Embracing Blessing of Massive Scale in Big Data

Guang Cheng
Department of Statistics
Purdue University

PURDUE
UNIVERSITY

July 31, 2017
A joint work with Shih-Kang Chao and Stanislav Volgushev

With the advance of technology, it is increasingly common that data set is so large that it cannot be stored in a single machine

- Social media (views, likes, comments, images...)
- Meteorological and environmental surveillance
- Transactions in e-commerce
- Others...

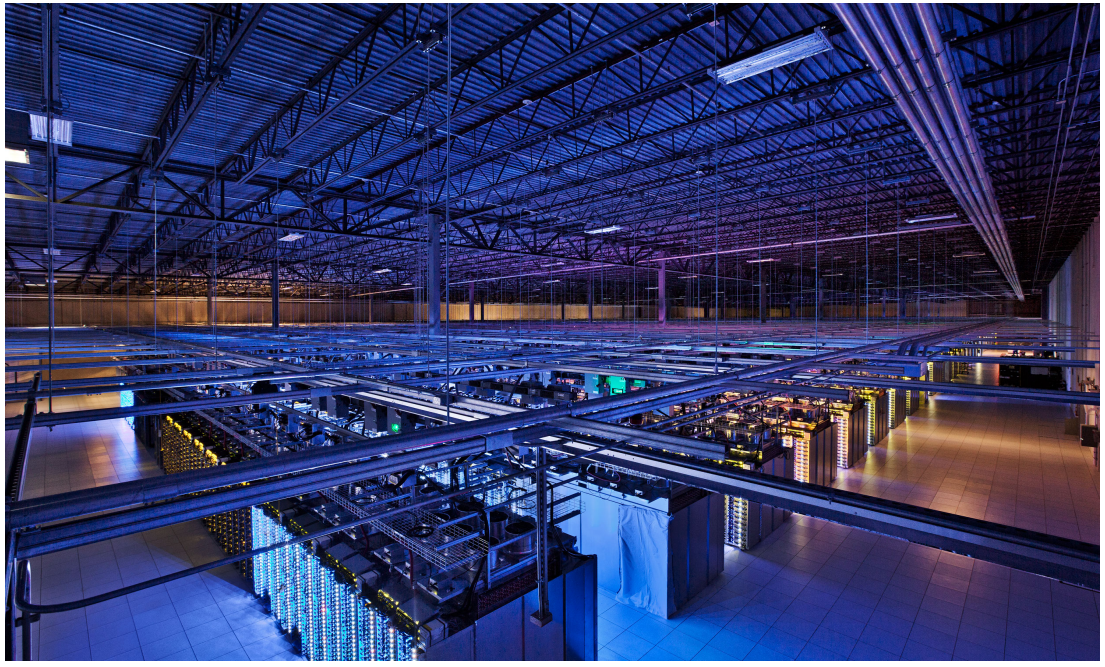


Figure: A server room in Council Bluffs, Iowa. Photo: Google/Connie Zhou.

Divide and Conquer (D&C)

- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Divide and Conquer (D&C)

- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Divide and Conquer (D&C)

- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Divide and Conquer (D&C)

- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Divide and Conquer (D&C)

- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Divide and Conquer (D&C)

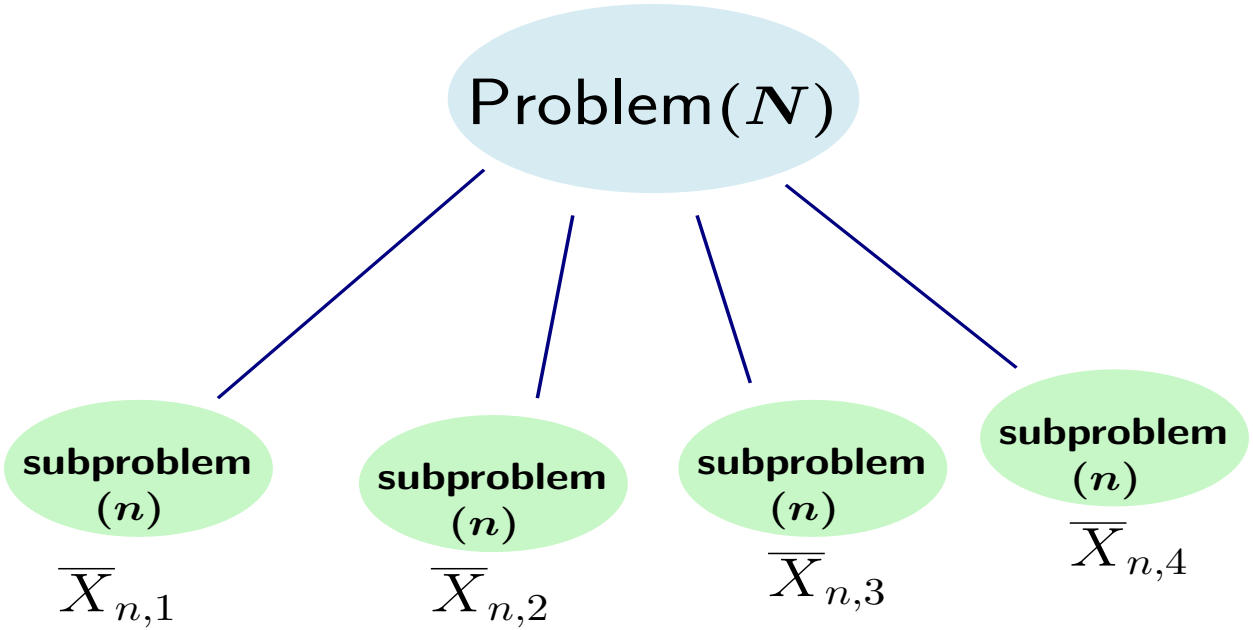
- To take advantage of the opportunities in massive data, we need to deal with storing (disc) and computational (memory, processor) bottlenecks.
- Divide and Conquer:
  - Randomly divide N samples into S groups;
  - Conduct parallel computing based on each subset, implemented by e.g., Hadoop;
  - Individual sub-estimates or sub-inference results are aggregated in the end.
- Data that are stored in distributed locations can be analyzed in a similar manner.

Does D&C fit for statistical analysis?

Sometimes it does, but sometimes it doesn't...

In the following, we give two simple examples.

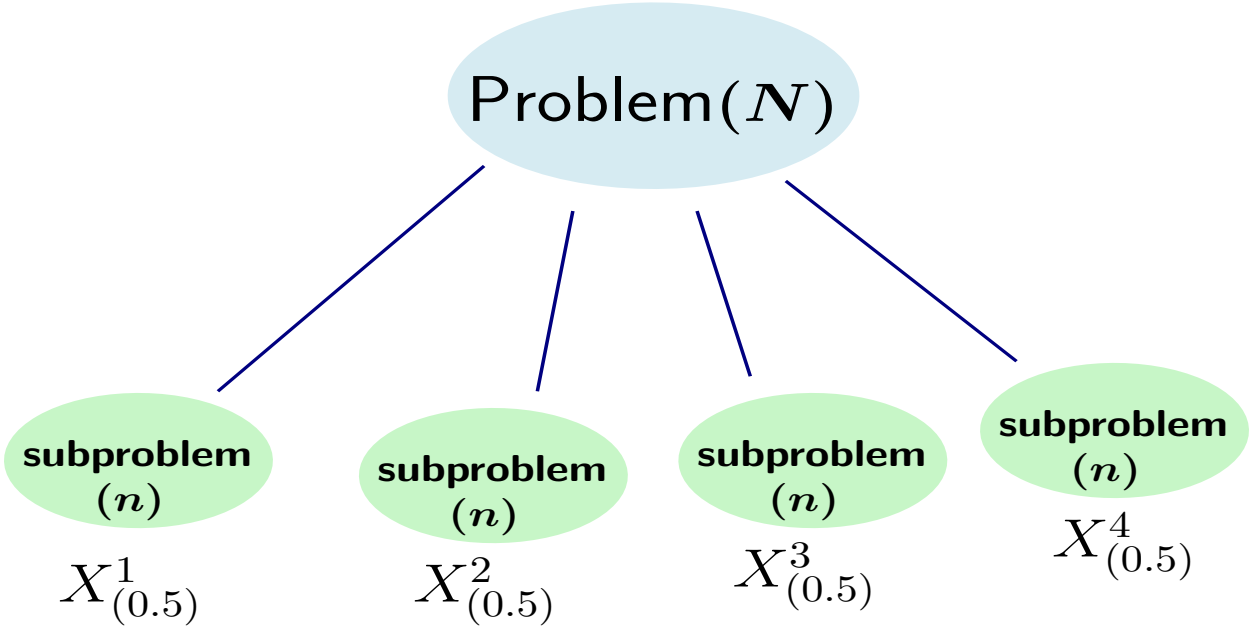
Example 1: Sample Mean



$$\frac{1}{4}\sum_{s=1}^{4}\overline{X}_s = \frac{1}{4n}\sum_{s=1}^{4}\sum_{i=1}^{n}X_{is} = \frac{1}{N}\sum_{i=1}^{N}X_i = \overline{X}_N.$$

It fits!

Example 2: Sample Median

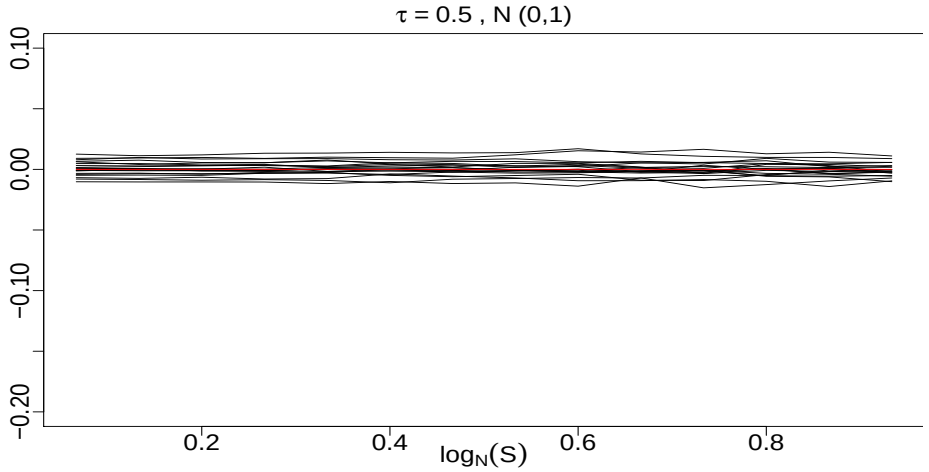


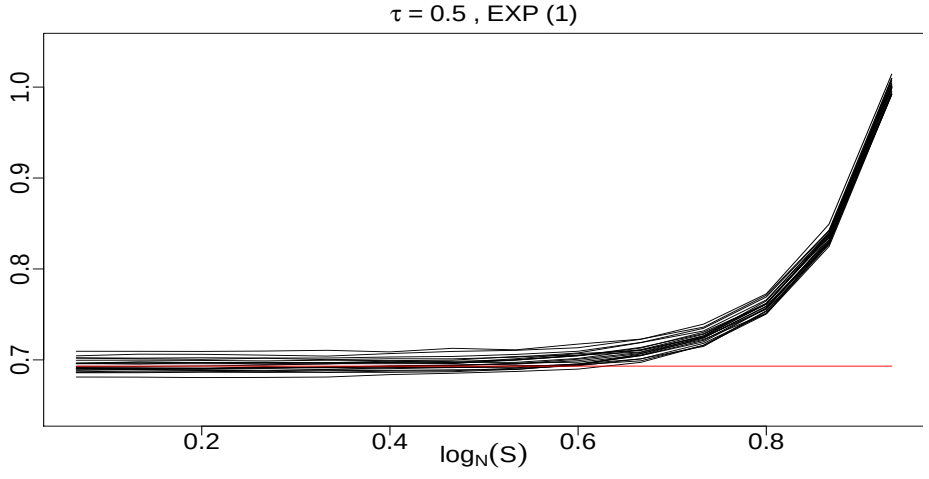
$X^s_{(0.5)}$ = the middle value of ordered n samples in s group;
$X_{(0.5)}$ = overall median

$$\frac{1}{4}\sum_{s=1}^{4} X^s_{(0.5)} \overset{??}{=} X_{(0.5)}$$

Example 2: Sample Median

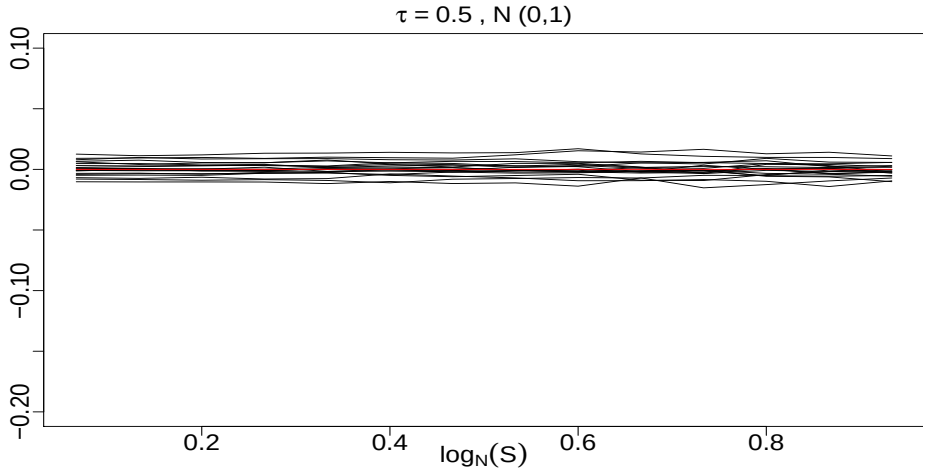
Simulation 1: $X_i \sim N(0,1)$; Simulation 2: $X_i \sim$ Exponential(1).
$N = 2^{15}$. True median v.s. simulated $S^{-1} \sum_{s=1}^{S} X^s_{(0.5)}$

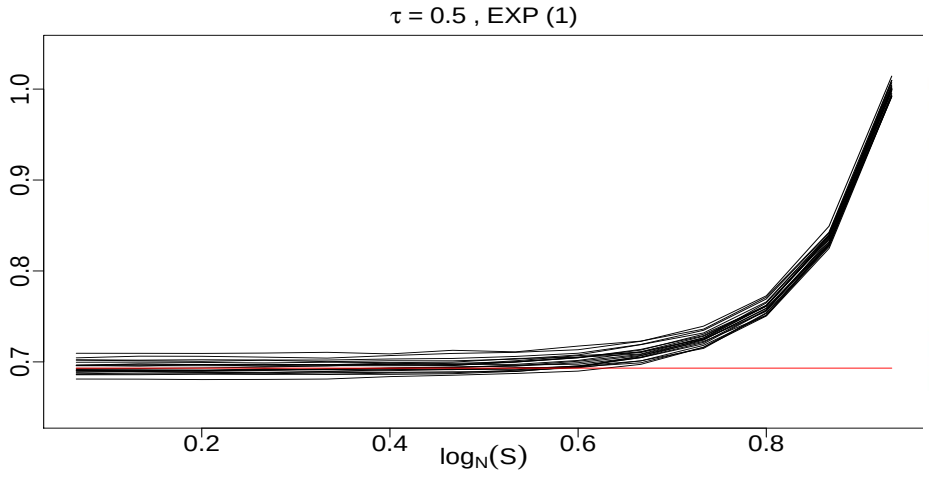




Example 2: Sample Median

Simulation 1: $X_i \sim N(0,1)$; Simulation 2: $X_i \sim$ Exponential(1). $N = 2^{15}$. True median v.s. simulated $S^{-1}\sum_{s=1}^{S} X^s_{(0.5)}$





Specific Goals

- When does the D&C algorithm work?
  - Especially for skewed and heavy tail distribution
- Statistical inference
  - Asymptotic distribution
- Inference for the "whole" underlying distribution?
  - Take advantage of massive size to discover subtle patterns hidden in the "whole" distribution

Specific Goals

- When does the D&C algorithm work?
  - Especially for skewed and heavy tail distribution
- Statistical inference
  - Asymptotic distribution
- Inference for the "whole" underlying distribution?
  - Take advantage of massive size to discover subtle patterns hidden in the "whole" distribution

Specific Goals

- When does the D&C algorithm work?
  - Especially for skewed and heavy tail distribution
- Statistical inference
  - Asymptotic distribution
- Inference for the "whole" underlying distribution?
  - Take advantage of massive size to discover subtle patterns hidden in the "whole" distribution

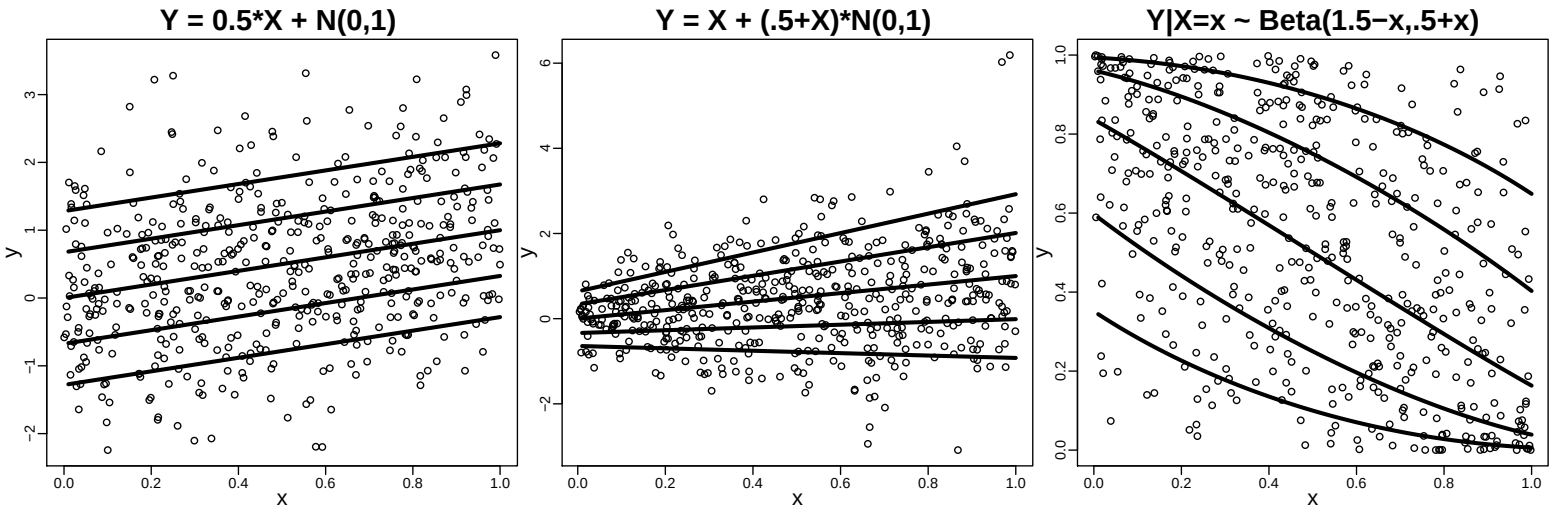
Outline

Quantile

Response Y, predictors X. For $\tau \in (0,1)$, conditional quantile curve $Q(\cdot;\tau)$ of $Y \in \mathbb{R}$ conditional on X is defined through

$$P(Y \leq Q(X;\tau)|X = x) = \tau \quad \forall x.$$

$Q(x;\tau)$ at $\tau = 0.1, 0.25, 0.5, 0.75, 0.9$ under different models

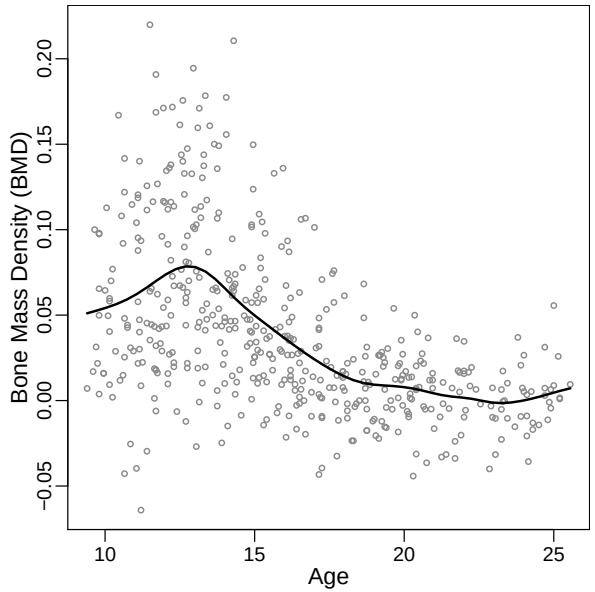


Quantile Regression v.s. Mean Regression

Mean Regression:

$Y_i = m(X_i) + \varepsilon_i,\ \mathbb{E}[\varepsilon | X = x] = 0$

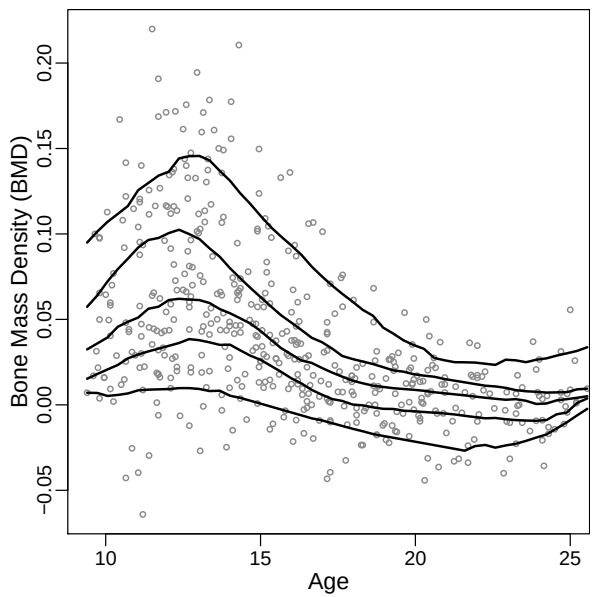
- m: regression function, object of interest.
- ε_i: "errors."



Quantile Regression:

$P(Y \leq Q(x;\tau) | X = x) = \tau$

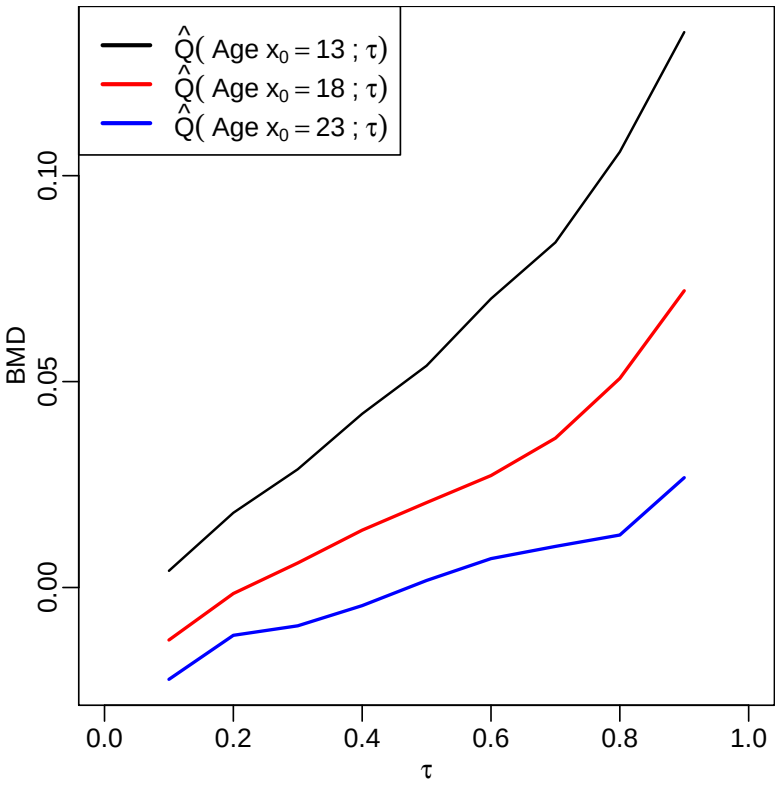
- No strict distinction between "signal" and "noise."
- Object of interest: properties of conditional distribution of $Y|X = x$.
- Contains much richer information than just mean trend.

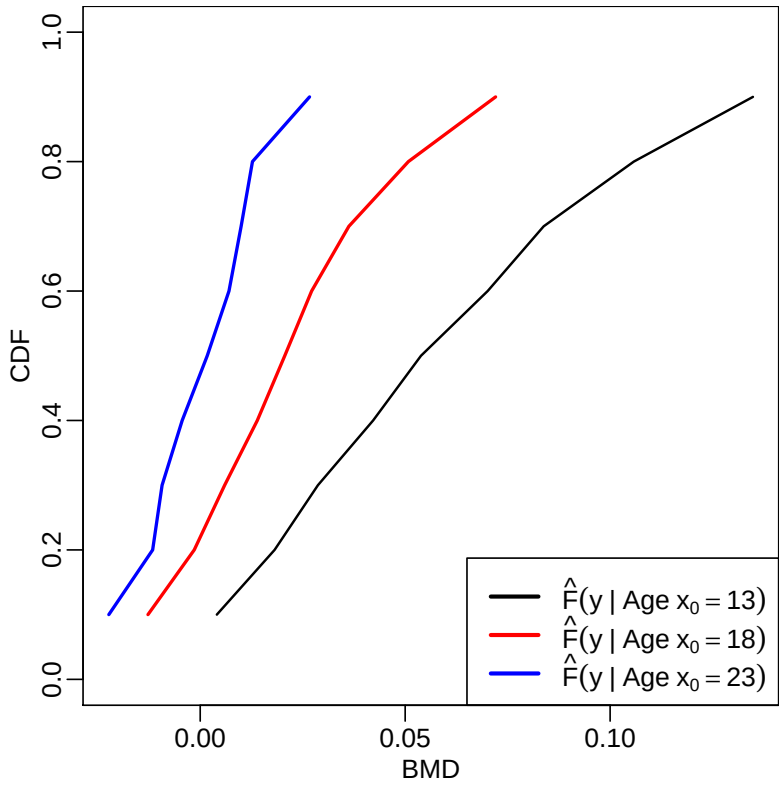


Quantile Curves v.s. Conditional Distribution

Let $F_{Y|X}(y|x)$ be the conditional dist. function of Y given X.

$$Q(x_0;\tau) = F^{-1}(\tau|x_0).$$





Quantile Regression as Optimization

Koenker and Bassett (1978): if $Q(x;\tau) = \boldsymbol{\beta}(\tau)^\top x$, estimate by

$$\widehat{\boldsymbol{\beta}}_{or}(\tau) := \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\tau(Y_i - \mathbf{b}^\top X_i) \tag{1.1}$$

where $\rho_\tau(u) := \tau u^+ + (1-\tau)u^-$ is the so-called "check function."

Remark: Optimization problem (1.1) is convex (but non-smooth), and can be solved by linear programming.

A General Series Approximation Framework:

$$Q(x;\tau) \approx \mathbf{Z}(x)^\top \boldsymbol{\beta}(\tau)$$

$m := \dim(\mathbf{Z})$ (it is possible that $m \to \infty$). Solve

$$\widehat{\boldsymbol{\beta}}_{or}(\tau) := \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\tau\{Y_i - \mathbf{b}^\top \mathbf{Z}(X_i)\}$$

- Examples of $\mathbf{Z}(x)$ include
  - linear model with fixed/increasing dimension;
  - B-splines, polynomials, trigonometric polynomials.

A General Series Approximation Framework:

$$Q(x;\tau) \approx \mathbf{Z}(x)^\top \boldsymbol{\beta}(\tau)$$

$m := \dim(\mathbf{Z})$ (it is possible that $m \to \infty$). Solve

$$\widehat{\boldsymbol{\beta}}_{or}(\tau) := \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\tau \{Y_i - \mathbf{b}^\top \mathbf{Z}(X_i)\}$$

- Examples of $\mathbf{Z}(x)$ include
  - linear model with fixed/increasing dimension;
  - B-splines, polynomials, trigonometric polynomials.

A General Series Approximation Framework:

$$Q(x;\tau) \approx \mathbf{Z}(x)^\top \boldsymbol{\beta}(\tau)$$

$m := \dim(\mathbf{Z})$ (it is possible that $m \to \infty$). Solve

$$\widehat{\boldsymbol{\beta}}_{or}(\tau) := \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\tau \{Y_i - \mathbf{b}^\top \mathbf{Z}(X_i)\}$$

- Examples of $\mathbf{Z}(x)$ include
  - linear model with fixed/increasing dimension;
  - B-splines, polynomials, trigonometric polynomials.

A General Series Approximation Framework:

$$Q(x;\tau) \approx \mathbf{Z}(x)^\top \boldsymbol{\beta}(\tau)$$

$m := \dim(\mathbf{Z})$ (it is possible that $m \to \infty$). Solve

$$\widehat{\boldsymbol{\beta}}_{or}(\tau) := \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\tau\{Y_i - \mathbf{b}^\top \mathbf{Z}(X_i)\}$$

- Examples of $\mathbf{Z}(x)$ include
  - linear model with fixed/increasing dimension;
  - B-splines, polynomials, trigonometric polynomials.

Motivation for Two-Step Algorithm

- To infer the "whole" conditional distribution by

$$F_{Y|X}(y|x_0) = \int_0^1 \mathbf{1}\{Q(x_0;\tau) < y\} d\tau,$$

  we need to compute $Q(x_0;\tau)$ for $\{\tau_1, \ldots, \tau_K\} \in [\tau_L, \tau_U]^K$;
- What is the minimal number of K?
- For each fixed τ_j, we have to distribute data to S machines since $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ is computationally infeasible.
- What is the maximum number of S?

Motivation for Two-Step Algorithm

- To infer the "whole" conditional distribution by

$$F_{Y|X}(y|x_0) = \int_0^1 \mathbf{1}\{Q(x_0;\tau) < y\} d\tau,$$

  we need to compute $Q(x_0;\tau)$ for $\{\tau_1, \ldots, \tau_K\} \in [\tau_L, \tau_U]^K$;
- What is the minimal number of K?
- For each fixed τ_j, we have to distribute data to S machines since $\widehat{\beta}_{or}(\tau)$ is computationally infeasible.
- What is the maximum number of S?

Motivation for Two-Step Algorithm

- To infer the "whole" conditional distribution by

$$F_{Y|X}(y|x_0) = \int_0^1 \mathbf{1}\{Q(x_0;\tau) < y\}d\tau,$$

  we need to compute $Q(x_0;\tau)$ for $\{\tau_1, \ldots, \tau_K\} \in [\tau_L, \tau_U]^K$;
- What is the minimal number of K?
- For each fixed τ_j, we have to distribute data to S machines since $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ is computationally infeasible.
- What is the maximum number of S?

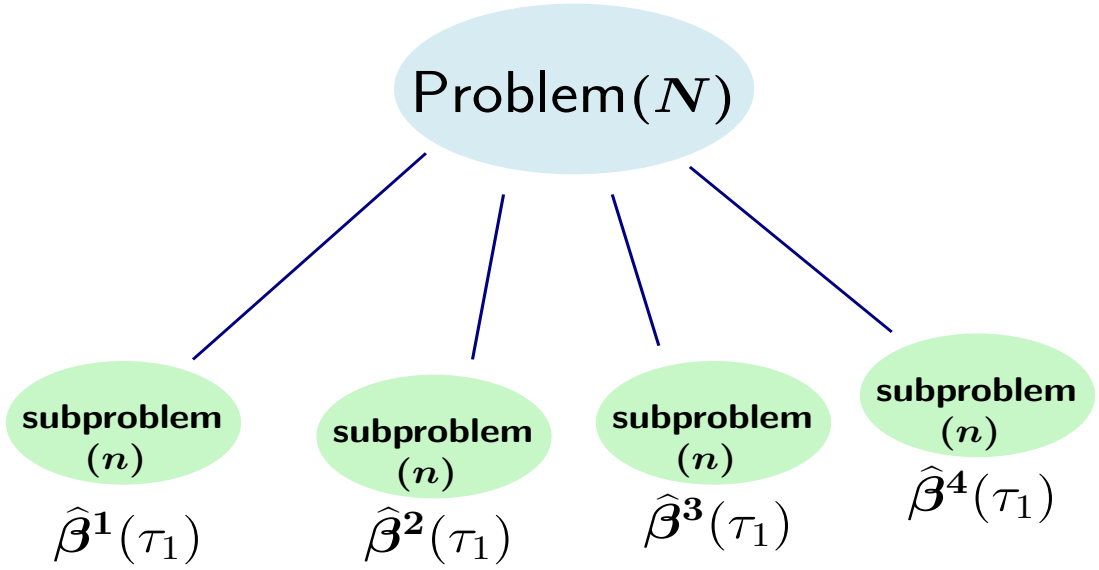
Motivation for Two-Step Algorithm

- To infer the "whole" conditional distribution by

$$F_{Y|X}(y|x_0) = \int_0^1 \mathbf{1}\{Q(x_0;\tau) < y\}d\tau,$$

  we need to compute $Q(x_0;\tau)$ for $\{\tau_1, \ldots, \tau_K\} \in [\tau_L, \tau_U]^K$;
- What is the minimal number of K?
- For each fixed τ_j, we have to distribute data to S machines since $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ is computationally infeasible.
- What is the maximum number of S?

Step I: D&C Algorithm at Any Fixed τ



$$\widehat{\boldsymbol{\beta}}^s(\tau) := \arg\min_{\mathbf{b}\in\mathbb{R}^m} \sum_{i=1}^{n} \rho_\tau\{Y_{is} - \mathbf{b}^\top \mathbf{Z}(X_{is})\}$$

$$\overline{\boldsymbol{\beta}}(\tau) := \frac{1}{S}\sum_{s=1}^{S} \widehat{\boldsymbol{\beta}}^s(\tau).$$

Step II: Quantile Projection Algorithm

We want to estimate $\boldsymbol{\beta}(\tau)$ as a function over τ based on $\{\overline{\boldsymbol{\beta}}(\tau_1), \ldots, \overline{\boldsymbol{\beta}}(\tau_K)\}$ obtained in Step I:

$$\widehat{\boldsymbol{\beta}}(\tau) := \widehat{\Xi}^\top \mathbf{B}(\tau). \tag{2.1}$$

- $\mathbf{B} := (B_1, ..., B_q)^\top$ is B-spline basis (along τ-direction) defined on $\{t_1, ..., t_G\} \subset \mathcal{T}$ with degree $r_\tau \in \mathbb{N}$;
- $\widehat{\Xi}$ is computed as $[\widehat{\boldsymbol{\alpha}}_1 \, \widehat{\boldsymbol{\alpha}}_2 \, ... \, \widehat{\boldsymbol{\alpha}}_m]$, where

$$\widehat{\boldsymbol{\alpha}}_j := \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^q} \sum_{k=1}^{K} \left(\overline{\beta}_j(\tau_k) - \boldsymbol{\alpha}^\top \mathbf{B}(\tau_k) \right)^2. \tag{2.2}$$

- Quantile projection requires only $\{\overline{\boldsymbol{\beta}}(\tau_1), ..., \overline{\boldsymbol{\beta}}(\tau_K)\}$ of size $m \times K$, without the need to access the raw data set.

Step II: Quantile Projection Algorithm

We want to estimate $\boldsymbol{\beta}(\tau)$ as a function over τ based on $\{\overline{\boldsymbol{\beta}}(\tau_1), \ldots, \overline{\boldsymbol{\beta}}(\tau_K)\}$ obtained in Step I:

$$\widehat{\boldsymbol{\beta}}(\tau) := \widehat{\Xi}^\top \mathbf{B}(\tau). \tag{2.1}$$

- $\mathbf{B} := (B_1, ..., B_q)^\top$ is B-spline basis (along τ-direction) defined on $\{t_1, ..., t_G\} \subset \mathcal{T}$ with degree $r_\tau \in \mathbb{N}$;
- $\widehat{\Xi}$ is computed as $[\widehat{\boldsymbol{\alpha}}_1 \, \widehat{\boldsymbol{\alpha}}_2 \, ... \, \widehat{\boldsymbol{\alpha}}_m]$, where

$$\widehat{\boldsymbol{\alpha}}_j := \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^q} \sum_{k=1}^{K} \left(\overline{\beta}_j(\tau_k) - \boldsymbol{\alpha}^\top \mathbf{B}(\tau_k) \right)^2. \tag{2.2}$$

- Quantile projection requires only $\{\overline{\boldsymbol{\beta}}(\tau_1), ..., \overline{\boldsymbol{\beta}}(\tau_K)\}$ of size $m \times K$, without the need to access the raw data set.

Step II: Quantile Projection Algorithm

We want to estimate $\boldsymbol{\beta}(\tau)$ as a function over τ based on $\{\overline{\boldsymbol{\beta}}(\tau_1), \ldots, \overline{\boldsymbol{\beta}}(\tau_K)\}$ obtained in Step I:

$$\widehat{\boldsymbol{\beta}}(\tau) := \widehat{\Xi}^\top \mathbf{B}(\tau). \tag{2.1}$$

- $\mathbf{B} := (B_1, ..., B_q)^\top$ is B-spline basis (along τ-direction) defined on $\{t_1, ..., t_G\} \subset \mathcal{T}$ with degree $r_\tau \in \mathbb{N}$;
- $\widehat{\Xi}$ is computed as $[\widehat{\boldsymbol{\alpha}}_1 \, \widehat{\boldsymbol{\alpha}}_2 \, ... \, \widehat{\boldsymbol{\alpha}}_m]$, where

$$\widehat{\boldsymbol{\alpha}}_j := \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^q} \sum_{k=1}^{K} \left(\overline{\beta}_j(\tau_k) - \boldsymbol{\alpha}^\top \mathbf{B}(\tau_k) \right)^2. \tag{2.2}$$

- Quantile projection requires only $\{\overline{\boldsymbol{\beta}}(\tau_1), ..., \overline{\boldsymbol{\beta}}(\tau_K)\}$ of size $m \times K$, without the need to access the raw data set.

Step II: Quantile Projection Algorithm

We want to estimate $\boldsymbol{\beta}(\tau)$ as a function over τ based on $\{\overline{\boldsymbol{\beta}}(\tau_1), \ldots, \overline{\boldsymbol{\beta}}(\tau_K)\}$ obtained in Step I:

$$\widehat{\boldsymbol{\beta}}(\tau) := \widehat{\Xi}^\top \mathbf{B}(\tau). \tag{2.1}$$

- $\mathbf{B} := (B_1, ..., B_q)^\top$ is B-spline basis (along τ-direction) defined on $\{t_1, ..., t_G\} \subset \mathcal{T}$ with degree $r_\tau \in \mathbb{N}$;
- $\widehat{\Xi}$ is computed as $[\widehat{\boldsymbol{\alpha}}_1 \, \widehat{\boldsymbol{\alpha}}_2 \, ... \, \widehat{\boldsymbol{\alpha}}_m]$, where

$$\widehat{\boldsymbol{\alpha}}_j := \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^q} \sum_{k=1}^{K} \left(\overline{\beta}_j(\tau_k) - \boldsymbol{\alpha}^\top \mathbf{B}(\tau_k) \right)^2. \tag{2.2}$$

- Quantile projection requires only $\{\overline{\boldsymbol{\beta}}(\tau_1), ..., \overline{\boldsymbol{\beta}}(\tau_K)\}$ of size $m \times K$, without the need to access the raw data set.

Estimation of $F_{Y|X}(y|x)$

Now we can estimate $F_{Y|X}$ as

$$\widehat{F}_{Y|X}(y|x_0) := \tau_L + \int_{\tau_L}^{\tau_U} \mathbf{1}\{\widehat{Q}(x_0;\tau) < y\} d\tau,$$

where $\widehat{Q}(x_0;\tau) = \mathbf{Z}(x_0)^\top \widehat{\boldsymbol{\beta}}(\tau)$, and $\tau_k \in [\tau_L, \tau_U]$ for $1 \leq k \leq K$.

Computational Cost of Two-Step Procedure

- The two-step procedure requires only one pass through the entire data;
- The computational cost and statistical performance are determined by the number of machines S and the number of quantiles K in the projection, which both grow as N.
  - Find an upper bound of S and an lower bound of K s.t. $\widehat{\beta}(\tau)$ are "close" to $\widehat{\beta}_{or}(\tau)$ in some statistical sense;
  - The sharp upper and lower bounds characterize the intrinsic computational limit.

Computational Cost of Two-Step Procedure

- The two-step procedure requires only one pass through the entire data;
- The computational cost and statistical performance are determined by the number of machines S and the number of quantiles K in the projection, which both grow as N.
  - Find an upper bound of S and an lower bound of K s.t. $\widehat{\boldsymbol{\beta}}(\tau)$ are "close" to $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ in some statistical sense;
  - The sharp upper and lower bounds characterize the intrinsic computational limit.

Computational Cost of Two-Step Procedure

- The two-step procedure requires only one pass through the entire data;
- The computational cost and statistical performance are determined by the number of machines S and the number of quantiles K in the projection, which both grow as N.
  - Find an upper bound of S and an lower bound of K s.t. $\widehat{\boldsymbol{\beta}}(\tau)$ are "close" to $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ in some statistical sense;
  - The sharp upper and lower bounds characterize the intrinsic computational limit.

Computational Cost of Two-Step Procedure

- The two-step procedure requires only one pass through the entire data;
- The computational cost and statistical performance are determined by the number of machines S and the number of quantiles K in the projection, which both grow as N.
  - Find an upper bound of S and an lower bound of K s.t. $\widehat{\boldsymbol{\beta}}(\tau)$ are "close" to $\widehat{\boldsymbol{\beta}}_{or}(\tau)$ in some statistical sense;
  - The sharp upper and lower bounds characterize the intrinsic computational limit.

Oracle Rule

Chao, Volgushev and C. (2016) showed that

$$a_N \mathbf{u}^\top \big(\widehat{\boldsymbol{\beta}}_{or}(\tau) - \boldsymbol{\beta}(\tau)\big) \rightsquigarrow \mathbb{G}(\tau) \text{ in } \ell^\infty(\mathcal{T}) \text{ for } \mathcal{T} \subset (0,1), \qquad (3.1)$$

where $\mathbb{G}$ is a mean-zero Gaussian process.

- We say that $\widehat{\boldsymbol{\beta}}(\tau)$ satisfies oracle rule if it shares the same weak convergence as $\widehat{\boldsymbol{\beta}}_{or}(\tau)$, i.e., (3.1);
- Statistical inferential accuracy (based on $\widehat{F}_{Y|X}(\cdot|x_0)$) for $F_{Y|X}(\cdot|x_0)$ achieves its oracle level if the above oracle rule holds.

Oracle Rule

Chao, Volgushev and C. (2016) showed that

$$a_N \mathbf{u}^\top \big(\widehat{\boldsymbol{\beta}}_{or}(\tau) - \boldsymbol{\beta}(\tau)\big) \rightsquigarrow \mathbb{G}(\tau) \text{ in } \ell^\infty(\mathcal{T}) \text{ for } \mathcal{T} \subset (0,1), \qquad (3.1)$$

where $\mathbb{G}$ is a mean-zero Gaussian process.

- We say that $\widehat{\boldsymbol{\beta}}(\tau)$ satisfies oracle rule if it shares the same weak convergence as $\widehat{\boldsymbol{\beta}}_{or}(\tau)$, i.e., (3.1);
- Statistical inferential accuracy (based on $\widehat{F}_{Y|X}(\cdot|x_0)$) for $F_{Y|X}(\cdot|x_0)$ achieves its oracle level if the above oracle rule holds.

Oracle Rule

Chao, Volgushev and C. (2016) showed that

$$a_N \mathbf{u}^\top \big(\widehat{\boldsymbol{\beta}}_{or}(\tau) - \boldsymbol{\beta}(\tau)\big) \rightsquigarrow \mathbb{G}(\tau) \text{ in } \ell^\infty(\mathcal{T}) \text{ for } \mathcal{T} \subset (0,1), \qquad (3.1)$$

where $\mathbb{G}$ is a mean-zero Gaussian process.

- We say that $\widehat{\boldsymbol{\beta}}(\tau)$ satisfies oracle rule if it shares the same weak convergence as $\widehat{\boldsymbol{\beta}}_{or}(\tau)$, i.e., (3.1);
- Statistical inferential accuracy (based on $\widehat{F}_{Y|X}(\cdot|x_0)$) for $F_{Y|X}(\cdot|x_0)$ achieves its oracle level if the above oracle rule holds.

Two Leading Models

Linear model: $m = \dim(\mathbf{Z}(x))$ is fixed and $Q(x;\tau) = \mathbf{Z}(x)^\top \boldsymbol{\beta}(\tau)$;

Univariate spline model: $m = \dim(\mathbf{Z}(x)) \to \infty$ with the spline approximation error defined as $c_N(\boldsymbol{\gamma}_N) := \left|Q(x;\tau) - \mathbf{Z}(x)^\top \boldsymbol{\gamma}_N(\tau)\right|$,

$$\boldsymbol{\gamma}_N(\tau) := \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}^m} \mathbb{E}\big[(\mathbf{Z}^\top\boldsymbol{\gamma} - Q(X;\tau))^2 f(Q(X;\tau)|X)\big]. \quad (3.2)$$

"Theoretical" Phase Transitions

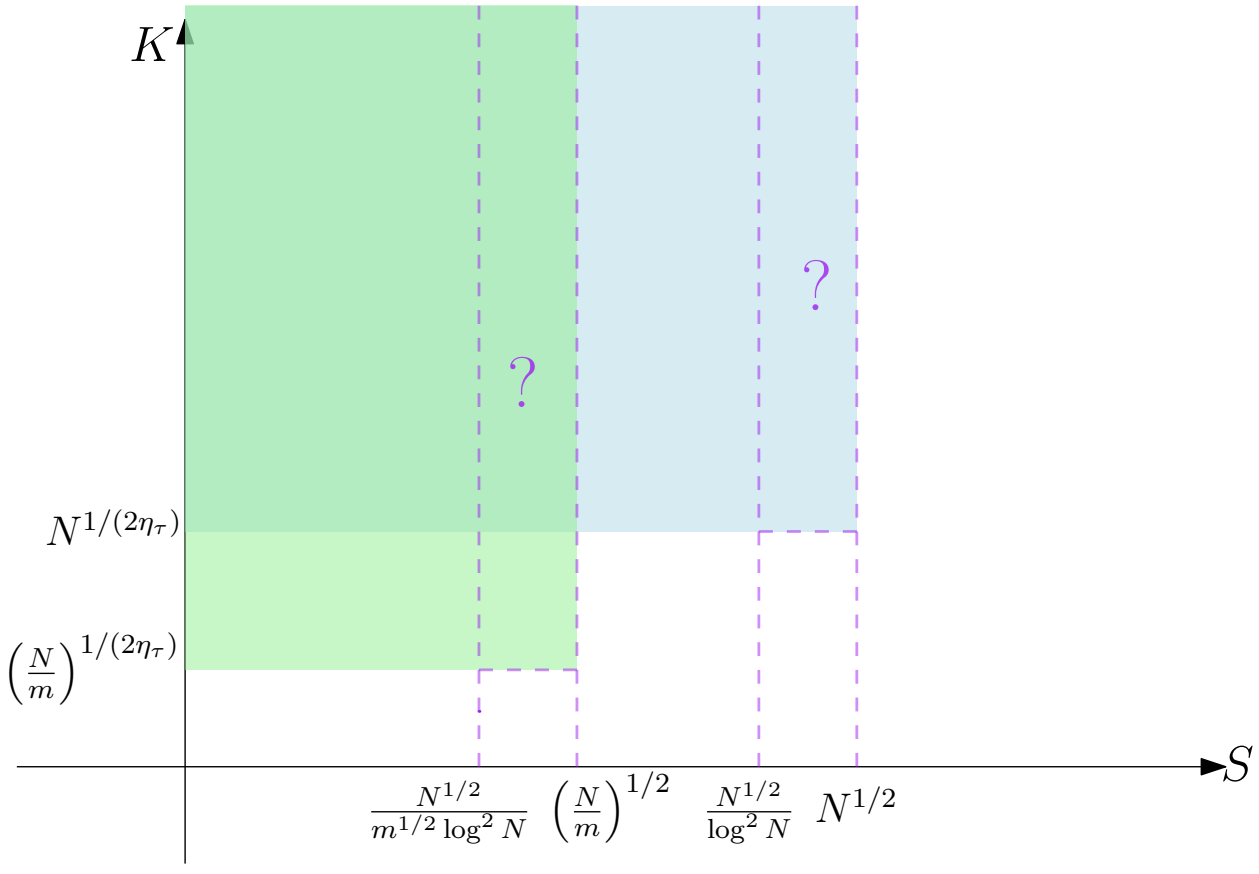


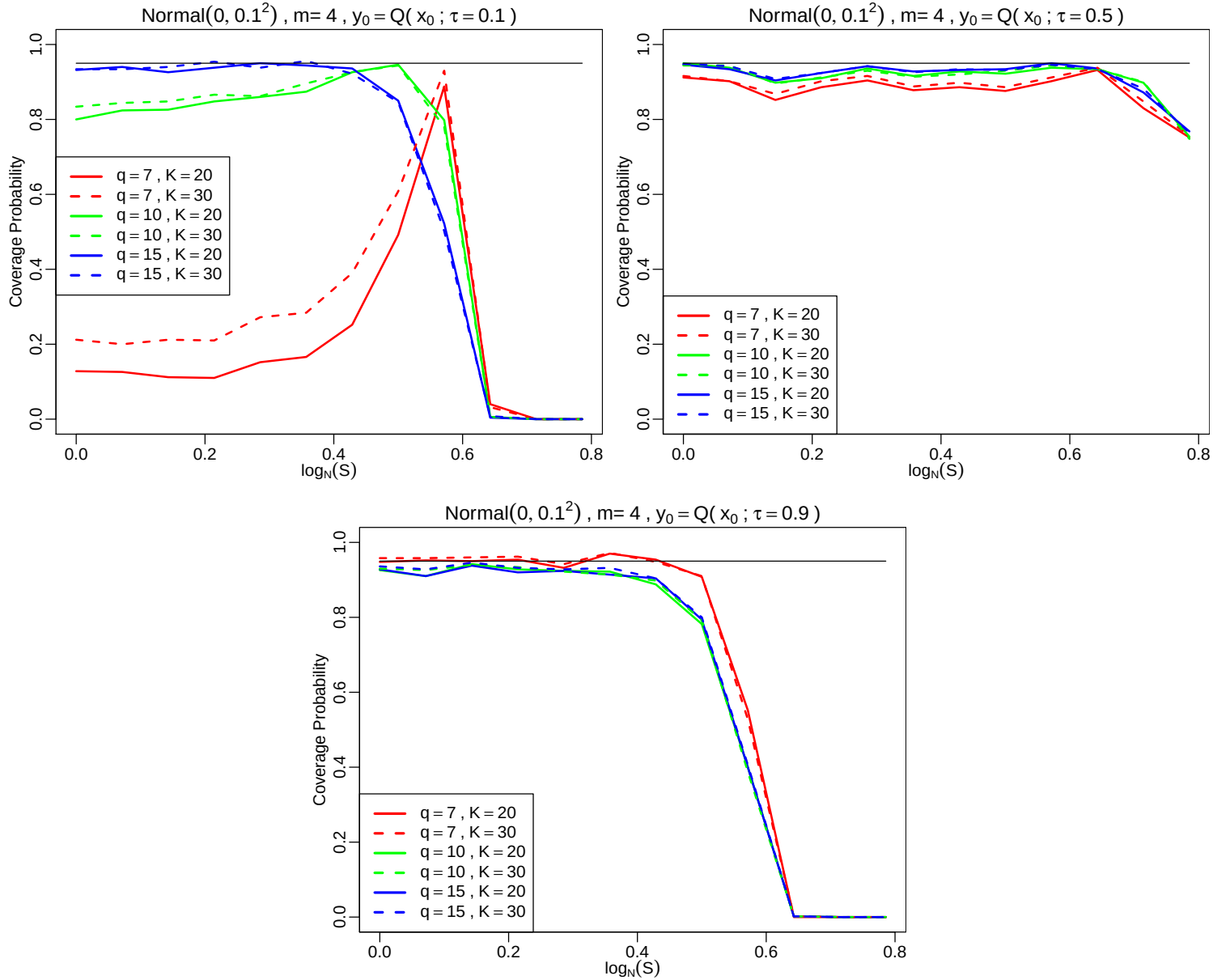
Figure: Regions (S, K) for the oracle rule of linear model and spline nonparametric model. "?" region is the discrepancy between the sufficient and necessary conditions.

Simulations for Nonparametric Spline Model

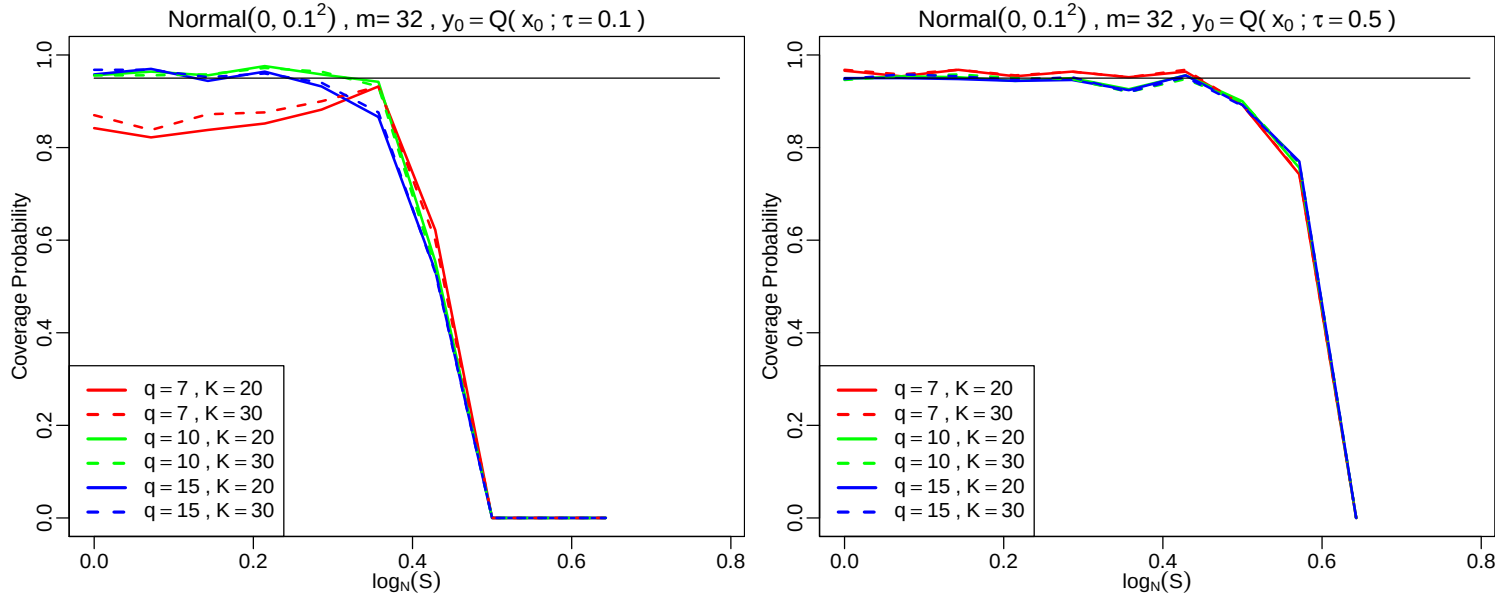
- $\mathbf{B}$: cubic B-spline with $q = \dim(\mathbf{B})$ defined on $G = 4 + q$ equidistant knots. Require $K > q$ so that $\widehat{\boldsymbol{\beta}}(\tau)$ is computable (see (2.2));
- X_i follows a multivariate uniform distribution $\mathcal{U}([0,1]^{m-1})$ with covariance matrix $\Sigma_X := \mathbb{E}[X_i X_i^\top]$ with $\Sigma_{jk} = 0.1^2 0.7^{|j-k|}$ for $j, k = 1, ..., m-1$;
- $x_0 = (1, (m-1)^{-1/2} \mathbf{1}_{m-1}^\top)^\top$ and $N = 2^{14}$;
- $y_0 = Q(x_0; \tau)$ so that $F_{Y|X}(y_0|x_0) = \tau$;
- Our theorem suggest the following CI for τ:

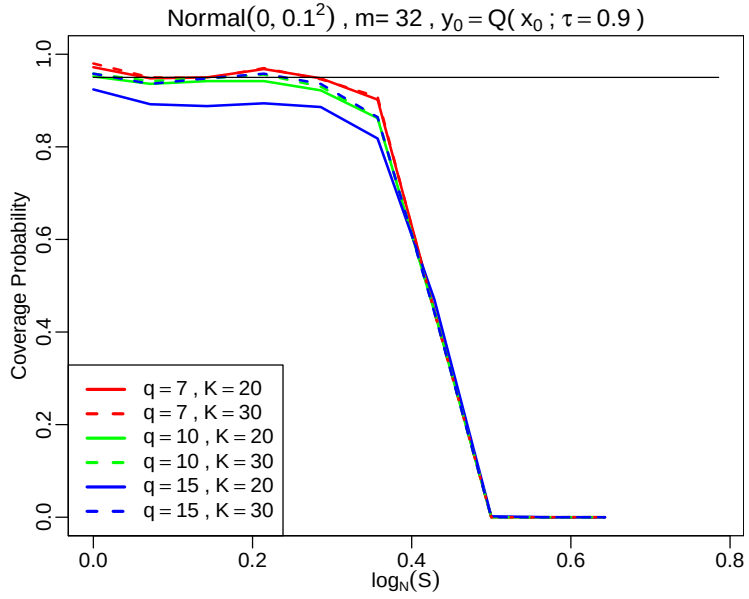
$$\left[\widehat{F}_{Y|X}(Q(x_0;\tau)|x_0) \pm N^{-1/2}\sqrt{\tau(1-\tau)x_0^\top \Sigma_X^{-1} x_0}\,\Phi^{-1}(1-\alpha/2)\right]. \tag{4.1}$$

$F_{Y|X}(y|x)$, $\varepsilon \sim \mathcal{N}(0, 0.1^2)$ and $m = 4$

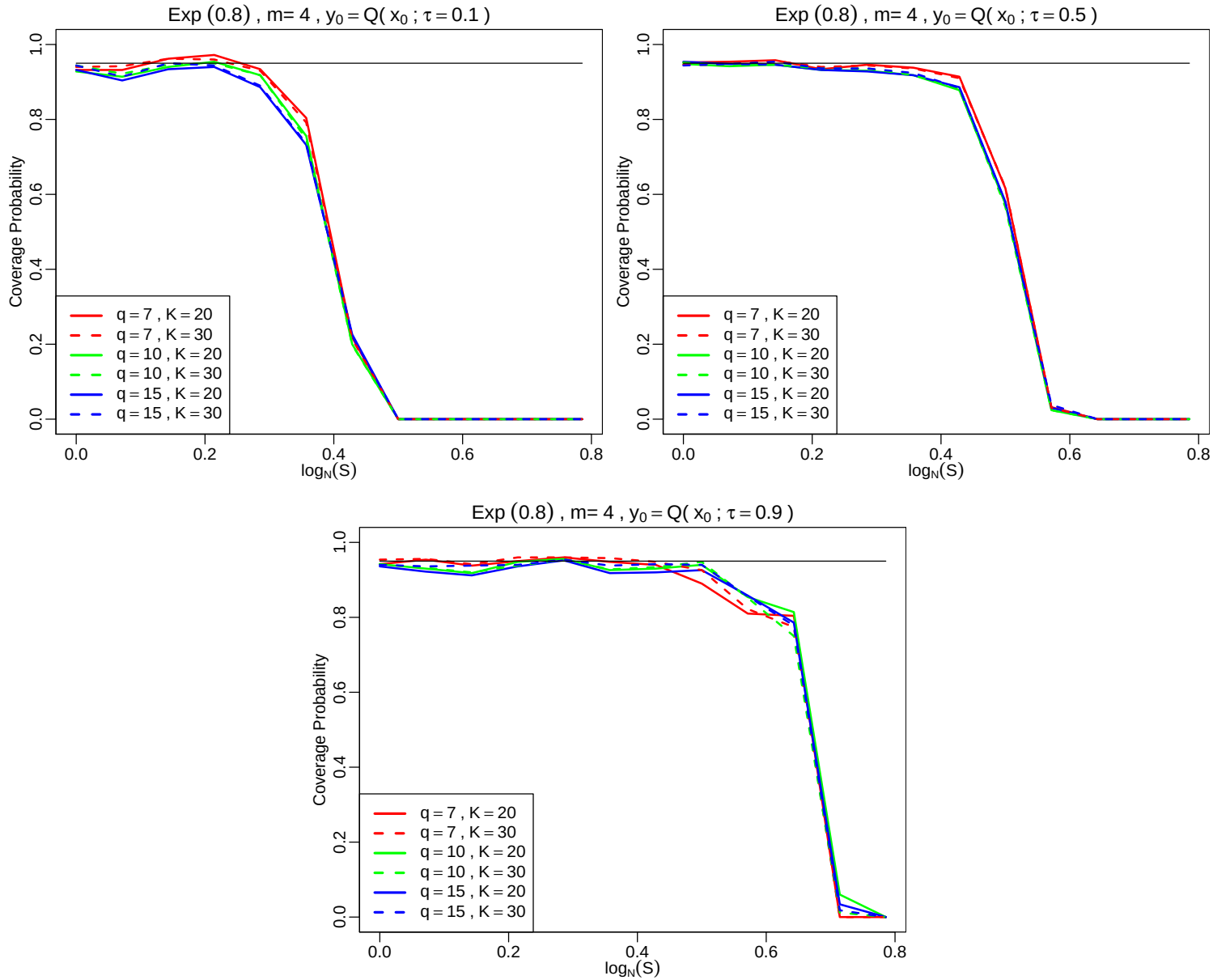


$F_{Y|X}(y|x)$, $\varepsilon \sim \mathcal{N}(0, 0.1^2)$ and $m = 32$

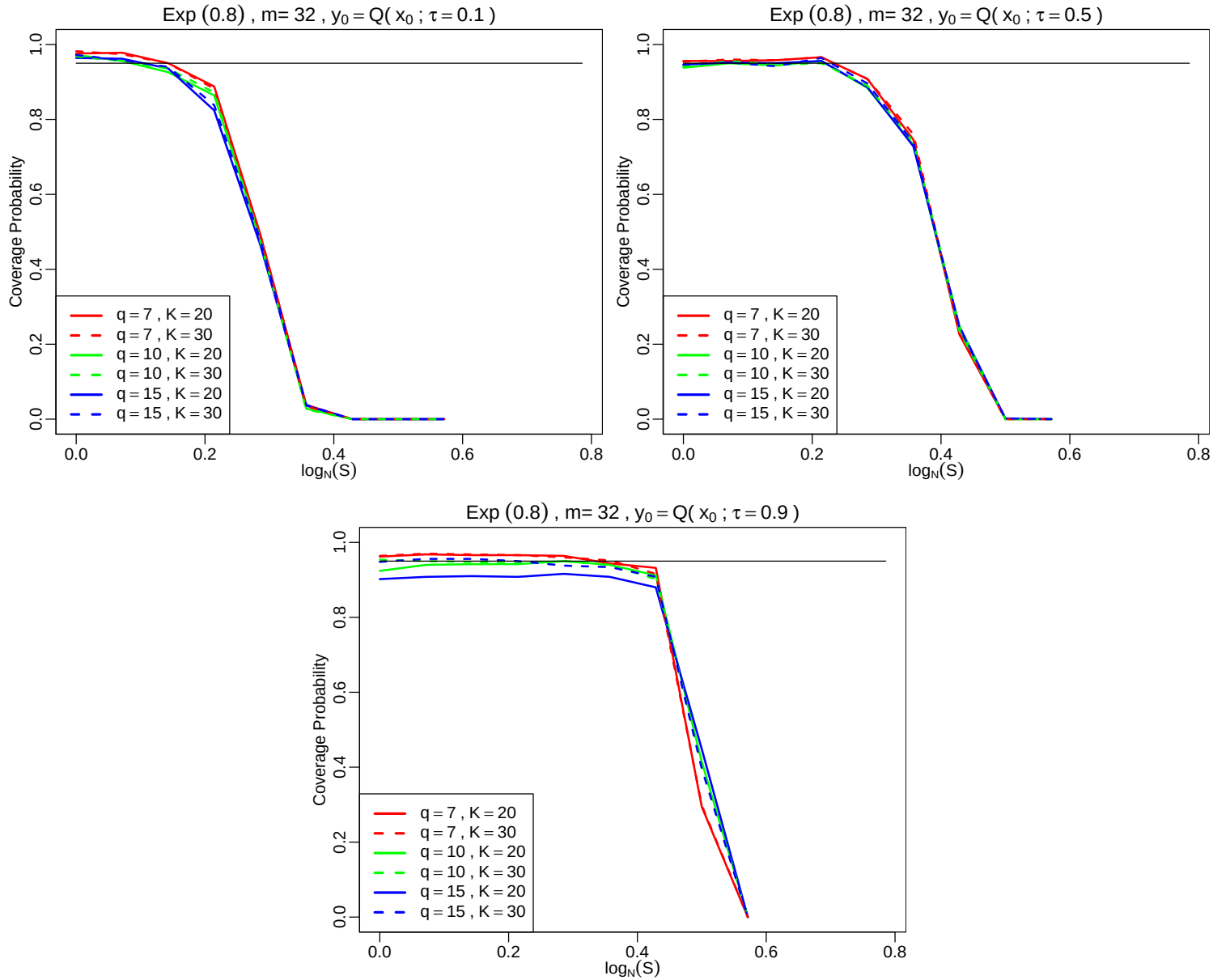




$F_{Y|X}(y|x)$, $\varepsilon \sim \mathbf{Exp}(0.8)$ and $m = 4$



$F_{Y|X}(y|x)$, $\varepsilon \sim \mathbf{Exp}(0.8)$ and $m = 32$



Empirical Observations from Simulations

- Either $S > S^*$ or $q < q^*$ leads to the collapse of the oracle rule.
- Increase in q and K improves the coverage probability;
- Coverage is no longer symmetry around τ even when model error is normal;

Thanks for your attention