Stepwise Variable Selection in Nonparametric Additive Models

Chong Gu

Department of Statistics
Purdue University

June 5, 2014

Outline

1. Problem and Method
   - Nonparametric Sparse Regression
   - Iterative Variable Selection
   - Estimation, Selection, and Screening
   - Forward Addition
2. Empirical Studies
   - Simulation Settings
   - Empirical Performances

Outline

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i = (x_{i\langle 1\rangle}, \ldots, x_{i\langle p\rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j \rangle}) + \epsilon_i$, $i = 1, \dots, n$, where $x_i = (x_{i\langle 1 \rangle}, \dots, x_{i\langle p \rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i = (x_{i\langle 1\rangle}, \ldots, x_{i\langle p\rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i = (x_{i\langle 1\rangle}, \ldots, x_{i\langle p\rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i = (x_{i\langle 1\rangle}, \ldots, x_{i\langle p\rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Sparse Additive Regression Model

- Consider $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j=1}^{p} \eta_j(x_{i\langle j \rangle}) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i = (x_{i\langle 1 \rangle}, \ldots, x_{i\langle p \rangle}) \in [0,1]^p$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Task: Only $d < p$ of the η_j's are nonzero, which we try to identify.
- Existing algorithms are largely LASSO-variants, using some L_1-type penalties to weed out inactive variables.
  - COSSO (Lin & Zhang 2006), SpAM (Ravikumar et al 2007), penGAM (Meier et al 2009), etc.
  - When $p \gg n$, variable-screening is needed to pare down the variable list for the algorithms to work (Fan et al 2011).
- LASSO-type algorithms select variables and estimate the model at the same time, but variable selection and model estimation are different tasks with different objectives.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{S}} \eta_j(x_{i\langle j\rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{S}} \eta_j(x_{i\langle j\rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{P}} \eta_j(x_{i\langle j \rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{S}} \eta_j(x_{i\langle j \rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
    1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{P}} \eta_j(x_{i\langle j \rangle})$ to the data.
    2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{S}} \eta_j(x_{i\langle j \rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
    3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{S}} \eta_j(x_{i\langle j\rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j\in\mathcal{S}} \eta_j(x_{i\langle j\rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Iterative Variable Selection

- Working with a pair of dynamic variable sets $\mathcal{P} \supseteq \mathcal{S}$, one may separate variable selection from estimation.
  1. Estimation: Given a variable pool $\mathcal{P}$, fit $\hat{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{P}} \eta_j(x_{i\langle j \rangle})$ to the data.
  2. Variable Selection: Pick selection set $\mathcal{S} \subseteq \mathcal{P}$ such that $\tilde{\eta}(x)$ of form $\eta(x_i) = \eta_\emptyset + \sum_{j \in \mathcal{S}} \eta_j(x_{i\langle j \rangle})$ nearly achieves the goodness-of-fit of $\hat{\eta}(x)$.
  3. Screening and Updating: Rank variables in $\mathcal{P}^c$ to obtain $\tilde{\mathcal{P}}$, and augment $\mathcal{S}$ by $\tilde{\mathcal{P}}$ to update $\mathcal{P} = \mathcal{S} \cup \tilde{\mathcal{P}}$.
- The size $\tilde{p}$ of $\tilde{P}$ is capped at a moderate number, say 5; $\tilde{p}$ varies with the amount of agreement between consecutive $\mathcal{S}$'s.
- Initial screening is needed to obtain an initial $\mathcal{P}$ of manageable size, but no variables are lost.
- For $p \gg n$, the numerical burden is primarily on variable screening, which is trivially parallelizable.

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^n \Big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\Big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes

$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$

with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).

- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j \in \mathcal{P}} \eta_j(x_{i\langle j \rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j \rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j \rangle})\big)^2 dx_{\langle j \rangle}$$
with $\eta(x) = d_0 + \sum_j d_j \phi(x_{\langle j \rangle}) + \sum_{i=1}^{n} c_i \big(\sum_j \theta_j R(x_{i\langle j \rangle}, x_{\langle j \rangle})\big)$, where $\phi(x)$ and $R(x, y)$ are known functions (Kimeldorf & Wahba 1971).
- It is clear that $\eta_j(x_{\langle j \rangle}) = d_j \phi(x_{\langle j \rangle}) + \theta_j \sum_i c_i R(x_{i\langle j \rangle}, x_{\langle j \rangle})$.
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^n \Big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\Big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- One may calculate an asymptotically efficient approximation of form $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{k=1}^q c_k\big(\sum_j \theta_j R(z_{k\langle j\rangle}, x_{\langle j\rangle})\big)$, for $\{z_k\} \subset \{x_i\}$ a random subset of size $q \asymp n^{2/9}$ (Gu & Kim 2002).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\Big)^2 + \lambda \sum_j \theta_j^{-1}\int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^{n} c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).
- $\sum_j \theta_j^{-1}\int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle} = \sum_j \theta_j\, \mathbf{c}^T Q_j \mathbf{c}$; $\int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle} = \theta_j^2\, \mathbf{c}^T Q_j \mathbf{c}$.
- First set $\tilde\theta_j^{-1} \propto \mathrm{tr} Q_j$ and fit $\tilde\eta$, then set $\theta_j \propto \tilde\theta_j^2\, \tilde{\mathbf{c}}^T Q_j \tilde{\mathbf{c}}$.

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
  $$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
  with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).

- $\sum_j \theta_j^{-1}\int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle} = \sum_j \theta_j\, \mathbf{c}^T Q_j \mathbf{c}$; $\int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle} = \theta_j^2\, \mathbf{c}^T Q_j \mathbf{c}$.
- First set $\tilde{\theta}_j^{-1} \propto \mathrm{tr} Q_j$ and fit $\tilde{\eta}$, then set $\theta_j \propto \tilde{\theta}_j^2\, \tilde{\mathbf{c}}^T Q_j \tilde{\mathbf{c}}$.

Cubic Spline Additive Models

- For estimation given $\mathcal{P}$, one may use cubic spline additive models.
- To fit $Y_i = \eta(x_i) + \epsilon_i = \eta_\emptyset + \sum_{j\in\mathcal{P}} \eta_j(x_{i\langle j\rangle}) + \epsilon_i$, one minimizes
$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta_\emptyset - \sum_j \eta_j(x_{i\langle j\rangle})\big)^2 + \lambda \sum_j \theta_j^{-1} \int_0^1 \big(\eta_j''(x_{\langle j\rangle})\big)^2 dx_{\langle j\rangle}$$
with $\eta(x) = d_0 + \sum_j d_j\phi(x_{\langle j\rangle}) + \sum_{i=1}^n c_i\big(\sum_j \theta_j R(x_{i\langle j\rangle}, x_{\langle j\rangle})\big)$, where $\phi(x)$ and $R(x,y)$ are known functions (Kimeldorf & Wahba 1971).
- Fixing θ_j, one may select λ using GCV (Craven & Wahba 1979).
- For θ_j, use the starting value algorithm of Gu & Wahba (1991).
- The computational cost is comparable to that with $p = 1$.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n} \sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n} \sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n} \sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n} \sum_i \left(g(x_i) - h(x_i)\right)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n}\sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n}\sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n} \sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Square Error Projection

- To select $\mathcal{S} \subseteq \mathcal{P}$, one may use square error projection, which is designed to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to get $\tilde{\eta}$. Inspect $\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c)$ for some η_c degenerate. (Gu 2004)
- We use $\mathrm{KL}(g, h) = \mathrm{SE}(g, h) = \frac{1}{n}\sum_i \big(g(x_i) - h(x_i)\big)^2$ and $\eta_c = \bar{Y}$.
- Forward Variable Selection: Given $\hat{\eta}$ fitted to variables in $\mathcal{P}$,
  1. Set $\mathcal{S} = \emptyset$, $\mathrm{SE} = \mathrm{SE}(\eta_c, \eta_c) = 0$, and $\mathrm{SE}_0 = \mathrm{SE}(\hat{\eta} - \eta_c)$.
  2. For $j \in \mathcal{P} \setminus \mathcal{S}$, add $x_{\langle j \rangle}$ to $\mathcal{S}$ one at a time and calculate the resulting $\mathrm{SE}(\tilde{\eta}, \eta_c)$; retain the largest $\mathrm{SE}(\tilde{\eta}^*, \eta_c)$ associated with $x_{\langle * \rangle}$.
  3. If $(\mathrm{SE}(\tilde{\eta}^*, \eta_c) - \mathrm{SE})/\mathrm{SE}_0 > \delta_0$, add $x_{\langle * \rangle}$ to $\mathcal{S}$ and update $\mathrm{SE} = \mathrm{SE}(\tilde{\eta}^*, \eta_c)$. Otherwise, stop.
  4. If $\mathrm{SE}/\mathrm{SE}_0 < 1 - \delta$, go to Step 2. Otherwise, stop.
- Default values $(\delta, \delta_0) = (.01, .002)$ are used to control selection size.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k \in \mathcal{S} \cup \{j\}} \eta_k(x_{i\langle k \rangle})$ via the minimization of
$$\frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \sum_k \eta_k(x_{i\langle k \rangle}) \right)^2 + \lambda \sum_k \theta_k^{-1} \int_0^1 \left(\eta_k''(x_{\langle j \rangle}) \right)^2 dx_{\langle k \rangle},$$
and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k\in\mathcal{S}\cup\{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of
$$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\right)^2 + \lambda\sum_k \theta_k^{-1}\int_0^1 \left(\eta_k''(x_{\langle j\rangle})\right)^2 dx_{\langle k\rangle},$$
and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k\in\mathcal{S}\cup\{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of
$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\big)^2 + \lambda \sum_k \theta_k^{-1}\int_0^1 \big(\eta_k''(x_{\langle j\rangle})\big)^2 dx_{\langle k\rangle},$$
and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S}=\emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k\in\mathcal{S}\cup\{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of
$$\frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\Big)^2 + \lambda\sum_k \theta_k^{-1}\int_0^1 \big(\eta_k''(x_{\langle j\rangle})\big)^2 dx_{\langle k\rangle},$$
and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k \in \mathcal{S} \cup \{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of

$$\frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\Big)^2 + \lambda \sum_k \theta_k^{-1} \int_0^1 \big(\eta_k''(x_{\langle j\rangle})\big)^2 dx_{\langle k\rangle},$$

  and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k \in \mathcal{S} \cup \{j\}} \eta_k(x_{i\langle k \rangle})$ via the minimization of

$$\frac{1}{n}\sum_{i=1}^n \Big(Y_i - \sum_k \eta_k(x_{i\langle k \rangle})\Big)^2 + \lambda \sum_k \theta_k^{-1} \int_0^1 \big(\eta_k''(x_{\langle j \rangle})\big)^2 dx_{\langle k \rangle},$$

  and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k\in\mathcal{S}\cup\{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of
$$\frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\big)^2 + \lambda\sum_k \theta_k^{-1}\int_0^1 \big(\eta_k''(x_{\langle j\rangle})\big)^2 dx_{\langle k\rangle},$$
and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Variable Screening

- To screen variables given $\mathcal{S}$, fit $\hat{\eta}_j$ of form $\eta_\emptyset + \sum_{k\in\mathcal{S}\cup\{j\}} \eta_k(x_{i\langle k\rangle})$ via the minimization of

$$\frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \sum_k \eta_k(x_{i\langle k\rangle})\Big)^2 + \lambda\sum_k \theta_k^{-1}\int_0^1 \big(\eta_k''(x_{\langle j\rangle})\big)^2 dx_{\langle k\rangle},$$

  and rank variables by $\mathrm{SE}(\hat{\eta}_j, \eta_c)$.
- For the initial screening, use $\mathcal{S} = \emptyset$ in above.
- Doable, fully nonparametric, but can be time-consuming!
- Initialization: Perform the following to obtain the initial $\mathcal{P}$.
  1. Form $\mathcal{P}_0$ using the top p^* variables from the initial screening.
  2. Given $\hat{\eta}$ fitted to $\mathcal{P}_0$, select $\mathcal{S}_0 \subseteq \mathcal{P}_0$.
  3. Rank variables in $\mathcal{S}_0^c$ to obtain $\tilde{\mathcal{P}}$, and form $\mathcal{P} = \mathcal{S}_0 \cup \tilde{\mathcal{P}}$.
- The size of $\mathcal{P}_0$ can be large, say $p^* = n/4$.
- The size of $\tilde{\mathcal{P}}$ should be moderate, say 5.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.

- For each variable $x_{i\langle j \rangle}$ in $\mathcal{S}^c$, permute $i^* = \pi(i)$ to obtain $x_{i\langle * \rangle} = x_{i^*\langle j \rangle}$.
- Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$, project $\hat{\eta}_*$ into $\mathcal{S}$, and obtain $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$.

- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of “null” $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.

- For each variable $x_{i\langle j \rangle}$ in $\mathcal{S}^c$, permute $i^* = \pi(i)$ to obtain $x_{i\langle * \rangle} = x_{i^*\langle j \rangle}$.
- Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$, project $\hat{\eta}_*$ into $\mathcal{S}$, and obtain $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$.

- The empirical threshold only works after $\mathcal{S}$ has captured most of the “signal” in the data.

Forward Addition

- Forward Addition: Given the converged $\mathcal{S}$ from iterative selection, weaker variables may be further added.
  1. Screen variables in $\mathcal{S}^c$ to obtain the top one, say $x_{\langle * \rangle}$.
  2. Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$ and project $\hat{\eta}_*$ into $\mathcal{S}$ to obtain $\tilde{\eta}_*$.
  3. If $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ exceeds a threshold, add $x_{\langle * \rangle}$ to $\mathcal{S}$, go to Step 1. Otherwise stop.
- A pool of "null" $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$ values can be obtained by decoupling Y_i from x_i via permutation, based on which one may set the threshold.
  - For each variable $x_{i\langle j \rangle}$ in $\mathcal{S}^c$, permute $i^* = \pi(i)$ to obtain $x_{i\langle * \rangle} = x_{i^*\langle j \rangle}$.
  - Fit $\hat{\eta}_*$ to $\mathcal{S} \cup \{x_{\langle * \rangle}\}$, project $\hat{\eta}_*$ into $\mathcal{S}$, and obtain $\mathrm{SE}(\hat{\eta}_*, \tilde{\eta}_*)$.
- The empirical threshold only works after $\mathcal{S}$ has captured most of the "signal" in the data.

Outline

Simulation Settings

- Simulations are conducted on some standard test examples in the literature. (Lin & Zhang 2006, Fan et al 2011, etc)

- Two test functions based on univariate g_1, g_2, g_3, g_4, and $\epsilon \sim N(0,1)$:

$$
\begin{aligned}
Y &= 5g_1(x_{\langle 1\rangle}) + 3g_2(x_{\langle 1\rangle}) + 4g_3(x_{\langle 1\rangle}) + 6g_4(x_{\langle 1\rangle}) + \sqrt{1.74}\,\epsilon; \\
Y &= g_1(x_{\langle 1\rangle}) + g_2(x_{\langle 2\rangle}) + g_3(x_{\langle 3\rangle}) + g_4(x_{\langle 4\rangle}) \\
&\quad + 1.5\{g_1(x_{\langle 5\rangle}) + g_2(x_{\langle 6\rangle}) + g_3(x_{\langle 7\rangle}) + g_4(x_{\langle 8\rangle})\} \\
&\quad + 2\{g_1(x_{\langle 9\rangle}) + g_2(x_{\langle 10\rangle}) + g_3(x_{\langle 11\rangle}) + g_4(x_{\langle 12\rangle}) + \sqrt{.5184}\,\epsilon.
\end{aligned}
$$

- Three designs: $x_{\langle j\rangle} = (w_j + \alpha u)/(1+\alpha)$, for $w_j, u \sim U(0,1)$ and $\alpha = 0, 1, 3$.
- One hundred replicates each with $n = 400$ and $p = 100, 1000$.

Simulation Settings

- Simulations are conducted on some standard test examples in the literature. (Lin & Zhang 2006, Fan et al 2011, etc)

- Two test functions based on univariate g_1, g_2, g_3, g_4, and $\epsilon \sim N(0,1)$:

$$
\begin{aligned}
Y &= 5g_1(x_{\langle 1\rangle}) + 3g_2(x_{\langle 1\rangle}) + 4g_3(x_{\langle 1\rangle}) + 6g_4(x_{\langle 1\rangle}) + \sqrt{1.74}\,\epsilon; \\
Y &= g_1(x_{\langle 1\rangle}) + g_2(x_{\langle 2\rangle}) + g_3(x_{\langle 3\rangle}) + g_4(x_{\langle 4\rangle}) \\
&\quad + 1.5\big\{g_1(x_{\langle 5\rangle}) + g_2(x_{\langle 6\rangle}) + g_3(x_{\langle 7\rangle}) + g_4(x_{\langle 8\rangle})\big\} \\
&\quad + 2\big\{g_1(x_{\langle 9\rangle}) + g_2(x_{\langle 10\rangle}) + g_3(x_{\langle 11\rangle}) + g_4(x_{\langle 12\rangle}) + \sqrt{.5184}\,\epsilon.
\end{aligned}
$$

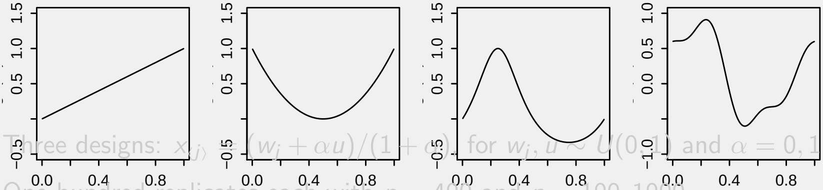
- Three designs: $x_{\langle j\rangle} = (w_j + \alpha u)/(1+\alpha)$, for $w_j, u \sim U(0,1)$ and $\alpha = 0, 1, 3$.
- One hundred replicates each with $n = 400$ and $p = 100, 1000$.

Simulation Settings

- Simulations are conducted on some standard test examples in the literature. (Lin & Zhang 2006, Fan et al 2011, etc)

- Two test functions based on univariate g_1, g_2, g_3, g_4, and $\epsilon \sim N(0,1)$:

$$
\begin{aligned}
Y &= 5g_1(x_{\langle 1\rangle}) + 3g_2(x_{\langle 1\rangle}) + 4g_3(x_{\langle 1\rangle}) + 6g_4(x_{\langle 1\rangle}) + \sqrt{1.74}\,\epsilon;\\
Y &= g_1(x_{\langle 1\rangle}) + g_2(x_{\langle 2\rangle}) + g_3(x_{\langle 3\rangle}) + g_4(x_{\langle 4\rangle})\\
&\quad + 1.5\{g_1(x_{\langle 5\rangle}) + g_2(x_{\langle 6\rangle}) + g_3(x_{\langle 7\rangle}) + g_4(x_{\langle 8\rangle})\}\\
&\quad + 2\{g_1(x_{\langle 9\rangle}) + g_2(x_{\langle 10\rangle}) + g_3(x_{\langle 11\rangle}) + g_4(x_{\langle 12\rangle}) + \sqrt{.5184}\,\epsilon.
\end{aligned}
$$



- Three designs: $x_{\langle j\rangle} = (w_j + \alpha u)/(1+\alpha)$, for $w_j, u \sim U(0,1)$ and $\alpha = 0, 1, 3$.
- One hundred replicates each with $n = 400$ and $p = 100, 1000$.

Simulation Settings

- Simulations are conducted on some standard test examples in the literature. (Lin & Zhang 2006, Fan et al 2011, etc)

- Two test functions based on univariate g_1, g_2, g_3, g_4, and $\epsilon \sim N(0,1)$:

$$
\begin{aligned}
Y &= 5g_1(x_{\langle 1\rangle}) + 3g_2(x_{\langle 1\rangle}) + 4g_3(x_{\langle 1\rangle}) + 6g_4(x_{\langle 1\rangle}) + \sqrt{1.74}\,\epsilon;\\
Y &= g_1(x_{\langle 1\rangle}) + g_2(x_{\langle 2\rangle}) + g_3(x_{\langle 3\rangle}) + g_4(x_{\langle 4\rangle})\\
&\quad + 1.5\{g_1(x_{\langle 5\rangle}) + g_2(x_{\langle 6\rangle}) + g_3(x_{\langle 7\rangle}) + g_4(x_{\langle 8\rangle})\}\\
&\quad + 2\{g_1(x_{\langle 9\rangle}) + g_2(x_{\langle 10\rangle}) + g_3(x_{\langle 11\rangle}) + g_4(x_{\langle 12\rangle}) + \sqrt{.5184}\,\epsilon.
\end{aligned}
$$

- Three designs: $x_{\langle j\rangle} = (w_j + \alpha u)/(1+\alpha)$, for $w_j, u \sim U(0,1)$ and $\alpha = 0, 1, 3$.
- One hundred replicates each with $n = 400$ and $p = 100, 1000$.

Simulation Settings

- Simulations are conducted on some standard test examples in the literature. (Lin & Zhang 2006, Fan et al 2011, etc)

- Two test functions based on univariate g_1, g_2, g_3, g_4, and $\epsilon \sim N(0,1)$:

$$\begin{aligned} Y &= 5g_1(x_{\langle 1\rangle}) + 3g_2(x_{\langle 1\rangle}) + 4g_3(x_{\langle 1\rangle}) + 6g_4(x_{\langle 1\rangle}) + \sqrt{1.74}\,\epsilon; \\ Y &= g_1(x_{\langle 1\rangle}) + g_2(x_{\langle 2\rangle}) + g_3(x_{\langle 3\rangle}) + g_4(x_{\langle 4\rangle}) \\ &\quad + 1.5\{g_1(x_{\langle 5\rangle}) + g_2(x_{\langle 6\rangle}) + g_3(x_{\langle 7\rangle}) + g_4(x_{\langle 8\rangle})\} \\ &\quad + 2\{g_1(x_{\langle 9\rangle}) + g_2(x_{\langle 10\rangle}) + g_3(x_{\langle 11\rangle}) + g_4(x_{\langle 12\rangle}) + \sqrt{.5184}\,\epsilon. \end{aligned}$$

- Three designs: $x_{\langle j\rangle} = (w_j + \alpha u)/(1+\alpha)$, for $w_j, u \sim U(0,1)$ and $\alpha = 0, 1, 3$.
- One hundred replicates each with $n = 400$ and $p = 100, 1000$.

Empirical Performances

- Selection results from 100 replicates each, for $p = 1000$ and $p = 100$:

- The numbers look okay for the independent and the mildly correlated designs, but not so good for the highly correlated design.
- But are TP and FP the only performance measures here?

Empirical Performances

- Selection results from 100 replicates each, for $p = 1000$ and $p = 100$:

	$d = 4$		$d = 12$	
	TP	FP	TP	FP
$\alpha = 0$	4.00, (4,4)	0.03, (0,1)	10.82, (9,12)	0.03, (0,1)
	4.00, (4,4)	0.03, (0,1)	11.58, (9,12)	0.03, (0,1)
$\alpha = 1$	3.99, (3,4)	0.15, (0,2)	9.18, (6,11)	0.18, (0,2)
	4.00, (4,4)	0.09, (0,2)	9.97, (8,12)	0.12, (0,2)
$\alpha = 3$	3.23, (2,4)	0.19, (0,2)	5.54, (4,8)	0.11, (0,2)
	3.79, (2,4)	0.02, (0,1)	6.42, (5,9)	0.08, (0,1)

- The numbers look okay for the independent and the mildly correlated designs, but not so good for the highly correlated design.
- But are TP and FP the only performance measures here?

Empirical Performances

- Selection results from 100 replicates each, for $p = 1000$ and $p = 100$:

	$d = 4$		$d = 12$	
	TP	FP	TP	FP
$\alpha = 0$	4.00, (4,4)	0.03, (0,1)	10.82, (9,12)	0.03, (0,1)
	4.00, (4,4)	0.03, (0,1)	11.58, (9,12)	0.03, (0,1)
$\alpha = 1$	3.99, (3,4)	0.15, (0,2)	9.18, (6,11)	0.18, (0,2)
	4.00, (4,4)	0.09, (0,2)	9.97, (8,12)	0.12, (0,2)
$\alpha = 3$	3.23, (2,4)	0.19, (0,2)	5.54, (4,8)	0.11, (0,2)
	3.79, (2,4)	0.02, (0,1)	6.42, (5,9)	0.08, (0,1)

- The numbers look okay for the independent and the mildly correlated designs, but not so good for the highly correlated design.
- But are TP and FP the only performance measures here?

Empirical Performances

- Selection results from 100 replicates each, for $p = 1000$ and $p = 100$:

	$d = 4$		$d = 12$	
	TP	FP	TP	FP
$\alpha = 0$	4.00, (4,4)	0.03, (0,1)	10.82, (9,12)	0.03, (0,1)
	4.00, (4,4)	0.03, (0,1)	11.58, (9,12)	0.03, (0,1)
$\alpha = 1$	3.99, (3,4)	0.15, (0,2)	9.18, (6,11)	0.18, (0,2)
	4.00, (4,4)	0.09, (0,2)	9.97, (8,12)	0.12, (0,2)
$\alpha = 3$	3.23, (2,4)	0.19, (0,2)	5.54, (4,8)	0.11, (0,2)
	3.79, (2,4)	0.02, (0,1)	6.42, (5,9)	0.08, (0,1)

- The numbers look okay for the independent and the mildly correlated designs, but not so good for the highly correlated design.
- But are TP and FP the only performance measures here?

Empirical Performances

- For estimation performance, one may compare the MSE of the selected model with that of the perfect selection.
- When FP=0, one may use square error projection to assess the prediction performance of the selected model.

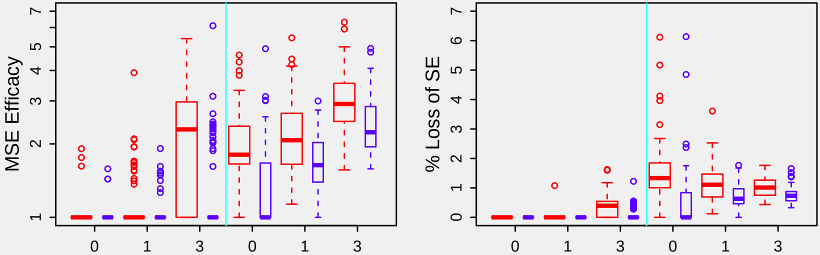
Empirical Performances

- For estimation performance, one may compare the MSE of the selected model with that of the perfect selection.
- When FP=0, one may use square error projection to assess the prediction performance of the selected model.

Empirical Performances

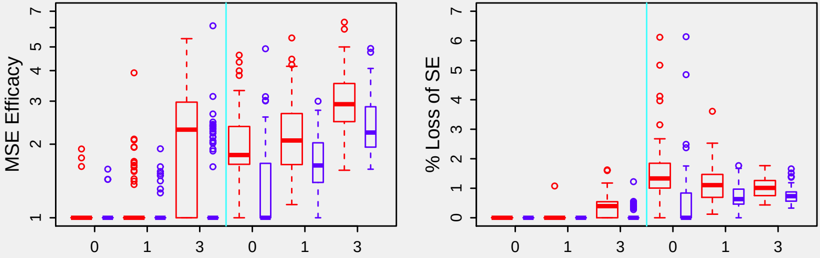
- For estimation performance, one may compare the MSE of the selected model with that of the perfect selection.
- When FP=0, one may use square error projection to assess the prediction performance of the selected model.

- Left: Ratio of $\sum_i \left(\hat{\eta}(x_i) - \eta(x_i)\right)^2$.
- Right: $SE(\hat{\eta}, \tilde{\eta})/SE(\hat{\eta}, \eta_c)$, for $\hat{\eta}$ the "perfect" fit and $\tilde{\eta}$ its projection in $\mathcal{S}$.



- For estimation performance, one may compare the MSE of the selected model with that of the perfect selection.
- When FP=0, one may use square error projection to assess the prediction performance of the selected model.

- Left: Ratio of $\sum_i \left(\hat\eta(x_i) - \eta(x_i)\right)^2$.
- Right: $SE(\hat\eta, \tilde\eta)/SE(\hat\eta, \eta_c)$, for $\hat\eta$ the "perfect" fit and $\tilde\eta$ its projection in $\mathcal{S}$.



Behind the Scene

```
[1] "p=100"
[1] "initial"
[1] "X12" "X8"  "X71" "X10" "X11" "X74"
[1] "phase 1"
[1] "X12" "X8"  "X11" "X10" "X7"
[1] "phase 2"
[1] "X12" "X8"  "X11" "X10" "X7"  "X3"
[1] "X12" "X8"  "X11" "X10" "X7"  "X3"  "X4"

[1] "p=1000"
[1] "initial"
[1] "X968" "X662" "X367" "X242" "X195" "X3"
[1] "phase 1"
[1] "X12"  "X8"   "X11"  "X195" "X4"
[1] "X12" "X8"  "X11" "X10" "X4"  "X6"
[1] "X12" "X8"  "X11" "X10" "X4"
[1] "phase 2"
[1] "X12" "X8"  "X11" "X10" "X4"  "X3"
```

Discussion

Comparison with existing methods (a.k.a., Fan et al 2011).

- Comparable performance, but *much* slower.
- Can handle highly correlated x's (to a degree).
- Can handle interaction terms?

Discussion

Comparison with existing methods (a.k.a., Fan et al 2011).

- Comparable performance, but *much* slower.
- Can handle highly correlated x's (to a degree).
- Can handle interaction terms?

Discussion

Comparison with existing methods (a.k.a., Fan et al 2011).

- Comparable performance, but *much* slower.
- Can handle highly correlated x's (to a degree).
- Can handle interaction terms?

Discussion

Comparison with existing methods (a.k.a., Fan et al 2011).

- Comparable performance, but *much* slower.
- Can handle highly correlated x's (to a degree).
- Can handle interaction terms?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

Discussion

Issues to further explore.

- Size of $\tilde{\mathcal{P}}$ increasing with p, collinearity, or both?
- Adaptive choices of (δ, δ_0)?
- Extension to non-Gaussian regression.
- Theory?
- Speeding up screening – skipping CV, saving basis?

The End

Thank You!