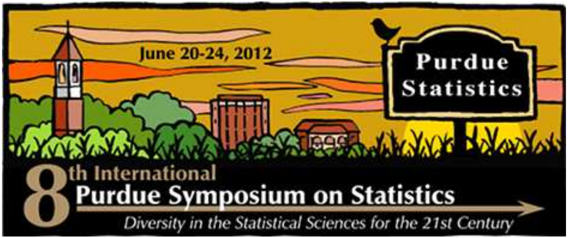
Smoothing Spline ANOVA Models

Chong Gu

Department of Statistics
Purdue University

June 21, 2012



Outline

1. Introduction
   - Cubic Spline and Penalized Likelihood
   - Functional ANOVA Decomposition
   - R Package gss
2. Estimation and Inference
   - Splines as Bayes Estimates
   - Efficient Approximation
   - Cross-Validation
   - Bayesian Confidence Intervals
   - Kullback-Leibler Projection
3. Regression Models
   - Non-Gaussian Regression
   - Regression with Correlated Data
4. Density and Hazard Estimation
   - Density and Conditional Density
   - Hazard and Relative Risk

Outline

1. Introduction
   - Cubic Spline and Penalized Likelihood
   - Functional ANOVA Decomposition
   - R Package gss
2. Estimation and Inference
   - Splines as Bayes Estimates
   - Efficient Approximation
   - Cross-Validation
   - Bayesian Confidence Intervals
   - Kullback-Leibler Projection
3. Regression Models
   - Non-Gaussian Regression
   - Regression with Correlated Data
4. Density and Hazard Estimation
   - Density and Conditional Density
   - Hazard and Relative Risk

Outline

Outline

Outline

1. Introduction
   - Cubic Spline and Penalized Likelihood
   - Functional ANOVA Decomposition
   - R Package gss
2. Estimation and Inference
   - Splines as Bayes Estimates
   - Efficient Approximation
   - Cross-Validation
   - Bayesian Confidence Intervals
   - Kullback-Leibler Projection
3. Regression Models
   - Non-Gaussian Regression
   - Regression with Correlated Data
4. Density and Hazard Estimation
   - Density and Conditional Density
   - Hazard and Relative Risk

Cubic Smoothing Spline

- Problem: Observing $Y_i = \eta(x_i) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i \in [0,1]$ and $\epsilon_i \sim N(0, \sigma^2)$, one is to estimate $\eta(x)$.
- Method: Minimize $\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \eta(x_i)\right)^2 + \lambda \int_0^1 \left(\eta''(x)\right)^2 dx$.
- Solution: Piecewise cubic polynomial, with $\eta^{(3)}(x)$ jumping at knots $\xi_1 < \xi_2 < \cdots < \xi_q$ (ordered distinctive x_i); linear on $[0, \xi_1] \cup [\xi_q, 1]$.
- Smoothing Parameter: As $\lambda \uparrow$, $\eta(x)$ gets smoother; at $\lambda = 0_+$, one interpolates data, at $\lambda = \infty$, $\eta(x) = \beta_0 + \beta_1 x$.

Cubic Smoothing Spline

- Problem: Observing $Y_i = \eta(x_i) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i \in [0, 1]$ and $\epsilon_i \sim N(0, \sigma^2)$, one is to estimate $\eta(x)$.
- Method: Minimize $\frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \eta(x_i)\big)^2 + \lambda \int_0^1 \big(\eta''(x)\big)^2 dx$.
- Solution: Piecewise cubic polynomial, with $\eta^{(3)}(x)$ jumping at knots $\xi_1 < \xi_2 < \cdots < \xi_q$ (ordered distinctive x_i); linear on $[0, \xi_1] \cup [\xi_q, 1]$.
- Smoothing Parameter: As $\lambda \uparrow$, $\eta(x)$ gets smoother; at $\lambda = 0_+$, one interpolates data, at $\lambda = \infty$, $\eta(x) = \beta_0 + \beta_1 x$.

Cubic Smoothing Spline

- Problem: Observing $Y_i = \eta(x_i) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i \in [0, 1]$ and $\epsilon_i \sim N(0, \sigma^2)$, one is to estimate $\eta(x)$.
- Method: Minimize $\frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \eta(x_i)\big)^2 + \lambda\int_0^1 \big(\eta''(x)\big)^2 dx$.
- Solution: Piecewise cubic polynomial, with $\eta^{(3)}(x)$ jumping at knots $\xi_1 < \xi_2 < \cdots < \xi_q$ (ordered distinctive x_i); linear on $[0, \xi_1] \cup [\xi_q, 1]$.
- Smoothing Parameter: As $\lambda \uparrow$, $\eta(x)$ gets smoother; at $\lambda = 0_+$, one interpolates data, at $\lambda = \infty$, $\eta(x) = \beta_0 + \beta_1 x$.

Cubic Smoothing Spline

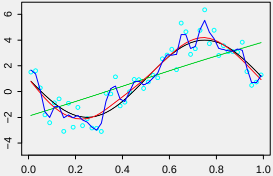
- Problem: Observing $Y_i = \eta(x_i) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i \in [0, 1]$ and $\epsilon_i \sim N(0, \sigma^2)$, one is to estimate $\eta(x)$.
- Method: Minimize $\frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \eta(x_i)\big)^2 + \lambda\int_0^1\big(\eta''(x)\big)^2 dx$.
- Solution: Piecewise cubic polynomial, with $\eta^{(3)}(x)$ jumping at knots $\xi_1 < \xi_2 < \cdots < \xi_q$ (ordered distinctive x_i); linear on $[0, \xi_1] \cup [\xi_q, 1]$.
- Smoothing Parameter: As $\lambda \uparrow$, $\eta(x)$ gets smoother; at $\lambda = 0_+$, one interpolates data, at $\lambda = \infty$, $\eta(x) = \beta_0 + \beta_1 x$.

Cubic Smoothing Spline

- Problem: Observing $Y_i = \eta(x_i) + \epsilon_i$, $i = 1, \ldots, n$, where $x_i \in [0, 1]$ and $\epsilon_i \sim N(0, \sigma^2)$, one is to estimate $\eta(x)$.
- Method: Minimize $\frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \eta(x_i)\big)^2 + \lambda \int_0^1 \big(\eta''(x)\big)^2 dx$.
- Solution: Piecewise cubic polynomial, with $\eta^{(3)}(x)$ jumping at knots $\xi_1 < \xi_2 < \cdots < \xi_q$ (ordered distinctive x_i); linear on $[0, \xi_1] \cup [\xi_q, 1]$.
- Smoothing Parameter: As $\lambda \uparrow$, $\eta(x)$ gets smoother; at $\lambda = 0_+$, one interpolates data, at $\lambda = \infty$, $\eta(x) = \beta_0 + \beta_1 x$.

Role of Smoothing Parameter

- Simulation: $n = 50$, equally spaced x_i, $\eta(x) = 1 + 3\sin(2\pi x - \pi)$, $\sigma^2 = 1$.
- Cubic spline estimates calculated with $\log_{10} n\lambda = 0$, -3, and -6.



Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.
- Equivalence of penalized and constrained MLE:
$$\min L(\eta) + \lambda J(\eta) \;\Leftrightarrow\; \min L(\eta), \text{ subject to } J(\eta) \le \rho,$$
with λ being the Lagrange multiplier.

Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.

- Equivalence of penalized and constrained MLE:

$$\min L(\eta) + \lambda J(\eta) \;\Leftrightarrow\; \min L(\eta), \text{ subject to } J(\eta) \le \rho,$$

with λ being the Lagrange multiplier.

Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.

Examples of Penalized Likelihood Estimation

- Gaussian Regression: $Y_i \sim N\big(\eta(x_i), \sigma^2\big)$; $L(\eta) = \frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2$, with σ^2 absorbed into λ.
- Logistic Regression: $Y_i \sim \text{Bin}\big(m_i, p(x_i)\big)$; for $\eta(x) = \log\frac{p(x)}{1-p(x)}$, $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \big\{Y_i\eta(x_i) - m_i \log(1+e^{\eta(x)})\big\}$.
- Density Estimation: $X_i \sim f(x)$ on $\mathcal{X}$, where $f(x) = e^{\eta(x)}/\int_{\mathcal{X}} e^{\eta(x)}dx$; $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \eta(X_i) + \log\int_{\mathcal{X}} e^{\eta(x)}dx$.

- Equivalence of penalized and constrained MLE:

$$\min L(\eta) + \lambda J(\eta) \iff \min L(\eta), \text{ subject to } J(\eta) \le \rho,$$

with λ being the Lagrange multiplier.

Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.

Examples of Penalized Likelihood Estimation

- Gaussian Regression: $Y_i \sim N\big(\eta(x_i), \sigma^2\big)$; $L(\eta) = \frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2$, with σ^2 absorbed into λ.
- Logistic Regression: $Y_i \sim \mathrm{Bin}\big(m_i, p(x_i)\big)$; for $\eta(x) = \log\frac{p(x)}{1-p(x)}$, $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \big\{Y_i\eta(x_i) - m_i \log(1 + e^{\eta(x)})\big\}$.
- Density Estimation: $X_i \sim f(x)$ on $\mathcal{X}$, where $f(x) = e^{\eta(x)} / \int_{\mathcal{X}} e^{\eta(x)} dx$; $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)} dx$.

- Equivalence of penalized and constrained MLE:

$$\min L(\eta) + \lambda J(\eta) \;\Leftrightarrow\; \min L(\eta), \text{ subject to } J(\eta) \leq \rho,$$

with λ being the Lagrange multiplier.

Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.

Examples of Penalized Likelihood Estimation

- Gaussian Regression: $Y_i \sim N\big(\eta(x_i), \sigma^2\big)$; $L(\eta) = \frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2$, with σ^2 absorbed into λ.
- Logistic Regression: $Y_i \sim \text{Bin}\big(m_i, p(x_i)\big)$; for $\eta(x) = \log \frac{p(x)}{1-p(x)}$, $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \big\{Y_i \eta(x_i) - m_i \log(1 + e^{\eta(x)})\big\}$.
- Density Estimation: $X_i \sim f(x)$ on $\mathcal{X}$, where $f(x) = e^{\eta(x)} / \int_{\mathcal{X}} e^{\eta(x)} dx$; $L(\eta) = -\frac{1}{n}\sum_{i=1}^n \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)} dx$.

- Equivalence of penalized and constrained MLE:

$$\min L(\eta) + \lambda J(\eta) \;\Leftrightarrow\; \min L(\eta), \text{ subject to } J(\eta) \le \rho,$$

with λ being the Lagrange multiplier.

Penalized Likelihood Method

- Problem: Estimate $\eta(x)$ on generic domain $\mathcal{X}$ using stochastic data.
- Method: Minimize $L(\eta) + \lambda J(\eta)$, where $L(\eta)$ is minus log likelihood and $J(\eta)$ is a roughness functional such as $\int_0^1 \big(\eta''(x)\big)^2 dx$.

- Equivalence of penalized and constrained MLE:
$$\min\ L(\eta) + \lambda J(\eta) \ \Leftrightarrow\ \min\ L(\eta),\ \text{subject to } J(\eta) \le \rho,$$
with λ being the Lagrange multiplier.

Outline

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,
$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1A_2f + (I - A_1)A_2f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$
where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,
$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1A_2f + (I - A_1)A_2f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$
where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,

$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1 A_2 f + (I - A_1)A_2 f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$

  where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,

$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1 A_2 f + (I - A_1)A_2 f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$

  where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,
$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1A_2f + (I - A_1)A_2f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$
where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Examples of Averaging Operators

- $\mathcal{X} = \{1, \dots, K\}$: $Af = \frac{1}{K}\sum_x f(x)$; $Af = f(1)$.
- $\mathcal{X} = [a, b]$: $Af = \frac{1}{b-a}\int_a^b f(x)dx$; $Af = f(a)$.

Functional ANOVA Decomposition

- ANOVA: $Y_{ijk} = \mu_{ij} + \epsilon_{ijk}$, where $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$.
- ANOVA (cont.): Rewrite μ_{ij} as function on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ discrete, $f(x) = f(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}) = f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$.
- ANOVA (cont.): On generic domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$,

$$\begin{aligned} f(x) &= (I - A_1 + A_1)(I - A_2 + A_2)f \\ &= A_1A_2f + (I - A_1)A_2f + A_1(I - A_2)f + (I - A_1)(I - A_2)f \\ &= f_\emptyset + f_1(x_{\langle 1 \rangle}) + f_2(x_{\langle 2 \rangle}) + f_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}), \end{aligned}$$

  where A_1, A_2 are averaging operators on $\mathcal{X}_1$, $\mathcal{X}_2$, respectively.
- Side Conditions: $A_1 f_1 = A_1 f_{12} = A_2 f_2 = A_2 f_{12} = 0$.

Examples of Averaging Operators

- $\mathcal{X} = \{1, \ldots, K\}$: $Af = \frac{1}{K}\sum_x f(x)$; $Af = f(1)$.
- $\mathcal{X} = [a, b]$: $Af = \frac{1}{b-a}\int_a^b f(x)dx$; $Af = f(a)$.

Examples of Two-Way ANOVA Decomposition

- $\mathcal{X} = [0,1]^2$, $A_1 f = \int_0^1 f\, dx_{\langle 1\rangle}$, $A_2 f = \int_0^1 f\, dx_{\langle 2\rangle}$:

$$f_\emptyset = A_1 A_2 f = \int_0^1 \int_0^1 f\, dx_{\langle 1\rangle} dx_{\langle 2\rangle},$$
$$f_1 = (I - A_1) A_2 f = \int_0^1 \Big(f - \int_0^1 f\, dx_{\langle 1\rangle}\Big) dx_{\langle 2\rangle},$$
$$f_2 = A_1 (I - A_2) f = \int_0^1 \Big(f - \int_0^1 f\, dx_{\langle 2\rangle}\Big) dx_{\langle 1\rangle},$$
$$f_{12} = (I - A_1)(I - A_2) f = f - \int_0^1 f\, dx_{\langle 2\rangle} - \int_0^1 f\, dx_{\langle 1\rangle} + \int_0^1 \int_0^1 f\, dx_{\langle 1\rangle} dx_{\langle 2\rangle}$$

- $\mathcal{X} = \{1, \dots, K\} \times [0,1]$, $A_1 f = \frac{1}{K} \sum_{x_{\langle 1\rangle}} f(x_{\langle 1\rangle})$, $A_2 f = \int_0^1 f dx_{\langle 2\rangle}$:

$$f_\emptyset = A_1 A_2 f = \frac{1}{K} \sum_{x_{\langle 1\rangle}} \int_0^1 f\, dx_{\langle 2\rangle},$$
$$f_1 = (I - A_1) A_2 f = \int_0^1 \Big(f - \frac{1}{K} \sum_{x_{\langle 1\rangle}} f\Big) dx_{\langle 2\rangle},$$
$$f_2 = A_1 (I - A_2) f = \frac{1}{K} \sum_{x_{\langle 1\rangle}} \Big(f - \int_0^1 f\, dx_{\langle 2\rangle}\Big),$$
$$f_{12} = (I - A_1)(I - A_2) f$$
$$= f - \int_0^1 f\, dx_{\langle 2\rangle} - \frac{1}{K} \sum_{x_{\langle 1\rangle}} f + \frac{1}{K} \sum_{x_{\langle 1\rangle}} \int_0^1 f\, dx_{\langle 2\rangle}.$$

Examples of Two-Way ANOVA Decomposition

- $\mathcal{X} = [0,1]^2$, $A_1 f = \int_0^1 f \, dx_{\langle 1 \rangle}$, $A_2 f = \int_0^1 f \, dx_{\langle 2 \rangle}$:

$$
\begin{aligned}
f_\emptyset &= A_1 A_2 f = \int_0^1 \int_0^1 f \, dx_{\langle 1 \rangle} dx_{\langle 2 \rangle}, \\
f_1 &= (I - A_1) A_2 f = \int_0^1 \left(f - \int_0^1 f \, dx_{\langle 1 \rangle} \right) dx_{\langle 2 \rangle}, \\
f_2 &= A_1 (I - A_2) f = \int_0^1 \left(f - \int_0^1 f \, dx_{\langle 2 \rangle} \right) dx_{\langle 1 \rangle}, \\
f_{12} &= (I - A_1)(I - A_2) f = f - \int_0^1 f \, dx_{\langle 2 \rangle} - \int_0^1 f \, dx_{\langle 1 \rangle} + \int_0^1 \int_0^1 f \, dx_{\langle 1 \rangle} dx_{\langle 2 \rangle}
\end{aligned}
$$

- $\mathcal{X} = \{1, \dots, K\} \times [0,1]$, $A_1 f = \frac{1}{K} \sum_{x_{\langle 1 \rangle}} f(x_{\langle 1 \rangle})$, $A_2 f = \int_0^1 f dx_{\langle 2 \rangle}$:

$$
\begin{aligned}
f_\emptyset &= A_1 A_2 f = \frac{1}{K} \sum_{x_{\langle 1 \rangle}} \int_0^1 f \, dx_{\langle 2 \rangle}, \\
f_1 &= (I - A_1) A_2 f = \int_0^1 \left(f - \frac{1}{K} \sum_{x_{\langle 1 \rangle}} f \right) dx_{\langle 2 \rangle}, \\
f_2 &= A_1 (I - A_2) f = \frac{1}{K} \sum_{x_{\langle 1 \rangle}} \left(f - \int_0^1 f \, dx_{\langle 2 \rangle} \right), \\
f_{12} &= (I - A_1)(I - A_2) f \\
&= f - \int_0^1 f \, dx_{\langle 2 \rangle} - \frac{1}{K} \sum_{x_{\langle 1 \rangle}} f + \frac{1}{K} \sum_{x_{\langle 1 \rangle}} \int_0^1 f \, dx_{\langle 2 \rangle}.
\end{aligned}
$$

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.
- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

Tensor Product Splines

- Write $\eta(x) = \sum_\beta \eta_\beta$ for an ANOVA decomposition, with generic index β.
- Schematically, $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$, where $J_\beta(\eta_\beta)$ measure the roughness of η_β and θ_β adjust their relative weights in $J(\eta)$.
- Technically, $J_\beta(\eta_\beta)$ are square (semi) norms in tensor product reproducing kernel Hilbert spaces constructed on product domains.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

Tensor Product Splines

- Write $\eta(x) = \sum_\beta \eta_\beta$ for an ANOVA decomposition, with generic index β.
- Schematically, $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$, where $J_\beta(\eta_\beta)$ measure the roughness of η_β and θ_β adjust their relative weights in $J(\eta)$.
- Technically, $J_\beta(\eta_\beta)$ are square (semi) norms in tensor product reproducing kernel Hilbert spaces constructed on product domains.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

Tensor Product Splines

- Write $\eta(x) = \sum_\beta \eta_\beta$ for an ANOVA decomposition, with generic index β.
- Schematically, $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$, where $J_\beta(\eta_\beta)$ measure the roughness of η_β and θ_β adjust their relative weights in $J(\eta)$.
- Technically, $J_\beta(\eta_\beta)$ are square (semi) norms in tensor product reproducing kernel Hilbert spaces constructed on product domains.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

Tensor Product Splines

- Write $\eta(x) = \sum_\beta \eta_\beta$ for an ANOVA decomposition, with generic index β.
- Schematically, $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$, where $J_\beta(\eta_\beta)$ measure the roughness of η_β and θ_β adjust their relative weights in $J(\eta)$.
- Technically, $J_\beta(\eta_\beta)$ are square (semi) norms in tensor product reproducing kernel Hilbert spaces constructed on product domains.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Smoothing Spline ANOVA Models

- Two-Way ANOVA decomposition on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ implies one-way ANOVA decomposition on $\mathcal{X}$, with $A = A_1 A_2$.
- Multi-way ANOVA decompositions can be defined directly, or recursively via 2-way decompositions.
- ANOVA structures can be built into penalized likelihood estimation, yielding tensor product splines.

Tensor Product Splines

- Write $\eta(x) = \sum_\beta \eta_\beta$ for an ANOVA decomposition, with generic index β.
- Schematically, $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$, where $J_\beta(\eta_\beta)$ measure the roughness of η_β and θ_β adjust their relative weights in $J(\eta)$.
- Technically, $J_\beta(\eta_\beta)$ are square (semi) norms in tensor product reproducing kernel Hilbert spaces constructed on product domains.

- ANOVA structures help model interpretation.
- Elimination of higher order terms lessens curse of dimensionality.

Outline

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of ssanova

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of ssanova

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of ssanova

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of `ssanova`

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of ssanova

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Implicit and Explicit Model Configurations

- Vector: Cubic, or periodic cubic via `type=list(x=list("per",c(a,b)))`.
- 2-D Matrix: Thin-plate, or spherical via `type=list(x="sphere")`.
- Factor: Shrinkage.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of ssanova

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Implicit and Explicit Model Configurations

- Vector: Cubic, or periodic cubic via `type=list(x=list("per",c(a,b)))`.
- 2-D Matrix: Thin-plate, or spherical via `type=list(x="sphere")`.
- Factor: Shrinkage.

Gaussian Regression with gss

- `gss` suites have syntax similar to `lm` and `glm`.
- Gaussian Regression: `ssanova` suite.

Basic Syntax of `ssanova`

- Fitting: `fit <- ssanova(y~x)`
- Evaluating: `est <- predict(fit,data.frame(x=xx),se=TRUE)`, with `est$fit` containing $\hat{\eta}(x)$ at `xx` and `est$se` containing standard errors.
- ANOVA terms are specified via model formulas: `y~x1*x2`, `y~x1+x2`, etc.

- Different "splines" can be configured on different margins.

Implicit and Explicit Model Configurations

- Vector: Cubic, or periodic cubic via `type=list(x=list("per",c(a,b)))`.
- 2-D Matrix: Thin-plate, or spherical via `type=list(x="sphere")`.
- Factor: Shrinkage.

Example: Water Acidity in Lakes

- Data: Water acidity measurements were recorded for 112 lakes in the Blue Ridge along with calcium concentrations.
- Modeling: A model of the following form was considered,

$$\text{pH} = \eta_{\emptyset} + \eta_c(\text{cal}) + \eta_g(\text{geog}) + \eta_{c,g}(\text{cal}, \text{geog}) + \epsilon,$$

  but $\eta_{c,g}$ is "insignificant" so an additive model is fitted.
- Notes: i) η_c is linear; ii) η_g is 2-D, isotropically invariant.

Example: Water Acidity in Lakes

- Data: Water acidity measurements were recorded for 112 lakes in the Blue Ridge along with calcium concentrations.
- Modeling: A model of the following form was considered,

$$\text{pH} = \eta_\emptyset + \eta_c(\text{cal}) + \eta_g(\text{geog}) + \eta_{c,g}(\text{cal}, \text{geog}) + \epsilon,$$

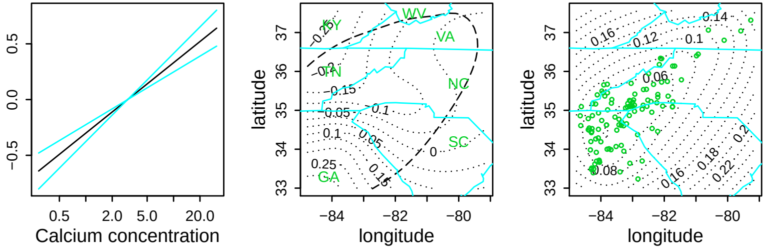
  but $\eta_{c,g}$ is "insignificant" so an additive model is fitted.
- Notes: i) η_c is linear; ii) η_g is 2-D, isotropically invariant.

Example: Water Acidity in Lakes

- Data: Water acidity measurements were recorded for 112 lakes in the Blue Ridge along with calcium concentrations.
- Modeling: A model of the following form was considered,

$$\text{pH} = \eta_\emptyset + \eta_c(\text{cal}) + \eta_g(\text{geog}) + \eta_{c,g}(\text{cal}, \text{geog}) + \epsilon,$$

  but $\eta_{c,g}$ is "insignificant" so an additive model is fitted.
- Notes: i) η_c is linear; ii) η_g is 2-D, isotropically invariant.

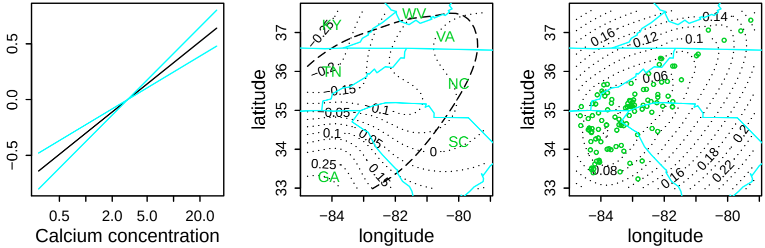


Example: Water Acidity in Lakes

- Data: Water acidity measurements were recorded for 112 lakes in the Blue Ridge along with calcium concentrations.
- Modeling: A model of the following form was considered,

$$\text{pH} = \eta_\emptyset + \eta_c(\text{cal}) + \eta_g(\text{geog}) + \eta_{c,g}(\text{cal}, \text{geog}) + \epsilon,$$

  but $\eta_{c,g}$ is “insignificant” so an additive model is fitted.
- Notes: i) η_c is linear; ii) η_g is 2-D, isotropically invariant.



A Sample Session with ssanova

First load data frame LakeAcidity with elements ph, cal, and geog.

```
data(LakeAcidity); lake <- LakeAcidity
```

Next fit a model of form $\eta = \eta_\emptyset + \eta_c + \eta_g + \eta_{c,g}$.

```
fit0 <- ssanova(ph~log(cal)*geog,data=lake)
```

Check for the "significance" of interaction, then fit an additive model.

```
project(fit0, inc=c("log(cal)","geog"))
fit1 <- ssanova(ph~log(cal)+geog,data=lake)
```

Now evaluate η_g on a grid, then plot

```
xx <- seq(-.04,.04,len=31)
grid <- cbind(rep(xx,31),rep(xx,rep(31,31)))
est.g <- predict(fit1,data.frame(geog=I(grid)),
                 se=TRUE,inc="geog")
ff <- matrix(est.g$fit,31,31)
filled.contour(xx,xx,ff,asp=1,plot.axes={points(acid$geog)})
```

A Sample Session with ssanova

First load data frame LakeAcidity with elements ph, cal, and geog.

```
data(LakeAcidity); lake <- LakeAcidity
```

Next fit a model of form $\eta = \eta_\emptyset + \eta_c + \eta_g + \eta_{c,g}$.

```
fit0 <- ssanova(ph~log(cal)*geog,data=lake)
```

Check for the "significance" of interaction, then fit an additive model.

```
project(fit0, inc=c("log(cal)","geog"))
fit1 <- ssanova(ph~log(cal)+geog,data=lake)
```

Now evaluate η_g on a grid, then plot

```
xx <- seq(-.04,.04,len=31)
grid <- cbind(rep(xx,31),rep(xx,rep(31,31)))
est.g <- predict(fit1,data.frame(geog=I(grid)),
                 se=TRUE,inc="geog")
ff <- matrix(est.g$fit,31,31)
filled.contour(xx,xx,ff,asp=1,plot.axes={points(acid$geog)})
```

A Sample Session with ssanova

First load data frame `LakeAcidity` with elements `ph`, `cal`, and `geog`.

```
data(LakeAcidity); lake <- LakeAcidity
```

Next fit a model of form $\eta = \eta_\emptyset + \eta_c + \eta_g + \eta_{c,g}$.

```
fit0 <- ssanova(ph~log(cal)*geog,data=lake)
```

Check for the “significance” of interaction, then fit an additive model.

```
project(fit0, inc=c("log(cal)","geog"))
fit1 <- ssanova(ph~log(cal)+geog,data=lake)
```

Now evaluate η_g on a grid, then plot

```
xx <- seq(-.04,.04,len=31)
grid <- cbind(rep(xx,31),rep(xx,rep(31,31)))
est.g <- predict(fit1,data.frame(geog=I(grid)),
                 se=TRUE,inc="geog")
ff <- matrix(est.g$fit,31,31)
filled.contour(xx,xx,ff,asp=1,plot.axes={points(acid$geog)})
```

A Sample Session with ssanova

First load data frame LakeAcidity with elements ph, cal, and geog.

```
data(LakeAcidity); lake <- LakeAcidity
```

Next fit a model of form $\eta = \eta_\emptyset + \eta_c + \eta_g + \eta_{c,g}$.

```
fit0 <- ssanova(ph~log(cal)*geog,data=lake)
```

Check for the "significance" of interaction, then fit an additive model.

```
project(fit0, inc=c("log(cal)","geog"))
fit1 <- ssanova(ph~log(cal)+geog,data=lake)
```

Now evaluate η_g on a grid, then plot

```
xx <- seq(-.04,.04,len=31)
grid <- cbind(rep(xx,31),rep(xx,rep(31,31)))
est.g <- predict(fit1,data.frame(geog=I(grid)),
                 se=TRUE,inc="geog")
ff <- matrix(est.g$fit,31,31)
filled.contour(xx,xx,ff,asp=1,plot.axes={points(acid$geog)})
```

Outline

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B \eta$ on $\mathcal{X} = \{1, \dots, K\}$; minimizer of $\frac{1}{n}\sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, where the "inverse" $R_J(x, y)$ of $J(\eta)$ is a reproducing kernel.

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B \eta$ on $\mathcal{X} = \{1, \dots, K\}$; minimizer of $\frac{1}{n} \sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

Normal Priors

- $B = I - \frac{1}{K} \mathbf{1}\mathbf{1}^T$: $J(\eta) = \sum_x \big(\eta(x) - \bar{\eta}\big)^2$; $Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$, $\boldsymbol{\alpha} \sim N(\mathbf{0}, cB)$.
- Ordinal Factor: $J(\eta) = \sum_{x=2}^K \big(\eta(x) - \eta(x-1)\big)^2$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, where the "inverse" $R_J(x, y)$ of $J(\eta)$ is a reproducing kernel.

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B\eta$ on $\mathcal{X} = \{1, \ldots, K\}$; minimizer of $\frac{1}{n}\sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

Normal Priors

- $B = I - \frac{1}{K}\mathbf{1}\mathbf{1}^T$: $J(\eta) = \sum_x \big(\eta(x) - \bar{\eta}\big)^2$; $Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$, $\boldsymbol{\alpha} \sim N(\mathbf{0}, cB)$.
- Ordinal Factor: $J(\eta) = \sum_{x=2}^K \big(\eta(x) - \eta(x-1)\big)^2$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x,y)$, where the "inverse" $R_J(x,y)$ of $J(\eta)$ is a reproducing kernel.

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B\eta$ on $\mathcal{X} = \{1, \ldots, K\}$; minimizer of $\frac{1}{n}\sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, where the "inverse" $R_J(x, y)$ of $J(\eta)$ is a reproducing kernel.

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B\eta$ on $\mathcal{X} = \{1, \dots, K\}$; minimizer of $\frac{1}{n}\sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, where the "inverse" $R_J(x, y)$ of $J(\eta)$ is a reproducing kernel.

Gaussian Process Priors

- Cubic Spline: $R_J(x, y) = k_2(x)k_2(y) - k_4(|x - y|)$ is the "inverse" of $J(\eta) = \int_0^1 \big(\eta''(x)\big)^2 dx$, with $k_\nu(x) = B_\nu(x)/\nu!$ scaled Bernoulli polynomials.
- Thin-Plate Spline: "Inverse" of $\int\int \Big(\frac{\partial^2 f}{\partial z_{\langle 1\rangle}^2} + 2\frac{\partial^2 f}{\partial z_{\langle 1\rangle}\partial z_{\langle 2\rangle}} + \frac{\partial^2 f}{\partial z_{\langle 2\rangle}^2}\Big)^2 dz_{\langle 1\rangle} dz_{\langle 2\rangle}$.

Empirical Bayes Models

- Shrinkage Estimate: Set $J(\eta) = \eta^T B\eta$ on $\mathcal{X} = \{1, \ldots, K\}$; minimizer of $\frac{1}{n}\sum_i \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is Bayes under prior $\eta \sim N(\mathbf{0}, cB^+)$.

- Univariate Spline: $\lambda J(\eta)$ acts like minus log likelihood of Gaussian process prior on $\mathcal{X}$, with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, where the "inverse" $R_J(x, y)$ of $J(\eta)$ is a reproducing kernel.

Gaussian Process Priors

- Cubic Spline: $R_J(x, y) = k_2(x)k_2(y) - k_4(|x - y|)$ is the "inverse" of $J(\eta) = \int_0^1 \big(\eta''(x)\big)^2 dx$, with $k_\nu(x) = B_\nu(x)/\nu!$ scaled Bernoulli polynomials.
- Thin-Plate Spline: "Inverse" of $\int\int\Big(\frac{\partial^2 f}{\partial z_{\langle 1\rangle}^2} + 2\frac{\partial^2 f}{\partial z_{\langle 1\rangle}\partial z_{\langle 2\rangle}} + \frac{\partial^2 f}{\partial z_{\langle 2\rangle}^2}\Big)^2 dz_{\langle 1\rangle}dz_{\langle 2\rangle}$.

Models on Product Domains

- On $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, one constructs product kernels from marginal ones,
$$R(x,y) = R^{\langle 1\rangle}(x_{\langle 1\rangle}, y_{\langle 1\rangle})R^{\langle 2\rangle}(x_{\langle 2\rangle}, y_{\langle 2\rangle}),$$
thus models are specified by marginal kernels and model formulas.
- With an ANOVA decomposition $\eta = \sum_\beta \eta_\beta$, the "inverse" of $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$ is $R_J(x,y) = \sum_\beta \theta_\beta R_\beta(x,y)$, where $R_\beta(x,y)$ are "inverses" of $J_\beta(\eta)$; η_β have independent priors.
- The minimizer of $\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is of form,
$$\eta(x) = \sum_{\nu=1}^m d_\nu \phi_\nu(x) + \sum_{i=1}^n c_i R_J(x_i, x),$$
where $\{\phi_\nu(x)\}$ is a basis of $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$.
- One needs explicit forms of $\phi_\nu(x)$ and $R_J(x,y)$ for computation, but not of $J(\eta)$.

Models on Product Domains

- On $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, one constructs product kernels from marginal ones,
$$R(x,y) = R^{\langle 1 \rangle}(x_{\langle 1 \rangle}, y_{\langle 1 \rangle}) R^{\langle 2 \rangle}(x_{\langle 2 \rangle}, y_{\langle 2 \rangle}),$$
thus models are specified by marginal kernels and model formulas.
- With an ANOVA decomposition $\eta = \sum_\beta \eta_\beta$, the "inverse" of $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$ is $R_J(x,y) = \sum_\beta \theta_\beta R_\beta(x,y)$, where $R_\beta(x,y)$ are "inverses" of $J_\beta(\eta)$; η_β have independent priors.
- The minimizer of $\frac{1}{n}\sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is of form,
$$\eta(x) = \sum_{\nu=1}^m d_\nu \phi_\nu(x) + \sum_{i=1}^n c_i R_J(x_i, x),$$
where $\{\phi_\nu(x)\}$ is a basis of $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$.
- One needs explicit forms of $\phi_\nu(x)$ and $R_J(x,y)$ for computation, but not of $J(\eta)$.

Models on Product Domains

- On $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, one constructs product kernels from marginal ones,
$$R(x, y) = R^{\langle 1 \rangle}(x_{\langle 1 \rangle}, y_{\langle 1 \rangle}) R^{\langle 2 \rangle}(x_{\langle 2 \rangle}, y_{\langle 2 \rangle}),$$
thus models are specified by marginal kernels and model formulas.
- With an ANOVA decomposition $\eta = \sum_\beta \eta_\beta$, the "inverse" of $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$ is $R_J(x, y) = \sum_\beta \theta_\beta R_\beta(x, y)$, where $R_\beta(x, y)$ are "inverses" of $J_\beta(\eta)$; η_β have independent priors.
- The minimizer of $\frac{1}{n} \sum_{i=1}^n \big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is of form,
$$\eta(x) = \textstyle\sum_{\nu=1}^m d_\nu \phi_\nu(x) + \sum_{i=1}^n c_i R_J(x_i, x),$$
where $\{\phi_\nu(x)\}$ is a basis of $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$.
- One needs explicit forms of $\phi_\nu(x)$ and $R_J(x, y)$ for computation, but not of $J(\eta)$.

Models on Product Domains

- On $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, one constructs product kernels from marginal ones,
$$R(x,y) = R^{\langle 1 \rangle}(x_{\langle 1 \rangle}, y_{\langle 1 \rangle}) R^{\langle 2 \rangle}(x_{\langle 2 \rangle}, y_{\langle 2 \rangle}),$$
thus models are specified by marginal kernels and model formulas.
- With an ANOVA decomposition $\eta = \sum_\beta \eta_\beta$, the "inverse" of $J(\eta) = \sum_\beta \theta_\beta^{-1} J_\beta(\eta_\beta)$ is $R_J(x,y) = \sum_\beta \theta_\beta R_\beta(x,y)$, where $R_\beta(x,y)$ are "inverses" of $J_\beta(\eta)$; η_β have independent priors.
- The minimizer of $\frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \eta(x_i)\big)^2 + \lambda J(\eta)$ is of form,
$$\eta(x) = \sum_{\nu=1}^{m} d_\nu \phi_\nu(x) + \sum_{i=1}^{n} c_i R_J(x_i, x),$$
where $\{\phi_\nu(x)\}$ is a basis of $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$.
- One needs explicit forms of $\phi_\nu(x)$ and $R_J(x,y)$ for computation, but not of $J(\eta)$.

Outline

Asymptotic Convergence

- As $n \to \infty$ and $\lambda \to 0$, $(V + \lambda J)(\hat{\eta} - \eta) = O_p(\lambda^p + n^{-1}\lambda^{-1/r})$, where $V(\hat{\eta} - \eta)$ is a quadratic loss, $r > 1$, and $p \in [1, 2]$.

- Efficient Approximation: The minimizers $\hat{\eta}^*$ of $L(\eta) + \lambda J(\eta)$ in
$$\mathcal{H}^* = \mathcal{N}_J \oplus \text{span}\{R_J(z_j, \cdot), j = 1, \dots, q\}$$
share the same convergence rates with $\hat{\eta} \in \mathcal{H} = \{\eta : J(\eta) < \infty\}$, with $\{z_j\} \subseteq \{x_i\}$ a random subset and $q\lambda^{2/r} \to \infty$.
- For $r = 4$, $p = 2$, the optimal $O_p(n^{-8/9})$ is achieved at $\lambda \asymp n^{-4/9}$; it suffices to have $q \asymp n^{2/9+\epsilon}$, $\forall \epsilon > 0$. Computation is $O(nq^2)$.

Asymptotic Convergence

- As $n \to \infty$ and $\lambda \to 0$, $(V + \lambda J)(\hat{\eta} - \eta) = O_p(\lambda^p + n^{-1}\lambda^{-1/r})$, where $V(\hat{\eta} - \eta)$ is a quadratic loss, $r > 1$, and $p \in [1, 2]$.

Convergence Rates

- Gaussian Regression: $V(g) = \int_{\mathcal{X}} g^2 f\, dx$, for $f(x)$ the limiting density of x_i.
- Density Estimation: $V(g) = \int_{\mathcal{X}} g^2 f\, dx - \left(\int_{\mathcal{X}} g\, f\, dx\right)^2$, for $f(x)$ the truth.
- Cubic Spline: For $J(\eta) = \int_0^1 \left(\eta''(x)\right)^2 dx$ on $\mathcal{X} = [0, 1]$, $r = 4$. If $J(\eta) < \infty$ holds "barely," $p = 1$; if $\int_0^1 \left(\eta^{(4)}(x)\right)^2 dx < \infty$, $p = 2$.

- Efficient Approximation: The minimizers $\hat{\eta}^*$ of $L(\eta) + \lambda J(\eta)$ in
$$\mathcal{H}^* = \mathcal{N}_J \oplus \mathrm{span}\{R_J(z_j, \cdot), j = 1, \ldots, q\}$$
share the same convergence rates with $\hat{\eta} \in \mathcal{H} = \{\eta : J(\eta) < \infty\}$, with $\{z_j\} \subseteq \{x_i\}$ a random subset and $q\lambda^{2/r} \to \infty$.
- For $r = 4$, $p = 2$, the optimal $O_p(n^{-8/9})$ is achieved at $\lambda \asymp n^{-4/9}$; it suffices to have $q \asymp n^{2/9+\epsilon}$, $\forall \epsilon > 0$. Computation is $O(nq^2)$.

Asymptotic Convergence

- As $n \to \infty$ and $\lambda \to 0$, $(V + \lambda J)(\hat{\eta} - \eta) = O_p(\lambda^p + n^{-1}\lambda^{-1/r})$, where $V(\hat{\eta} - \eta)$ is a quadratic loss, $r > 1$, and $p \in [1, 2]$.

Convergence Rates

- Gaussian Regression: $V(g) = \int_{\mathcal{X}} g^2 f\, dx$, for $f(x)$ the limiting density of x_i.
- Density Estimation: $V(g) = \int_{\mathcal{X}} g^2 f\, dx - \left(\int_{\mathcal{X}} g\, f\, dx\right)^2$, for $f(x)$ the truth.
- Cubic Spline: For $J(\eta) = \int_0^1 \big(\eta''(x)\big)^2 dx$ on $\mathcal{X} = [0, 1]$, $r = 4$. If $J(\eta) < \infty$ holds "barely," $p = 1$; if $\int_0^1 \big(\eta^{(4)}(x)\big)^2 dx < \infty$, $p = 2$.

- Efficient Approximation: The minimizers $\hat{\eta}^*$ of $L(\eta) + \lambda J(\eta)$ in
$$\mathcal{H}^* = \mathcal{N}_J \oplus \text{span}\{R_J(z_j, \cdot), j = 1, \dots, q\}$$
share the same convergence rates with $\hat{\eta} \in \mathcal{H} = \{\eta : J(\eta) < \infty\}$, with $\{z_j\} \subseteq \{x_i\}$ a random subset and $q\lambda^{2/r} \to \infty$.
- For $r = 4$, $p = 2$, the optimal $O_p(n^{-8/9})$ is achieved at $\lambda \asymp n^{-4/9}$; it suffices to have $q \asymp n^{2/9+\epsilon}$, $\forall \epsilon > 0$. Computation is $O(nq^2)$.

Asymptotic Convergence

- As $n \to \infty$ and $\lambda \to 0$, $(V + \lambda J)(\hat{\eta} - \eta) = O_p(\lambda^p + n^{-1}\lambda^{-1/r})$, where $V(\hat{\eta} - \eta)$ is a quadratic loss, $r > 1$, and $p \in [1, 2]$.

Convergence Rates

- Gaussian Regression: $V(g) = \int_{\mathcal{X}} g^2 f \, dx$, for $f(x)$ the limiting density of x_i.
- Density Estimation: $V(g) = \int_{\mathcal{X}} g^2 f \, dx - \left(\int_{\mathcal{X}} g f \, dx \right)^2$, for $f(x)$ the truth.
- Cubic Spline: For $J(\eta) = \int_0^1 \left(\eta''(x)\right)^2 dx$ on $\mathcal{X} = [0, 1]$, $r = 4$. If $J(\eta) < \infty$ holds "barely," $p = 1$; if $\int_0^1 \left(\eta^{(4)}(x)\right)^2 dx < \infty$, $p = 2$.

- Efficient Approximation: The minimizers $\hat{\eta}^*$ of $L(\eta) + \lambda J(\eta)$ in
$$\mathcal{H}^* = \mathcal{N}_J \oplus \text{span}\left\{ R_J(z_j, \cdot), j = 1, \ldots, q \right\}$$
share the same convergence rates with $\hat{\eta} \in \mathcal{H} = \{\eta : J(\eta) < \infty\}$, with $\{z_j\} \subseteq \{x_i\}$ a random subset and $q\lambda^{2/r} \to \infty$.
- For $r = 4$, $p = 2$, the optimal $O_p(n^{-8/9})$ is achieved at $\lambda \asymp n^{-4/9}$; it suffices to have $q \asymp n^{2/9+\epsilon}$, $\forall \epsilon > 0$. Computation is $O(nq^2)$.

Asymptotic Convergence

- As $n \to \infty$ and $\lambda \to 0$, $(V + \lambda J)(\hat{\eta} - \eta) = O_p(\lambda^p + n^{-1}\lambda^{-1/r})$, where $V(\hat{\eta} - \eta)$ is a quadratic loss, $r > 1$, and $p \in [1, 2]$.

Convergence Rates

- Gaussian Regression: $V(g) = \int_{\mathcal{X}} g^2 f\, dx$, for $f(x)$ the limiting density of x_i.
- Density Estimation: $V(g) = \int_{\mathcal{X}} g^2 f\, dx - \big(\int_{\mathcal{X}} g\, f\, dx\big)^2$, for $f(x)$ the truth.
- Cubic Spline: For $J(\eta) = \int_0^1 \big(\eta''(x)\big)^2 dx$ on $\mathcal{X} = [0, 1]$, $r = 4$. If $J(\eta) < \infty$ holds "barely," $p = 1$; if $\int_0^1 \big(\eta^{(4)}(x)\big)^2 dx < \infty$, $p = 2$.

- Efficient Approximation: The minimizers $\hat{\eta}^*$ of $L(\eta) + \lambda J(\eta)$ in
$$\mathcal{H}^* = \mathcal{N}_J \oplus \text{span}\big\{R_J(z_j, \cdot), j = 1, \ldots, q\big\}$$
share the same convergence rates with $\hat{\eta} \in \mathcal{H} = \{\eta : J(\eta) < \infty\}$, with $\{z_j\} \subseteq \{x_i\}$ a random subset and $q\lambda^{2/r} \to \infty$.
- For $r = 4$, $p = 2$, the optimal $O_p(n^{-8/9})$ is achieved at $\lambda \asymp n^{-4/9}$; it suffices to have $q \asymp n^{2/9+\epsilon}$, $\forall \epsilon > 0$. Computation is $O(nq^2)$.

Outline

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

Kullback-Leibler Distances

- Gaussian Regression: $\mathrm{KL}(\eta, \eta_\lambda) \propto \frac{1}{n}\sum_i \big(\eta_\lambda(x_i) - \eta(x_i)\big)^2 \approx V(\eta_\lambda - \eta)$.
- Density Estimation: $\mathrm{SKL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda)(f - f_\lambda)dx \approx V(\eta_\lambda - \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

Kullback-Leibler Distances

- Gaussian Regression: $\mathrm{KL}(\eta, \eta_\lambda) \propto \frac{1}{n} \sum_i \big(\eta_\lambda(x_i) - \eta(x_i)\big)^2 \approx V(\eta_\lambda - \eta)$.
- Density Estimation: $\mathrm{SKL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda)(f - f_\lambda) dx \approx V(\eta_\lambda - \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

Kullback-Leibler Distances

- Gaussian Regression: $\mathrm{KL}(\eta, \eta_\lambda) \propto \frac{1}{n}\sum_i \big(\eta_\lambda(x_i) - \eta(x_i)\big)^2 \approx V(\eta_\lambda - \eta)$.
- Density Estimation: $\mathrm{SKL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda)(f - f_\lambda)dx \approx V(\eta_\lambda - \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Cross-Validation in Density Estimation

- $\mathrm{KL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda) f\, dx - \log \int_{\mathcal{X}} e^{\eta} dx + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx$.
- $\int_{\mathcal{X}} \eta_\lambda f\, dx$ is estimated by $\frac{1}{n}\sum_i \eta_\lambda^{[i]}(X_i)$, for $\eta_\lambda^{[i]}$ the "delete-1" estimates.
- $V(\lambda) = -\frac{1}{n}\sum_i \eta_\lambda(X_i) + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx + \frac{1}{n}\sum_i \big(\eta_\lambda - \eta_\lambda^{[i]}\big)(X_i)$

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Cross-Validation in Density Estimation

- $\mathrm{KL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda) f \, dx - \log \int_{\mathcal{X}} e^{\eta} dx + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx$.
- $\int_{\mathcal{X}} \eta_\lambda f \, dx$ is estimated by $\frac{1}{n} \sum_i \eta_\lambda^{[i]}(X_i)$, for $\eta_\lambda^{[i]}$ the "delete-1" estimates.
- $V(\lambda) = -\frac{1}{n} \sum_i \eta_\lambda(X_i) + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx + \frac{1}{n} \sum_i \big(\eta_\lambda - \eta_\lambda^{[i]}\big)(X_i)$

Smoothing Parameter Selection

- To measure the performance of $\hat{\eta} = \eta_\lambda$ as estimate of η, one may use $\mathrm{KL}(\eta, \eta_\lambda)$ or $\mathrm{SKL}(\eta, \eta_\lambda) = \mathrm{KL}(\eta, \eta_\lambda) + \mathrm{KL}(\eta_\lambda, \eta)$.

- Cross-Validation: Typically, $\mathrm{KL}(\eta, \eta_\lambda) = A(\eta_\lambda) + B(\eta_\lambda, \eta) + C(\eta)$, with $A(\eta_\lambda)$ computable, $B(\eta_\lambda, \eta)$ estimable, and $C(\eta)$ disposable. Minimize $V(\lambda) = A(\eta_\lambda) + \hat{B}(\eta_\lambda, \eta) = L(\eta_\lambda) + P(\eta_\lambda)$ to select λ.

Cross-Validation in Density Estimation

- $\mathrm{KL}(\eta, \eta_\lambda) = \int_{\mathcal{X}} (\eta - \eta_\lambda) f\, dx - \log \int_{\mathcal{X}} e^{\eta} dx + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx$.
- $\int_{\mathcal{X}} \eta_\lambda f\, dx$ is estimated by $\frac{1}{n} \sum_i \eta_\lambda^{[i]}(X_i)$, for $\eta_\lambda^{[i]}$ the "delete-1" estimates.
- $V(\lambda) = -\frac{1}{n} \sum_i \eta_\lambda(X_i) + \log \int_{\mathcal{X}} e^{\eta_\lambda} dx + \frac{1}{n} \sum_i \big(\eta_\lambda - \eta_\lambda^{[i]}\big)(X_i)$

Cross-Validation

- Cross-validation does not define optimality; it merely chases it.
- Cross-validation tends to overfit/undersmooth occasionally. Adding a fudge factor, $\alpha > 1$, often helps: $V(\lambda) = L(\eta_\lambda) + \alpha P(\eta_\lambda)$.

Cross-Validation

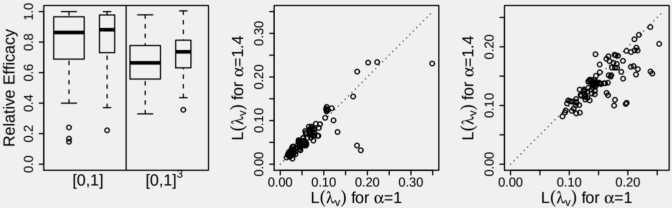
- Cross-validation does not define optimality; it merely chases it.
- Cross-validation tends to overfit/undersmooth occasionally. Adding a fudge factor, $\alpha > 1$, often helps: $V(\lambda) = L(\eta_\lambda) + \alpha P(\eta_\lambda)$.

Cross-Validation

- Cross-validation does not define optimality; it merely chases it.
- Cross-validation tends to overfit/undersmooth occasionally. Adding a fudge factor, $\alpha > 1$, often helps: $V(\lambda) = L(\eta_\lambda) + \alpha P(\eta_\lambda)$.

Empirical Performance of CV in Density Estimation

- Simulations: $n = 100$ on $[0,1]$, $n = 300$ on $[0,1]^3$; 100 replicates each.
- $L(\lambda) = \text{SKL}(\eta, \eta_\lambda)$ is minimized at λ_o. $V(\lambda)$ is minimized at λ_v for $\alpha = 1, 1.4$. $L(\lambda_o)/L(\lambda_v)$ defines the relative efficacy of $V(\lambda)$.



Outline

Bayesian Confidence Intervals

- Under a Gaussian process prior on $\mathcal{X}$ with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x,y)$, $L(\eta) + (\lambda/2)J(\eta)$ is posterior likelihood.
- With $L(\eta)$ quadratic as in Gaussian regression, the posterior is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$, which, together with $\mathrm{Var}\big[\eta(x)\big]$, yield Bayesian confidence intervals for $\eta(x)$.
- For $L(\eta)$ non-quadratic, substitute $L(\eta)$ by its quadratic approximation $Q_{\hat{\eta}}(\eta)$ at $\hat{\eta}$, and the approximate likelihood $Q_{\hat{\eta}}(\eta) + (\lambda/2)J(\eta)$ is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$.
- Posterior means and variances can be calculated for partial sums of $\eta(x)$, say $\eta_1 + \eta_2$ of $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_3 + \eta_{12}$.
- For an `ssanova fit`, one may use

```
predict(fit,data.frame(...),se,inc=c(...))
```

  with `se=TRUE` asking for standard errors and `inc` specifing terms to be included in the (partial) sum.

Bayesian Confidence Intervals

- Under a Gaussian process prior on $\mathcal{X}$ with $E\big[\eta(x)\big]=0$ and $E\big[\eta(x)\eta(y)\big]\propto R_J(x,y)$, $L(\eta)+(\lambda/2)J(\eta)$ is posterior likelihood.
- With $L(\eta)$ quadratic as in Gaussian regression, the posterior is Gaussian with $E\big[\eta(x)\big]=\hat{\eta}(x)$, which, together with $\mathrm{Var}\big[\eta(x)\big]$, yield Bayesian confidence intervals for $\eta(x)$.
- For $L(\eta)$ non-quadratic, substitute $L(\eta)$ by its quadratic approximation $Q_{\hat{\eta}}(\eta)$ at $\hat{\eta}$, and the approximate likelihood $Q_{\hat{\eta}}(\eta)+(\lambda/2)J(\eta)$ is Gaussian with $E\big[\eta(x)\big]=\hat{\eta}(x)$.
- Posterior means and variances can be calculated for partial sums of $\eta(x)$, say $\eta_1+\eta_2$ of $\eta=\eta_\emptyset+\eta_1+\eta_2+\eta_3+\eta_{12}$.
- For an `ssanova fit`, one may use

```
predict(fit,data.frame(...),se,inc=c(...))
```

  with `se=TRUE` asking for standard errors and `inc` specifing terms to be included in the (partial) sum.

Bayesian Confidence Intervals

- Under a Gaussian process prior on $\mathcal{X}$ with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x,y)$, $L(\eta) + (\lambda/2)J(\eta)$ is posterior likelihood.
- With $L(\eta)$ quadratic as in Gaussian regression, the posterior is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$, which, together with $\mathrm{Var}\big[\eta(x)\big]$, yield Bayesian confidence intervals for $\eta(x)$.
- For $L(\eta)$ non-quadratic, substitute $L(\eta)$ by its quadratic approximation $Q_{\hat{\eta}}(\eta)$ at $\hat{\eta}$, and the approximate likelihood $Q_{\hat{\eta}}(\eta) + (\lambda/2)J(\eta)$ is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$.
- Posterior means and variances can be calculated for partial sums of $\eta(x)$, say $\eta_1 + \eta_2$ of $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_3 + \eta_{12}$.
- For an `ssanova fit`, one may use

  ```
  predict(fit,data.frame(...),se,inc=c(...))
  ```

  with `se=TRUE` asking for standard errors and `inc` specifing terms to be included in the (partial) sum.

Bayesian Confidence Intervals

- Under a Gaussian process prior on $\mathcal{X}$ with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x, y)$, $L(\eta) + (\lambda/2)J(\eta)$ is posterior likelihood.
- With $L(\eta)$ quadratic as in Gaussian regression, the posterior is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$, which, together with $\mathrm{Var}\big[\eta(x)\big]$, yield Bayesian confidence intervals for $\eta(x)$.
- For $L(\eta)$ non-quadratic, substitute $L(\eta)$ by its quadratic approximation $Q_{\hat{\eta}}(\eta)$ at $\hat{\eta}$, and the approximate likelihood $Q_{\hat{\eta}}(\eta) + (\lambda/2)J(\eta)$ is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$.
- Posterior means and variances can be calculated for partial sums of $\eta(x)$, say $\eta_1 + \eta_2$ of $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_3 + \eta_{12}$.
- For an `ssanova fit`, one may use

  `predict(fit,data.frame(...),se,inc=c(...))`

  with `se=TRUE` asking for standard errors and `inc` specifing terms to be included in the (partial) sum.

Bayesian Confidence Intervals

- Under a Gaussian process prior on $\mathcal{X}$ with $E\big[\eta(x)\big] = 0$ and $E\big[\eta(x)\eta(y)\big] \propto R_J(x,y)$, $L(\eta) + (\lambda/2)J(\eta)$ is posterior likelihood.
- With $L(\eta)$ quadratic as in Gaussian regression, the posterior is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$, which, together with $\mathrm{Var}\big[\eta(x)\big]$, yield Bayesian confidence intervals for $\eta(x)$.
- For $L(\eta)$ non-quadratic, substitute $L(\eta)$ by its quadratic approximation $Q_{\hat{\eta}}(\eta)$ at $\hat{\eta}$, and the approximate likelihood $Q_{\hat{\eta}}(\eta) + (\lambda/2)J(\eta)$ is Gaussian with $E\big[\eta(x)\big] = \hat{\eta}(x)$.
- Posterior means and variances can be calculated for partial sums of $\eta(x)$, say $\eta_1 + \eta_2$ of $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_3 + \eta_{12}$.
- For an `ssanova fit`, one may use

  ```
  predict(fit,data.frame(...),se,inc=c(...))
  ```

  with `se=TRUE` asking for standard errors and `inc` specifing terms to be included in the (partial) sum.

Outline

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.

- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,
$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$
where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using
```
project(fit,inc=c(...))
```
with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$.

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Additive Models: $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_{12}$.
- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,
$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$
where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using
```
project(fit,inc=c(...))
```
with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$.

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- Additive Models: $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_{12}$.
- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,

$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$

  where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using

  ```
  project(fit,inc=c(...))
  ```

  with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$.

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,
$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$
where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using
```
project(fit,inc=c(...))
```
with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$.

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.

- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,
$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$
where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using
```
project(fit,inc=c(...))
```
with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) / \mathrm{KL}(\hat{\eta}, \eta_c)$.

Kullback-Leibler Projection

- Need to "test" $H_0 : \eta \in \mathcal{H}_0$ versus $H_a : \eta \in \mathcal{H}_0 \oplus \mathcal{H}_1$.
- With $\mathcal{H}_0$ infinite-D, Neyman-Pearson tests are unavailable.
- KL Projection: Given $\hat{\eta} \in \mathcal{H}_0 \oplus \mathcal{H}_1$, minimize $\mathrm{KL}(\hat{\eta}, \eta)$ over $\eta \in \mathcal{H}_0$ to obtain $\tilde{\eta}$, then inspect the "entropy" decomposition,
$$\mathrm{KL}(\hat{\eta}, \eta_c) = \mathrm{KL}(\hat{\eta}, \tilde{\eta}) + \mathrm{KL}(\tilde{\eta}, \eta_c),$$
where $\eta_c \in \mathcal{H}_0$ can be a constant regression or a uniform density.
- If $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta})/\mathrm{KL}(\hat{\eta}, \eta_c)$ is small, one loses little by cutting out $\mathcal{H}_1$.
- For an `ssanova fit`, one may calculate KL projection using
```
project(fit,inc=c(...))
```
with `inc` listing terms in $\mathcal{H}_0$; this returns a list with element `ratio` containing $\rho = \mathrm{KL}(\hat{\eta}, \tilde{\eta})/\mathrm{KL}(\hat{\eta}, \eta_c)$.

Outline

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

- In gss, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

Exponential Family Regression

- Logistic Regression: $f(y) = \binom{m}{y} p^y (1-p)^{m-y}$; $\eta = \log \frac{p}{1-p}$; $l(\eta; y) = -y\eta + m\log(1+e^{\eta})$.
- Poisson Regression: $f(y) = \lambda^y e^{-\lambda}/y!$; $\eta = \log\lambda$; $l(\eta; y) = -y\eta + e^{\eta}$.
- Gamma Regression: $f(y) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} y^{\alpha-1} e^{-y/\beta}$; $\eta = \log(\alpha\beta)$; $l(\eta; y) = ye^{-\eta} + \eta$.

- In gss, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

Exponential Family Regression

- Logistic Regression: $f(y) = \binom{m}{y} p^y (1-p)^{m-y}$; $\eta = \log \frac{p}{1-p}$; $l(\eta; y) = -y\eta + m\log(1+e^{\eta})$.
- Poisson Regression: $f(y) = \lambda^y e^{-\lambda}/y!$; $\eta = \log \lambda$; $l(\eta; y) = -y\eta + e^{\eta}$.
- Gamma Regression: $f(y) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} y^{\alpha-1} e^{-y/\beta}$; $\eta = \log(\alpha\beta)$; $l(\eta; y) = ye^{-\eta} + \eta$.

- In gss, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

Exponential Family Regression

- Logistic Regression: $f(y) = \binom{m}{y} p^y (1-p)^{m-y}$; $\eta = \log\frac{p}{1-p}$; $l(\eta; y) = -y\eta + m\log(1+e^{\eta})$.
- Poisson Regression: $f(y) = \lambda^y e^{-\lambda}/y!$; $\eta = \log\lambda$; $l(\eta; y) = -y\eta + e^{\eta}$.
- Gamma Regression: $f(y) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} y^{\alpha-1} e^{-y/\beta}$; $\eta = \log(\alpha\beta)$; $l(\eta; y) = ye^{-\eta} + \eta$.

- In `gss`, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

Exponential Family Regression

- Logistic Regression: $f(y) = \binom{m}{y} p^y (1-p)^{m-y}$; $\eta = \log \frac{p}{1-p}$; $l(\eta; y) = -y\eta + m\log(1+e^\eta)$.
- Poisson Regression: $f(y) = \lambda^y e^{-\lambda}/y!$; $\eta = \log\lambda$; $l(\eta; y) = -y\eta + e^\eta$.
- Gamma Regression: $f(y) = \frac{1}{\beta^\alpha \Gamma(\alpha)} y^{\alpha-1} e^{-y/\beta}$; $\eta = \log(\alpha\beta)$; $l(\eta; y) = ye^{-\eta} + \eta$.

- In `gss`, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Regression with Exponential Family Responses

- For $Y \sim f(y)$, one may use $L(\eta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\eta(x_i); Y_i)$ in $L(\eta) + \lambda J(\eta)$, where $l(\eta; y) \propto -\log f(y)$ for η the link.

Exponential Family Regression

- Logistic Regression: $f(y) = \binom{m}{y} p^y (1-p)^{m-y}$; $\eta = \log \frac{p}{1-p}$; $l(\eta; y) = -y\eta + m\log(1+e^{\eta})$.
- Poisson Regression: $f(y) = \lambda^y e^{-\lambda}/y!$; $\eta = \log\lambda$; $l(\eta; y) = -y\eta + e^{\eta}$.
- Gamma Regression: $f(y) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} y^{\alpha-1} e^{-y/\beta}$; $\eta = \log(\alpha\beta)$; $l(\eta; y) = ye^{-\eta} + \eta$.

- In `gss`, use something like `gssanova(y~x,family="binomial")`; only one link is used for each family, with η free of constraint.
- `predict` and `project` also work for `gssanova` fits, with calculations taking place on the link scale.

Example: Colorectal Cancer Mortality Rate

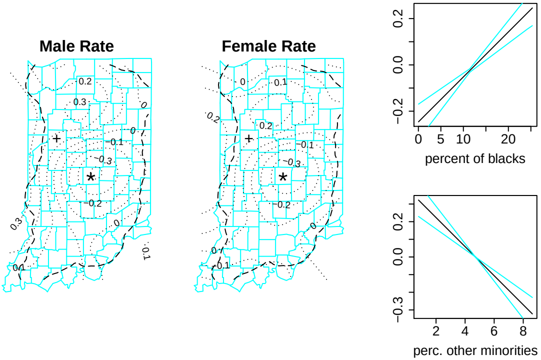
- Data: Colorectal cancer mortality during 2000 – 2004 were registered in the 92 counties of Indiana; Census 2000 is also available.
- Modeling: $Y \sim \text{Poisson}(e^{\eta} p)$, with p the population, and η is modeled as a function of sex, location, racial composition, etc.

Example: Colorectal Cancer Mortality Rate

- Data: Colorectal cancer mortality during 2000 – 2004 were registered in the 92 counties of Indiana; Census 2000 is also available.
- Modeling: $Y \sim \text{Poisson}(e^{\eta} p)$, with p the population, and η is modeled as a function of sex, location, racial composition, etc.

Example: Colorectal Cancer Mortality Rate

- Data: Colorectal cancer mortality during 2000 – 2004 were registered in the 92 counties of Indiana; Census 2000 is also available.
- Modeling: $Y \sim \text{Poisson}(e^{\eta} p)$, with p the population, and η is modeled as a function of sex, location, racial composition, etc.



Example: Colorectal Cancer Mortality Rate

First load data frame ColoCan with elements pop, event, sex, brt, ort, scrn, and geog, then fit an initial model.

```
data(ColoCan)
fit0 <- gssanova(event~sex*(geog+brt+ort+scrn),"poisson",
                  offset=log(pop),data=ColoCan,nbasis=40)
```

Check for the "significance" of some terms, then fit an additive model.

```
project(fit.cc.0,c("sex","geog","sex:geog","brt","ort"))
fit1 <- gssanova(event~sex*geog+brt+ort,"poisson",
                  offset=log(pop),data=ColoCan,nbasis=40)
```

Now compare mortality rates in Tippecanoe and Marion counties.

```
est1 <- predict(fit.cc,ColoCan[c(49,79,141,171),],
                inc=c("sex","geog","sex:geog"))
exp(est1[2]-est1[1])
exp(est1[4]-est1[3])
```

Example: Colorectal Cancer Mortality Rate

First load data frame `ColoCan` with elements `pop`, `event`, `sex`, `brt`, `ort`, `scrn`, and `geog`, then fit an initial model.

```
data(ColoCan)
fit0 <- gssanova(event~sex*(geog+brt+ort+scrn),"poisson",
                 offset=log(pop),data=ColoCan,nbasis=40)
```

Check for the "significance" of some terms, then fit an additive model.

```
project(fit.cc.0,c("sex","geog","sex:geog","brt","ort"))
fit1 <- gssanova(event~sex*geog+brt+ort,"poisson",
                 offset=log(pop),data=ColoCan,nbasis=40)
```

Now compare mortality rates in Tippecanoe and Marion counties.

```
est1 <- predict(fit.cc,ColoCan[c(49,79,141,171),],
                inc=c("sex","geog","sex:geog"))
exp(est1[2]-est1[1])
exp(est1[4]-est1[3])
```

Example: Colorectal Cancer Mortality Rate

First load data frame `ColoCan` with elements `pop`, `event`, `sex`, `brt`, `ort`, `scrn`, and `geog`, then fit an initial model.

```
data(ColoCan)
fit0 <- gssanova(event~sex*(geog+brt+ort+scrn),"poisson",
                 offset=log(pop),data=ColoCan,nbasis=40)
```

Check for the "significance" of some terms, then fit an additive model.

```
project(fit.cc.0,c("sex","geog","sex:geog","brt","ort"))
fit1 <- gssanova(event~sex*geog+brt+ort,"poisson",
                 offset=log(pop),data=ColoCan,nbasis=40)
```

Now compare mortality rates in Tippecanoe and Marion counties.

```
est1 <- predict(fit.cc,ColoCan[c(49,79,141,171),],
                inc=c("sex","geog","sex:geog"))
exp(est1[2]-est1[1])
exp(est1[4]-est1[3])
```

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\left\{-\int_0^t \lambda(u)du\right\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}e^z = \nu t^{\nu-1}e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t}\frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}\frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu)dt$, where $\eta = \mu$ is a function of covariate u.
- In `gss`, use something like `gssanova(y~u,family="weibull")`, where `y` is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\left\{-\int_0^t \lambda(u)du\right\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}e^z = \nu t^{\nu-1}e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t}\frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}\frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu)dt$, where $\eta = \mu$ is a function of covariate u.
- In gss, use something like `gssanova(y~u,family="weibull")`, where y is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\{-\int_0^t \lambda(u)du\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t} e^z = \nu t^{\nu-1} e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t} \frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t} \frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu) dt$, where $\eta = \mu$ is a function of covariate u.
- In gss, use something like `gssanova(y~u,family="weibull")`, where y is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\{-\int_0^t \lambda(u)du\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t} e^z = \nu t^{\nu-1} e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t} \frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t} \frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \leq C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu) dt$, where $\eta = \mu$ is a function of covariate u.
- In gss, use something like `gssanova(y~u,family="weibull")`, where y is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\left\{-\int_0^t \lambda(u)du\right\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}e^z = \nu t^{\nu-1}e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t}\frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}\frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu)dt$, where $\eta = \mu$ is a function of covariate u.
- In `gss`, use something like `gssanova(y~u,family="weibull")`, where `y` is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Regression with Accelerated Life Models

- Consider life time T with survival function $P(T > t) = S(t)$ and hazard function $\lambda(t) = -d \log S(t)/dt$; $S(t) = \exp\left\{-\int_0^t \lambda(u)du\right\}$.
- When $\log T \sim f(z) = f\left(\frac{\log t - \mu}{\sigma}\right)$, one has an accelerated life model.

Accelerated Life Models

- Weibull: $f(z) = we^{-w}$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}e^z = \nu t^{\nu-1}e^{-\nu\mu}$, for $\nu = 1/\sigma$.
- Log Normal: $f(z) = \phi(z)$; $\lambda(t) = \frac{\nu}{t}\frac{\phi(z)}{1-\Phi(z)}$.
- Log Logistic: $f(z) = w/(1+w)^2$, for $w = e^z$; $\lambda(t) = \frac{\nu}{t}\frac{e^z}{1+e^z}$.

- Observing (X, δ, Z), where $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, one may use $l(\eta; Y_i) = \delta \log \lambda(X_i; \eta(u_i), \nu) - \int_{Z_i}^{X_i} \lambda(t; \eta(u_i), \nu)dt$, where $\eta = \mu$ is a function of covariate u.
- In `gss`, use something like `gssanova(y~u,family="weibull")`, where `y` is a matrix of 2 or 3 columns, (X_i, δ_i, Z_i).

Example: Stanford Heart Transplant

- Load data frame `stan` with elements `time`, `status`, and `age`.

```
data(stan)
```

- Fit a Weibull regression model

```
fit <- gssanova(cbind(time+.01,status)~age,"weibull",
                data=stan)
```

- With $\lambda(t,u)=\nu t^{\nu-1}e^{-\nu(\eta_\emptyset+\eta_u)}=\lambda_0(t)\lambda_1(u)$, the Weibull model is a proportional hazard model.
- One may evaluate the fitted η_u,

```
gd <- seq(min(stan$age),max(stan$age),length=51)
est <- predict(fit,data.frame(age=gd),se=TRUE,inc="age")
```

and plot the relative risk $\lambda_1(u)=e^{-\nu\eta_u}$,

```
plot(gd,exp(-fit$nu*est$fit),type="l",ylim=c(0,5))
lines(gd,exp(-fit$nu*(est$fit-1.96*est$se)),col=5)
lines(gd,exp(-fit$nu*(est$fit+1.96*est$se)),col=5)
```

Example: Stanford Heart Transplant

- Load data frame `stan` with elements `time`, `status`, and `age`.

```
data(stan)
```

- Fit a Weibull regression model

```
fit <- gssanova(cbind(time+.01,status)~age,"weibull",
                data=stan)
```

- With $\lambda(t,u) = \nu t^{\nu-1} e^{-\nu(\eta_\emptyset+\eta_u)} = \lambda_0(t)\lambda_1(u)$, the Weibull model is a proportional hazard model.
- One may evaluate the fitted η_u,

```
gd <- seq(min(stan$age),max(stan$age),length=51)
est <- predict(fit,data.frame(age=gd),se=TRUE,inc="age")
```

and plot the relative risk $\lambda_1(u) = e^{-\nu\eta_u}$,

```
plot(gd,exp(-fit$nu*est$fit),type="l",ylim=c(0,5))
lines(gd,exp(-fit$nu*(est$fit-1.96*est$se)),col=5)
lines(gd,exp(-fit$nu*(est$fit+1.96*est$se)),col=5)
```

Example: Stanford Heart Transplant

- Load data frame `stan` with elements `time`, `status`, and `age`.

```
data(stan)
```

- Fit a Weibull regression model

```
fit <- gssanova(cbind(time+.01,status)~age,"weibull",
                data=stan)
```

- With $\lambda(t,u) = \nu t^{\nu-1} e^{-\nu(\eta_\emptyset+\eta_u)} = \lambda_0(t)\lambda_1(u)$, the Weibull model is a proportional hazard model.
- One may evaluate the fitted η_u,

```
gd <- seq(min(stan$age),max(stan$age),length=51)
est <- predict(fit,data.frame(age=gd),se=TRUE,inc="age")
```

and plot the relative risk $\lambda_1(u) = e^{-\nu\eta_u}$,

```
plot(gd,exp(-fit$nu*est$fit),type="l",ylim=c(0,5))
lines(gd,exp(-fit$nu*(est$fit-1.96*est$se)),col=5)
lines(gd,exp(-fit$nu*(est$fit+1.96*est$se)),col=5)
```

Example: Stanford Heart Transplant

- Load data frame `stan` with elements `time`, `status`, and `age`.

```
data(stan)
```

- Fit a Weibull regression model

```
fit <- gssanova(cbind(time+.01,status)~age,"weibull",
                data=stan)
```

- With $\lambda(t,u) = \nu t^{\nu-1} e^{-\nu(\eta_\emptyset+\eta_u)} = \lambda_0(t)\lambda_1(u)$, the Weibull model is a proportional hazard model.
- One may evaluate the fitted η_u,

```
gd <- seq(min(stan$age),max(stan$age),length=51)
est <- predict(fit,data.frame(age=gd),se=TRUE,inc="age")
```

and plot the relative risk $\lambda_1(u) = e^{-\nu\eta_u}$,

```
plot(gd,exp(-fit$nu*est$fit),type="l",ylim=c(0,5))
lines(gd,exp(-fit$nu*(est$fit-1.96*est$se)),col=5)
lines(gd,exp(-fit$nu*(est$fit+1.96*est$se)),col=5)
```

Example: Stanford Heart Transplant

- Load data frame `stan` with elements `time`, `status`, and `age`.

```
data(stan)
```

- Fit a Weibull regression model

```
fit <- gssanova(cbind(time+.01,status)~age,"weibull",
                data=stan)
```

- With $\lambda(t,u) = \nu t^{\nu-1} e^{-\nu(\eta_\emptyset+\eta_u)} = \lambda_0(t)\lambda_1(u)$, the Weibull model is a proportional hazard model.
- One may evaluate the fitted η_u,

```
gd <- seq(min(stan$age),max(stan$age),length=51)
est <- predict(fit,data.frame(age=gd),se=TRUE,inc="age")
```

  and plot the relative risk $\lambda_1(u) = e^{-\nu\eta_u}$,

```
plot(gd,exp(-fit$nu*est$fit),type="l",ylim=c(0,5))
lines(gd,exp(-fit$nu*(est$fit-1.96*est$se)),col=5)
lines(gd,exp(-fit$nu*(est$fit+1.96*est$se)),col=5)
```

Outline

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^{n} l_i\big(\eta(x_i) + \mathbf{z}_i^T\mathbf{b}; Y_i\big) + \mathbf{b}^T\Sigma\mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

Random Effects

- The random effects $\mathbf{z}^T\mathbf{b}$ are also known as variance components; $\mathrm{Var}(\mathbf{Y}) = \sigma^2(I + ZBZ^T)$, for $Z^T = (\mathbf{z}_1, \dots, \mathbf{z}_n)$.
- For longitudinal data from subjects $s_i \in \mathcal{S}$, $\mathbf{z}_i^T\mathbf{b} = b_{s_i}$, $b_s \sim N(0, \tau^2)$.
- For clustered data from clusters $c \in \mathcal{C}$, $\mathbf{z}_i^T\mathbf{b} = b_{c_i}$, $b_c \sim N(0, \tau_c^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^n l_i\big(\eta(x_i) + \mathbf{z}_i^T\mathbf{b}; Y_i\big) + \mathbf{b}^T\Sigma\mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

Random Effects

- The random effects $\mathbf{z}^T \mathbf{b}$ are also known as variance components; $\mathrm{Var}(\mathbf{Y}) = \sigma^2(I + ZBZ^T)$, for $Z^T = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$.
- For longitudinal data from subjects $s_i \in \mathcal{S}$, $\mathbf{z}_i^T \mathbf{b} = b_{s_i}$, $b_s \sim N(0, \tau^2)$.
- For clustered data from clusters $c \in \mathcal{C}$, $\mathbf{z}_i^T \mathbf{b} = b_{c_i}$, $b_c \sim N(0, \tau_c^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^n l_i\big(\eta(x_i) + \mathbf{z}_i^T \mathbf{b}; Y_i\big) + \mathbf{b}^T \Sigma \mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

Random Effects

- The random effects $\mathbf{z}^T\mathbf{b}$ are also known as variance components; $\mathrm{Var}(\mathbf{Y}) = \sigma^2(I + ZBZ^T)$, for $Z^T = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$.
- For longitudinal data from subjects $s_i \in \mathcal{S}$, $\mathbf{z}_i^T\mathbf{b} = b_{s_i}$, $b_s \sim N(0, \tau^2)$.
- For clustered data from clusters $c \in \mathcal{C}$, $\mathbf{z}_i^T\mathbf{b} = b_{c_i}$, $b_c \sim N(0, \tau_c^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^n l_i\big(\eta(x_i) + \mathbf{z}_i^T\mathbf{b}; Y_i\big) + \mathbf{b}^T\Sigma\mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T \mathbf{b}$, and minimize
$$\frac{1}{n} \sum_{i=1}^{n} l_i\big(\eta(x_i) + \mathbf{z}_i^T \mathbf{b}; Y_i\big) + \mathbf{b}^T \Sigma \mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

Random Effects

- The random effects $\mathbf{z}^T\mathbf{b}$ are also known as variance components; $\text{Var}(\mathbf{Y}) = \sigma^2(I + ZBZ^T)$, for $Z^T = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$.
- For longitudinal data from subjects $s_i \in \mathcal{S}$, $\mathbf{z}_i^T\mathbf{b} = b_{s_i}$, $b_s \sim N(0, \tau^2)$.
- For clustered data from clusters $c \in \mathcal{C}$, $\mathbf{z}_i^T\mathbf{b} = b_{c_i}$, $b_c \sim N(0, \tau_c^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^n l_i\big(\eta(x_i) + \mathbf{z}_i^T\mathbf{b}; Y_i\big) + \mathbf{b}^T\Sigma\mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

- Using `random=~1|id`, with factor `id` containing $s_i \in \mathcal{S}$, one sets $Z = \text{diag}(\mathbf{1}, \ldots, \mathbf{1})$ and $\Sigma \propto \gamma I$, for $\gamma = 1/\tau^2$.
- Using `random=~id|id`, one sets $Z = \text{diag}(\mathbf{1}, \ldots, \mathbf{1})$ and $\Sigma \propto \text{diag}(\gamma_c)$.

Random Effects and Mixed-Effect Models

- Consider $Y_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b} + \epsilon_i$, $\mathbf{b} \sim N(\mathbf{0}, \sigma^2 B)$, $\epsilon \sim N(0, \sigma^2)$.

Random Effects

- The random effects $\mathbf{z}^T\mathbf{b}$ are also known as variance components; $\text{Var}(\mathbf{Y}) = \sigma^2(I + ZBZ^T)$, for $Z^T = (\mathbf{z}_1, \dots, \mathbf{z}_n)$.
- For longitudinal data from subjects $s_i \in \mathcal{S}$, $\mathbf{z}_i^T\mathbf{b} = b_{s_i}$, $b_s \sim N(0, \tau^2)$.
- For clustered data from clusters $c \in \mathcal{C}$, $\mathbf{z}_i^T\mathbf{b} = b_{c_i}$, $b_c \sim N(0, \tau_c^2)$.

- In general, replace $\eta(x_i)$ by $\zeta_i = \eta(x_i) + \mathbf{z}_i^T\mathbf{b}$, and minimize
$$\frac{1}{n}\sum_{i=1}^n l_i\big(\eta(x_i) + \mathbf{z}_i^T\mathbf{b}; Y_i\big) + \mathbf{b}^T\Sigma\mathbf{b} + \lambda J(\eta),$$
where $\Sigma \propto B^{-1}$, usually structured, contains correlation parameters.

- Using `random=~1|id`, with factor `id` containing $s_i \in \mathcal{S}$, one sets $Z = \text{diag}(\mathbf{1}, \dots, \mathbf{1})$ and $\Sigma \propto \gamma I$, for $\gamma = 1/\tau^2$.
- Using `random=~id|id`, one sets $Z = \text{diag}(\mathbf{1}, \dots, \mathbf{1})$ and $\Sigma \propto \text{diag}(\gamma_c)$.

Example: Treatment of Bacteriuria

- Load bacteriuria with elements infect, trt, and time.

```
data(bacteriuria); bact <- bacteriuria
```

- Fit logistic regression model with subject random effect.

```
id.z <- (1:820)[bact$id%in%c(3,38)]
fit0 <- gssanova(infect~trt*time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the "significance" of interaction, then fit an additive model.

```
project(fit0,c("trt","time"))
fit1 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the magnitudes of b_s, noticing disparity between trt levels.

```
var(fit1$b[1:36]); var(fit1$b[37:72])
```

- Attach different γ for the two groups, $\Sigma \propto \mathrm{diag}(\gamma_1 I, \gamma_2 I)$.

```
fit2 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~trt|id,id.basis=id.z)
```

Example: Treatment of Bacteriuria

- Load `bacteriuria` with elements `infect`, `trt`, and `time`.

```
data(bacteriuria); bact <- bacteriuria
```

- Fit logistic regression model with subject random effect.

```
id.z <- (1:820)[bact$id%in%c(3,38)]
fit0 <- gssanova(infect~trt*time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the "significance" of interaction, then fit an additive model.

```
project(fit0,c("trt","time"))
fit1 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the magnitudes of b_s, noticing disparity between `trt` levels.

```
var(fit1$b[1:36]); var(fit1$b[37:72])
```

- Attach different γ for the two groups, $\Sigma \propto \mathrm{diag}(\gamma_1 I, \gamma_2 I)$.

```
fit2 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~trt|id,id.basis=id.z)
```

Example: Treatment of Bacteriuria

- Load bacteriuria with elements infect, trt, and time.

```
data(bacteriuria); bact <- bacteriuria
```

- Fit logistic regression model with subject random effect.

```
id.z <- (1:820)[bact$id%in%c(3,38)]
fit0 <- gssanova(infect~trt*time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the "significance" of interaction, then fit an additive model.

```
project(fit0,c("trt","time"))
fit1 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the magnitudes of b_s, noticing disparity between trt levels.

```
var(fit1$b[1:36]); var(fit1$b[37:72])
```

- Attach different γ for the two groups, $\Sigma \propto \mathrm{diag}(\gamma_1 I, \gamma_2 I)$.

```
fit2 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~trt|id,id.basis=id.z)
```

Example: Treatment of Bacteriuria

- Load `bacteriuria` with elements `infect`, `trt`, and `time`.

```
data(bacteriuria); bact <- bacteriuria
```

- Fit logistic regression model with subject random effect.

```
id.z <- (1:820)[bact$id%in%c(3,38)]
fit0 <- gssanova(infect~trt*time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the “significance” of interaction, then fit an additive model.

```
project(fit0,c("trt","time"))
fit1 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the magnitudes of b_s, noticing disparity between `trt` levels.

```
var(fit1$b[1:36]); var(fit1$b[37:72])
```

- Attach different γ for the two groups, $\Sigma \propto \mathrm{diag}(\gamma_1 I, \gamma_2 I)$.

```
fit2 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~trt|id,id.basis=id.z)
```

Example: Treatment of Bacteriuria

- Load bacteriuria with elements infect, trt, and time.

```
data(bacteriuria); bact <- bacteriuria
```

- Fit logistic regression model with subject random effect.

```
id.z <- (1:820)[bact$id%in%c(3,38)]
fit0 <- gssanova(infect~trt*time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the "significance" of interaction, then fit an additive model.

```
project(fit0,c("trt","time"))
fit1 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~1|id,id.basis=id.z)
```

- Check the magnitudes of b_s, noticing disparity between trt levels.

```
var(fit1$b[1:36]); var(fit1$b[37:72])
```

- Attach different γ for the two groups, $\Sigma \propto \text{diag}(\gamma_1 I, \gamma_2 I)$.

```
fit2 <- gssanova(infect~trt+time,"binomial",data=bact,
                 random=~trt|id,id.basis=id.z)
```

Outline

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.
- In `gss`, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.

- In gss, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.

- In `gss`, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.

- In `gss`, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.
- On $\mathcal{X}_1 \times \mathcal{X}_2$, $\eta = \eta_1 + \eta_2$ implies $X_{\langle 1 \rangle} \perp X_{\langle 2 \rangle}$.
- On $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, $\eta = \eta_1 + \eta_2 + \eta_3 + \eta_{12} + \eta_{13}$ implies $X_{\langle 2 \rangle} \perp X_{\langle 3 \rangle} | X_{\langle 1 \rangle}$.
- In `gss`, use something like `ssden(~x)`.

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.
  - On $\mathcal{X}_1 \times \mathcal{X}_2$, $\eta = \eta_1 + \eta_2$ implies $X_{\langle 1\rangle} \perp X_{\langle 2\rangle}$.
  - On $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, $\eta = \eta_1 + \eta_2 + \eta_3 + \eta_{12} + \eta_{13}$ implies $X_{\langle 2\rangle} \perp X_{\langle 3\rangle} | X_{\langle 1\rangle}$.
- In gss, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.

- In `gss`, use something like `ssden(~x)`.

Density Estimation

- For $X_i \sim f(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)}}$, use $L(\eta) = -\frac{1}{n}\sum_i \eta(X_i) + \log \int_{\mathcal{X}} e^{\eta(x)}$ in $L(\eta) + \lambda J(\eta)$, for $\mathcal{X}$ generic.
  - With $\eta = \eta_\emptyset + \eta_x$, $f(x) = e^{\eta_x(x)} / \int_{\mathcal{X}} e^{\eta_x(x)}$.
  - $\mathcal{X}$ is part of data, and should be finite so that $\int_{\mathcal{X}} 1 < \infty$.
- On $\mathcal{X} = \prod_\gamma \mathcal{X}_\gamma$, ANOVA structures imply conditional independence.

- In gss, use something like `ssden(~x)`.

```
data(buffalo); buff <- buffalo
fit0 <- ssden(~buff,domain=data.frame(buff=c(0,150)))
fit1 <- ssden(~buff,domain=data.frame(buff=c(20,130)))
plot(0:150,dssden(fit0,0:150),type="l"); abline(h=0,lty=2)
lines(20:130,dssden(fit1,20:130),col=4)
pp <- qssden(fit1,(0:5)/5); pssden(fit1,pp)
```

Example: AIDS Incubation

- Load `aids` with elements `incu`, `infe`, and `age`.

```
data(aids); aids1 <- aids[aids$age>=60,]
```

- Generate quadrature on $\mathcal{X} = \{X_{\langle 1 \rangle} \leq X_{\langle 2 \rangle}\}$.

```
qd.pt <- expand.grid(incu=2*(1:50)-1,infe=2*(1:50)-1)
qd.pt <- qd.pt[qd.pt$incu<=qd.pt$infe,]
qd.wt <- rep(1,nrow(qd.pt));
qd.wt[qd.pt$incu==qd.pt$infe] <- .5
my.qd <- list(pt=qd.pt,wt=qd.wt)
```

- Calculate fits with and without η_{12}.

```
dm <- data.frame(incu=c(0,100),infe=c(0,100))
fit0 <- ssden(~incu*infe,data=aids1,domain=dm,quad=my.qd)
project(fit0,c("incu","infe"))
fit1 <- ssden(~incu+infe,data=aids1,domain=dm,quad=my.qd)
```

- $X_{\langle 1 \rangle}$ and $X_{\langle 2 \rangle}$ are always dependent with truncation; $\eta_{12} = 0$ implies pre-truncation independence.

Example: AIDS Incubation

- Load `aids` with elements `incu`, `infe`, and `age`.

```
data(aids); aids1 <- aids[aids$age>=60,]
```

- Generate quadrature on $\mathcal{X} = \{X_{\langle 1\rangle} \leq X_{\langle 2\rangle}\}$.

```
qd.pt <- expand.grid(incu=2*(1:50)-1,infe=2*(1:50)-1)
qd.pt <- qd.pt[qd.pt$incu<=qd.pt$infe,]
qd.wt <- rep(1,nrow(qd.pt));
qd.wt[qd.pt$incu==qd.pt$infe] <- .5
my.qd <- list(pt=qd.pt,wt=qd.wt)
```

- Calculate fits with and without η_{12}.

```
dm <- data.frame(incu=c(0,100),infe=c(0,100))
fit0 <- ssden(~incu*infe,data=aids1,domain=dm,quad=my.qd)
project(fit0,c("incu","infe"))
fit1 <- ssden(~incu+infe,data=aids1,domain=dm,quad=my.qd)
```

- $X_{\langle 1\rangle}$ and $X_{\langle 2\rangle}$ are always dependent with truncation; $\eta_{12} = 0$ implies pre-truncation independence.

Example: AIDS Incubation

- Load `aids` with elements `incu`, `infe`, and `age`.

```
data(aids); aids1 <- aids[aids$age>=60,]
```

- Generate quadrature on $\mathcal{X} = \{X_{\langle 1 \rangle} \leq X_{\langle 2 \rangle}\}$.

```
qd.pt <- expand.grid(incu=2*(1:50)-1,infe=2*(1:50)-1)
qd.pt <- qd.pt[qd.pt$incu<=qd.pt$infe,]
qd.wt <- rep(1,nrow(qd.pt));
qd.wt[qd.pt$incu==qd.pt$infe] <- .5
my.qd <- list(pt=qd.pt,wt=qd.wt)
```

- Calculate fits with and without η_{12}.

```
dm <- data.frame(incu=c(0,100),infe=c(0,100))
fit0 <- ssden(~incu*infe,data=aids1,domain=dm,quad=my.qd)
project(fit0,c("incu","infe"))
fit1 <- ssden(~incu+infe,data=aids1,domain=dm,quad=my.qd)
```

- $X_{\langle 1 \rangle}$ and $X_{\langle 2 \rangle}$ are always dependent with truncation; $\eta_{12} = 0$ implies pre-truncation independence.

Example: AIDS Incubation

- Load aids with elements incu, infe, and age.

```
data(aids); aids1 <- aids[aids$age>=60,]
```

- Generate quadrature on $\mathcal{X} = \{X_{\langle 1\rangle} \leq X_{\langle 2\rangle}\}$.

```
qd.pt <- expand.grid(incu=2*(1:50)-1,infe=2*(1:50)-1)
qd.pt <- qd.pt[qd.pt$incu<=qd.pt$infe,]
qd.wt <- rep(1,nrow(qd.pt));
qd.wt[qd.pt$incu==qd.pt$infe] <- .5
my.qd <- list(pt=qd.pt,wt=qd.wt)
```

- Calculate fits with and without η_{12}.

```
dm <- data.frame(incu=c(0,100),infe=c(0,100))
fit0 <- ssden(~incu*infe,data=aids1,domain=dm,quad=my.qd)
project(fit0,c("incu","infe"))
fit1 <- ssden(~incu+infe,data=aids1,domain=dm,quad=my.qd)
```

- $X_{\langle 1\rangle}$ and $X_{\langle 2\rangle}$ are always dependent with truncation; $\eta_{12} = 0$ implies pre-truncation independence.

Example: AIDS Incubation

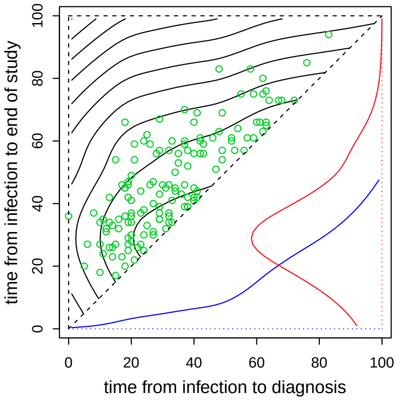
- Evaluate conditional densities $f(x_{\langle 1\rangle}|x_{\langle 2\rangle})$ and $f(x_{\langle 2\rangle}|x_{\langle 1\rangle})$.

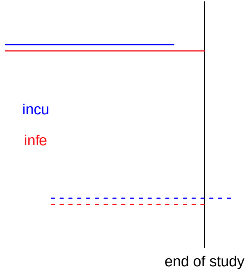
```
xx <- 2*(1:50)-1
f.incu <- cdssden(fit1,xx,data.frame(infe=50))$pdf
f.infe <- cdssden(fit1,xx,data.frame(incu=50))$pdf
```

Example: AIDS Incubation

- Evaluate conditional densities $f(x_{\langle 1 \rangle}|x_{\langle 2 \rangle})$ and $f(x_{\langle 2 \rangle}|x_{\langle 1 \rangle})$.

```
xx <- 2*(1:50)-1
f.incu <- cdssden(fit1,xx,data.frame(infe=50))$pdf
f.infe <- cdssden(fit1,xx,data.frame(incu=50))$pdf
```





Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n}\sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n}\sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.

- With $\eta = \eta_\emptyset + \eta_x + \eta_y + \eta_{xy}$, $f(y|x) = e^{\eta_y(y)+\eta_{xy}(x,y)} / \int_{\mathcal{Y}} e^{\eta_y(y)+\eta_{xy}(x,y)}$.
- $\mathcal{X}$, $\mathcal{Y}$ are generic, and η_y, η_{xy} may be futher decomposed.

- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n}\sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.
- With $\eta = \eta_\emptyset + \eta_x + \eta_y + \eta_{xy}$, $f(y|x) = e^{\eta_y(y)+\eta_{xy}(x,y)} / \int_{\mathcal{Y}} e^{\eta_y(y)+\eta_{xy}(x,y)}$.
- $\mathcal{X}$, $\mathcal{Y}$ are generic, and η_y, η_{xy} may be futher decomposed.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n} \sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.
- With $\eta = \eta_\emptyset + \eta_x + \eta_y + \eta_{xy}$, $f(y|x) = e^{\eta_y(y)+\eta_{xy}(x,y)} / \int_{\mathcal{Y}} e^{\eta_y(y)+\eta_{xy}(x,y)}$.
- $\mathcal{X}$, $\mathcal{Y}$ are generic, and η_y, η_{xy} may be futher decomposed.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n}\sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.
- With $\eta = \eta_\emptyset + \eta_x + \eta_y + \eta_{xy}$, $f(y|x) = e^{\eta_y(y)+\eta_{xy}(x,y)} / \int_{\mathcal{Y}} e^{\eta_y(y)+\eta_{xy}(x,y)}$.
- $\mathcal{X}$, $\mathcal{Y}$ are generic, and η_y, η_{xy} may be futher decomposed.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `sslrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n} \sum_i \{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Regression with Cross-Classified Responses

- For $\mathcal{Y} = \{0, 1\}$, this reduces to logistic regression.
- In general, this "unwinds" contingency tables along an "x-axis" and estimates table probabilities as functions of covariates.

Conditional Density Estimation

- To estimate $f(y|x) = \frac{e^{\eta(x,y)}}{\int_{\mathcal{Y}} e^{\eta(x,y)}}$ on $\mathcal{X} \times \mathcal{Y}$ from (X_i, Y_i), use $L(\eta) = -\frac{1}{n}\sum_i \left\{\eta(X_i, Y_i) - \log \int_{\mathcal{Y}} e^{\eta(X_i,y)}\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sscden(~x*y,~y)`, where `~x*y` specifies model terms and `~y` designates y-variables; terms in η_x are removed.
- For $\mathcal{Y}$ all discrete, use something like `ssllrm(~x*y1*y2,~y1+y2)`, where `y1`, `y2` are factors.

Regression with Cross-Classified Responses

- For $\mathcal{Y} = \{0, 1\}$, this reduces to logistic regression.
- In general, this "unwinds" contingency tables along an "x-axis" and estimates table probabilities as functions of covariates.

Example: Penny Thickness

- Load `penny` with elements `mil` and `year`.

  ```
  data(penny); ydm <- data.frame(mil=c(49,61))
  ```

- Calculate fits with and without break.

  ```
  fit0 <- sscden(~year*mil,~mil,data=penny,ydomain=ydm)
  z <- factor(penny$year>1974.5)
  fit1 <- sscden(~(year+z)*mil,~mil,data=penny,ydomain=ydm)
  ```

- For quantiles of $f(y|x)$, use `qsscden(fit,pp,data.frame(...))`.

Example: Penny Thickness

- Load `penny` with elements `mil` and `year`.

```
data(penny); ydm <- data.frame(mil=c(49,61))
```

- Calculate fits with and without break.

```
fit0 <- sscden(~year*mil,~mil,data=penny,ydomain=ydm)
z <- factor(penny$year>1974.5)
fit1 <- sscden(~(year+z)*mil,~mil,data=penny,ydomain=ydm)
```

- For quantiles of $f(y|x)$, use `qsscden(fit,pp,data.frame(...))`.

Example: Penny Thickness

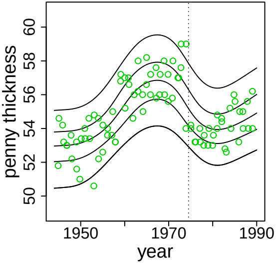
- Load `penny` with elements `mil` and `year`.

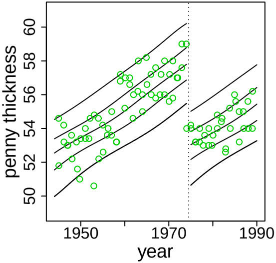
```
data(penny); ydm <- data.frame(mil=c(49,61))
```

- Calculate fits with and without break.

```
fit0 <- sscden(~year*mil,~mil,data=penny,ydomain=ydm)
z <- factor(penny$year>1974.5)
fit1 <- sscden(~(year+z)*mil,~mil,data=penny,ydomain=ydm)
```

- For quantiles of $f(y|x)$, use `qsscden(fit,pp,data.frame(...))`.





Example: Penny Thickness

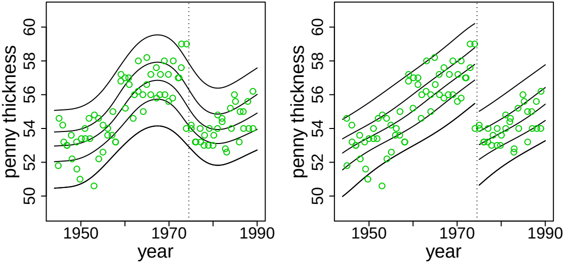
- Load `penny` with elements `mil` and `year`.

```
data(penny); ydm <- data.frame(mil=c(49,61))
```

- Calculate fits with and without break.

```
fit0 <- sscden(~year*mil,~mil,data=penny,ydomain=ydm)
z <- factor(penny$year>1974.5)
fit1 <- sscden(~(year+z)*mil,~mil,data=penny,ydomain=ydm)
```

- For quantiles of $f(y|x)$, use `qsscden(fit,pp,data.frame(...))`.



Outline

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t|U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \leq C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In gss, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\{-\int_0^t e^{\eta(s,u)} ds\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\}$ in $L(\eta) + \lambda J(\eta)$.
- With an additive $\eta = \eta_\emptyset + \eta_t + \eta_u$, one has a proportional hazard model.
- Adding random effects $\mathbf{z}^T\mathbf{b}$ to $\eta(t,u)$, one has frailty models.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- With an additive $\eta = \eta_\emptyset + \eta_t + \eta_u$, one has a proportional hazard model.
- Adding random effects $\mathbf{z}^T\mathbf{b}$ to $\eta(t,u)$, one has frailty models.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \leq C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.
- Use `hzdrate.sshzd(fit,data.frame(x=tt,u=uu),se,inc)` to evaluate fits, much like `predict` for `ssanova`/`gssanova` fits.
- Use `hzdcurve.sshzd(fit,tt,data.frame(u=uu),se)` to evaluate $e^{\eta(t,u)}$ for t on a grid.
- Use `survexp.sshzd(fit,tt,data.frame(u=uu))` to evaluate $S(t,u)$.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \le C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.
- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.
- Use `hzdrate.sshzd(fit,data.frame(x=tt,u=uu),se,inc)` to evaluate fits, much like `predict` for `ssanova`/`gssanova` fits.
- Use `hzdcurve.sshzd(fit,tt,data.frame(u=uu),se)` to evaluate $e^{\eta(t,u)}$ for t on a grid.
- Use `survexp.sshzd(fit,tt,data.frame(u=uu))` to evaluate $S(t,u)$.

Hazard Estimation

- For life time T with covariate U, consider $S(t,u) = P(T > t | U = u)$, $\lambda(t,u) = e^{\eta(t,u)} = -\partial \log S(t,u)/\partial t$; $S(t,u) = \exp\left\{-\int_0^t e^{\eta(s,u)} ds\right\}$.
- Observing $(X_i, \delta_i, Z_i, U_i)$, for $X = \min(T, C)$, $\delta = I_{[X \leq C]}$, and $Z < X$, use $L(\eta) = -\frac{1}{n}\sum_i \left\{\delta_i \eta(X_i, U_i) - \int_{Z_i}^{X_i} e^{\eta(t,U_i)} dt\right\}$ in $L(\eta) + \lambda J(\eta)$.

- In `gss`, use something like `sshzd(Surv(x,delta,z)~x*u)`, where η_t must be included in model terms; here t is represented by `x` in `x*u`.

- Use `hzdrate.sshzd(fit,data.frame(x=tt,u=uu),se,inc)` to evaluate fits, much like `predict` for `ssanova`/`gssanova` fits.
- Use `hzdcurve.sshzd(fit,tt,data.frame(u=uu),se)` to evaluate $e^{\eta(t,u)}$ for t on a grid.
- Use `survexp.sshzd(fit,tt,data.frame(u=uu))` to evaluate $S(t,u)$.

Example: Stanford Heart Transplant

- Calculate fits with and without η_{tu}.

```
fit0 <- sshzd(Surv(futime,status)~futime*age,data=stan)
project(fit0,inc=c("futime","age"))
fit1 <- sshzd(Surv(futime,status)~futime+age,data=stan)
```

  where `futime` is on the $t^* = \sqrt{t}$ scale; $\tilde{\lambda}(t,u) = \lambda(t^*,u)/2t^*$.

- Evaluate e^{η_u} on a grid, with `e.a$fit` on e^{η_u} and `e.a$se` on η_u.

```
e.a <- hzdrate.sshzd(fit1,data.frame(age=11:64),
                     se=TRUE,inc=c("age"))
```

- Evaluate $e^{\eta_\emptyset + \eta_t}$ on a grid.

```
e.b <- hzdrate.sshzd(fit1,data.frame(futime=0:60),
                     se=TRUE,inc=c("1","futime"))
```

Example: Stanford Heart Transplant

- Calculate fits with and without η_{tu}.

```
fit0 <- sshzd(Surv(futime,status)~futime*age,data=stan)
project(fit0,inc=c("futime","age"))
fit1 <- sshzd(Surv(futime,status)~futime+age,data=stan)
```

  where `futime` is on the $t^* = \sqrt{t}$ scale; $\tilde{\lambda}(t,u) = \lambda(t^*,u)/2t^*$.

- Evaluate e^{η_u} on a grid, with `e.a$fit` on e^{η_u} and `e.a$se` on η_u.

```
e.a <- hzdrate.sshzd(fit1,data.frame(age=11:64),
                     se=TRUE,inc=c("age"))
```

- Evaluate $e^{\eta_\emptyset+\eta_t}$ on a grid.

```
e.b <- hzdrate.sshzd(fit1,data.frame(futime=0:60),
                     se=TRUE,inc=c("1","futime"))
```

Example: Stanford Heart Transplant

- Calculate fits with and without η_{tu}.

```
fit0 <- sshzd(Surv(futime,status)~futime*age,data=stan)
project(fit0,inc=c("futime","age"))
fit1 <- sshzd(Surv(futime,status)~futime+age,data=stan)
```

  where `futime` is on the $t^* = \sqrt{t}$ scale; $\tilde{\lambda}(t,u) = \lambda(t^*,u)/2t^*$.

- Evaluate e^{η_u} on a grid, with `e.a$fit` on e^{η_u} and `e.a$se` on η_u.

```
e.a <- hzdrate.sshzd(fit1,data.frame(age=11:64),
                     se=TRUE,inc=c("age"))
```

```
plot(11:64,e.a$fit,type="l",ylim=c(0,5))
lines(11:64,e.a$fit*exp(1.96*e.a$se),col=5)
lines(11:64,e.a$fit/exp(1.96*e.a$se),col=5)
```

- Evaluate $e^{\eta_\emptyset+\eta_t}$ on a grid.

```
e.b <- hzdrate.sshzd(fit1,data.frame(futime=0:60),
                     se=TRUE,inc=c("1","futime"))
```

Example: Stanford Heart Transplant

- Calculate fits with and without η_{tu}.

```
fit0 <- sshzd(Surv(futime,status)~futime*age,data=stan)
project(fit0,inc=c("futime","age"))
fit1 <- sshzd(Surv(futime,status)~futime+age,data=stan)
```

where `futime` is on the $t^* = \sqrt{t}$ scale; $\tilde{\lambda}(t,u) = \lambda(t^*,u)/2t^*$.

- Evaluate e^{η_u} on a grid, with `e.a$fit` on e^{η_u} and `e.a$se` on η_u.

```
e.a <- hzdrate.sshzd(fit1,data.frame(age=11:64),
                     se=TRUE,inc=c("age"))
```

```
plot(11:64,e.a$fit,type="l",ylim=c(0,5))
lines(11:64,e.a$fit*exp(1.96*e.a$se),col=5)
lines(11:64,e.a$fit/exp(1.96*e.a$se),col=5)
```

- Evaluate $e^{\eta_\emptyset + \eta_t}$ on a grid.

```
e.b <- hzdrate.sshzd(fit1,data.frame(futime=0:60),
                     se=TRUE,inc=c("1","futime"))
```

Estimation of Relative Risk

- For $\lambda(t,u)=\lambda_0(t)\lambda_1(u)$, one may treat $\lambda_0(t)$ as a nuisance and estimate $\lambda_1(u)=e^{\eta_u(u)}$ via partial likelihood, using in $L(\eta)+\lambda J(\eta)$ $L(\eta)=-\frac{1}{n}\sum_{i=1}^{n}\delta_i\{\eta(U_i)-\log\sum_{j=1}^{n}I_{[Z_j<X_i\leq X_j]}e^{\eta(U_j)}\}$.
- In gss, use something like `sscox(Surv(x,delta,z)~u)`.

Estimation of Relative Risk

- For $\lambda(t,u) = \lambda_0(t)\lambda_1(u)$, one may treat $\lambda_0(t)$ as a nuisance and estimate $\lambda_1(u) = e^{\eta_u(u)}$ via partial likelihood, using in $L(\eta) + \lambda J(\eta)$ $L(\eta) = -\frac{1}{n}\sum_{i=1}^{n} \delta_i \{\eta(U_i) - \log \sum_{j=1}^{n} I_{[Z_j < X_i \leq X_j]} e^{\eta(U_j)}\}$.
- In `gss`, use something like `sscox(Surv(x,delta,z)~u)`.

Estimation of Relative Risk

- For $\lambda(t,u) = \lambda_0(t)\lambda_1(u)$, one may treat $\lambda_0(t)$ as a nuisance and estimate $\lambda_1(u) = e^{\eta_u(u)}$ via partial likelihood, using in $L(\eta) + \lambda J(\eta)$ $L(\eta) = -\frac{1}{n}\sum_{i=1}^{n} \delta_i \{\eta(U_i) - \log \sum_{j=1}^{n} I_{[Z_j < X_i \le X_j]} e^{\eta(U_j)}\}$.
- In `gss`, use something like `sscox(Surv(x,delta,z)~u)`.

Example: Stanford Heart Transplant

- Calculate the fit, which is invariant to the scaling of t.

```
fit2 <- sscox(Surv(time,status)~age,data=stan)
```

- Evaluate e^{η_u} on data points, then estimate $\lambda_0(t)$.

```
risk <- predict(fit2,stan)
fit3 <- sshzd(Surv(futime,status)~futime,data=stan,
              offset=log(risk))
```

- Evaluate $\lambda_1(u)$ and $\lambda_0(t)$ on grids.

```
e.a0 <- predict(fit2,data.frame(age=11:64),se=T,inc=c("age"))
e.b0 <- hzdcurve.sshzd(fit3,0:60,se=TRUE)
```

Estimation of Relative Risk

- For $\lambda(t,u) = \lambda_0(t)\lambda_1(u)$, one may treat $\lambda_0(t)$ as a nuisance and estimate $\lambda_1(u) = e^{\eta_u(u)}$ via partial likelihood, using in $L(\eta) + \lambda J(\eta)$ $L(\eta) = -\frac{1}{n}\sum_{i=1}^{n} \delta_i \big\{\eta(U_i) - \log \sum_{j=1}^{n} I_{[Z_j < X_i \le X_j]} e^{\eta(U_j)}\big\}$.
- In `gss`, use something like `sscox(Surv(x,delta,z)~u)`.

Example: Stanford Heart Transplant

- Calculate the fit, which is invariant to the scaling of t.

```
fit2 <- sscox(Surv(time,status)~age,data=stan)
```

- Evaluate e^{η_u} on data points, then estimate $\lambda_0(t)$.

```
risk <- predict(fit2,stan)
fit3 <- sshzd(Surv(futime,status)~futime,data=stan,
              offset=log(risk))
```

- Evaluate $\lambda_1(u)$ and $\lambda_0(t)$ on grids.

```
e.a0 <- predict(fit2,data.frame(age=11:64),se=T,inc=c("age"))
e.b0 <- hzdcurve.sshzd(fit3,0:60,se=TRUE)
```

Estimation of Relative Risk

- For $\lambda(t,u) = \lambda_0(t)\lambda_1(u)$, one may treat $\lambda_0(t)$ as a nuisance and estimate $\lambda_1(u) = e^{\eta_u(u)}$ via partial likelihood, using in $L(\eta) + \lambda J(\eta)$ $L(\eta) = -\frac{1}{n}\sum_{i=1}^{n} \delta_i \{\eta(U_i) - \log \sum_{j=1}^{n} I_{[Z_j < X_i \leq X_j]} e^{\eta(U_j)}\}$.
- In gss, use something like `sscox(Surv(x,delta,z)~u)`.

Example: Stanford Heart Transplant

- Calculate the fit, which is invariant to the scaling of t.

```
fit2 <- sscox(Surv(time,status)~age,data=stan)
```

- Evaluate e^{η_u} on data points, then estimate $\lambda_0(t)$.

```
risk <- predict(fit2,stan)
fit3 <- sshzd(Surv(futime,status)~futime,data=stan,
              offset=log(risk))
```

- Evaluate $\lambda_1(u)$ and $\lambda_0(t)$ on grids.

```
e.a0 <- predict(fit2,data.frame(age=11:64),se=T,inc=c("age"))
e.b0 <- hzdcurve.sshzd(fit3,0:60,se=TRUE)
```

Thank You!