Web-based Supplementary Materials for "Spatial regression with covariate measurement error: A semi-parametric approach," by Md Hamidul Huque, Howard D. Bondell, Raymond J. Carroll, and Louise M. Ryan.

Web Appendix A

Estimation of $\widehat{\theta}_2$ and its asymptotic distribution

We assume that the smoothing parameters are small relative to the sample size, i.e., $n^{1/2}\delta_j \to 0$ for $j = 1, 2$. From (4) we estimate θ_2 by minimizing the penalized sum of squares $\boldsymbol{J}(\theta_2) = n^{-1}\sum_{i=1}^{n}\{W_i - B_2^{\mathrm{T}}(S_i)\theta_2\}^2 + \delta_2\theta_2^{\mathrm{T}}D_2\theta_2$. We have

$$
\begin{aligned}
&\frac{\partial \boldsymbol{J}(\theta_2)}{\partial \theta_2} = 0 \\
&\Rightarrow 2n^{-1}\textstyle\sum_{i=1}^{n}\{W_i - B_2^{\mathrm{T}}(S_i)\theta_2\}B_2(S_i) + 2\delta_2 D_2\theta_2 = 0 \\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)W_i - \{n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)B_2^{\mathrm{T}}(S_i) - \delta_2 D_2\}\theta_2 = 0 \\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)W_i - \Lambda_{n2}^{-1}\theta_2 = 0 \\
&\Rightarrow \widehat{\theta}_2 = \Lambda_{n2}n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)W_i.
\end{aligned}
$$

Now, the asymptotics of $\widehat{\theta}_2$ can be given by

$$
\begin{aligned}
n^{1/2}(\widehat{\theta}_2 - \theta_2) \quad &= \quad n^{1/2}(\Lambda_{n2}n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)W_i - \theta_2) \\
&= n^{1/2}(\Lambda_{n2}n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)[B_2^{\mathrm{T}}(S_i)\theta_2 + U_i] - \theta_2) \\
&= n^{1/2}\Lambda_{n2}(n^{-1}\textstyle\sum_{i=1}^{n}B_2(S_i)B_2^{\mathrm{T}}(S_i) - \Lambda_{n2}^{-1})\theta_2 + \Lambda_{n2}n^{-1/2}\textstyle\sum_{i=1}^{n}B_2(S_i)U_i \\
&= \Lambda_{n2}n^{-1/2}\textstyle\sum_{i=1}^{n}B_2(S_i)U_i + o_p(1).
\end{aligned}
$$

Thus,

$$
(\widehat{\theta}_2 - \theta_2) \quad = \quad n^{-1}\Lambda_{n2}\textstyle\sum_{i=1}^{n}B_2(S_i)U_i + o_p(n^{-1/2}). \tag{15}
$$

This is because as, $n^{1/2}\delta_j \to 0$, then

$$n^{-1}\textstyle\sum_{i=1}^{n} B_2(S_i)B_2^{\mathrm{T}}(S_i) - \Lambda_{n2}^{-1} = \delta_2 D_2 = o_p(n^{-1/2}). \qquad (16)$$

Estimation of $\widehat{\theta}_1$

From (3), we estimate β_1 and θ_1 by minimizing the penalized sum of squares $\boldsymbol{J}(\beta,\theta_1) = n^{-1}\sum_{i=1}^{n}\{Y_i - B_2^{\mathrm{T}}(S_i)\theta_2\beta_1 - B_1^{\mathrm{T}}(S_i)\theta_1\}^2 + \delta_1\theta_1^{\mathrm{T}}D_1\theta_1$. That is,

$$\begin{aligned}
&\frac{\partial \boldsymbol{J}(\beta,\theta_1)}{\partial \beta_1} = 0\\
&\Rightarrow 2n^{-1}\textstyle\sum_{i=1}^{n}\{Y_i - B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - B_1^{\mathrm{T}}(S_i)\theta_1\}B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 = 0\\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}\{B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 Y_i - \widehat{\theta}_2 B_2^{\mathrm{T}}(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - \widehat{\theta}_2 B_2^{\mathrm{T}}(S_i)B_1^{\mathrm{T}}(S_i)\theta_1\} = 0\\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}\{B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 Y_i - \widehat{\theta}_2 B_2^{\mathrm{T}}(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - \widehat{\theta}_2 B_2^{\mathrm{T}}(S_i)B_1^{\mathrm{T}}(S_i)\theta_1\} = 0,
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\partial \boldsymbol{J}(\beta,\theta_1)}{\partial \theta_1} = 0\\
&\Rightarrow 2n^{-1}\textstyle\sum_{i=1}^{n}\{Y_i - B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - B_1^{\mathrm{T}}(S_i)\theta_1\}B_1(S_i) + 2\delta_1 D_1\theta_1 = 0\\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}\{B_1(S_i)Y_i - B_1(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - B_1(S_i)B_1^{\mathrm{T}}(S_i)\theta_1\} + \delta_1 D_1\theta_1 = 0\\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}B_1(S_i)Y_i - n^{-1}\sum_{i=1}^{n}B_1(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - (n^{-1}\sum_{i=1}^{n}B_1(S_i)B_1^{\mathrm{T}}(S_i) - \delta_1 D_1)\theta_1 = 0\\
&\Rightarrow n^{-1}\textstyle\sum_{i=1}^{n}B_1(S_i)Y_i - n^{-1}\sum_{i=1}^{n}B_1(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - \Lambda_{n1}^{-1}\theta_1 = 0\\
&\Rightarrow n^{-1}\{\textstyle\sum_{i=1}^{n}B_1(S_i)Y_i - n^{-1}\sum_{i=1}^{n}B_1(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\beta_1 - \Lambda_{n1}^{-1}\theta_1\} = 0.
\end{aligned}$$

This leads to the estimating equations

$$\begin{aligned}
0 \;=\; & n^{-1}\textstyle\sum_{i=1}^{n}\left\{\begin{matrix} B_1(S_1)Y_i \\ B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 Y_i \end{matrix}\right\}\\
& -n^{-1}\textstyle\sum_{i=1}^{n}\left\{\begin{matrix} \Lambda_{n1}^{-1} & B_1(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 \\ B_1^{\mathrm{T}}(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 & \widehat{\theta}_2^{\mathrm{T}}B_2^{\mathrm{T}}(S_i)B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 \end{matrix}\right\}\begin{pmatrix}\theta_1\\ \beta_1\end{pmatrix}.
\end{aligned}$$

Writing $\widehat{G}_2(S_i) = B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2$, we see that the estimating equations are

$$0 = \sum_{i=1}^n \{B_1(S_i)Y_i - \Lambda_{n1}^{-1}\widehat{\theta}_1 - B_1(S_i)\widehat{G}_2(S_i)\widehat{\beta}_1\}; \tag{17}$$

$$0 = \sum_{i=1}^n \widehat{G}_2(S_i)\{Y_i - B_1^{\mathrm{T}}(S_i)\widehat{\theta}_1 - \widehat{G}_2(S_i)\widehat{\beta}_1\}. \tag{18}$$

Now from (17) we see that

$$\widehat{\theta}_1 = V_n - R_n\widehat{\theta}_2\widehat{\beta}_1,$$

where,

$$V_n = \Lambda_{n1} n^{-1}\sum_{i=1}^n B_1(S_i)Y_i,$$

$$R_n = \Lambda_{n1} n^{-1}\sum_{i=1}^n B_1(S_i)B_2^{\mathrm{T}}(S_i).$$

Estimation of $\widehat{\beta}_1$

Substituting the value of $\widehat{\theta}_1$ in (18) we have

$$0 = \sum_{i=1}^n \widehat{G}_2(S_i)\{Y_i - B_1^{\mathrm{T}}(S_i)V_n + B_1^{\mathrm{T}}(S_i)R_n\widehat{\theta}_2\widehat{\beta}_1 - \widehat{G}_2(S_i)\widehat{\beta}_1\}.$$

From this,

$$\widehat{\beta}_1 = \frac{n^{-1}\sum_{i=1}^n \widehat{G}_2(S_i)\{Y_i - B_1^{\mathrm{T}}(S_i)V_n\}}{n^{-1}\sum_{i=1}^n \widehat{G}_2(S_i)\{\widehat{G}_2(S_i) - B_1^{\mathrm{T}}(S_i)R_n\widehat{\theta}_2\}}. \tag{19}$$

The numerator of (19) is

$$\begin{aligned}
& n^{-1}\sum_{i=1}^n \widehat{G}_2(S_i)\{Y_i - B_1^{\mathrm{T}}(S_i)V_n\} \\
&= n^{-1}\sum_{i=1}^n Y_i B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 - n^{-1}\sum_{i=1}^n \widehat{\theta}_2^{\mathrm{T}} B_2(S_i)B_1^{\mathrm{T}}(S_i)V_n \\
&= n^{-1}\sum_{i=1}^n Y_i B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 - n^{-1}\sum_{i=1}^n \widehat{\theta}_2^{\mathrm{T}} \{B_1(S_i)B_2^{\mathrm{T}}(S_i)\}^{\mathrm{T}} V_n \\
&= n^{-1}\sum_{i=1}^n Y_i B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 - \widehat{\theta}_2^{\mathrm{T}} R_n^{\mathrm{T}}\Lambda_{n1}^{-1}V_n \\
&= n^{-1}\sum_{i=1}^n Y_i B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 - V_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n\widehat{\theta}_2 \\
&= n^{-1}\sum_{i=1}^n \left\{Y_i B_2^{\mathrm{T}}(S_i) - V_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n\right\}\widehat{\theta}_2 \\
&= n^{-1}\sum_{i=1}^n Y_i\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2.
\end{aligned}$$

The denominator of (19) is

$$
\begin{aligned}
& n^{-1}\textstyle\sum_{i=1}^{n}\widehat{G}_2(S_i)\{\widehat{G}_2(S_i) - B_1^{\mathrm{T}}(S_i)R_n\widehat{\theta}_2 \\
& = \widehat{\theta}_2^{\mathrm{T}} n^{-1}\textstyle\sum_{i=1}^{n} B_2(S_i)\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2 \\
& = \widehat{\theta}_2^{\mathrm{T}}\{n^{-1}\textstyle\sum_{i=1}^{n} B_2(S_i)B_2^{\mathrm{T}}(S_i) - n^{-1}\textstyle\sum_{i=1}^{n} B_2(S_i)B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2 \\
& = \widehat{\theta}_2^{\mathrm{T}}(\mathcal{T}_n - R_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n)\widehat{\theta}_2.
\end{aligned}
$$

where, $\mathcal{T}_n = n^{-1}\sum_{i=1}^{n} B_2(S_i)B_2^{\mathrm{T}}(S_i)$ (say).

Hence,

$$
\widehat{\beta}_1 \quad = \quad \frac{n^{-1}\sum_{i=1}^{n} Y_i\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2}{\widehat{\theta}_2^{\mathrm{T}}(\mathcal{T}_n - R_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n)\widehat{\theta}_2}.
$$

Proof of Theorem 1

To ease exposition, let us define the following expressions:

$$
\begin{aligned}
\mathcal{A}_n &= n^{-1}\textstyle\sum_{i=1}^{n}\{G_2(S_i)\beta_1 + G_1(S_i)\}\{B_2(S_i) - R_n^{\mathrm{T}}B_1(S_i)\}^{\mathrm{T}}; \\
\mathcal{C}_n &= \mathcal{T}_n - R_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n; \\
\mathcal{D}_{ni} &= \{B_2(S_i) - R_n^{\mathrm{T}}B_1(S_i)\}^{\mathrm{T}}\theta_2; \\
\mathcal{F}_{ni} &= \theta_2^{\mathrm{T}}\mathcal{C}_n\Lambda_{n2}B_2(S_i) + B_2^{\mathrm{T}}(S_i)\Lambda_{n2}\mathcal{C}_n\theta_2; \\
\mathcal{G}_{ni} &= \mathcal{D}_{ni}/\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2; \\
\mathcal{H}_{ni} &= \mathcal{A}_n\Lambda_{n2}B_2(S_i)/\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 - \mathcal{A}_n\theta_2\mathcal{F}_{ni}/(\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2)^2.
\end{aligned}
\tag{20}
$$

Consider that the S_i's are fixed and known and recall that, $Y_i = G_2(S_i)\beta_1 + G_1(S_i) + \epsilon_i$. Substituting the expression for Y_i into the numerator of (19) and simplifying using the expression from (20), we have

$$n^{-1}\textstyle\sum_{i=1}^{n} Y_i\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2$$

$$= n^{-1}\textstyle\sum_{i=1}^{n}(B_2^{\mathrm{T}}(S_i)\theta_2\beta_1 + B_1^{\mathrm{T}}(S_i)\theta_1 + \epsilon_i)\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2$$

$$= n^{-1}\textstyle\sum_{i=1}^{n}(B_2^{\mathrm{T}}(S_i)\theta_2\beta_1 + B_1^{\mathrm{T}}(S_i)\theta_1)\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2 +$$

$$n^{-1}\textstyle\sum_{i=1}^{n}\epsilon_i\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2$$

$$= \mathcal{A}_n\widehat{\theta}_2 + n^{-1}\textstyle\sum_{i=1}^{n}\mathcal{D}_{ni}\epsilon_i$$

$$= \mathcal{A}_n\theta_2 + \mathcal{A}_n(\widehat{\theta}_2 - \theta_2) + n^{-1}\textstyle\sum_{i=1}^{n}\mathcal{D}_{ni}\epsilon_i.$$

Applying (15) to the above equation, we have

$$\mathcal{A}_n\theta_2 + n^{-1}\textstyle\sum_{i=1}^{n}\{\mathcal{A}_n\Lambda_{n2}B_2(S_i)U_i + \mathcal{D}_{ni}\epsilon_i\} + o_p(n^{-1/2}).$$

Again, the denominator of (19) is

$$\widehat{\theta}_2^{\mathrm{T}}(\mathcal{T}_n - R_n^{\mathrm{T}}\Lambda_{n1}^{-1}R_n)\widehat{\theta}_2 = \widehat{\theta}_2^{\mathrm{T}}\mathcal{C}_n\widehat{\theta}_2.$$

Now applying (15), the denominator becomes,

$$(\theta_2 + n^{-1}\textstyle\sum_{i=1}^{n}\Lambda_{n2}B_2(S_i)U_i) + o_p(n^{-1/2})^{\mathrm{T}}\mathcal{C}_n(\theta_2 + n^{-1}\textstyle\sum_{i=1}^{n}\Lambda_{n2}B_2(S_i)U_i) + o_p(n^{-1/2})$$

$$= \theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 + n^{-1}\textstyle\sum_{i=1}^{n}\theta_2^{\mathrm{T}}\mathcal{C}_n\Lambda_{n2}B_2(S_i)U_i + n^{-1}\textstyle\sum_{i=1}^{n}U_i^{\mathrm{T}}B_2(S_i)^{\mathrm{T}}\Lambda_{n2}\mathcal{C}_n\theta_2$$

$$+(n^{-1}\textstyle\sum_{i=1}^{n}\Lambda_{n2}B_2(S_i)U_i)^{\mathrm{T}}(n^{-1}\textstyle\sum_{i=1}^{n}\Lambda_{n2}B_2(S_i)U_i) + o_p(n^{-1/2})$$

$$= \theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 + n^{-1}\textstyle\sum_{i=1}^{n}\mathcal{F}_{ni}U_i + o_p(n^{-1/2}).$$

Then, by a Taylor series expansion,

$$\begin{aligned}
\widehat{\beta}_1 &= \frac{\mathcal{A}_n\theta_2 + n^{-1}\sum_{i=1}^{n}\{\mathcal{A}_n\Lambda_{n2}B_2(S_i)U_i + \mathcal{D}_{ni}\epsilon_i\}}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 + n^{-1}\sum_{i=1}^{n}\mathcal{F}_{ni}U_i} + o_p(n^{-1/2}) \\
&= \left\{\frac{\mathcal{A}_n\theta_2}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2} + \frac{n^{-1}\sum_{i=1}^{n}\{\mathcal{A}_n\Lambda_{n2}B_2(S_i)U_i + \mathcal{D}_{ni}\epsilon_i\}}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2}\right\}\left\{1 + \frac{\sum_{i=1}^{n}\mathcal{F}_{ni}U_i}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2}\right\}^{-1} + o_p(n^{-1/2}) \\
&= \frac{\mathcal{A}_n\theta_2}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2} + \frac{n^{-1}\sum_{i=1}^{n}\{\mathcal{A}_n\Lambda_{n2}B_2(S_i)U_i + \mathcal{D}_{ni}\epsilon_i\}}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2} \\
&\qquad - \frac{\mathcal{A}_n\theta_2}{(\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2)^2}n^{-1}\textstyle\sum_{i=1}^{n}\mathcal{F}_{ni}U_i + o_p(n^{-1/2}).
\end{aligned}$$

Thus

$$\widehat{\beta}_1 - \frac{\mathcal{A}_n\theta_2}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2} = n^{-1}\textstyle\sum_{i=1}^n(\mathcal{G}_{ni}\epsilon_i + \mathcal{H}_{ni}U_i) + o_p(n^{-1/2}).$$

Now considering the fact from equation (16) that $n^{-1}\sum_{i=1}^n B_2(S_i)B_2^{\mathrm{T}}(S_i) = \Lambda_{n2}^{-1} + o(n^{-1/2})$ and using this in the expression for $\mathcal{A}_n\theta_2$, we have

$$\begin{aligned}\mathcal{A}_n\theta_2 &= n^{-1}\textstyle\sum_{i=1}^n\beta_1\theta_2^{\mathrm{T}}B_2(S_i)\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\theta_2 \\ &\quad +\theta_1^{\mathrm{T}}n^{-1}\textstyle\sum_{i=1}^n B_1(S_i)\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\theta_2 \\ &= \beta_1\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 + \theta_1^{\mathrm{T}}\{\Lambda_{n1}^{-1} - n^{-1}\textstyle\sum_{i=1}^n B_1(S_i)B_1^{\mathrm{T}}(S_i)\}R_n\theta_2 \\ &= \beta_1\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2 + o_p(n^{-1/2}).\end{aligned}$$

Therefore,

$$\frac{\mathcal{A}_n\theta_2}{\theta_2^{\mathrm{T}}\mathcal{C}_n\theta_2} = \beta_1 + o_p(n^{-1/2}).$$

Hence,

$$n^{1/2}(\widehat{\beta}_1 - \beta_1) \sim \mathrm{Normal}(0, \sigma^2),$$

where

$$\sigma^2 = \lim_{n\to\infty} n^{-1}\textstyle\sum_{i=1}^n(\sigma_\epsilon^2\mathcal{G}_{ni}^2 + \sigma_u^2\mathcal{H}_{ni}^2). \tag{21}$$

Thus, $\hat{\beta}_1$ is a consistent estimate for β_1.

Derivation of smoothing matrix

From (3), we have

$$
\begin{aligned}
\widehat{Y}_i &= B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\widehat{\beta}_1 + B_1^{\mathrm{T}}(S_i)\widehat{\theta}_1 \\
&= B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2\widehat{\beta}_1 + B_1^{\mathrm{T}}(S_i)[V_n - R_n\widehat{\theta}_2\widehat{\beta}_1] \\
&= [B_2(S_i) - B_1^{\mathrm{T}}(S_i)R_n]^{\mathrm{T}}\widehat{\theta}_2\widehat{\beta}_2 + B_1^{\mathrm{T}}(S_i)V_n \\
&= [B_2(S_i) - B_1^{\mathrm{T}}(S_i)R_n]^{\mathrm{T}}\widehat{\theta}_2\left(\frac{n^{-1}\sum_{i=1}^n Y_i\{B_2^{\mathrm{T}}(S_i) - B_1^{\mathrm{T}}(S_i)R_n\}\widehat{\theta}_2}{\widehat{\theta}_2^{\mathrm{T}}\mathcal{C}_n\widehat{\theta}_2}\right) \\
&\qquad + B_1^{\mathrm{T}}(S_i)\Lambda_{n1}n^{-1}\sum_{i=1}^n B_1(S_i)Y_i \\
&= \frac{D_{ni}n^{-1}\sum_{i=1}^n D_{ni}Y_i}{\widehat{\theta}_2^{\mathrm{T}}\mathcal{C}_n\widehat{\theta}_2} + B_1^{\mathrm{T}}(S_i)\Lambda_{n1}n^{-1}\sum_{i=1}^n B_1(S_i)Y_i.
\end{aligned}
$$

Similarly from (4), we have

$$
\begin{aligned}
\widehat{W}_i &= B_2^{\mathrm{T}}(S_i)\widehat{\theta}_2 \\
&= B_2^{\mathrm{T}}(S_i)\Lambda_{n2}n^{-1}\sum_{i=1}^n B_2(S_i)W_i.
\end{aligned}
$$

Web Table 1

[Table 1 about here.]

R Code

Function for data generation

```
dataSimulation<-function(nsamp=500,b0=1,b1=2, sigma.Delta=0.25,
        sigma.G=0.2,sigma.Y=0.5,range.par.Y=0.1)
{
# nsamp= Sample size for data generation.
# range.par.Y= Value of the range parameter for outcome variable.
# sigma.Delta= Measurement error variance.
# sigma.G=Variance for smooth spatial surface.
# sigma.Y=Residual error variance.
# b0=Intercept of the regression model.
# b1= Slope parameter for regression model.
# Load necessary R library
        require(nlme)
# generate grid of nsamp points uniformly on square
        s1 <- runif(nsamp)
        s2 <- runif(nsamp)
        spatDat <- data.frame(cbind(s1,s2))
#Covariance structure for Y following exponential correlation
        cs1Exp.Y <- corExp(range.par.Y, form = ~ s1 + s2)
        cs1Exp.Y <- Initialize(cs1Exp.Y, spatDat)
        R.Y.matrix <- corMatrix(cs1Exp.Y)
# Get true inverse covariance matrices, square roots, eigenvalues,
# used to generate data and also to compute bias correction factors
        eigen.decomp.corr.Y <- eigen(R.Y.matrix)
        gamma.Y.mat <- cbind(eigen.decomp.corr.Y$vectors)
        lambda.Y.vec <- eigen.decomp.corr.Y$val
        decomp.Y <- gamma.Y.mat%*%diag(sqrt(lambda.Y.vec))%*%
                        t(gamma.Y.mat)
# generate true covariate X is is generated using a bivariate bump function
        X1 <- 1/(1+s1) + 3*exp(-50*(s1-.3)^2) + 2*exp(-25*(s1-.7)^2)
        X2 <- 1/(1+s2) + 3*exp(-50*(s2-.3)^2) + 2*exp(-25*(s2-.7)^2)
        X <-X1*X2
# generate observed covariate measured with error
# Z ~ N (X, variance = sigma.Delta)
        Z <- X + sqrt(sigma.Delta)*rnorm(nsamp)
# Generate smooth spatial surface
        G1<-sqrt(sigma.G)*decomp.Y%*%rnorm(nsamp)
# generate outcome Y | X ~ N (b0 + b1 X+G1(si)+variance)
        Y <- b0 + b1*X+G1+sqrt(sigma.Y)*rnorm(nsamp)
        data<-as.data.frame(cbind(s1=s1,s2=s2,X=X,Z=Z,Y=as.vector(Y)))
        data
}
```

Necessary functions for implementing the proposed method

```
### Function for selecting knots in two dimensional space
defaultKnots2D <- function(x1,x2,num.knots)
{
        require("cluster")
        # Set default value for num.knots
        if (missing(num.knots))
        num.knots <- max(10,min(50,round(length(x1)/4)))
        # Delete repeated values from x
        X <- cbind(x1,x2)
        dup.inds <- (1:nrow(X))[dup.matrix(X)==T]
        if (length(dup.inds) > 0)
        X <- X[-dup.inds,]
        # Obtain and output knots chosen using
        # coverage design principles
        knots <- clara(X,num.knots)$medoids
        return(knots)
}
### Function for penalty matrices using thin plate basis
# Set up thin plate spline generalised covariance function:
tps.cov <- function(r,m=2,d=1)
{
        r <- as.matrix(r)
        num.row <- nrow(r)
        num.col <- ncol(r)
        r <- as.vector(r)
        nzi <- (1:length(r))[r!=0]
        ans <- rep(0,length(r))
        if ((d+1)%%2!=0)
        ans[nzi] <- (abs(r[nzi]))^(2*m-d)*log(abs(r[nzi])) # d is even
        else
        ans[nzi] <- (abs(r[nzi]))^(2*m-d)
        if (num.col>1) ans <- matrix(ans,num.row,num.col)     # d is odd
        return(ans)
}
# Set up function for matrix square-roots:
matrix.sqrt <- function(A)
{
        sva <- svd(A)
        if (min(sva$d)>=0)
        Asqrt <- t(sva$v %*% (t(sva$u) * sqrt(sva$d)))
        else
        stop("Matrix square root is not defined")
        return(Asqrt)
}
Ztps <- function(x,knots)
{
        # Obtain  matrix of inter-knot distances:
        numKnots <- nrow(knots)
        dist.mat <- matrix(0,numKnots,numKnots)
        dist.mat[lower.tri(dist.mat)] <- dist(as.matrix(knots))
        dist.mat <- dist.mat + t(dist.mat)
        Omega <- tps.cov(dist.mat,d=2)
        # Obtain preliminary Z matrix of knot to data covariances:
        x.knot.diffs.1 <- outer(x[,1],knots[,1],"-")
        x.knot.diffs.2 <- outer(x[,2],knots[,2],"-")
        x.knot.dists <- sqrt(x.knot.diffs.1^2+x.knot.diffs.2^2)
        prelim.Z <- tps.cov(x.knot.dists,m=2,d=2)
        # Transform to canonical form:
        sqrt.Omega <- matrix.sqrt(Omega)
        Z <- t(solve(sqrt.Omega,t(prelim.Z)))
        output<-list(basis=prelim.Z, penalty=Omega ,Z=Z)
```

```
        return(output)
}
#Function for generalized cross validation for delta2
gcv.delta2<-function(nsamp,Z,B2,BTB.2, D2, delta.2)
{
        Lambda.2.inv<-1/nsamp*BTB.2+delta.2*D2
        eigen.decomp.Lambda.2 <- eigen(Lambda.2.inv)
        eigen.Lamdba.2.mat <- cbind(eigen.decomp.Lambda.2$vectors)
        eigen.Lambda.2.vec <- eigen.decomp.Lambda.2$val
        Lambda.2<-eigen.Lamdba.2.mat%*%diag(1/eigen.Lambda.2.vec)%*%
                                t(eigen.Lamdba.2.mat)
        theta.2.Hat<-1/nsamp*Lambda.2%*%(t(B2)%*%Z)
        RSS.delta2<-t(Z-B2%*%theta.2.Hat)%*%(Z-B2%*%theta.2.Hat)
        smooth.delta2<-1/nsamp*B2%*%Lambda.2%*%t(B2)
        gcv.delta2<-1/nsamp*
            as.vector(RSS.delta2)/(1-1/nsamp*sum(diag(smooth.delta2)))^2
        gcv.delta2
}
#Function for generalized cross validation for delta1
gcv.delta1<-function(nsamp,Y,Z,B1,B2,BTB.1,BTB.2,D1, D2, delta.2,delta.1)
{
        Lambda.2.inv<-1/nsamp*BTB.2+delta.2*D2
        eigen.decomp.Lambda.2 <- eigen(Lambda.2.inv)
        eigen.Lamdba.2.mat <- cbind(eigen.decomp.Lambda.2$vectors)
        eigen.Lambda.2.vec <- eigen.decomp.Lambda.2$val
        Lambda.2<-eigen.Lamdba.2.mat%*%diag(1/eigen.Lambda.2.vec)%*%
                        t(eigen.Lamdba.2.mat)
        theta.2.Hat<-1/nsamp*Lambda.2%*%(t(B2)%*%Z)
        RSS.delta2<-t(Z-B2%*%theta.2.Hat)%*%(Z-B2%*%theta.2.Hat)
        smooth.delta2<-1/nsamp*B2%*%Lambda.2%*%t(B2)
        gcv.delta2<-1/nsamp*
            as.vector(RSS.delta2)/(1-1/nsamp*sum(diag(smooth.delta2)))^2
        Lambda.1.inv<-1/nsamp*BTB.1+delta.1*D1
        eigen.decomp.Lambda.1 <- eigen(Lambda.1.inv)
        eigen.Lamdba.1.mat <- cbind(eigen.decomp.Lambda.1$vectors)
        eigen.Lambda.1.vec <- eigen.decomp.Lambda.1$val
        Lambda.1<-eigen.Lamdba.1.mat%*%diag(1/eigen.Lambda.1.vec)%*%
                        t(eigen.Lamdba.1.mat)
        ### Generating the required components for equation (13)
        Vn<-1/nsamp*Lambda.1%*%(t(B1)%*%Y)
        Rn<-1/nsamp*Lambda.1%*%crossprod(B1,B2)
        Tn<-1/nsamp*BTB.2
        Cn<-(Tn-t(Rn)%*%Lambda.1.inv%*%Rn)
        Dn<-(B2-B1%*%Rn)%*%theta.2.Hat
        numerator<-1/nsamp*t(Y)%*%Dn
        denominator<-t(theta.2.Hat)%*%Cn%*%theta.2.Hat
        # Estimates of the regression parameter
        b1.hat.basis<-numerator/denominator
        theta.1.Hat<-Vn-Rn%*%theta.2.Hat*as.vector(b1.hat.basis)
        RSS.Y<-
          t(Y-B2%*%theta.2.Hat*as.vector(b1.hat.basis)-B1%*%theta.1.Hat)%*%
          (Y-B2%*%theta.2.Hat*as.vector(b1.hat.basis)-B1%*%theta.1.Hat)
        smooth.mat<-1/(nsamp*as.vector(denominator))*Dn%*%t(Dn)+
                1/nsamp*(B1%*%Lambda.1%*%t(B1))
        gcv.delta1<-as.vector(RSS.Y)/(1-1/nsamp*sum(diag(smooth.mat)))^2
        gcv<-gcv.delta1
        return(gcv)
}
```

Codes for Data analysis

```
# Set the number of knots
        q1=125
        q2=150
# Get the data
        set.seed(123456)
        data<-dataSimulation()
# Extract the coordinates (s1,s2), covariate Z and outcome Y
        s1<-data$s1
        s2<-data$s2
        Z<-data$Z
        Y<-data$Y
        nsamp<-nrow(data)
# Load relevant R library
        require(cluster)
        require(SemiPar)
# Section of knot locations
        Knots1<-defaultKnots2D(s1,s2,q1)
        Knots2<-defaultKnots2D(s1,s2,q2)
# Generating thin plate spline basis B1(.) & B2(.) with q1 and q2 knots
# respectively
        B1<-cbind(rep(1,nsamp),s1,s2,Ztps(cbind(s1,s2),Knots1)$basis)
        B2<-cbind(rep(1,nsamp),s1,s2,Ztps(cbind(s1,s2),Knots2)$basis)
### Generating penulty matrices
        k1<-nrow(Knots1)+3
        k2<-nrow(Knots2)+3
        D1<-matrix(0,k1,k1)
        D2<-matrix(0,k2,k2)
        D1[4:k1,4:k1]<-Ztps(cbind(s1,s2),Knots1)$penalty
        D2[4:k2,4:k2]<-Ztps(cbind(s1,s2),Knots2)$penalty
### Generating Lambda1 and Lamda2
        BTB.1 <- crossprod(B1,B1)
        BTB.2 <- crossprod(B2,B2)
# Estimating delta.2
        delta.range<-seq(-11, 11, length.out =60)
        V<-rep(0,60)
        for (i in 1:60){
        V[i]<-gcv.delta2(nsamp,Z,B2,BTB.2, D2,exp(delta.range[i]))}
        index.delta2<-(1:60)[V==min(V)]
        delta.2<-10^delta.range[index.delta2]
# Estimating delta.1
        U<-rep(0,60)
        for (i in 1:60){
        U[i]<-gcv.delta1(nsamp,Y,Z,B1,B2,BTB.1,BTB.2,D1,D2,delta.2,
                                exp(delta.range[i]))}
        index.delta1<-(1:60)[U==min(U)]
        delta.1<-10^delta.range[index.delta1]
#Get lambda.1 using eigen value decomposition
        Lambda.1.inv<-1/nsamp*BTB.1+delta.1*D1
        eigen.decomp.Lambda.1 <- eigen(Lambda.1.inv)
        eigen.Lamdba.1.mat <- cbind(eigen.decomp.Lambda.1$vectors)
        eigen.Lambda.1.vec <- eigen.decomp.Lambda.1$val
        Lambda.1<-eigen.Lamdba.1.mat%*%diag(1/eigen.Lambda.1.vec)%*%
                t(eigen.Lamdba.1.mat)
#Get lambda.2 using eigen value decomposition
        Lambda.2.inv<-1/nsamp*BTB.2+delta.2*D2
        eigen.decomp.Lambda.2 <- eigen(Lambda.2.inv)
        eigen.Lamdba.2.mat <- cbind(eigen.decomp.Lambda.2$vectors)
        eigen.Lambda.2.vec <- eigen.decomp.Lambda.2$val
        Lambda.2<-eigen.Lamdba.2.mat%*%diag(1/eigen.Lambda.2.vec)%*%
```

```
                t(eigen.Lamdba.2.mat)
### Generating the required components for equation (13)
        theta.2.Hat<-1/nsamp*Lambda.2%*%(t(B2)%*%Z)
        Vn<-1/nsamp*Lambda.1%*%(t(B1)%*%Y)
        Rn<-1/nsamp*Lambda.1%*%crossprod(B1,B2)
        Tn<-1/nsamp*BTB.2
        Cn<-(Tn-t(Rn)%*%Lambda.1.inv%*%Rn)
        Dn<-(B2-B1%*%Rn)%*%theta.2.Hat
        numerator<-1/nsamp*t(Y)%*%Dn
        denominator<-t(theta.2.Hat)%*%Cn%*%theta.2.Hat
# Estimates of the regression parameter
        b1.hat.basis<-numerator/denominator
# Estimates of the empirical variance for the beta estimates
        theta.1.Hat<-Vn-Rn%*%theta.2.Hat*as.vector(b1.hat.basis)
        RSS.Y<-
          t(Y-B2%*%theta.2.Hat*as.vector(b1.hat.basis)-B1%*%theta.1.Hat)%*%
           (Y-B2%*%theta.2.Hat*as.vector(b1.hat.basis)-B1%*%theta.1.Hat)
        smooth.mat<-1/nsamp*(1/as.vector(denominator)*Dn%*%t(Dn)+
                B1%*%Lambda.1%*%(t(B1)))
        sigma.error.est<-as.vector(RSS.Y)/(nsamp-2*sum(diag(smooth.mat))+
                sum(diag(crossprod(smooth.mat,smooth.mat))))
        RSS.Z<-t(Z-B2%*%theta.2.Hat)%*%(Z-B2%*%theta.2.Hat)
        smooth.delta2<-1/nsamp*B2%*%Lambda.2%*%t(B2)
        sigma.u.est<-as.vector(RSS.Z)/(nsamp-2*sum(diag(smooth.delta2))+
                sum(diag(smooth.delta2%*%t(smooth.delta2))))

        Gn<-Dn/(as.vector(denominator))
        Fn<-2*B2%*%Lambda.2%*%Cn%*%theta.2.Hat
        An<-(1/nsamp)*t(B2%*%theta.2.Hat*as.vector(b1.hat.basis)+
                B1%*%theta.1.Hat)%*%(B2-B1%*%Rn)
        Hn<-t(An%*%Lambda.2%*%t(B2)/as.vector(denominator))-
                as.vector(An%*%theta.2.Hat)*Fn/(as.vector(denominator))^2
        sigma.beta.est<-1/(nsamp^2)*(sigma.error.est*t(Gn)%*%Gn+
                sigma.u.est*t(Hn)%*%Hn)
#Codes for the simulated standard error
        nboot<-100
        sim.b1.hat<-NULL
        for (i in 1:nboot)
        {
        sim.sigma.Y<-rnorm(nsamp,mean=0,sd=sqrt(sigma.error.est))
        sim.sigma.Delta<-rnorm(nsamp,mean=0,sd=sqrt(sigma.u.est))
        sim.numerator<-An%*%theta.2.Hat+
        1/nsamp*(An%*%Lambda.2%*%t(B2)%*%sim.sigma.Delta+
                t(Dn)%*%sim.sigma.Y)
        sim.denominator<-t(theta.2.Hat)%*%Cn%*%theta.2.Hat+
        1/nsamp*(t(Fn)%*%sim.sigma.Delta)
        sim.new.b1.hat<-sim.numerator/sim.denominator
        sim.b1.hat<-rbind(sim.b1.hat, sim.new.b1.hat)
        }
#Estimated regression coefficient
        b1.hat<-b1.hat.basis
        b1.hat
#Estimation of standard error
        se.b1.hat<-sqrt(as.vector(sigma.beta.est))
        se.b1.hat
#Estimation of simulated standard error
        sim.se.b1.hat<-sqrt(apply(sim.b1.hat,2,var))
        sim.se.b1.hat
```

Table 1

Simulation results with varying number of knots for covariate model (q_2) for our proposed method. In each case, the number of knots for the residual model (q_1) were fixed at 125. Reported numbers are averaged over 1000 simulations. Data were simulated with sample size 500, regression coefficient 2, measurement error variance 0.25 and varying range parameters for spatial correlations.

Range*	Number of knots for covariate model		
(τ_{G_1})	130	140	170
	Estimated coefficient		
0.1	2.016	2.013	2.007
0.3	2.019	2.016	2.010
0.5	2.025	2.023	2.018
	Empirical standard error		
0.1	0.060	0.061	0.068
0.3	0.058	0.059	0.066
0.5	0.052	0.053	0.060
	Average of estimated standard errors		
0.1	0.048	0.048	0.049
0.3	0.047	0.047	0.047
0.5	0.045	0.045	0.046
	Average of simulated standard errors		
0.1	0.048	0.048	0.049
0.3	0.047	0.047	0.047
0.5	0.045	0.045	0.046

τ_{G_1}: values of the range parameter following exponential correlation in $G_1(s_i)$.