OPTIMAL MODEL AVERAGING OF VARYING COEFFICIENT MODELS

CONG LI

School of Economics, Shanghai University of Finance and Economics, 777 Guoding Road, Shanghai, 200433, PR China Key Laboratory of Mathematical Economics (SUFE), Ministry of Education, Shanghai, 200433, PR China.

QI LI

ISEM, Capital University of Economics and Business, Beijing, 100070, PR China; Department of Economics, Texas A&M University, College Station, TX 77843, USA.

JEFFREY S. RACINE

Department of Economics and Graduate Program in Statistics, McMaster University, racinej@mcmaster.ca; Department of Economics and Finance, La Trobe University; Info-Metrics Institute, American University; Rimini Center for Economic Analysis; Center for Research in Econometric Analysis of TimE Series (CREATES), Aarhus University.

DAIQIANG ZHANG

Department of Economics, Texas A&M University, College Station, TX 77843, USA.

Abstract. We consider the problem of model averaging over a set of semiparametric varying coefficient models where the varying coefficients can be functions of continuous and categorical variables. We propose a Mallows model averaging procedure that is capable of delivering model averaging estimators with solid finite-sample performance. Theoretical underpinnings are provided, finite-sample performance is assessed via Monte Carlo simulation, and an illustrative application is presented. The approach is very simple to implement in practice and R code is provided in an appendix.

Date: January 20, 2017.

1991 *Mathematics Subject Classification.* C14 Semiparametric and Nonparametric Methods.

Key words and phrases. Kernel Smoothing.

We are indebted to Peter Hall for his deep and broad contributions to the statistics community. His contributions to semi- and nonparametric estimation have had a profound impact on the field, and we dedicate this paper to his memory. Racine would like to gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC:www.nserc.ca), the Social Sciences and Humanities Research Council of Canada (SSHRC:www.sshrc.ca), and the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca). Qi Li's research is partially supported by China National Science Foundation, projects #71601130 and #71133001.

1. Introduction

Practitioners who wish to tackle *model uncertainty* have a variety of approaches at their disposal. The most promising involve *model selection* and *model averaging*. *Model selection* proceeds from the premise that all models are, at best, approximations and involves selecting one model from among a set of candidate models. It is understood that, in practice, it is unlikely that the *true* model is among the set of candidate models, hence the model selected is the least misspecified among the set of models considered, in some known statistical sense. In essence, the practitioner who adopts model selection applies weight 1 to one candidate model and weight 0 to all others using a *selection criterion*. Model selection has a long history, and a variety of methods have been proposed, each based on distinct estimation criteria. These include Akaike's *An Information Criterion* (AIC; Akaike (1970), Akaike (1973)), Mallows' C_p (Mallows (1973)), the *Bayesian Information Criterion* (BIC; Schwarz (1978)), *delete-one cross-validation* (Stone (1974)), *generalized cross-validation* (Craven & Wahba (1979)), and the *Focused Information Criterion* (FIC) (Claeskens & Hjort (2003)), to name but a few.

Model averaging, on the other hand, produces a model that is a *weighted average* defined over a set of candidate models for which the weights are chosen by a statistical procedure having known properties (i.e., an *averaging criterion*). There is a longstanding literature on Bayesian model averaging; see Hoeting, Madigan, Raftery & Volinsky (1999) for a comprehensive review. There is also a rapidly-growing literature on frequentist methods for model averaging, including Buckland, Burnhamn & Augustin (1997), Hansen (2007), Wan, Zhang & Zou (2010), Hansen & Racine (2012), Zhang, Zou & Carroll (2015) and Zhang, Yu, Zou & Liang (2016), among others.

Practitioners who adopt the model averaging approach typically construct a weighted average defined over a set of *parametric* candidates. An alternative approach, one that we consider here, is to instead construct a weighted average defined over a set of more flexible *semiparametric* candidates. From a practical perspective, one might hope that by using more flexible estimators for the set of candidate models then perhaps fewer candidate models might be needed, or that perhaps the approximation capabilities of the resulting model might be improved. Though one might be tempted to perhaps average over fully nonparametric models, such models suffer from the so-called *curse of dimensionality* and are restricted to only a few predictors at most. Semiparametric models

strike a balance between flexibility and efficiency thereby attenuating the curse of dimensionality. Furthermore, being semiparametric in nature, one can easily incorporate prior parametric information if it exists. Our approach involves averaging over the so-called varying coefficient specification; see Beran & Hall (1992), Hastie & Tibshirani (1993), Cai, Fan & Yao (2000), Li, Huang, Li & Fu (2002) and the references therein. The varying coefficient specification is particularly appealing in this context in part because a range of models turn out to be special cases including fully parametric models and Robinson's (1988) partially linear model, by way of illustration. Our approach adopts Mallows' C_p criterion (Mallows 1973) for selecting the averaging weights, and allows for the coefficients to be functions of either continuous data types, categorical data types, or a mix of both.

Our theoretical results (based on the Mallows criterion) apply both to nested and non-nested regression models, and allow for heterogeneous errors. Hansen (2014) examines the asymptotic risk of nested least-squares averaging estimators based on minimizing a generalized Mallows criterion in a linear model with heteroskedasticity. Liu, Okui & Yoshimura (2016) adopt the Mallows criterion to choose the weight vector in the model averaging estimator for linear regression models with heteroskedastic errors. By averaging over semiparametric specifications we generalize existing approaches and provide practitioners with a straightforward and powerful approach to handling model uncertainty.

The rest of this paper proceeds as follows. Section 2 presents the varying coefficient specification defined over mixed datatypes, Mallows-driven weight choice, and asymptotic optimality of the proposed approach. Section 3 examines the finite-sample performance of the proposed approach relative to alternative model averaging estimators and model selection estimators, while Section 4 considers an illustrative example and a comparison of hold-out data performance of a range of averaging and selection criteria. Proofs of the main theorems are provided in Appendix A while R code can be found in Appendix B. Section 5 presents some brief concluding remarks.

2. Model Averaging Estimation

2.1. **Model and estimators.** We consider a varying coefficient model

$$Y_i = \mu_i + \epsilon_i = X_i'\beta(Z_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

where $X_i = (X_{i1}, \ldots, X_{ip})'$ is a $p \times 1$ vector, $Z_i = (Z_{i1}, \ldots, Z_{iq})'$ is a $q \times 1$ vector, $\beta(Z_i) = (\beta_1(Z_i), \ldots, \beta_p(Z_i))'$ is a $p \times 1$ unknown vector function, $\mu_i = X_i'\beta(Z_i)$, and $\epsilon_1, \ldots, \epsilon_n$ are independent and possibly heteroscedastic random errors with $E(\epsilon_i|X_i, Z_i) = 0$ and $E(\epsilon_i^2|X_i, Z_i) = \sigma_i^2$.

Our goal is to estimate μ_i for the purposes of prediction which is the focus of the literature on model averaging estimation; see Hansen (2007) and Lu & Su (2015) by way of illustration. To this end, we use S_n candidate varying coefficient models to approximate (1), where the number of models, S_n, is allowed to diverge to infinity as $n \to \infty$. The s_{th} candidate model is

$$Y_i = \mu_i + \epsilon_i = X_{i,(s)}'\beta_{(s)}(Z_{i,(s)}) + \epsilon_i, \quad i = 1, \ldots, n \tag{2}$$

where $X_{i,(s)}'$ is a p_s-dimensional subset of X_i, $Z_{i,(s)}$ is a q_s-dimensional subset of Z_i, and $\beta_{(s)}(Z_{i,(s)})$ is the corresponding $p_s \times 1$ unknown function.

To provide an optimal weighting scheme, we first need to estimate each candidate model. Premultiplying (1) by $X_{i,(s)}$ and taking $E(\cdot|Z_{i,(s)} = z_{(s)})$ leads to $E[X_{i,(s)}Y_i|Z_{i,(s)} = z_{(s)}] = E[X_{i,(s)}X_{i,(s)}']\beta_{(s)}(z_{(s)})$, yielding

$$\beta_{(s)}(z_{(s)}) = [E(X_{i,(s)}X_{i,(s)}'|z_{(s)})]^{-1}E[X_{i,(s)}Y_i|z_{(s)}]. \tag{3}$$

Let $k_{(s)}\left(\frac{Z_{j,(s)}-z_{(s)}}{h_{(s)}}\right) = k_{(s),1}\left(\frac{Z_{j,(s),1}-z_{(s),1}}{h_{(s),1}}\right) \times \ldots \times k_{(s),q_s}\left(\frac{Z_{j,(s),q_s}-z_{(s),q_s}}{h_{(s),q_s}}\right)$ denote a *product kernel function*, where $k_{(s),r}\left(\frac{Z_{j,(s),r}-z_{(s),r}}{h_{(s),r}}\right)$ is a univariate kernel function and h_r is a bandwidth for $r = 1, \ldots, q_s$. When the data consist of a mix of categorical and continuous datatypes, we use this product kernel device matching the appropriate data type with its kernel; see Hall, Racine & Li (2004) for details, and also Hall, Li & Racine (2007) and Hall & Racine (2015) for related extensions. Then (3) suggests the following local constant least square estimator,

$$\widehat{\beta}_{(s)}(z_{(s)}) = \left[\sum_{j=1}^{n} X_{j,(s)}X_{j,(s)}'k_{(s)}\left(\frac{Z_{j,(s)} - z_{(s)}}{h_{(s)}}\right)\right]^{-1} \sum_{j=1}^{n} X_{j,(s)}Y_j k_{(s)}\left(\frac{Z_{j,(s)} - z_{(s)}}{h_{(s)}}\right). \tag{4}$$

Letting $X_{(s)} = (X_{1,(s)}, \ldots, X_{n,(s)})'$, $Z_{(s)} = (Z_{1,(s)}, \ldots, Z_{n,(s)})'$, $Y = (Y_1, \ldots, Y_n)'$, and $K_{[z_{(s)}]}$ be a diagonal matrix of dimension n with the jth diagonal element being $k_{(s)}\left(\frac{Z_{j,(s)}-z_{(s)}}{h_{(s)}}\right)$, we can rewrite (4) as

$$\widehat{\beta}_{(s)}(z_{(s)}) = \left(X_{(s)}'K_{[z_{(s)}]}X_{(s)}\right)^{-1} X_{(s)}'K_{[z_{(s)}]}Y. \tag{5}$$

Then, we can estimate $\mu_{i,(s)}$ by

$$\widehat{\mu}_{i,(s)} = X'_{i,(s)}\widehat{\beta}_{(s)}(Z_{i,(s)}) = X'_{i,(s)}\left(X'_{(s)}K_{[Z_{i,(s)}]}X_{(s)}\right)^{-1}X'_{(s)}K_{[Z_{i,(s)}]}Y, \tag{6}$$

and rewrite it in matrix notation as $\widehat{\mu}_{(s)} = P_{(s)}Y$, where $P_{(s)}$ is a square matrix of dimension n with the ith row being $X'_{i,(s)}\left(X'_{(s)}K_{[Z_{i,(s)}]}X_{(s)}\right)^{-1}X'_{(s)}K_{[Z_{i,(s)}]}$, and $\widehat{\mu}_{(s)} = (\widehat{\mu}_{1,(s)}, \ldots, \widehat{\mu}_{n,(s)})'$.

Let the weight vector $w = (w_1, \ldots, w_{S_n})^T$ belong to the set $\mathcal{W} = \{w \in [0,1]^{S_n} : 0 \leq w_s \leq 1, s = 1, \ldots, S_n\}$, and let $P(w) = \sum_{s=1}^{S_n} w_s P_{(s)}$. Then, the model averaging estimator of μ is specified as

$$\widehat{\mu}(w) = \sum_{s=1}^{S_n} w_s \widehat{\mu}_{(s)} = P(w)Y. \tag{7}$$

Usually, the weight vector is restricted so that $\sum_{s=1}^{S_n} w_s = 1$. If all candidate models are equally competitive, this restriction is plausible in terms of allowing the data to determine the relative contribution of each candidate model to the final weighted model. On the other hand, if there is no prior information such that the candidate models are equally competitive, relaxing the aggregate weight restriction is likely to lower the prediction error (Ando & Li (2014)). Therefore, for the purpose of allowing for more general cases, this paper removes this restriction.

2.2. **Weight Choice Criterion and Asymptotic Optimality.** Up to now, the weight vector in $\widehat{\mu}(w)$ is unspecified. Motivated by the Mallows criterion for model averaging estimators (e.g. Hansen (2007)), we will now outline how we choose this weight vector. Let $\Omega = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. Define the predictive squared loss by

$$L_n(w) = n^{-1}\|\widehat{\mu}(w) - \mu\|^2, \tag{8}$$

and the conditional expected loss by

$$R_n(w) = E[L_n(w)|X, Z] = n^{-1}\|P(w)\mu - \mu\|^2 + n^{-1}\,\text{trace}[\Omega P(w)'P(w)]. \tag{9}$$

Let the Mallows-type criterion function be

$$C_n(w) = n^{-1}\|P(w)\mu - Y\|^2 + 2n^{-1}\,\text{trace}[P(w)\Omega]. \tag{10}$$

It is easy to show that

$$R_n(w) = E[C_n(w)|X, Z] - n^{-1}\,\mathrm{trace}(\Omega),$$

which suggests that for the optimal choice of w in the sense of minimizing $R_n(w)$, we can minimize $C_n(w)$ to choose w by noting the fact that $n^{-1}\,\mathrm{trace}(\Omega)$ does not depend on w.

Assuming that Ω is known, the optimal weight choice is given by

$$\widehat{w} = \arg\min_{w\in\mathcal{W}} C_n(w), \tag{11}$$

which implies that the optimal model averaging estimator of μ is $\widehat{\mu}(\widehat{w}) = P(\widehat{w})Y$, and we refer to $\widehat{\mu}(\widehat{w})$ as a *Mallows model average of varying coefficient models*. In order to provide regularity conditions for the optimal choice of the weight vector, we need to introduce some notation. Let $\xi_n = \inf_{w\in\mathcal{W}} nR_n(w)$, and let w_s^o be an $S_n \times 1$ vector in which the sth element is one and all others are zeros. We now list the conditions required for the asymptotic optimality of $\widehat{w}$ defined in (11). For some integer $N \geq 1$,

$$\max_i E(\epsilon_i^{4N}|X_i, Z_i) < \infty, \tag{12}$$

$$S_n^{4N+1}\xi_n^{-2N}\sum_{s=1}^{S_n}[nR_n(w_s^o)]^N \to 0, \tag{13}$$

$$\sup_{s\in\{1,\ldots,S_n\}}\max_i\sum_{j=1}^{n}|P_{(s),ij}| = O(1) \text{ and } \sup_{s\in\{1,\ldots,S_n\}}\max_j\sum_{i=1}^{n}|P_{(s),ij}| = O(1). \tag{14}$$

The first two conditions are commonplace in the literature on model averaging estimation (e.g., Hansen (2007); Hansen & Racine (2012); Wan et al. (2010); Ando & Li (2014)). Condition (12) imposes a finite moment bound and is satisfied by Gaussian noise. Condition (13) requires $\xi_n \to \infty$, implying that there is no finite approximating model whose bias is zero. Moreover, this condition also constrains the rates at which S_n and $nR_n(w_s^o)$ approach ∞. Condition (14) is similar to Assumption (i) of Speckman (1988) that bounds the kernel function in partially linear models.

Theorem 2.1. *Under conditions (12)-(14),*

$$\frac{L_n(\widehat{w})}{\inf_{w\in\mathcal{W}} L_n(w)} \to 1$$

in probability as $n \to \infty$.

Theorem 2.1 shows that the practitioner may do as well asymptotically as if they knew the true μ_i. That is, the weight vector $\widehat{w}$ is asymptotically optimal in the sense that the average loss with $\widehat{w}$ is asymptotically equivalent to that using the infeasible optimal weight vector.

By now we have assumed that Ω is known. In practice, however, Ω will be unknown. To make the Mallows-type criterion (10) computationally feasible, we estimate the unknown Ω based on residuals from model averaging estimation by

$$\widehat{\Omega}(w) = \text{diag}(\hat{\epsilon}_1^2(w), \ldots, \hat{\epsilon}_n^2(w)), \tag{15}$$

where $\hat{\epsilon}_i(w) = y_i - \hat{\mu}_i(w)$.[1] Replacing Ω with $\widehat{\Omega}$ in $C_n(w)$, we obtain the feasible criterion

$$\widehat{C}_n(w) = n^{-1}\|P(w)\mu - Y\|^2 + 2n^{-1}\,\text{trace}[P(w)\widehat{\Omega}(w)]. \tag{16}$$

Correspondingly, the new optimal weights are defined as

$$\widetilde{w} = \arg\min w \in \mathcal{W}\widehat{C}_n(w). \tag{17}$$

We now show that the weight vector $\widetilde{w}$ is still asymptotically optimal. Let $\rho_{ii}^{(s)}$ be the i^{th} diagonal element of $P_{(s)}$, and let $\overline{p} = \max_{1\leq s\leq S_n} p_s$. The conditions required for the asymptotic optimality of $\widetilde{w}$ are as follows.

$$\text{There exists a constant c such that } |\rho_{ii}^{(s)}| \leq cn^{-1}|\,\text{trace}(P_{(s)})|,\ \forall s = 1, \ldots, S_n, \tag{18}$$

$$n^{-1}\overline{p}^2 = O(1). \tag{19}$$

Condition (18) is commonly used to ensure the asymptotic optimality of cross-validation (e.g., Andrews (1991) and Hansen & Racine (2012)). Condition (19), which is the same as Condition (12) of Wan et al. (2010), allows the p_s's to increase as $n \to \infty$, but restricts their rate of increase.

[1] An alternative strategy for estimating Ω can be based on the largest model that includes all covariates (e.g., see the estimation of the variance of homoscedastic error terms in Hansen (2007)). The current approach is motivated by avoiding placing excessive confidence on a single model. More details regarding this alternative estimation strategy and its proofs are available upon request.

Theorem 2.2. *Under conditions (12)-(19),*

$$\frac{L_n(\widetilde{w})}{\inf_{w\in\mathcal{W}} L_n(w)} \to 1 \tag{20}$$

in probability as $n \to \infty$.

It is easy to prove that theorems 2.1 and 2.2 apply to the mixed data setting in which $Z = (Z_c, Z_d)$ with Z_c being a continuous vector and Z_d being a discrete vector, because our proofs are valid as long as the model averaging estimator is linear in Y when Z consists of multivariate mixed discrete and continuous covariates, which remains the case.

3. Monte Carlo Simulations

In this section we investigate the finite-sample performance of the proposed Mallows model averaging ('MMA') method. We consider simulating data from an infinite-order varying coefficient regression model of the form $y_i = \sum_{j=1}^{\infty} \theta_j(z_i)x_{ij} + \epsilon_i$, $i = 1, \ldots, n$. The x_{ij} are independent and identically distributed $N(0, 1)$ random variates, while z_i is distributed $U[-1, 1]$. The error ϵ_i is distributed $N(0, \sigma^2)$ and is independent of the x_{ij}.

The parameters are determined by the rule $\theta_j(z_i) = \sqrt{2\alpha}j^{-\alpha-1/2}\exp(z_i)$. The sample size is varied from $n = 50, 100, 200$, and 400. The parameter α is varied from 0.10, 0.25, and 0.50. Larger values of α imply that the coefficients $\theta_j(z)$ decline more quickly with j. The number of models M_n is determined by the rule $M_n = 3n^{1/3}$ (so $M_n = 11, 14, 18$, and 22 for the four sample sizes considered herein). We rescale the DGP to have unit variance and set σ equal to $0.25, 0.50, 1.00$ and 2.00 so that the expected R^2 for the unknown true model is given by $1/(1+\sigma^2)$ and is $0.95, 0.80, 0.50$, and 0.20, respectively.

The simulations use nested regression models with variables $\{x_{ij}, j = 1, \ldots, M_n\}$. We consider six estimators: (1) Mallows model averaging ('MMA'), (2) smoothed AIC model averaging ('SAIC'), (3) smoothed BIC model averaging ('SBIC'), (4) AIC model selection ('AIC'), (5) BIC model selection ('BIC'), and (6) Mallows' C_p model selection. To evaluate the estimators, we compute the risk (expected squared error). We do this by computing means (medians) across 1,000 simulation draws.

The SAIC and SBIC weights for the $j = 1, 2, \ldots, M$ models are given by

$$w_j = \exp(-AIC_j/2) / \sum_{j=1}^{M} \exp(-AIC_j/2),$$

$$w_j = \exp(-BIC_j/2) / \sum_{j=1}^{M} \exp(-BIC_j/2)$$

where AIC_j and BIC_j are given by $\log(\hat{\sigma}_j^2) + 2n^{-1}\,\mathrm{trace}(\mathbf{P}_{(j)})$ and $\log(\hat{\sigma}_j^2) + n^{-1}\,\mathrm{trace}(\mathbf{P}_{(j)})\log(n)$, respectively. The C_p criterion is given by $\hat{\sigma}^2(n + 2\,\mathrm{trace}(\mathbf{P}_{(j)}))$ where $\hat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}\hat{\epsilon}_i^2$.

Let $H = (\hat{\mu}_{(1)} - y, \ldots, \hat{\mu}_{(M_n)} - y)$ and let $b = \{\mathrm{trace}(P_{(1)}\hat{\Omega}_{(M_n)}), \ldots, \mathrm{trace}(P_{(M_n)}\hat{\Omega}_{(M_n)})\}^T$, where $\hat{\Omega}_{(M_n)}$ is a diagonal matrix formed from the squared residuals from the model indexed by the largest j (i.e. M_n). Note that we can rewrite $\hat{C}_n(w)$ as $\hat{C}_n(w) = w^T H^T H w + 2w^T b$, which is a quadratic function of the weight vector w and the optimization can be done by standard software packages such as the `R` package `quadprog` (code underlying this simulation can be found in Appendix B). Note that using the largest model to estimate the error covariance matrix is advocated by Hansen (2007) and Liu & Okui (2013), and in small samples this approach performs admirably.

Simulation results are summarized in Table 1, which reports the mean relative MSE row normalized so that the method with lowest mean MSE has entry 1.00. R^2 is higher for smaller values of σ; for larger values of α the $\theta_j(z)$ coefficients decay more rapidly with j. MMA, SAIC, and SBIC are model averaging methods; AIC, BIC and C_p are model selection methods.

3.1. **Discussion.** Clearly no one method dominates over the range of sample sizes, signal to noise ratio, and range of parameter decay considered above. AIC and C_p have similar risk. However, from a minimax perspective, the proposed method is competitive among its peers. In particular, if one considers the range of risk relative to the best performing method is any experiment (row of Table 1), it would appear that the proposed approach dominates its peers. The proposed method ought to appeal to practitioners interested in model average estimators defined over the flexible and popular varying coefficient specification.

TABLE 1. Mean Relative MSE (row normalized so that the method with lowest mean MSE has entry 1.00). R^2 is higher for smaller values of σ; for larger values of α the $\theta_j(z)$ coefficients decay more rapidly with j. MMA, SAIC, and SBIC are model averaging methods; AIC, BIC and C_p are model selection methods.

n	α	σ	MMA	SAIC	SBIC	AIC	BIC	C_p
50	0.10	0.25	1.00	1.31	1.36	1.00	1.62	1.02
50	0.10	0.50	1.00	1.17	1.20	1.05	1.60	1.07
50	0.10	1.00	1.04	1.00	1.01	1.20	1.38	1.20
50	0.10	2.00	1.21	1.02	1.00	1.52	1.11	1.47
50	0.25	0.25	1.00	1.37	1.45	1.01	1.52	1.03
50	0.25	0.50	1.00	1.13	1.16	1.07	1.53	1.08
50	0.25	1.00	1.11	1.00	1.00	1.31	1.44	1.30
50	0.25	2.00	1.33	1.03	1.00	1.64	1.15	1.58
50	0.50	0.25	1.00	1.25	1.33	1.07	1.37	1.08
50	0.50	0.50	1.04	1.00	1.02	1.18	1.43	1.18
50	0.50	1.00	1.24	1.01	1.00	1.51	1.43	1.49
50	0.50	2.00	1.35	1.04	1.00	1.76	1.10	1.68
100	0.10	0.25	1.00	1.26	1.29	1.00	1.62	1.01
100	0.10	0.50	1.00	1.16	1.19	1.03	1.61	1.04
100	0.10	1.00	1.00	1.01	1.02	1.11	1.43	1.12
100	0.10	2.00	1.10	1.01	1.00	1.36	1.19	1.35
100	0.25	0.25	1.00	1.34	1.41	1.01	1.61	1.02
100	0.25	0.50	1.00	1.16	1.19	1.06	1.60	1.07
100	0.25	1.00	1.04	1.00	1.01	1.20	1.50	1.20
100	0.25	2.00	1.15	1.01	1.00	1.46	1.24	1.44
100	0.50	0.25	1.00	1.27	1.35	1.07	1.62	1.07
100	0.50	0.50	1.00	1.00	1.03	1.14	1.49	1.14
100	0.50	1.00	1.15	1.01	1.00	1.40	1.52	1.39
100	0.50	2.00	1.18	1.02	1.00	1.53	1.17	1.50
200	0.10	0.25	1.00	1.23	1.26	1.00	1.47	1.00
200	0.10	0.50	1.00	1.16	1.18	1.02	1.51	1.02
200	0.10	1.00	1.00	1.04	1.04	1.08	1.48	1.08
200	0.10	2.00	1.06	1.00	1.00	1.27	1.29	1.26
200	0.25	0.25	1.00	1.32	1.37	1.01	1.55	1.01
200	0.25	0.50	1.00	1.17	1.20	1.04	1.59	1.05
200	0.25	1.00	1.00	1.00	1.01	1.13	1.56	1.14
200	0.25	2.00	1.09	1.01	1.00	1.35	1.39	1.35
200	0.50	0.25	1.00	1.27	1.35	1.06	1.70	1.07
200	0.50	0.50	1.00	1.03	1.06	1.14	1.64	1.14
200	0.50	1.00	1.10	1.00	1.00	1.33	1.61	1.33
200	0.50	2.00	1.09	1.02	1.00	1.39	1.32	1.37
400	0.10	0.25	1.00	1.22	1.24	1.00	1.33	1.00
400	0.10	0.50	1.00	1.16	1.18	1.00	1.41	1.00
400	0.10	1.00	1.00	1.06	1.07	1.05	1.46	1.05
400	0.10	2.00	1.04	1.00	1.00	1.19	1.39	1.19
400	0.25	0.25	1.00	1.31	1.35	1.00	1.37	1.00
400	0.25	0.50	1.00	1.19	1.22	1.02	1.53	1.02
400	0.25	1.00	1.00	1.03	1.04	1.10	1.55	1.10
400	0.25	2.00	1.07	1.00	1.00	1.27	1.53	1.27
400	0.50	0.25	1.00	1.30	1.38	1.04	1.67	1.04
400	0.50	0.50	1.00	1.07	1.10	1.12	1.67	1.12
400	0.50	1.00	1.07	1.00	1.00	1.28	1.67	1.28
400	0.50	2.00	1.06	1.01	1.00	1.29	1.48	1.29

4. Empirical Illustration

In what follows we estimate a Mincer (earnings) equation using Wooldridge's (2002) 'wage1' data which contains $n = 526$ observations on a range of variables. We consider modeling expected (log) hourly wages ('*lwage*') based on a number of commonly employed predictors, namely

(1) *educ*: years of education
(2) *exper*: years potential experience
(3) *tenure*: years with current employer
(4) *female*: "Female" if female, "Male" otherwise
(5) *married*: "Married" if Married, "Nonmarried" otherwise

We treat the predictors *educ*, *exper* and *tenure* as belonging to X and *female* and *married* as belonging to Z. We consider varying coefficient models that differ in terms of the contents of X. Let d be the order of a (orthogonal) polynomial formed from each of *educ*, *exper* and *tenure*. When $d = 1$ there are 3 columns in X (*educ*, *exper* and *tenure*) and if we consider all possible combinations of the predictors taken 1, 2, and 3 at a time then there are $M = \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$ candidate models. When $d = 2$ there are 6 columns in X hence $M = 63$ candidate models, and when $d = 3$ there are 9 columns in X hence $M = 511$ candidate models. We also consider standard local constant ('LC'), local linear ('LL'), and varying coefficient ('VC') models defined over the full set of predictors by way of comparison.

We conduct a simulation in which the data is repeatedly shuffled and split into two parts 1000 times, based on an estimation sample of size $n_1 = 500$ and an independent validation sample of size $n_2 = 26$. For each estimation sample we fit the cross-validated semiparametric varying coefficient model and each of the parametric and nonparametric models listed above. For each model we then compute predicted square error ('PSE') for the independent validation data set given by PSE $= n_2^{-1} \sum_{i=1}^{n_2} (Y_i - \hat{Y}_i)^2$ where $\hat{Y}_i$ is the prediction for a given model. The mean relative hold-out PSE is presented in Table 2, row normalized so that the method with lowest mean MSE has entry 1.00, while the mean PSE is presented in Table 3.

Table 2 reveals some interesting features. First, note from row 1 (i.e., $d = 1$) that when we average across models in which the parametric component X is linear, then the fully nonparametric local linear estimator is the best performer dominating both model averaging and model selection, which for some might be unexpected. However, when we move to a larger number of candidate models allowing for quadratic ($d = 2$) and cubic ($d = 3$) terms to enter in the parametric component X, this appears to be sufficient for the model averaging estimator to dominate its peers. Furthermore, Table 3 reveals that there is no further MSE improvement in either the selection or averaging methods when we move from $d = 2$ to $d = 3$, hence a relatively modest number of candidate models appear to be sufficient for the proposed model averaging method to dominate its peers.

TABLE 2. Mean Relative PSE

		Model Average			Model Selection			Model Specification		
d	M	MMA	SAIC	SBIC	AIC	BIC	C_p	LC	LL	VC
1	7	1.043	1.080	1.081	1.041	1.051	1.041	1.041	1.000	1.040
2	63	1.000	1.056	1.057	1.008	1.054	1.008	1.082	1.039	1.089
3	511	1.000	1.061	1.062	1.029	1.056	1.029	1.075	1.039	1.093

TABLE 3. Mean PSE

		Model Average			Model Selection			Model Specification		
d	M	MMA	SAIC	SBIC	AIC	BIC	C_p	LC	LL	VC
1	7	0.167	0.173	0.173	0.167	0.169	0.167	0.167	0.160	0.167
2	63	0.151	0.160	0.160	0.153	0.159	0.153	0.164	0.157	0.165
3	511	0.152	0.161	0.161	0.156	0.160	0.156	0.163	0.158	0.166

5. Concluding Remarks

In this paper we present a semiparametric approach to model averaging that possesses a number of desirable features. Theoretical underpinnings are provided, and its finite-sample performance indicates that it ought to be of interest to practitioners who wish to tackle model uncertainty. An illustrative application indicates that the method is capable of delivering models with impressive approximation capabilities. In particular, it can be seen how averaging over a set of semiparametric models can outperform fully nonparametric specifications in applied settings, which ought to excite the practitioner. R code for implementing the proposed approach is presented in the appendix, and is available upon request from the authors.

Appendix A. Appendix

Proof of Theorem 2.1. The proof is similar to that of Theorem 1 of Wan et al. (2010). Let the largest singular values of a matrix A be $\lambda(A)$. By (12), we have

$$\lambda(\Omega) = O(1). \tag{A.1}$$

Under Condition (14), by an inequality of Reisz (e.g., Speckman (1988)), we obtain

$$\lambda[P_{(s)}P'_{(s)}] \leq \lambda^2[P_{(s)}] \leq \max_i \sum_{j=1}^{n} |P_{(s),ji}| \max_j \sum_{i=1}^{n} |P_{(s),ji}| = O(1). \tag{A.2}$$

Hence,

$$\lambda(P_{(s)}) = \lambda(P(w_s^o)) = O(1) \text{ for any } s \in \{1, \ldots, S_n\}. \tag{A.3}$$

Let $A(w) = I - P(w)$. Note that

$$C_n(w) = L_n(w) + n^{-1}\|\epsilon\|^2 + 2n^{-1}\langle \epsilon, A(w)\mu\rangle + 2n^{-1}\left\{\text{trace}[P(w)\Omega] - \langle \epsilon, P(w)\epsilon\rangle\right\}$$

Theorem (2.1) is valid if the following is true: As $n \to \infty$,

$$\sup_{w\in\mathcal{W}} |\langle \epsilon, A(w)\mu\rangle|/[nR_n(w)] \xrightarrow{p} 0, \tag{A.4}$$

$$\sup_{w\in\mathcal{W}} |\,\text{trace}[P(w)\Omega] - \langle \epsilon, P(w)\epsilon\rangle|/[nR_n(w)] \xrightarrow{p} 0, \tag{A.5}$$

$$\sup w \in \mathcal{W}|L_n(w)/R_n(w) - 1| \xrightarrow{p} 0, \tag{A.6}$$

First, we consider (A.4). $\forall \delta > 0$. By triangle inequality, Chebyshev's inequality, Theorem 2 of Whittle (1960), (A.1), and (13), we obtain

$$
\begin{aligned}
&Pr\left\{\sup_{w\in\mathcal{W}} |\langle \epsilon, A(w)\mu\rangle|/[nR_n(w)] > \delta\right\} \\
&\quad \le Pr\left\{\sup_{w\in\mathcal{W}} \sum_{s=1}^{S_n} w_s |\epsilon'(I - P_{(s)})\mu| > \delta\xi_n\right\} \\
&\quad \le Pr\left\{S_n \max_{1\le s\le S_n} |\epsilon'(I - P_{(s)})\mu| > \delta\xi_n\right\} \\
&\quad = Pr\left\{\left\{|\langle \epsilon, A(w_1^o)\mu\rangle| > \delta\xi_n S_n^{-1}\right\} \bigcup \left\{|\langle \epsilon, A(w_2^o)\mu\rangle| > \delta\xi_n S_n^{-1}\right\} \bigcup \ldots \bigcup \left\{|\langle \epsilon, A(w_{S_n}^o)\mu\rangle| > \delta\xi_n S_n^{-1}\right\}\right\} \\
&\quad \le \sum_{s=1}^{S_n} Pr\left\{|\langle \epsilon, A(w_s^o)\mu\rangle| > \delta\xi_n S_n^{-1}\right\} \text{ by triangle inequality} \\
&\quad \le S_n^{2N} \sum_{s=1}^{S_n} E\left\{\frac{\langle \epsilon, A(w_s^o)\mu\rangle^{2N}}{\delta^{2N}\xi_n^{2N}}\right\} \text{ by Chebyshev's inequality} \\
&\quad \le C_1 \delta^{-2N} \xi_n^{-2N} S_n^{2N} \sum_{s=1}^{S_n} \|\Omega^{1/2} A(w_s^o)\mu\|^{2N} \text{ by (7) in Theorem 2 of Whittle (1960)} \\
&\quad \le C_1 \delta^{-2N} \xi_n^{-2N} S_n^{2N} \lambda(\Omega)^N \sum_{s=1}^{S_n} \|A(w_s^o)\mu\|^{2N} \\
&\quad \le C_1 \delta^{-2N} \xi_n^{-2N} S_n^{2N} \lambda(\Omega)^N \sum_{s=1}^{S_n} [nR_n(w_s^o)]^N \to 0, \quad \text{as n} \to \infty \text{ by (A.1) and (13)}
\end{aligned}
$$

where C_1 is a constant, the second to last inequality follows the result that $\mu' A\mu \le \lambda(A)\mu'\mu$ and $\lambda(AA) = \lambda(A)^2$ for any symmetric square matrix A, and the last inequality follows $nR_n(w_s^o) \ge \|A(w_s^o)\mu\|^2$ which is implied by (9).

Similarly for (A.5), we have

$$
\begin{aligned}
&Pr\left\{\sup_{w\in\mathcal{W}}\left|\text{trace}[P(w)\Omega]-\langle\epsilon,P(w)\epsilon\rangle\right|/[nR_n(w)]>\delta\right\}\\
&\quad=Pr\left\{\sup_{w\in\mathcal{W}}\left|\sum_{s=1}^{S_n}w_s[\text{trace}(P_{(s)}\Omega)-\langle\epsilon,P_{(s)}\epsilon\rangle]\right|/[nR_n(w)]>\delta\right\}\\
&\quad\le Pr\left\{\max_{1\le s\le S_n}\left|\text{trace}(P_{(s)}\Omega)-\langle\epsilon,P_{(s)}\epsilon\rangle\right|/[nR_n(w)]>\delta S_n^{-1}\right\}\\
&\quad\le\sum_{s=1}^{S_n}Pr\left\{\left|\text{trace}[P(w_s^o)\Omega]-\langle\epsilon,P(w_s^o)\epsilon\rangle\right|>\delta\xi_nS_n^{-1}\right\}\\
&\quad\le S_n^{2N}\sum_{s=1}^{S_n}E\left\{\frac{[\text{trace}[P(w_s^o)\Omega]-\langle\epsilon,P(w_s^o)\epsilon\rangle]^{2N}}{\delta^{2N}\xi_n^{2N}}\right\}\\
&\quad\le C_2\delta^{-2N}\xi_n^{-2N}S_n^{2N}\lambda(\Omega)^N\sum_{s=1}^{S_n}\left\{\text{trace}\left[\Omega P(w_s^o)'P(w_s^o)\right]\right\}^N\text{ by (8) in Theorem 2 of Whittle (1960)}\\
&\quad\le C_2'\delta^{-2N}\xi_n^{-2N}S_n^{2N}\lambda(\Omega)^N\sum_{s=1}^{S_n}[nR_n(w_s^o)]^N\to 0,\quad\text{as n}\to\infty.
\end{aligned}
\tag{A.7}
$$

where C_2 and C_2' are constants, the last inequality follows $nR_n(w_s^o)\ge\text{trace}\left[\Omega P(w_s^o)'P(w_s^o)\right]$ implied by (9).

Note that (A.6) is equivalent to

$$
\sup_{w\in\mathcal{W}}\left|\frac{n^{-1}\|P(w)\epsilon\|^2-n^{-1}\,\text{trace}[\Omega P(w)'P(w)]-2n^{-1}\langle A(w)\mu,P(w)\epsilon\rangle}{R_n(w)}\right|\xrightarrow{p}0.
$$

Thus, (A.6) holds if as $n\to\infty$, we have

$$
\sup_{w\in\mathcal{W}}\left|\frac{\langle A(w)\mu,P(w)\epsilon\rangle}{nR_n(w)}\right|\xrightarrow{p}0,\tag{A.8}
$$

and

$$
\sup_{w\in\mathcal{W}}\left|\frac{\|P(w)\epsilon\|^2-\text{trace}[\Omega P(w)'P(w)]}{nR_n(w)}\right|\xrightarrow{p}0.\tag{A.9}
$$

By (A.3), we have

$$
\begin{aligned}
&Pr\left\{\sup_{w\in\mathcal{W}}\left|\frac{\langle A(w)\mu, P(w)\epsilon\rangle}{nR_n(w)}\right| > \delta\right\} \\
&\le Pr\left\{\sup_{w\in\mathcal{W}}\sum_{m=1}^{S_n}\sum_{s=1}^{S_n} w_t w_s \left|\epsilon' P_{(s)}(I-P_{(m)})\mu\right| > \delta\xi_n\right\} \\
&\le Pr\left\{S_n^2 \max_{1\le m\le S_n}\max_{1\le s\le S_n}\left|\epsilon' P_{(s)}(I-P_{(m)})\mu\right| > \delta\xi_n\right\} \\
&\le S_n^2\sum_{t=1}^{S_n}\sum_{s=1}^{S_n} E\left[\frac{\langle P(w_t^o)\epsilon, A(w_s^o)\mu\rangle^{2N}}{\delta^{2N}\xi_n^{2N}}\right] \\
&\le C_3 S_n^{4N}\delta^{-2N}\xi_n^{-2N}\sum_{m=1}^{S_n}\sum_{s=1}^{S_n}\left\|P(w_m^o)\Omega^{1/2}A(w_s^o)\mu)\right\|^{2N} \\
&\le C_3 S_n^{4N}\lambda[\Omega^{1/2}P(w_m^o)'P(w_m^o)\Omega^{1/2}]^N\delta^{-2N}\xi_n^{-2N}\sum_{m=1}^{S_n}\sum_{s=1}^{S_n}\|A(w_s^o)\mu)\|^{2N} \\
&\le C_3 S_n^{4N+1}\lambda(\Omega)^N\lambda[P(w_m^o)]^{2N}\delta^{-2N}\xi_n^{-2N}\sum_{s=1}^{S_n}[nR_n(w_s^o)]^N \to 0, \quad \text{as n}\to\infty,
\end{aligned}
$$

where C_3 is a constant, and

$$
\begin{aligned}
&Pr\left\{\sup_{w\in\mathcal{W}}\left|\frac{\|P(w)\epsilon\|^2 - \text{trace}[\Omega P(w)'P(w)]}{nR_n(w)}\right| > \delta\right\} \\
&\le Pr\left\{\sup_{w\in\mathcal{W}}\sum_{t=1}^{S_n}\sum_{s=1}^{S_n} w_t w_s|\epsilon' P'_{(t)}P_{(s)}\epsilon - \text{trace}[\Omega P'_{(s)}P_{(t)}]| > \delta\xi_n\right\} \\
&\le Pr\left\{S_n^2\max_{1\le t\le S_n}\max_{1\le s\le S_n}|\epsilon' P'_{(t)}P_{(s)}\epsilon - \text{trace}[\Omega P'_{(s)}P_{(t)}]| > \delta\xi_n\right\} \\
&\le S_n^{4N}\sum_{t=1}^{S_n}\sum_{s=1}^{S_n} E\left\{\frac{[\langle\Omega^{-1/2}\epsilon, \Omega^{1/2}P(w_t^o)'P(w_s^o)\Omega^{1/2}\Omega^{-1/2}\epsilon\rangle - \text{trace}(\Omega P(w_t^o)'P(w_s^o))]^{2N}}{\delta^{2N}\xi_n^{2N}}\right\} \\
&\le C_4 S_n^{4N}\delta^{-2N}\xi_n^{-2N}\sum_{t=1}^{S_n}\sum_{s=1}^{S_n}\text{trace}(\Omega P(w_t^o)'P(w_s^o)P(w_s^o)'P(w_t^o)\Omega)^N \\
&= C_4 S_n^{4N}\delta^{-2N}\xi_n^{-2N}\sum_{t=1}^{S_n}\sum_{s=1}^{S_n}\text{trace}(\Omega P(w_t^o)'P(w_t^o)P(w_t^o)^{-1}P(w_s^o)P(w_s^o)'P(w_t^o)\Omega)^N \\
&\le C_4' S_n^{4N+1}\lambda(\Omega)^N\lambda[P(w_s^o)]^{2N}\delta^{-2N}\xi_n^{-2N}\sum_{t=1}^{S_n}[nR_n(w_t^o)]^N \to 0, \quad \text{as n}\to\infty,
\end{aligned}
$$

where C_4 and C_4' are constants. Thus we obtain (A.8) and (A.9). □

Proof of Theorem 2.2. Obviously,

$$\widehat{C}_n(w) = C_n(w) + 2n^{-1}\,\mathrm{trace}[P(w)\widehat{\Omega}(w)] - 2n^{-1}\,\mathrm{trace}[P(w)\Omega].$$

Therefore, (20) holds if and only if

$$\sup_{w\in\mathcal{W}} |\,\mathrm{trace}[P(w)\widehat{\Omega}(w)] - \mathrm{trace}[P(w)\Omega]|/[nR_n(w)] = o_p(1). \tag{A.10}$$

Let $H_{(s)} = \mathrm{diag}(\rho_{11}^{(s)}, \ldots, \rho_{nn}^{(s)})$ and $H(w) = \sum_{s=1}^{S_n} w_s H_{(s)}$. Then we obtain that

$$\begin{aligned}
&\sup_{w\in\mathcal{W}} |\,\mathrm{trace}[P(w)\widehat{\Omega}(w)] - \mathrm{trace}[P(w)\Omega|/[nR_n(w)] \\
&= \sup_{w\in\mathcal{W}} |[y - P(w)y]'H(w)[y - P(w)y] - \mathrm{trace}[H(w)\Omega]|/[nR_n(w)] \\
&= \sup_{w\in\mathcal{W}} |[\epsilon + \mu - P(w)y]'H(w)[\epsilon + \mu - P(w)y] - \mathrm{trace}[H(w)\Omega]|/[nR_n(w)] \\
&\le \sup_{w\in\mathcal{W}} \frac{|\epsilon'H(w)\epsilon - \mathrm{trace}[H(w)\Omega]|}{[nR_n(w)]} + 2\sup_{w\in\mathcal{W}} \frac{|\epsilon'H(w)[P(w)y - \mu]|}{[nR_n(w)]} \\
&+ \sup_{w\in\mathcal{W}} \frac{|[P(w)y - \mu]'H(w)[P(w)y - \mu]|}{[nR_n(w)]} \\
&\le \sup_{w\in\mathcal{W}} \frac{|\epsilon'H(w)\epsilon - \mathrm{trace}[H(w)\Omega]|}{[nR_n(w)]} \\
&+ 2\sup_{w\in\mathcal{W}} \frac{|\epsilon'H(w)[P(w)\mu - \mu]|}{[nR_n(w)]} \\
&+ 2\sup_{w\in\mathcal{W}} \frac{|\epsilon'H(w)P(w)\epsilon - \mathrm{trace}[H(w)P(w)\Omega]|}{[nR_n(w)]} \\
&+ 2\sup_{w\in\mathcal{W}} \frac{|\,\mathrm{trace}[H(w)P(w)\Omega]|}{[nR_n(w)]} \\
&+ \sup_{w\in\mathcal{W}} \frac{|[P(w)y - \mu]'H(w)[P(w)y - \mu]|}{[nR_n(w)]} \\
&\equiv D_1 + D_2 + D_3 + D_4 + D_5,
\end{aligned}$$

where the definitions of D_i ($i \in 1, \ldots, 5$) should be apparent. Define $\rho = \max_{1 \le s \le S_n} \max_{1 \le i \le n} |\rho_{ii}^{(s)}|$. Then

$$\begin{aligned} \rho &\le cn^{-1} \max_s |\operatorname{trace}[P_{(s)}]| \\ &\le cn^{-1} \max_s p_m |\lambda[P_{(s)}]| \\ &\le cn^{-1} \bar{p} \max_s |\lambda[P_{(s)}]| \\ &= O(n^{-1}\bar{p}), \end{aligned} \tag{A.11}$$

where the first inequality follows (18), and the last equality follows $\max_s |\lambda(P_{(s)})| = O(1)$ implied by (A.3). Note that (13) implies

$$\xi_n \to \infty, \text{ and } S_n^{2N+1} \xi_n^{-2N} = o(1). \tag{A.12}$$

Then, for any $\delta > 0$, we obtain

$$\begin{aligned} Pr(D_1 > \delta) &\le Pr\left\{ \sup_{w \in \mathcal{W}} \sum_{s=1}^{S_n} |\epsilon' H_{(s)} \epsilon - \operatorname{trace}(H_{(s)}\Omega)| > \delta \xi_n \right\} \\ &\le Pr\left\{ S_n \max_{1 \le s \le S_n} |\epsilon' H_{(s)} \epsilon - \operatorname{trace}(H_{(s)}\Omega)| > \delta \xi_n \right\} \\ &\le \sum_{s=1}^{S_n} Pr\left\{ |\epsilon' H_{(s)} \epsilon - \operatorname{trace}(H_{(s)}\Omega)| > \delta \xi_n S_n^{-1} \right\} \\ &\le \delta^{-2N} \xi_n^{-2N} S_n^{2N} \sum_{s=1}^{S_n} E[\epsilon' H_{(s)} \epsilon - \operatorname{trace}(H_{(s)}\Omega)]^{2N} \\ &\le C_5 \delta^{-2N} \xi_n^{-2N} S_n^{2N} \sum_{s=1}^{S_n} \operatorname{trace}[\Omega^{1/2} H_{(s)} \Omega H_{(s)} \Omega^{1/2}]^N \\ &\le C_5 \delta^{-2N} \xi_n^{-2N} \lambda^{2N}(\Omega) n^N \rho^{2N} S_n^{2N+1} \\ &= \xi_n^{-2N} S_n^{2N+1} O[(n^{-1}\bar{p}^2)^N] = o(1)O(1) = o(1), \end{aligned} \tag{A.13}$$

where C_5 is a positive constant, and the second to last inequality follows from (A.12) and (19).

$$
\begin{aligned}
Pr(D_3/2 > \delta) &\leq \sum_{s=1}^{S_n} Pr[|\epsilon' H_{(s)} P_{(s)} \epsilon - \text{trace}[H_{(s)} P_{(s)} \Omega]| > \delta \xi_n S_n^{-1}] \\
&\leq \delta^{-2N} \xi_n^{-2N} S_n^{2N} \sum_{s=1}^{S_n} E[\epsilon' H_{(s)} P_{(s)} \epsilon - \text{trace}(H_{(s)} P_{(s)} \Omega)]^{2N} \\
&\leq C_6 \delta^{-2N} \xi_n^{-2N} S_n^{2N} \sum_{s=1}^{S_n} \text{trace}[\Omega^{1/2} H_{(s)} P_{(s)} \Omega P'_{(s)} H_{(s)} \Omega^{1/2}]^N \\
&\leq C_6 \delta^{-2N} \xi_n^{-2N} \lambda^{2N}(\Omega) n^N \rho^{2N} S_n^{2N+1} \max_s [\lambda(P_{(s)} P'_{(s)})]^N \\
&= \xi_n^{-2N} S_n^{2N+1} O[(n^{-1} \bar{p}^2)^N] O(1) = o(1) O(1) O(1) = o(1),
\end{aligned}
\tag{A.14}
$$

where C_6 is a positive constant, and the second last equality follows (A.12) and (19), and the fact that $\max_s \lambda(P_{(s)} P'_{(s)}) \leq \max_s \lambda^2(P_{(s)}) \leq [\max_s \lambda(P_{(s)})]^2 = O(1)$ due to (A.3).

Similarly, we have

$$
\begin{aligned}
D_2 &= 2 \sup_{w \in \mathcal{W}} \frac{|\epsilon' H(w)[P(w)\mu - \mu]|}{[nR_n(w)]} \\
&\leq 2 \sup_{w \in \mathcal{W}} \|\epsilon\| \rho \| P(w)\mu - \mu \| / [nR_n(w)] \\
&\leq 2 \|\epsilon\| \rho \| \xi_n^{-1/2} = \xi_n^{-1/2} O_p(n^{-1/2} \bar{p}) = o(1) O_p(1) = o_p(1),
\end{aligned}
$$

and

$$
\begin{aligned}
D_4 &= 2 \sup_{w \in \mathcal{W}} \frac{|\,\text{trace}[H(w)P(w)\Omega]|}{[nR_n(w)]} \\
&\leq 2 \xi_n^{-1} \rho \lambda(\Omega) \sup_{w \in \mathcal{W}} [\text{trace}(P(w))] \\
&\leq 2 \xi_n^{-1/2} \rho \lambda(\Omega) S_n \max_s [\text{trace}(P_{(s)})] \\
&\leq 2 \xi_n^{-1} \rho \lambda(\Omega) S_n \max_s [\lambda(P_{(s)})] \max_s [\text{rank}(P_{(s)})] = \xi_n^{-1} S_n O(n^{-1} \bar{p}^2) = o_p(1) o_p(1) = o_p(1),
\end{aligned}
$$

where the second last equality follows the fact that $\xi_n^{-1} S_n = o_p(1)$ implied by (A.12).

$$
\begin{aligned}
D_5 &= \sup_{w\in\mathcal{W}} \frac{|[P(w)y-\mu]'H(w)[P(w)y-\mu]|}{[nR_n(w)]} \\
&\le \rho \sup_{w\in\mathcal{W}} \frac{|[P(w)y-\mu]'[P(w)y-\mu]|}{[nR_n(w)]} \\
&= \rho \sup_{w\in\mathcal{W}} \left\{ \frac{|[P(w)\mu-\mu]'[P(w)\mu-\mu]|}{[nR_n(w)]} + \frac{\epsilon'\epsilon}{[nR_n(w)]} + \frac{|2\epsilon'[P(w)\mu-\mu]|}{[nR_n(w)]} \right\} \\
&\le \rho \sup_{w\in\mathcal{W}} \frac{|[P(w)\mu-\mu]'[P(w)\mu-\mu]|}{[nR_n(w)]} + \rho \sup_{w\in\mathcal{W}} \frac{\epsilon'\epsilon}{[nR_n(w)]} + \rho \sup_{w\in\mathcal{W}} \frac{|2\epsilon'[P(w)\mu-\mu]|}{[nR_n(w)]} \\
&\le \rho + \rho \sup_{w\in\mathcal{W}} \frac{\epsilon'\epsilon}{[nR_n(w)]} + \rho o_p(1) \quad \text{due to (A.7)} \\
&\le O(n^{-1/2}) + O(n^{-1/2})\epsilon'\epsilon\xi_n^{-1} \quad \text{due to (9)} \\
&= o_p(1) + O(n^{-1/2})O_p(n^{1/2})o_p(1) = o_p(1),
\end{aligned}
$$

where in the last inequality $\rho = O(n^{-1/2})$ follows Condition (19) and (A.11). Combining the above results, we have that $D_1 + D_2 + D_3 + D_4 + D_5 = o_p(1)$. □

References

Akaike, H. (1970), 'Statistical predictor identification', *Annals of the Institute of Statistics and Mathematics* **22**, 203–217.

Akaike, H. (1973), Information theory and an extension of the maximum likelihood principle, *in* B. Petroc & F. Csake, eds, 'Second International Symposium on Information Theory', Akademiai Kiado, Budapest, pp. 267–281.

Ando, T. & Li, K.-C. (2014), 'A model-averaging approach for high-dimensional regression', *Journal of the American Statistical Association* **109**(505), 254–265.

Andrews, D. W. (1991), 'Asymptotic optimality of generalized C_L, cross-validation, and generalized cross-validation in regression with heteroskedastic errors', *Journal of Econometrics* **47**(2), 359–377.

Beran, R. & Hall, P. (1992), 'Estimating coefficient distributions in random coefficient regressions', *The Annals of Statistics* **20**(4), 1970–1984.

Buckland, S. T., Burnhamn, K. P. & Augustin, N. H. (1997), 'Model selection: An integral part of inference', *Biometrics* **53**, 603618.

Cai, Z., Fan, J. & Yao, Q. W. (2000), 'Functional-coefficient regression models for nonlinear time series', *Journal of the American Statistical Association* **95**, 941–956.

Claeskens, G. & Hjort, N. L. (2003), 'The focused information criterion', *Journal of the American Statistical Association* **98**(464), 900–916.

Craven, P. & Wahba, G. (1979), 'Smoothing noisy data with spline functions', *Numerische Mathematik* **13**, 377–403.

Hall, P. G. & Racine, J. S. (2015), 'Infinite order cross-validated local polynomial regression', *Journal of Econometrics* **185**(2), 510 – 525.

Hall, P., Li, Q. & Racine, J. S. (2007), 'Nonparametric estimation of regression functions in the presence of irrelevant regressors', *The Review of Economics and Statistics* **89**, 784–789.

Hall, P., Racine, J. S. & Li, Q. (2004), 'Cross-validation and the estimation of conditional probability densities', *Journal of the American Statistical Association* **99**(468), 1015–1026.

Hansen, B. E. (2007), 'Least squares model averaging', *Econometrica* **75**(4), 1175–1189.

Hansen, B. E. (2014), 'Model averaging, asymptotic risk, and regressor groups', *Quantitative Economics* **5**(3), 495–530.

Hansen, B. E. & Racine, J. S. (2012), 'Jackknife model averaging', *Journal of Econometrics* **167**(1), 38–46.

Hastie, T. & Tibshirani, R. (1993), 'Varying-coefficient models', *Journal of the Royal Statistical Society. Series B* **55**, 757–796.

Hoeting, J. A., Madigan, D., Raftery, A. E. & Volinsky, C. T. (1999), 'Bayesian model averaging: A tutorial', *Statistical Science* **14**, 382417.

Li, Q., Huang, C. J., Li, D. & Fu, T. T. (2002), 'Semiparametric smooth coefficient models', *Journal of Business and Economics Statistics* **20**, 412–422.

Liu, Q. & Okui, R. (2013), 'Heteroscedasticity-robust C_p model averaging', *The Econometrics Journal* **16**(3), 463–472.

Liu, Q., Okui, R. & Yoshimura, A. (2016), 'Generalized least squares model averaging', *Econometric Reviews* pp. 1–61.

Lu, X. & Su, L. (2015), 'Jackknife model averaging for quantile regressions', *Journal of Econometrics* **188**(1), 40–58.

Mallows, C. L. (1973), 'Some comments on C_p', *Technometrics* **15**, 661–675.

Robinson, P. M. (1988), 'Root-n consistent semiparametric regression', *Econometrica* **56**, 931–954.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The Annals of Statistics* **6**, 461–464.

Speckman, P. (1988), 'Kernel smoothing in partial linear models', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 413–436.

Stone, C. J. (1974), 'Cross-validatory choice and assessment of statistical predictions (with discussion)', *Journal of the Royal Statistical Society* **36**, 111–147.

Wan, A. T., Zhang, X. & Zou, G. (2010), 'Least squares model averaging by Mallows criterion', *Journal of Econometrics* **156**(2), 277–283.

Whittle, P. (1960), 'Bounds for the moments of linear and quadratic forms in independent variables', *Theory of Probability & Its Applications* **5**(3), 302–305.

Wooldridge, J. M. (2002), *Econometric Analysis of Cross Section and Panel Data*, MIT Press, Cambridge.

Zhang, X., Yu, D., Zou, G. & Liang, H. (2016), 'Optimal model averaging estimation for generalized linear models and generalized linear mixed-effects models', *Journal of American Statistical Association* . In Press.

Zhang, X., Zou, G. & Carroll, R. (2015), 'Model averaging based on Kullback-Leibler distance', *Statistic Sinica* **25**, 1583–1598.

Appendix B. R Code (NOT FOR PUBLICATION)

The R code to replicate the Monte Carlo simulations is provided below. It can be run in serial mode or in parallel on a laptop, desktop, or cluster environment.

```
## Monte Carlo simulation and functions for model averaging and model
## selection for the varying coefficient specification. This requires
## that the latest version of the R package 'np' from github
## (https://github.com/JeffreyRacine/R-Package-np) along with the
## quadprog, foreach, and doParallel packages from CRAN.

rm(list=ls())

## To process Monte Carlo simulations in parallel (e.g. using multiple
## cores), modify the integer in makeCluster() to the desired number
## of cores, then uncomment/comment the appropriate foreach() command
## at or around lines 133-136. Can also use makeCluster(detectCores())
## to automate the number of cores used.

library(foreach)
library(doParallel)
cl<-makeCluster(detectCores())
registerDoParallel(cl)

library(np)
library(quadprog)
options(np.messages=FALSE)
clusterExport(cl,"options")

## alpha determines the rate of coefficient decline, larger alpha
## implies coefficients decline more quickly with j

alpha <- scan("alpha.dat")

## sigma determines R^2=1/(1+sigma^2)
## R^2 = (0.941, 0.80, 0.50, 0.20) when (c = 0.25, 0.50, 1.0, 2.0)

sigma <- scan("sigma.dat")

## Generate X matrix with large number of N(0,1) columns for the DGP
## (infinite order regression with decaying weights)

num.cols <- 1000

## Sample size, number of Monte Carlo replications, and ability to
## restart if halted

n <- scan("num_obs.dat")
Monte <- scan("num_monte.dat")
system("if test -e mse.out;then wc -l mse.out | awk '{print $1}' > num_monte_exist.dat;else echo 0 > num_monte_exist.dat;fi")
M.exist <- scan("num_monte_exist.dat")

## Number of candidate models is a function of the sample size (Hansen
## (2007))

M <- round(3*n**(1/3))

## Create headers (can be restarted and, when so, the headers are not
## written nor is the seed set)

if(M.exist==0) {
    M.remain <- Monte
    system("rm *.out")
    write(c("JMA","MMA","SAIC","SBIC","AIC","BIC","CV","Cp"),file="mse.out",ncol=8)
    write(c("DGP","JMA","MMA","SAIC","SBIC","AIC","BIC","CV","Cp"),file="r_squared.out",ncol=9)
} else {
    Monte <- Monte+1-M.exist
}

## Functions for computing R^2 and MSE

r.sq <- function(yhat,ybar) {
    sum((yhat-ybar)^2)/sum((y-ybar)^2)
}
```

```
mse <- function(yhat,dgp) {
    mean((yhat-dgp)^2)
}

## Function for computing the hat matrix of the varying coefficient
## specification (the diagonal matrix with 1e-10 proceeds with the
## inversion without throwing an error if the singular case is
## encountered)

hat.mat.npscoef <- function(x,z,bw) {
    W <- cbind(1,x)
    IM <- diag(1e-10,ncol(W),ncol(W))
    K <- npksum(txdat=z,bws=bw,return.kernel.weights=TRUE)$kw
    P <- sapply(1:NROW(x),function(i){tWK <- t(W*K[,i]);W[i,,drop=FALSE]%*%chol2inv(chol(tWK%*%W+IM))%*%tWK})
}

## Function to compute the trace of the hat matrix times \hat\Omega

trace.hat.mat.Omega.npscoef <- function(x,z,bw,resid) {
    sum(diag(hat.mat.npscoef(x,z,bw)*resid^2))
}

## Function for Cp model selection criterion

Cp.npscoef <- function(n,trace.hat.mat,resid) {
    sum(resid^2)*(1 + 2*trace.hat.mat/(n-trace.hat.mat))
}

## Function for BIC model selection

BIC.npscoef <- function(n,trace.hat.mat,sigmasq.ml) {
    log(sigmasq.ml)+trace.hat.mat*log(n)/n
}

## Function for AIC model selection

AIC.npscoef <- function(n,trace.hat.mat,sigmasq.ml) {
    log(sigmasq.ml)+2*trace.hat.mat/n
}

## Storage matrices and vectors (loo = 'leave-one-out')

yhat.loo.mat <- matrix(nrow=n,ncol=M)
yhat.mat <- matrix(nrow=n,ncol=M)
residual.mat <- matrix(nrow=n,ncol=M)

trace.Omega.vec <- numeric(M)
mse.vec <- numeric(M)
aic.vec <- numeric(M)
bic.vec <- numeric(M)
cv.vec <- numeric(M)
cp.vec <- numeric(M)
bw.vec <- numeric(M)
r.sq.vec <- numeric(M)

## Monte Carlo begins

## NB - foreach() is _not_ a loop, and the index i cannot be used in
## return vectors though it is available inside the loop it
## appears... it simply chunks things up... so don't make the mistake
## of expecting it to work like for() or you will be sorely
## disappointed (it is list based, for like 'apply')

## Uncomment for serial processing
##if(Monte>0) foreach(i=1:Monte,.combine='c') %do% {
## Uncomment for parallel processing
if(Monte>0) foreach(i=1:Monte,.packages=c('np','quadprog'),.combine='c',.inorder=FALSE) %dopar% {

    set.seed(i+M.exist)

    z <- runif(n,min=-1,max=1)
    X <- matrix(rnorm(n*num.cols),nrow=n,ncol=num.cols)

    ## Generate parameter vector theta of length num.cols
```

```
theta <- sqrt(2*alpha)*seq(1,num.cols)**(-alpha-1/2)

## Generate the DGP, convert to unit variance, add error with
## variance sigma^2 (expected r-squared will therefore be
## 1/(1+sigma^2))

dgp <- as.numeric(X%*%theta)*exp(z)
dgp <- dgp/sd(dgp)

y <- dgp + rnorm(n,sd=sigma)

ybar <- mean(y)
dgp.r.sq <- r.sq(dgp,ybar)

for(j in M:1) {

    ## Compute cross-validated bandwidths, the model fit, residuals,
    ## and delete-one fits and residuals

    bws <- npscoefbw(ydat=y,zdat=z,xdat=X[,1:j])

    model <- npscoef(bws=bws,residuals=TRUE)

    ## Need residuals from the 'largest' model for computing
    ## \hat\Omega (the full matrix X)

    if(j==M) model.largest <- model
    model.loo <- npscoef(bws=bws,delete.one=TRUE,residuals=TRUE)

    yhat <- fitted(model)
    r.sq.vec[j] <- r.sq(yhat,ybar)
    mse.vec[j] <- mse(fitted(model),dgp)
    bw.vec[j] <- bws$bw[1]

    ## For JMA

    yhat.loo.mat[,j] <- fitted(model.loo)
    yhat.mat[,j] <- fitted(model)

    ## For MMA

    residual.mat[,j] <- residuals(model)
    trace.Omega.vec[j] <- trace.hat.mat.Omega.npscoef(X[,1:j],z,model$bws$bw,residuals(model.largest))

    ## For model selection

    trace.hat.mat <- sum(diag(hat.mat.npscoef(X[,1:j],z,model$bws$bw)))

    aic.vec[j] <- AIC.npscoef(n,trace.hat.mat,model$MSE)
    bic.vec[j] <- BIC.npscoef(n,trace.hat.mat,model$MSE)
    cv.vec[j] <-  mean(residuals(model.loo)^2)
    cp.vec[j] <- Cp.npscoef(n,trace.hat.mat,residuals(model))

}

## Grab the mse and r-squared from the AIC, BIC, CV and Cp optimal
## models

aic.mse <- mse.vec[which.min(aic.vec)]
bic.mse <- mse.vec[which.min(bic.vec)]
cv.mse <- mse.vec[which.min(cv.vec)]
cp.mse <- mse.vec[which.min(cp.vec)]

aic.r.sq <- r.sq.vec[which.min(aic.vec)]
bic.r.sq <- r.sq.vec[which.min(bic.vec)]
cv.r.sq <- r.sq.vec[which.min(cv.vec)]
cp.r.sq <- r.sq.vec[which.min(cp.vec)]

## Now compute the JMA-optimal model. Compute weights, impose
## restriction of summing to one and being non-negative

## The w'Dmat w matrix (M x M)

Dmat <- t(yhat.loo.mat)%*%yhat.loo.mat
if(qr(Dmat)$rank<M) Dmat <- Dmat + diag(1e-10,M,M)
```

```
    ## The -2 dvec'w vector (1 X M)

    dvec <- t(y)%*%yhat.loo.mat

    ## The constraint matrix.  Amat has row one the adding up
    ## constraint, the following num.model rows the non-negativity,
    ## finally the following num.model the less than one constraints.

    Amat <- t(rbind(rep(1,M),diag(x=1,M,M),diag(x=-1,M,M)))

    ## The constraint vector

    bvec <- c(1,rep(0,M),rep(-1,M))

    ## meq tells us to treat the first constraint as an equality
    ## constraint, the rest as inequality ones

    ## JMA weight vector, fitted model, MSE and r-squared

    w.hat.jma <- solve.QP(Dmat,dvec,Amat,bvec=bvec,meq=1)$solution

    yhat.jma <- yhat.mat%*%w.hat.jma

    jma.mse <- mse(yhat.jma,dgp)
    jma.r.sq <- r.sq(yhat.jma,ybar)

    ## Mallows model average (can reuse constraint matrix/vector)

    ## The w'Dmat w matrix (M x M)

    Dmat <- t(residual.mat)%*%residual.mat
    if(qr(Dmat)$rank<M) Dmat <- Dmat + diag(1e-10,M,M)

    ## The 2 dvec'w vector (1 x M) (opposite sign from JMA dvec)

    dvec <- -trace.Omega.vec

    ## MMA weight vector, fitted model, MSE and r-squared

    w.hat.mma <- solve.QP(Dmat,dvec,Amat,bvec=bvec,meq=1)$solution

    yhat.mma <- yhat.mat%*%w.hat.mma

    mma.mse <- mse(yhat.mma,dgp)
    mma.r.sq <- r.sq(yhat.mma,ybar)

    ## SAIC, SBIC weight vectors, fitted model, MSE and r-squared

    w.hat.aic <- exp(-aic.vec/2)/sum(exp(-aic.vec/2))
    w.hat.bic <- exp(-bic.vec/2)/sum(exp(-bic.vec/2))

    yhat.saic <- yhat.mat%*%w.hat.aic
    yhat.sbic <- yhat.mat%*%w.hat.bic

    saic.mse <- mse(yhat.saic,dgp)
    sbic.mse <- mse(yhat.sbic,dgp)

    saic.r.sq <- r.sq(yhat.saic,ybar)
    sbic.r.sq <- r.sq(yhat.sbic,ybar)

    ## Write results to files as the Monte Carlo progresses (can compute
    ## summaries before experiment is completed).

    write(c(jma.mse,mma.mse,saic.mse,sbic.mse,aic.mse,bic.mse,cv.mse,cp.mse),"mse.out",ncol=8,append=TRUE)
    write(c(dgp.r.sq,jma.r.sq,mma.r.sq,saic.r.sq,sbic.r.sq,aic.r.sq,bic.r.sq,cv.r.sq,cp.r.sq),"r_squared.out",ncol=9,append=TRUE)
    write(mse.vec,"mse_models.out",ncol=M,append=TRUE)
    write(w.hat.jma,"jma_weights.out",ncol=M,append=TRUE)
    write(w.hat.mma,"mma_weights.out",ncol=M,append=TRUE)
    write(w.hat.aic,"saic_weights.out",ncol=M,append=TRUE)
    write(w.hat.bic,"sbic_weights.out",ncol=M,append=TRUE)
    write(bw.vec,"bw.out",ncol=length(bw.vec),append=TRUE)

}

stopCluster(cl)
```

Appendix C. R Code for the Illustrative Application (NOT FOR PUBLICATION)

The R code to replicate the illustrative application is provided below. It can be run in serial mode or in parallel on a laptop, desktop, or cluster environment. When the number of candidate models is large (e.g. 500+) the benefits of running in parallel are obvious.

```
num.reps <- 1000

set.seed(42)

library(foreach)
library(doParallel)
cl<-makeCluster(detectCores())
registerDoParallel(cl)

library(np)
library(quadprog)
options(np.messages=FALSE)

## Functions for computing R^2 and MSE

r.sq <- function(yhat,ybar) {
    sum((yhat-ybar)^2)/sum((y-ybar)^2)
}

## Function for computing the hat matrix of the varying coefficient
## specification (the diagonal matrix with 1e-10 proceeds with the
## inversion without throwing an error if the singular case is
## encountered)

hat.mat.npscoef <- function(x,z,bw) {
    W <- cbind(1,x)
    IM <- diag(1e-10,ncol(W),ncol(W))
    K <- npksum(txdat=z,bws=bw,return.kernel.weights=TRUE)$kw
    P <- sapply(1:NROW(x),function(i){tWK <- t(W*K[,i]);W[i,,drop=FALSE]%*%chol2inv(chol(tWK%*%W+IM))%*%tWK})
}

## Function to compute the trace of the hat matrix times \hat\Omega

trace.hat.mat.Omega.npscoef <- function(x,z,bw,resid) {
    sum(diag(hat.mat.npscoef(x,z,bw)*resid^2))
}

## Function for Cp model selection criterion

Cp.npscoef <- function(n,trace.hat.mat,resid) {
    sum(resid^2)*(1 + 2*trace.hat.mat/(n-trace.hat.mat))
}

## Function for BIC model selection

BIC.npscoef <- function(n,trace.hat.mat,sigmasq.ml) {
    log(sigmasq.ml)+trace.hat.mat*log(n)/n
}

## Function for AIC model selection

AIC.npscoef <- function(n,trace.hat.mat,sigmasq.ml) {
    log(sigmasq.ml)+2*trace.hat.mat/n
}

## Data

data(wage1)
attach(wage1)
y <- lwage
z <- data.frame(factor(female),factor(married))
## X must be a matrix...
d <- scan("poly_order.dat")
X <- cbind(poly(educ,d),poly(exper,d),poly(tenure,d))
K <- ncol(X)
M <- 0
for(k in 1:K) M <- M + ncol(combn(K,k))
indices.mat <- matrix(0,K,M)
```

```
j <- 1
for(k in 1:K) {
  indices.mat[1:nrow(combn(K,k)),j:(j+ncol(combn(K,k))-1)] <- combn(K,k)
  j <- j + ncol(combn(K,k))
}

n <- nrow(X)

## foreach() returns a list (or list of lists), it is not a
## replacement for for(), rather a kindof 'apply' replacement albeit
## in parallel (thanks Revolution R programmers!)

n.train <- 500
n.eval <- n - n.train

## Write results to files
write(c("JMA","MMA","SAIC","SBIC","AIC","BIC","CV","Cp"),
      file="r_squared.out",ncol=8)
write(c("AIC","BIC","CV","Cp"),
      file="model_selection.out",ncol=4)
write(c("JMA","MMA","SAIC","SBIC","AIC","BIC","CV","Cp"),
      file="mse.out",ncol=8)

for(m in 1:num.reps) {

    ii <- sample(n)
    ii.train <- ii[1:n.train]
    ii.eval <- ii[(1+n.train):n]

    y.train <- y[ii.train]
    ybar.train <- mean(y.train)
    y.eval <- y[ii.eval]

    z.train <- z[ii.train,]
    z.eval <- z[ii.eval,]

    ## Overhead for computing the largest model (done in serial, might be
    ## folded into post processing but might have to pass back big hat
    ## matrices)

    bws <- npscoefbw(ydat=y.train,zdat=z.train,xdat=X[ii.train,indices.mat[,M]])
    model.largest <- npscoef(bws=bws,residuals=TRUE)

    output <- foreach(i=1:M,.packages=c('np','quadprog')) %dopar% {

        X.train <- X[ii.train,indices.mat[,i]]
        X.eval <- X[ii.eval,indices.mat[,i]]

        bws <- npscoefbw(ydat=y.train,zdat=z.train,xdat=X.train)
        model <- npscoef(bws=bws,residuals=TRUE)
        model.loo <- npscoef(bws=bws,delete.one=TRUE,residuals=TRUE)
        trace.hat.mat <- sum(diag(hat.mat.npscoef(X.train,z.train,model$bws$bw)))

        list(r.sq.train.vec=r.sq(fitted(model),ybar.train),
             yhat.train.loo.mat=fitted(model.loo),
             yhat.train.mat=fitted(model),
             residual.train.mat=residuals(model),
             trace.Omega.train.vec=trace.hat.mat.Omega.npscoef(X.train,z.train,model$bws$bw,residuals(model.largest)),
             aic.train.vec=AIC.npscoef(n.train,trace.hat.mat,model$MSE),
             bic.train.vec=BIC.npscoef(n.train,trace.hat.mat,model$MSE),
             cv.train.vec=mean(residuals(model.loo)^2),
             cp.train.vec=Cp.npscoef(n.train,trace.hat.mat,residuals(model)),
             yhat.eval.mat=fitted(npscoef(ydat=y.train,zdat=z.train,xdat=X.train,ezdat=z.eval,exdat=X.eval,bws=bws$bw)),
             residual.eval.mat=y.eval-fitted(npscoef(ydat=y.train,zdat=z.train,xdat=X.train,ezdat=z.eval,exdat=X.eval,bws=bws$bw)))

    }

    ## Storage matrices and vectors (loo = 'leave-one-out')

    yhat.train.loo.mat <- matrix(nrow=n.train,ncol=M)
    yhat.train.mat <- matrix(nrow=n.train,ncol=M)
    residual.train.mat <- matrix(nrow=n.train,ncol=M)
    residual.eval.mat <- matrix(nrow=n.eval,ncol=M)
    yhat.eval.mat <- matrix(nrow=n.eval,ncol=M)

    trace.Omega.train.vec <- numeric(M)
```

```
aic.train.vec <- numeric(M)
bic.train.vec <- numeric(M)
cv.train.vec <- numeric(M)
cp.train.vec <- numeric(M)
r.sq.train.vec <- numeric(M)

## Extract elements of list into storage matrices and vectors

for(i in 1:M) {
    yhat.train.loo.mat[,i] <- output[[i]]$yhat.train.loo.mat
    yhat.train.mat[,i] <- output[[i]]$yhat.train.mat
    residual.train.mat[,i] <- output[[i]]$residual.train.mat
    residual.eval.mat[,i] <- output[[i]]$residual.eval.mat
    yhat.eval.mat[,i] <- output[[i]]$yhat.eval.mat

    trace.Omega.train.vec[i] <- output[[i]]$trace.Omega.train.vec
    aic.train.vec[i] <- output[[i]]$aic.train.vec
    bic.train.vec[i] <- output[[i]]$bic.train.vec
    cv.train.vec[i] <- output[[i]]$cv.train.vec
    cp.train.vec[i] <- output[[i]]$cp.train.vec
    r.sq.train.vec[i] <- output[[i]]$r.sq.train.vec
}

## Grab the r-squared from the AIC, BIC, CV and Cp optimal
## models

aic.r.sq <- r.sq.train.vec[which.min(aic.train.vec)]
bic.r.sq <- r.sq.train.vec[which.min(bic.train.vec)]
cv.r.sq <- r.sq.train.vec[which.min(cv.train.vec)]
cp.r.sq <- r.sq.train.vec[which.min(cp.train.vec)]

aic.mse <- mean((yhat.eval.mat[,which.min(aic.train.vec)]-y.eval)^2)
bic.mse <- mean((yhat.eval.mat[,which.min(bic.train.vec)]-y.eval)^2)
cv.mse <- mean((yhat.eval.mat[,which.min(cv.train.vec)]-y.eval)^2)
cp.mse <- mean((yhat.eval.mat[,which.min(cp.train.vec)]-y.eval)^2)

## Now compute the JMA-optimal model. Compute weights, impose
## restriction of summing to one and being non-negative

## The w'Dmat w matrix (M x M)

Dmat <- t(yhat.train.loo.mat)%*%yhat.train.loo.mat
if(qr(Dmat)$rank<M) Dmat <- Dmat + diag(1e-10,M,M)

## The -2 dvec'w vector (1 X M)

dvec <- t(y.train)%*%yhat.train.loo.mat

## The constraint matrix.  Amat has row one the adding up
## constraint, the following num.model rows the non-negativity,
## finally the following num.model the less than one constraints.

Amat <- t(rbind(rep(1,M),diag(x=1,M,M),diag(x=-1,M,M)))

## The constraint vector

bvec <- c(1,rep(0,M),rep(-1,M))

## meq tells us to treat the first constraint as an equality
## constraint, the rest as inequality ones

## JMA weight vector, fitted model and r-squared

w.hat.jma <- solve.QP(Dmat,dvec,Amat,bvec=bvec,meq=1)$solution

yhat.train.jma <- yhat.train.mat%*%w.hat.jma
yhat.eval.jma <- yhat.eval.mat%*%w.hat.jma

jma.r.sq <- r.sq(yhat.train.jma,ybar.train)
jma.mse <- mean((yhat.eval.jma-y.eval)^2)

## Mallows model average (can reuse constraint matrix/vector)

## The w'Dmat w matrix (M x M)

Dmat <- t(residual.train.mat)%*%residual.train.mat
```

```
    if(qr(Dmat)$rank<M) Dmat <- Dmat + diag(1e-10,M,M)

    ## The 2 dvec'w vector (1 x M) (opposite sign from JMA dvec)

    dvec <- -trace.Omega.train.vec

    ## MMA weight vector, fitted model and r-squared

    w.hat.mma <- solve.QP(Dmat,dvec,Amat,bvec=bvec,meq=1)$solution

    yhat.train.mma <- yhat.train.mat%*%w.hat.mma
    yhat.eval.mma <- yhat.eval.mat%*%w.hat.mma

    mma.r.sq <- r.sq(yhat.train.mma,ybar.train)
    mma.mse <- mean((yhat.eval.mma-y.eval)^2)

    ## SAIC, SBIC weight vectors, fitted model and r-squared

    w.hat.aic <- exp(-aic.train.vec/2)/sum(exp(-aic.train.vec/2))
    w.hat.bic <- exp(-bic.train.vec/2)/sum(exp(-bic.train.vec/2))

    yhat.train.saic <- yhat.train.mat%*%w.hat.aic
    yhat.train.sbic <- yhat.train.mat%*%w.hat.bic

    yhat.eval.saic <- yhat.eval.mat%*%w.hat.aic
    yhat.eval.sbic <- yhat.eval.mat%*%w.hat.bic

    saic.r.sq <- r.sq(yhat.train.saic,ybar.train)
    sbic.r.sq <- r.sq(yhat.train.sbic,ybar.train)

    saic.mse <- mean((yhat.eval.saic-y.eval)^2)
    sbic.mse <- mean((yhat.eval.sbic-y.eval)^2)

    write(c(jma.r.sq,mma.r.sq,saic.r.sq,sbic.r.sq,aic.r.sq,bic.r.sq,cv.r.sq,cp.r.sq),
          file="r_squared.out",ncol=8,append=TRUE)
    write(c(which.min(aic.train.vec),which.min(bic.train.vec),which.min(cv.train.vec),which.min(cp.train.vec)),
          file="model_selection.out",ncol=4,append=TRUE)
    write(c(jma.mse,mma.mse,saic.mse,sbic.mse,aic.mse,bic.mse,cv.mse,cp.mse),
          file="mse.out",ncol=8,append=TRUE)

}

stopCluster(cl)
```