Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Model Averaging - The R Package ma

Jeffrey Racine
McMaster University

Abstract

There exist a number of R packages that perform model averaging (see **AICcmodavg**, **BMA**, **glmulti** and **MuMLn** by way of illustration). Though existing packages contain a rich set of features, they rely on the practitioner to supply the set of candidate models and they tend to focus on parameters common to all candidate models (i.e., coefficients on linear predictors). The **ma** package takes a somewhat different approach. Using heuristics that can be modified by the user, we strive to automatically generate a rich set of candidate models that are subsequently averaged, and we avoid focusing on parameters per se, instead focusing on conditional mean models, their derivatives, and nonparametric inferential procedures. Categorical predictors can be modelled in a variety of ways including a kernel-smoothed varying coefficient approach. The resulting models are competitive with fully nonparametric methods and attenuate bias arising from model mis-specification.

Keywords: parametric, weighted average, R.

1. Introduction

Practitioners frequently wrestle with issues surrounding model uncertainty, and model *selection* is perhaps one of the most widely used solutions to this problem. Model selection deals with model uncertainty by selecting one model from among a set of candidate models based on 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 and Wahba (1979)), and the *Focused Information Criterion* (FIC) (Claeskens and Hjort (2003)), to name but a few.

Model *averaging* deals with model uncertainty not by having the user select one model from among a set of candidate models according to a criterion such as C_p, AIC or BIC, but instead by averaging over the set of candidate models in a particular manner. The goal is to

reduce estimation variance while controlling mis-specification bias. There is a longstanding literature on Bayesian model averaging; see Hoeting, Madigan, Raftery, and Volinsky (1999) for a comprehensive review. There is also a rapidly-growing literature on frequentist methods for model averaging, including Buckland, Burnhamn, and Augustin (1997), Hansen (2007), Wan, Zhang, and Zou (2010), and Hansen and Racine (2012), among others.

The R package **ma** aims to automate the process of model averaging thereby freeing the practitioner from tedious details. The **ma** package adopts a frequentist approach; for those interested in a Bayesian approach see the R package **BMA**. It is assumed that the practitioner is using regression techniques and is concerned about the impact of model mis-specification on inference and prediction. As far as possible, the interface is designed to mimic that for `lm()` with some notable exceptions outlined below.

2. Implementation of the Model Average Procedure

Consider a nonparametric regression model containing both categorical and continuous predictors where one is interested in modeling the unknown conditional mean in the following location-scale model,

$$Y = g(\mathbf{X}, \mathbf{Z}) + \sigma(\mathbf{X}, \mathbf{Z})\,\varepsilon, \tag{1}$$

where $g(\mathbf{x}, \mathbf{z}) = E[Y|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}]$ is an unknown function, $\mathbf{X} = (X_1, \ldots, X_q)^{\mathrm{T}}$ is a q-dimensional vector of continuous predictors, and $\mathbf{Z} = (Z_1, \ldots, Z_r)^{\mathrm{T}}$ is an r-dimensional vector of categorical predictors. Letting $\mathbf{z} = (z_s)_{s=1}^{r}$, we assume that z_s takes c_s different values in $D_s \equiv \{0, 1, \ldots, c_s - 1\}$, $s = 1, \ldots, r$, and let c_s be a finite positive constant.

2.1. Categorical Predictors and Candidate Model Selection

To handle the presence of categorical predictors, we might estimate $g(\cdot)$ by tensor-product polynomial splines weighted by categorical kernel functions as proposed by Ma, Racine, and Yang (2015). Let $\mathcal{B}(\mathbf{x})$ be the tensor-product polynomial splines and $L(\mathbf{Z}, \mathbf{z}, \lambda)$ be a product categorical kernel function. Then the nonparametric function $g(\mathbf{x}, \mathbf{z})$ can be approximated by $\mathcal{B}(\mathbf{x})^{\mathrm{T}} \beta(\mathbf{z})$, where $\beta(\mathbf{z})$ is a $\mathbf{K}_n \times 1$ vector with $\mathbf{K}_n \to \infty$ as $n \to \infty$. We estimate $\beta(\mathbf{z})$ by minimizing the following weighted least squares criterion,

$$\widehat{\beta}(\mathbf{z}) = \arg \min_{\beta \in R^{\mathbf{K}_n}} \sum_{i=1}^{n} \left\{ Y_i - \mathcal{B}(\mathbf{X}_i)^{\mathrm{T}} \beta \right\}^2 L(\mathbf{Z}_i, \mathbf{z}, \lambda). \tag{2}$$

Thus $g(\mathbf{x}, \mathbf{z})$ is estimated by $\widehat{g}(\mathbf{x}, \mathbf{z}) = \mathcal{B}(\mathbf{x})^{\mathrm{T}} \widehat{\beta}(\mathbf{z})$. If $\mathcal{B}(\mathbf{x}) = \mathbf{x}$ (i.e., the identity basis) the we are conducting standard linear (in predictors) weighted least squares estimation (i.e., $\widehat{g}(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathrm{T}} \widehat{\beta}(\mathbf{z})$). If in addition we introduced the categorical predictors as additive factors instead of using kernel weighting, we are conducting standard linear least squares with additive dummy variables (i.e., $\widehat{g}(\mathbf{x}, \mathbf{z}) = \mathbf{z}^{\mathrm{T}} \widehat{\alpha} + \mathbf{x}^{\mathrm{T}} \widehat{\beta}$); see the related work of Ma and Racine (2013). Each of these possibilities can be entertained among the set of candidate models implemented in the **ma** package.

2.2. Candidate Model Basis Function Choice

Note that $\mathcal{B}(\mathbf{x})$ need not necessarily be a tensor product (i.e., a nonparametric basis per Ma *et al.* (2015)). It could be additive (i.e, a *semiparametric* basis per Ma and Racine (2013)) or a perhaps a modified Taylor-type basis that admits differing orders in each dimension. Regardless of the nature of the basis, each candidate model used in the model average will differ in terms of its polynomial degree in each dimension, number of interior knots, and smoothing parameters for the kernel weighting function.

Denote the mth such model as $\widehat{g}_m(\mathbf{x}, \mathbf{z})$. Having fitted the M candidate models over which the averaging is to take place, we will require a model averaging criterion; the **ma** package allows for two such criteria which we briefly outline below.

2.3. Hansen's (2007) Mallows Model Average Criterion

The Mallows (Mallows 1973) criterion for the model average estimator (Hansen 2007) is

$$C_n(w) = w'\hat{\mathbf{E}}'\hat{\mathbf{E}}w + 2\sigma^2 K'w,$$

where $\hat{\mathbf{E}}$ is the $T \times M$ matrix with columns containing the residual vector from the mth candidate estimating equation, K the $M \times 1$ vector whose elements are the number of parameters (i.e., rank) in each model, and σ^2 the variance from the largest dimensional model. This criterion is used to select the weight vector $\hat{w}$, i.e.,

$$\hat{w} = \text{argmin}_w\, C_n(w).$$

Because $\text{argmin}_w\, C_n(w)$ has no closed-form solution, the weight vector is found numerically. The solution involves constrained minimization subject to non-negativity and summation constraints, which constitutes a classic quadratic programming problem. This criterion involves nothing more than computing the residuals for each candidate estimating equation, obtaining the rank of each candidate estimating equation, and solving a simple quadratic program. The Mallows model averaging (MMA) criterion $C_n(w)$ provides an estimate of the average squared error from the model average fit, and has been shown to be asymptotically optimal in the sense of achieving the lowest possible squared error in a class of model average estimators. See Hansen (2007) for further details.

2.4. Hansen and Racine's (2012) Jackknife Model Average Criterion

Hansen and Racine (2012) propose an alternative jackknife model averaging (JMA) criterion for the model average estimator given by

$$CV_n(w) = \frac{1}{n}(y - \tilde{\mathbf{Y}}w)'(y - \tilde{\mathbf{Y}}w),$$

where $\tilde{\mathbf{Y}}$ is the $T \times M$ matrix with columns containing the jackknife fitted value vector from the mth candidate estimating equation formed by deleting the tth observation when constructing the tth prediction. Like its Mallows counterpart, this involves solving a quadratic program where we minimize $(y - \tilde{\mathbf{Y}}w)'(y - \tilde{\mathbf{Y}}w) = y'y + w'\tilde{\mathbf{Y}}'\tilde{\mathbf{Y}}w - 2y'\tilde{\mathbf{Y}}w$ and the first term is ignorable as it does not involve the weight vector w. In the presence of homoskedastic errors, JMA and MMA are nearly equivalent, but when the errors are heteroskedastic, JMA delivers models with significantly lower MSE.

Whether using JMA or MMA, both involve solving a simple quadratic program (the R package **quadprog** is invoked for its solution).

3. R Function Interface and Defaults

The main function is called `lm.ma()`. Models are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. Note that, unlike `lm()` in which the formula interface specifies functional form, in `lm.ma()` the formula interface is strictly for listing the variables involved and the procedure will determine an appropriate model averaged functional form. Do not incorporate transformations, interactions and the like in the formula interface for lm.ma as these will most surely fail.

The function `lm.ma()` computes a model that is the weighted average of a set of least squares candidate models whose predictors are generated by common basis functions (additive, modified Taylor polynomial, or tensor products). The candidate models increase in complexity from linear bases (if `degree.min=1`) through higher order ones up to degree.max. All bases are of the Bernstein polynomial class, as opposed to raw polynomials, and allow for differing degrees across multivariate predictors regardless of the basis invoked. When `knots=TRUE`, interior *quantile* knots are used and the Bernstein polynomials become B-spline bases and we are then averaging over regression spline models (i.e., knots are placed at equi-probable quantiles for each predictor, so one interior knot would be placed at the median value of each predictor, three at the first, second, and third quartiles etc.). When the number of numeric predictors is two or more, the modified Taylor polynomial includes interaction terms (i.e., cross-partial derivatives) up to order degree minus one. Since we are averaging over models that are nonlinear in the predictors, derivatives will be vectors that potentially depend on the values of every predictor. An ad-hoc formula is used to determine the relationship between the largest (i.e., most complex) model, the sample size, and the number of predictors. This ad-hoc rule was set so that, as the sample size increases, we can approximate ever more complex functions while necessarily restricting the size of the largest model in small sample settings. Categorical predictors can enter additively and linearly (if `vc=FALSE`) or in a parsimonious manner by exploiting recent developments in semiparametric varying coefficient models along the lines of Ouyang, Li, and Racine (2013). With the options `knots=TRUE` and `vc=TRUE`, we are averaging over varying-coefficient regression splines.

The heuristic used for `degree.max`, the maximum value for the basis degree in each dimension, defaults to `max(2,ceiling(log(n)-2*log(1+k)))` where `k` is the number of numeric predictors and `n` the number of observations.

The heuristic used for `lambda.num.max`, the maximum value for the smoothing parameter grid in each dimension, defaults to `max(2,ceiling(log(n)-2*log(1+p)))` where `p` is the number of categorical predictors and `n` the number of observations.

When there exist more than two predictors, the spline degree increases by 2 by default (this can be controlled by the option `degree.by`) so we would estimate polynomials of degree 1,3,5, etc. The option `all.combinations=FALSE` can be invoked in multivariate settings to control the number of candidate models (this option has the effect of using the same degree for each predictor in each candidate model rather than considering all possible combinations).

3.1. Generic Accessor Functions

The function summary is used to obtain and print a summary of the results. The generic accessor functions `coef`, `fitted`, `predict`, `plot` (see `?plot.lm` for details) and `residuals` extract various useful features of the value returned by lm.ma.

3.2. Plotting lm.ma Objects.

The function `plot.lm.ma()` plots an `lm.ma` object. It can plot either the conditional mean (default) or partial derivatives of the conditional mean with respect to the predictors, whether of type `numeric` or `factor`. Nonparametric confidence intervals can be plotted via the option `plot.ci=TRUE`, while data (and a rug) can be included when `plot.data=TRUE` (`plot.rug=TRUE`).

4. Simulated Univariate Illustration

By way of illustration, consider the following chunk of R code. We simulate data from a nonlinear data generating process (DGP). The practitioner might consider, say, linear regression via `lm(y~x)` but this model would clearly be mis-specified as might other models they consider. Instead, consider the model produced by `lm.ma(y~x)`. The fitted model and data are plotted in Figure 1.

```
R> library(ma)
R> set.seed(42)
R> n <- 1000
R> x <- runif(n)
R> dgp <- sin(4*pi*x)
R> y <- dgp + rnorm(n,sd=0.25*sd(dgp))
R> model <- lm.ma(y~x,verbose=FALSE)
```

Figure 1 reveals that the model average estimator is more faithful to the underlying DGP than the naive linear regression model. Of course no serious practitioner would likely have settled for the naive linear model, but that is besides the point. The point to be made is that the model average approach is known to possess a number of appealing characteristics and by automating the choice of the candidate models we can provide an interface for model averaging that may be of use for practitioners.

4.1. Illustrative Application

Next we consider an application based on the Demographic and Health Survey data on childhood nutrition in India (Koenker 2011). This dataset involves 37,623 observations and we consider three numeric predictors and three categorical predictors. The dependent variable is the child's height (centimeters) and we consider predictors child's age (months), mother's BMI (kilograms per meter squared), mother's years of education, child's sex, whether or not child is a twin, and birth order of the child. We consider an additive basis with additive dummies for the candidate models. Results for the conditional mean are presented in partial plots (i.e., the off-axis predictors are held constant at their respective medians/modes) in Figure 2. We also present the marginal effects (i.e., derivatives) in Figure 3. The model summary appears in Table 1.

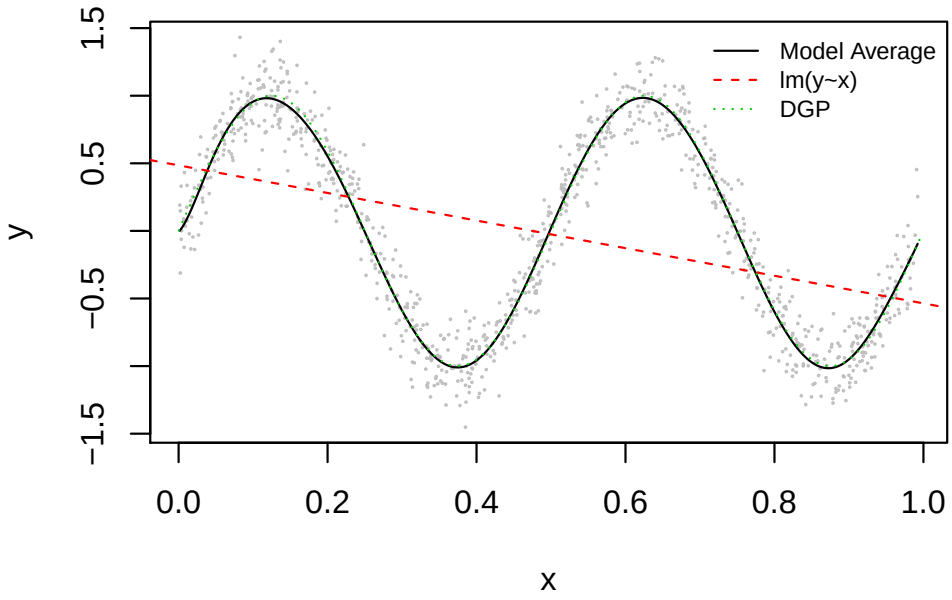

Figure 1: Simulated illustration, $n = 1000$ observations, model average estimate, linear (in predictor) estimate, and data generating process (DGP) (figure created via `plot(model,plot.data=TRUE)`).

Figures 2 and 3 would accord with one's expectations. There are clear nonlinearities at work with regards to the child's age that go beyond simple low order polynomials, while it is interesting to consider the marginal effect associated with, e.g., whether one has a twin and birth order.

5. ANOVA-Based Inference

An ANOVA-based test of significance is considered. The standard asymptotics are not applicable in this setting (though they are available to the user). Instead a nonparametric bootstrap procedure is implemented.

The bootstrap procedure is straightforward.

- Fit the model average model for all predictors (call this the unrestricted model).
- Fit the model average model for all predictors except the one whose significance is being tested (call this the restricted model).
- Compute the usual F-statistic from these two models. Call this statistic F.
- Use a residual-based bootstrap procedure based on the residuals from the restricted model, then bootstrap the F-statistic following the same steps as above. Call these $F_1^*, F_2^*, \ldots, F_B^*$.
- Compute the nonparametric P-value in the usual manner (i.e., $\hat{P} = B^{-1} \sum_{b=1}^{B} \mathbf{1}(F_b^* > F)$).

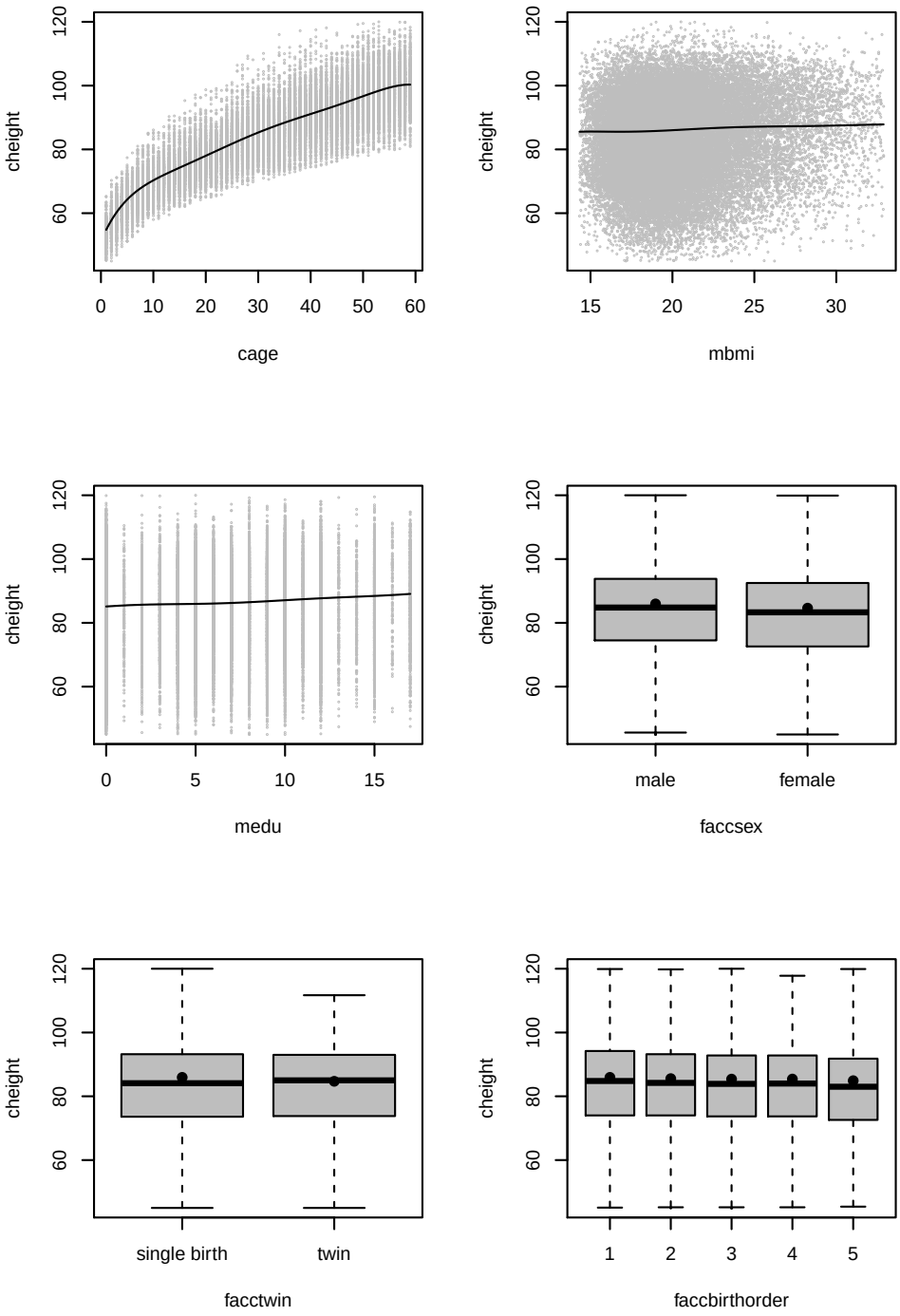

Figure 2: India childhood nutrition derivative estimate.

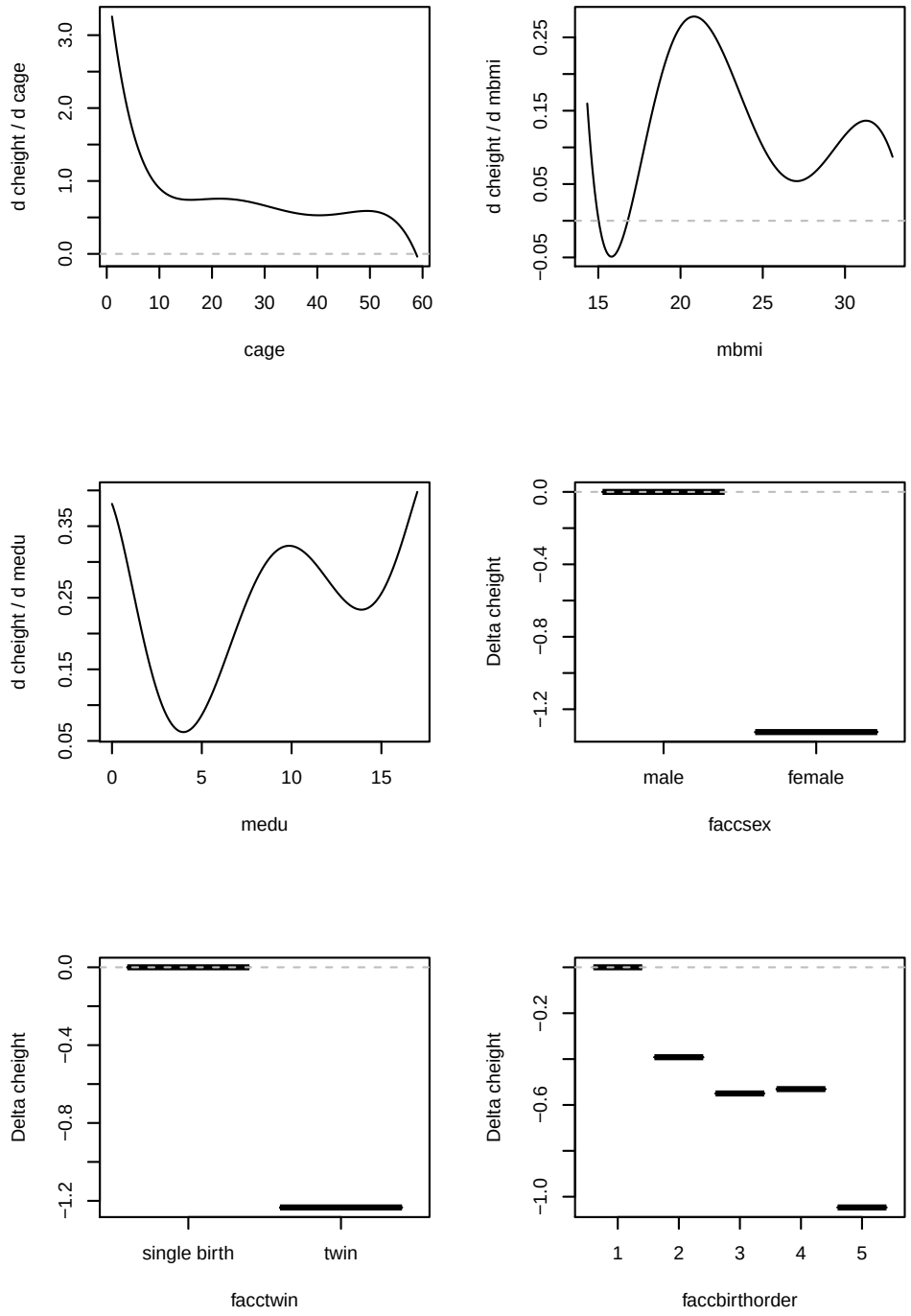

Figure 3: India childhood nutrition derivative estimate.

Table 1: Model Summary For Childhood Nutrition Model.

```
R> summary(model)

Call:
lm.ma.formula(formula = cheight ~ cage + mbmi + medu + faccsex +
    facctwin + faccbirthorder, basis = "additive", vc = FALSE,
    verbose = FALSE)

Model Averaging Linear Regression (Additive Dummy Specification)
Model average criterion: Jackknife (Hansen and Racine (2013))
Minimum degree: 0
Maximum degree: 10
Basis: additive
Number of observations: 37623
Number of numeric predictors: 3
Number of categorical predictors: 3
Equivalent number of parameters: 25.66
Residual standard error: 5.163 on 37597.34 degrees of freedom
Multiple R-squared: 0.8429
Estimation time:  54.6 seconds

Number of candidate models: 216
Non-zero model average weights: 0.0004 0.0036 0.0031 0.0021 0.0516 0.0729 0.1272 0.0058 0.1008 0.2854 0.3471
Non-zero weight model ranks: 13.0 15.0 21.0 21.0 22.0 22.0 24.0 24.0 24.0 26.0 28.0
```

- Reject H_0 if $\hat{P} < \alpha$ where α is the nominal level of the test.

A simulated example is presented in Table 2.

Extensive simulations reveal that the test is correctly sized and displays power that increases with the departure from the null and the sample size.

6. Parallel Computation

Parallel processing can be enabled by invoking the option `parallel=TRUE`. This can reduce computation by up to `1/parallel.cores`, particularly for multiple predictor models where computation of each candidate model may be non-trivial. Note that the default number of cores invoked is the number of cores present in the CPU on which the code is executed. Note also that, when `parallel=TRUE`, this applies to all computation *except* for solving the quadratic program which invokes a call to `solve.QP` from the R package **quadprog** that, at this time, does not support parallel processing.

7. Limitations and Caveats

The model average approach implemented in the **ma** package is capable of statistically consistent estimation of certain classes of smooth functions (e.g., analytic) at rates approaching or equal to those of correctly specified parametric models (extensive simulations highlighting this feature are available upon request). In this sense the approach is competitive with and even more efficient than some popular nonparametric approaches such as locally weighted

Table 2: Simulated ANOVA Test of Significance.

```
R> model <- lm.ma(y~x,
R+                compute.anova=TRUE,
R+                compute.anova.boot=TRUE,
R+                degree.min=1,
R+                verbose=FALSE)
R> summary(model)

Call:
lm.ma.formula(formula = y ~ x, compute.anova = TRUE, compute.anova.boot = TRUE,
    degree.min = 1, verbose = FALSE)

Model Averaging Linear Regression (Varying Coefficient Specification)
Model average criterion: Jackknife (Hansen and Racine (2013))
Minimum degree: 1
Maximum degree: 14
Basis: additive
Number of observations: 1000
Number of numeric predictors: 1
Equivalent number of parameters: 12.12
Residual standard error: 0.1768 on 987.88 degrees of freedom
Multiple R-squared: 0.9413
Estimation time:  3.5 seconds

Number of candidate models: 14
Non-zero model average weights: 0.0005 0.0006 0.0442 0.2169 0.7378
Non-zero weight model ranks: 2.0 6.0 8.0 10.0 13.0

Nonparametric significance test
P Value:
x < 2.22e-16 *** [F = 1425.059]
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

kernel estimation. This is achieved automatically and without input by the user. This stands in contrast to the model average approaches embodied in the R packages listed previously that rely on a set of candidate models provided by the user that may not be endowed with this feature *unless* the user also ties the dimensions of their candidate models to the available sample information, among other important aspects (such as including interactions among predictors and allowing for a sufficiently high polynomial order in the candidate models to capture a sufficiently rich set of nonlinearities).

It should come as no surprise that this approach, like its nonparametreic peers, will face the *curse of dimensionality*. To partially address this shortcoming, in order for our model average approach to deliver statistically consistent estimation of broad classes of functions, the complexity of the maximal basis function in the set of candidate models to be averaged over must increase with the amount of sample information present, among other important aspects. However, for a finite sample size, the maximum dimension of the basis is constrained by this limit. Some heuristics already mentioned seek to trade off consistency versus computational feasibility given the data at hand. Of course, default settings appropriate for one DGP may be inappropriate for another, so some sensitivity analysis may be called for to ensure that the final results are not upset by modifying the default configuration.

When using the default settings in high dimensional situations, the number of candidate models may grow unreasonably large (say 2,500 or more). Furthermore, the dimension of the maximal basis may similarly grow unreasonably large (say 5,000 or more). In such cases you might want to i) increase `S`, ii) increase `lambda.S` (if categorical predictors are present and `vc=TRUE`), iii) set and restrict `degree.max`, iv) set and restrict `lambda.num.max` if categorical predictors are present, v) reduce `segments.max` (if `knots=TRUE`), vi) set `all.combinations=FALSE`, or perhaps instead consider a semiparametric model (`basis="additive"` and `vc=FALSE` produces semiparametric additive candidate models - see the example in `?india` for an illustration).

It is therefore helpful to remember that the goal of model averaging is to outperform model assertion or model selection by reducing estimation variance while controlling mis-specification bias. The goal is not to obtain statistically consistent estimators - that remains the domain of fully nonparametric estimation. As Box (1979) famously remarked in a section titled "All models are wrong, but some are useful",

> Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law $PV = RT$ relating pressure P, volume V and temperature T of an "ideal" gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules.
>
> For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?".

We might rephrase this as "Any single model is wrong but potentially useful. However, a model which is itself an average of a set of models is potentially even more illuminating and useful still."

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*, pp. 267–281. Akademiai Kiado, Budapest.

Box G (1979). "Robustness in the strategy of scientific model building." In RL Launer, GN Wilkinson (eds.), *Robustness in Statistics*, pp. 201–236. Academic Press.

Buckland ST, Burnhamn KP, Augustin NH (1997). "Model Selection: An Integral Part of Inference." *Biometrics*, **53**, 603–618.

Claeskens G, Hjort NL (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.

Hansen BE (2007). "Least Squares Model Averaging." *Econometrica*, **75**, 1175–1189.

Hansen BE, Racine JS (2012). "Jackknife model averaging." *Journal of Econometrics*, **167**(1), 38–46.

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). "Bayesian Model Averaging: A Tutorial." *Statistical Science*, **14**, 382–417.

Koenker R (2011). "Additive models for quantile regression: Model selection and confidence bandaids." *Brazilian Journal of Probability and Statistics*, **25**, 239–262.

Ma S, Racine JS (2013). "Additive Regression Splines With Irrelevant Categorical and Continuous Regressors." *Statistica Sinica*, **23**, 515–541.

Ma S, Racine JS, Yang L (2015). "Spline Regression in the Presence of Categorical Predictors." *Journal of Applied Econometrics*, **30**, 703–717.

Mallows CL (1973). "Some comments on C_p." *Technometrics*, **15**, 661–675.

Ouyang D, Li Q, Racine JS (2013). "Categorical Semiparametric Varying Coefficient Models." *Journal of Applied Econometrics*, **28**(3), 551–579.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**, 461–464.

Stone CJ (1974). "Cross-Validatory Choice and Assessment of Statistical Predictions (with discussion)." *Journal of the Royal Statistical Society*, **36**, 111–147.

Wan ATK, Zhang X, Zou G (2010). "Least squares model averaging by Mallows criterion." *Journal of Econometrics*, **156**(2), 277–283.

Affiliation:

Jeffrey Racine
McMaster University
Department of Economics Kenneth Taylor Hall, Room 426 McMaster University 1280 Main Street West Hamilton, Ontario, Canada, L8S 4M4
E-mail: racinej@mcmaster.ca
URL: https://socialsciences.mcmaster.ca/people/racinej

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd