854	0.886	0.878	0.860	0.876
	1.6147	1.6119	1.6117	1.6116	1.6117	1.6116	1.6117
	0.006	0.006	0.006	0.006	0.006	0.006	0.006
MB	0.852	0.864	0.818	0.854	0.878	0.866	0.802
	1.6021	1.5326	1.4547	1.5855	1.7120	1.5955	1.4530
	0.013	0.013	0.012	0.014	0.015	0.013	0.012
MFMB	0.904	0.894	0.890	0.900	0.928	0.910	0.870
	1.8918	1.8097	1.7248	1.8602	2.006	1.8669	1.7170
	0.017	0.016	0.017	0.016	0.016	0.015	0.016
LMF	0.916	0.872	0.860	0.898	0.926	0.910	0.888
	1.8581	1.7730	1.6877	1.8286	1.9685	1.8334	1.6921
	0.016	0.015	0.014	0.016	0.017	0.015	0.015
MF	0.910	0.888	0.902	0.892	0.906	0.922	0.874
	1.8394	1.7531	1.6784	1.8117	1.9423	1.8139	1.6808
	0.016	0.015	0.014	0.016	0.017	0.016	0.015
PMF	0.900	0.884	0.880	0.906	0.912	0.912	0.884
	1.8734	1.7814	1.7013	1.8394	1.9705	1.8462	1.7076
	0.016	0.014	0.014	0.015	0.016	0.015	0.014

90% Prediction intervals: i.i.d. Normal errors.

x_f/π	0.15	0.3	0.5	0.75	1	1.25	1.5
Norm	0.886	0.892	0.872	0.896	0.896	0.878	0.894
	1.6296	1.6268	1.6266	1.6265	1.6266	1.6266	1.6266
	0.008	0.008	0.008	0.008	0.008	0.008	0.008
MB	0.872	0.836	0.856	0.868	0.890	0.860	0.846
	1.5881	1.5743	1.5114	1.6276	1.7526	1.6255	1.4487
	0.017	0.017	0.018	0.017	0.017	0.017	0.016
MFMB	0.914	0.904	0.906	0.898	0.938	0.898	0.892
	1.8663	1.8602	1.7735	1.9157	2.044	1.9043	1.7049
	0.021	0.022	0.022	0.020	0.020	0.020	0.020
LMF	0.902	0.868	0.904	0.912	0.910	0.912	0.870
	1.8418	1.8470	1.8034	1.8777	1.9907	1.8978	1.7110
	0.022	0.022	0.025	0.022	0.021	0.022	0.021
MF	0.898	0.884	0.886	0.914	0.938	0.904	0.874
	1.8134	1.8307	1.7847	1.8632	1.9704	1.8756	1.7054
	0.022	0.022	0.025	0.023	0.021	0.023	0.022
PMF	0.918	0.910	0.868	0.880	0.946	0.928	0.882
	1.8504	1.8633	1.8090	1.8954	1.9953	1.8995	1.7236
	0.022	0.022	0.024	0.022	0.021	0.022	0.021

90% Prediction intervals: i.i.d. Laplace errors.

x_f/π	0.15	0.3	0.5	0.75	1	1.25	1.5
Norm	0.906	0.890	0.890	0.884	0.908	0.900	0.870
	1.5937	1.5911	1.5908	1.5908	1.5908	1.5908	1.5908
	0.009	0.009	0.009	0.009	0.009	0.009	0.009
MB	0.846	0.878	0.860	0.882	0.894	0.862	0.804
	1.4846	1.4530	1.3485	1.5421	1.6795	1.5329	1.4012
	0.021	0.019	0.018	0.019	0.019	0.017	0.015
MFMB	0.928	0.946	0.886	0.964	0.932	0.912	0.846
	1.8161	1.7776	1.6409	1.8833	2.051	1.8695	1.7162
	0.031	0.025	0.023	0.026	0.024	0.022	0.021
LMF	0.916	0.934	0.908	0.928	0.918	0.898	0.846
	1.7555	1.7460	1.5870	1.8489	1.9798	1.7985	1.6652
	0.027	0.025	0.023	0.024	0.024	0.020	0.019
MF	0.908	0.932	0.882	0.910	0.906	0.910	0.860
	1.7344	1.7265	1.5561	1.8300	1.9345	1.7707	1.6355
	0.027	0.025	0.023	0.025	0.023	0.020	0.019
PMF	0.926	0.936	0.932	0.922	0.932	0.872	0.872
	1.7748	1.7636	1.5991	1.8550	1.9898	1.8083	1.6737
	0.026	0.024	0.022	0.023	0.023	0.019	0.019

90% Prediction intervals: non-i.i.d. errors.

Local Linear Estimation of a Conditional Distribution

- **Objective:** Nonparametric regression at boundary points
- Local regression applied for problems involving **conditional moment** estimation at both interior and boundary points e.g. $\mu(x) = E(Y|X = x)$
- **Our interest:** Estimate **conditional distribution** at boundary points using local linear regression
- **Known issues:** Estimated distribution may not be monotone increasing and may not lie in [0,1]
- **Proposed solution** corrects for monotonicity, superior performance demonstrated for both synthetic and real-life datasets versus existing estimators

Local Linear Setup

Conditional Mean:

$\mu(x) = E(Y|X = x)$

estimated by

Local Constant Estimator (Nadaraya-Watson) :

$\frac{\sum_{i=1}^{n} \tilde{K}_{i,x} Y_i}{\sum_{i=1}^{n} \tilde{K}_{i,x}}$

where $\tilde{K}_{i,x} = K\left(\frac{x - x_i}{b}\right)$

or by Local Linear Estimator:

$\frac{\sum_{j=1}^{n} w_j Y_j}{\sum_{j=1}^{n} w_j}$

where $w_i = \tilde{K}_{i,x}\left(1 - \hat{\beta}(x - x_i)\right)$ and $\hat{\beta} = \frac{\sum_{i=1}^{n} \tilde{K}_{i,x}(x - x_i)}{\sum_{i=1}^{n} \tilde{K}_{i,x}(x - x_i)^2}$

Local Linear Distribution

Conditional Distribution is a Mean:

$D_x(y) = E(W|X = x)$ where $W = \mathbf{1}\{Y \leq y\}$

Local Constant Distribution Estimator:

$\hat{D}_x^{LC}(y) = \frac{\sum_{i=1}^n \tilde{K}_{i,x} \mathbf{1}\{Y_i \leq y\}}{\sum_{i=1}^n \tilde{K}_{i,x}}$

where $\tilde{K}_{i,x} = K\left(\frac{x - x_i}{b}\right)$

Local Linear Distribution Estimator:

$\hat{D}_x^{LL}(y) = \frac{\sum_{j=1}^n w_j \mathbf{1}\{Y_j \leq y\}}{\sum_{j=1}^n w_j}$

where $w_i = \tilde{K}_{i,x}\left(1 - \hat{\beta}(x - x_i)\right)$ and $\hat{\beta} = \frac{\sum_{i=1}^n \tilde{K}_{i,x}(x - x_i)}{\sum_{i=1}^n \tilde{K}_{i,x}(x - x_i)^2}$

Smooth Version of Local Linear Estimator:

$\bar{D}_x^{LL}(y) = \frac{\sum_{j=1}^n w_j \Lambda(\frac{y - Y_j}{h_0})}{\sum_{j=1}^n w_j}$ where Λ is a smooth cdf.

Hansen Local Linear Estimator

Issues with LL-based distribution estimation:

($\star$) Weights in local linear estimation can be negative

- $\bar{D}_x^{LL}(y)$ not guaranteed to be in $[0,1]$
- $\bar{D}_x^{LL}(y)$ not guaranteed to be monotonic

Hansen proposal:

$$\bar{D}_x^{LLH}(y) = \frac{\sum_{i=1}^n w_i^\diamond \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^n w_i^\diamond}$$

$$w_i = \tilde{K}_{i,x}\left(1 - \hat{\beta}(x - x_i)\right)$$

$$\alpha = \hat{\beta}(x - x_i)$$

$$w_i^\diamond = \begin{cases} 0 & \text{when } \alpha > 1 \\ \tilde{K}_{i,x}(1-\alpha) & \text{when } \alpha \leq 1. \end{cases}$$

Monotone Local Linear Estimation (joint with S. Das)

- Recall that the derivative of $\bar{D}_x^{LL}(y)$ with respect to y is given by

$$\bar{d}_x^{LL}(y) = \frac{\frac{1}{h_0}\sum_{j=1}^n w_j \lambda(\frac{y-Y_j}{h_0})}{\sum_{j=1}^n w_j}$$

where $\lambda(y)$ is the derivative of $\Lambda(y)$.

- Define a nonnegative version of $\bar{d}_x^{LL}(y)$ as $\bar{d}_x^{LL+}(y) = \max(\bar{d}_x^{LL}(y), 0)$.
- To make the above a proper density function, renormalize it to area one, i.e., let

$$\bar{d}_x^{LLM}(y) = \frac{\bar{d}_x^{LL+}(y)}{\int_{-\infty}^{\infty} \bar{d}_x^{LL+}(s)ds}. \quad (1)$$

- Finally, define $\bar{D}_x^{LLM}(y) = \int_{-\infty}^{y} \bar{d}_x^{LLM}(s)ds$.

Note: Other algorithms for monotonicity correction are also possible which directly use the estimated distribution $\bar{D}_x^{LL}(y)$.

Results with synthetic data - (KS statistic)

Model:

$Y_i = \sin(2\pi x_i) + \sigma(x_i)\epsilon_i$ for $i = 1, 2, \ldots, 1001, x_i = \frac{i}{n}, \sigma(x_i) = 0.1,$ and $\epsilon_i = N(0,1)$, Prediction at i=1001

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	**0.23508**	0.252884	0.275132
7.4	0.241992	0.233996	0.23606
11.1	0.2767	**0.232064**	0.218948
14.8	0.31528	0.240476	0.20744
18.5	0.349924	0.2554	**0.2009**
22.2	0.38438	0.273648	0.204404
25.9	0.418316	0.288032	0.21502
29.6	0.448772	0.307672	0.231588
33.3	0.474796	0.326224	0.253472
37.0	0.502768	0.342884	0.275936
40.7	0.5264	0.360888	0.2993
44.4	0.54664	0.37786	0.320348
48.1	0.56692	0.393392	0.34248
51.8	0.58646	0.407108	0.359404

Results with synthetic data - (Point Prediction)

Model:

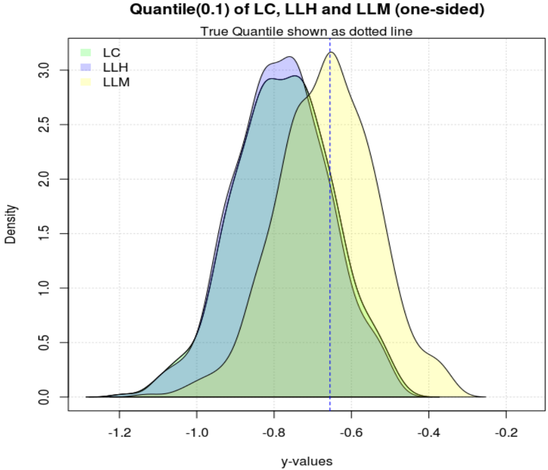
$Y_i = \sin(2\pi x_i) + \sigma(x_i)\epsilon_i$ for $i = 1, 2, \ldots, 1001, x_i = \frac{i}{n}, \sigma(x_i) = 0.1$, and $\epsilon_i = N(0, 1)$, Prediction at i=1001

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.01887676	0.01265856	-0.0087034	0.01453471	0.0004694887	0.01667712	0.00279478	0.01713243
7.4	-0.03782673	**0.01261435**	-0.01818502	0.0126929	0.0005444976	0.01323652	0.003247646	0.01340418
11.1	-0.05753609	0.01418224	-0.02725602	**0.01232877**	-0.001022256	0.01200918	0.0039133	0.01219628
14.8	-0.07724901	0.01672728	-0.03718728	0.01259729	-0.005397138	0.01148354	0.00354838	0.01167496
18.5	-0.09692561	0.0200906	-0.04758345	0.01327841	-0.01222596	**0.01130622**	0.002834568	0.01139095
22.2	-0.116533	0.02423279	-0.05831195	0.01431087	-0.02106315	0.01142789	0.002008806	0.01120327
25.9	-0.1359991	0.02911512	-0.06918129	0.0156254	-0.03138586	0.01185914	0.001102312	0.01106821
29.6	-0.1555938	0.03480583	-0.08021998	0.01722284	-0.04274234	0.01263368	8.912064e-05	0.01096947
33.3	-0.1752324	0.04128715	-0.09144259	0.01910772	-0.05473059	0.01375585	-0.001070282	0.01089842
37.0	-0.1947342	0.04848954	-0.1027918	0.02127558	-0.0670785	0.01521865	-0.002416635	0.01084951
40.7	-0.2145001	0.05656322	-0.1142845	0.02374615	-0.07967838	0.01704094	-0.003988081	0.01081946
44.4	-0.2343967	0.06548142	-0.1259372	0.02651703	-0.09236019	0.01919461	-0.005818943	**0.01080699**
48.1	-0.2543523	0.07522469	-0.1377167	0.02960364	-0.1050934	0.02168698	-0.007939144	0.01081259
51.8	-0.2740635	0.08563245	-0.1496325	0.03301117	-0.1178388	0.02451228	-0.01037417	0.01083832

Results with synthetic data - (Quantile Estimation)

Model:

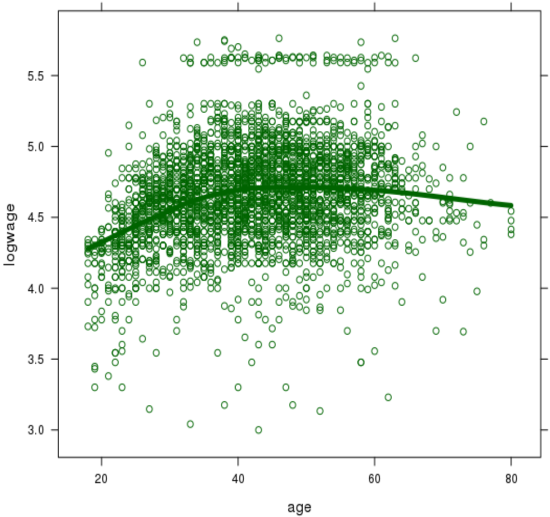
$Y_i = \sin(2\pi x_i) + \sigma(x_i)\epsilon_i$ for $i = 1, 2, \ldots, 1001, x_i = \frac{i}{n}, \sigma(x_i) = 0.3$, and $\epsilon_i = N(0,1)$, Prediction at i=1001



Results with real-life data

Model: **Wage** dataset from ISLR package in R.

Objective: point prediction over last 231 values of backward dataset.



Point Prediction with ISLR data

Method	Bias	MSE
LC	0.0004954944	0.08236025
LLH	-0.001962329	0.0808793
LLM	-6.005305e-05	0.08044857
LL	0.0002608775	0.08055141