Select relevant moderators using Bayesian regularized meta-regression

Caspar J. Van Lissa[1,2], Sara van Erp[1], & Eli-Boaz Clapper[1]

[1] Utrecht University, dept. Methodology & Statistics

[2] Open Science Community Utrecht

Author Note

This is a preprint paper, generated from Git Commit # 716cf43. This work was funded by a NWO Veni Grant (NWO Grant Number VI.Veni.191G.090), awarded to the lead author.

The authors made the following contributions. Caspar J. Van Lissa: Conceptualization, Formal Analysis, Funding acquisition, Methodology, Project administration, Software, Supervision, Writing – original draft, Writing – review & editing; Sara van Erp: Methodology, Software, Writing – original draft, Writing – review & editing; Eli-Boaz Clapper: Formal Analysis, Writing – original draft, Writing – review & editing.

Correspondence concerning this article should be addressed to Caspar J. Van Lissa, Padualaan 14, 3584CH Utrecht, The Netherlands. E-mail: c.j.vanlissa@uu.nl

Abstract

When analyzing a heterogeneous body of literature, there may be many potentially relevant between-studies differences. These differences can be coded as moderators, and accounted for using meta-regression. However, many applied meta-analyses lack the power to adequately account for multiple moderators, as the number of studies on any given topic is often low. The present study introduces Bayesian Regularized Meta-Analysis (BRMA), an algorithm that performs variable selection to identify relevant moderators from a larger number of candidates. This approach is suitable when heterogeneity is suspected, but it is not known which moderators most strongly influence the observed effect size. We present a simulation study to validate the performance of BRMA relative to state-of-the-art meta-regression (RMA). Results indicated that BRMA compared favorably to RMA on three metrics: predictive performance (a measure of generalizability), the ability to reject irrelevant moderators, and the ability to recover population parameters with low bias. BRMA had slightly lower ability to detect true effects of relevant moderators, but the overall proportion of Type I and Type II errors was equivalent to RMA. BRMA regression coefficients were slightly biased towards zero (by design), but its estimates of residual heterogeneity were unbiased. BRMA performed well with as few as 20 studies in the training data, suggesting its suitability as a small sample solution. We discuss how applied researchers can use BRMA to explore between-studies heterogeneity in meta-analysis.

Keywords: meta-analysis, machine learning, bayesian, lasso, horseshoe, regularized

Word count: 5356

Select relevant moderators using Bayesian regularized meta-regression

Meta-analysis is a quantitative form of evidence synthesis, whereby effect sizes from multiple similar studies are aggregated. In its simplest form, this aggregation consists of a weighted average of the observed effect sizes. Weighting accounts for the fact that some observed effect sizes are assumed to be more informative about the underlying population effect. The weights are based on specific assumptions; for example, the *fixed effect* model assumes that all observed effect sizes reflect one underlying true population effect. This assumption is well-suited to the situation where effect sizes from close replication studies are meta-analyzed (Higgins, Thompson, & Spiegelhalter, 2009). The *random effects* model, by contrast, assumes that population effect sizes follow a normal distribution. Each observed effect size provides information about the mean and standard deviation of this distribution of population effect sizes. This assumption is more appropriate when studies are conceptually similar and differences between them are random (Higgins et al., 2009).

Not all heterogeneity in effect sizes is random, however. Quantifiable between-study differences may introduce systematic heterogeneity. Such between-study differences are known as "moderators." For example, if studies have been replicated in Europe and the Americas, this difference could be captured by a binary moderator called "continent." Alternatively, if studies have used different dosages of the same drug, this may be captured by a continuous moderator called "dosage." Systematic heterogeneity in the observed effect sizes can be accounted for using *meta-regression* (see López-López, Marín-Martínez, Sánchez-Meca, Van den Noortgate, & Viechtbauer, 2014). This technique provides estimates of the effect of one or more study characteristics on the overall effect size, as well as of the overall effect size and residual heterogeneity after controlling for their influence.

One common application of meta-analysis is to summarize existing bodies of literature. In such situations, the number of moderators is often relatively high because similar research questions have been studied in different laboratories, using different methods, instruments, and samples. Each of these between-study differences could be coded as a moderator, and some of these moderators may explain systematic heterogeneity.

The influence of multiple moderators can be accounted for using meta-regression. However, like any regression-based approach, meta-regression requires a relatively high number of cases (studies) per parameter to obtain sufficient power to examine heterogeneity. In applied meta-analyses, the number of available studies is often too low to examine heterogeneity reliably (Riley, Higgins, & Deeks, 2011). At the same time, there are many potential sources of heterogeneity, as similar research questions are studied in different laboratories, using different methods, instruments, and samples. This leads to a problem known as the "curse of dimensionality": the number of candidate moderators is large relative to the number of cases in the data. Between-studies differences present a non-trivial challenge to data aggregation using classic meta-analytic methods. At the same time, they also provide an unexploited opportunity to learn which differences between studies have an impact on the effect size found, if adequate exploratory techniques are used.

Addressing this curse of dimensionality necessitates *variable selection*: the selection of a smaller subset of relevant moderators from a larger number of candidate moderators. One way to perform variable selection is by relying on theory. However, in many fields of science, theories exist at the individual level of analysis (e.g., in social science, at the level of individual people). These theories do not necessarily generalize to the study level of analysis. Using theories at the individual level for moderator selection at the study level amounts to committing the ecological fallacy: generalizing inferences across levels of analysis (Jargowsky, 2004). To illustrate what a theory at the study level of analysis might look like, consider the so-called *decline effect*. It is a phenomenon whereby effect sizes in a particular tranche of the literature seem to diminish over time (Schooler, 2011). It has been theorized that the decline effect can be attributed to regression to the mean: A finding initially draws attention from the research community because an anomalously large effect size has been published, and subsequent replications find smaller effect sizes.

Based on the decline effect, we might thus expect the variable “year of publication” to be a relevant moderator of study effect sizes. Note that this prediction is valid even if year is orthogonal to the outcome of interest within each study. Until more theory about the drivers of between-study heterogeneity is developed, however, this approach will have limited utility for variable selection.

An alternative solution is to rely on statistical methods for variable selection. This is a focal issue in the discipline of machine learning (Hastie, Tibshirani, & Friedman, 2009). There is precedent for the use of machine learning to perform variable selection in meta-analysis (Van Lissa, 2020). This work used the *random forest* algorithm; a non-parametric approach that largely ignores irrelevant moderators. One limitation of random forests is that non-parametric models are harder to interpret, particularly for a readership that is accustomed to linear models, where the effect of each predictor is described by a single parameter. An alternative method for variable selection that can be used in linear models is *regularization*: shrinking model parameters towards zero, such that irrelevant moderators are eliminated. The present paper introduces *Bayesian regularized meta-regression* (BRMA), an algorithm that uses Bayesian estimation with regularizing priors to perform variable selection in meta-analysis. The algorithm is implemented in the function `brma()` in the R-package `pema`.

Statistical underpinnings

To understand how BRMA estimates the relevant parameters and performs variable selection, it is instructional to first review the statistical underpinnings of the aforementioned classic approaches to meta-analysis. First is the fixed-effect model, which assumes that each observed effect size T_i is an estimate of an underlying true effect size Θ (Hedges & Vevea, 1998). The only cause of heterogeneity in observed effect sizes is presumed to be effect size-specific sampling variance, v_i, which is treated as known, and computed as the square of the standard error of the effect size. Thus, for a collection of k

studies, the observed effects sizes of individual studies i (for i = 1,2, . . . k) are given by:

$$T_i = \Theta + \epsilon_i \tag{1}$$

$$\text{where } \epsilon_i \sim N(0, v_i) \tag{2}$$

Under the fixed effect model, the estimated population effect size $\hat{\theta}$ is obtained by computing a weighted average of the observed effect sizes. If sampling error is assumed to be the only source of variance in the observed effect size, then it follows that studies with smaller standard errors estimate the underlying true effect size more precisely. The fixed-effect weights are thus simply the reciprocal of the sampling variance, $w_i = \frac{1}{v_i}$. The estimate of the true effect is a weighted average across observed effect sizes:

$$\hat{\theta} = \frac{\sum_{i=1}^{k} w_i T_i}{\sum_{i=1}^{k} w_i} \tag{3}$$

Whereas the fixed-effect model assumes that only one true population effect exists, the random-effects model assumes that true effects may vary for unknown reasons, and thus follow a (normal) distribution of their own (Hedges & Vevea, 1998). This heterogeneity of the true effects is represented by their variance, τ^2. The random effects model thus assumes that the heterogeneity in observed effects can be decomposed into sampling error and between-studies heterogeneity, resulting in the following equation for the observed effect sizes:

$$T_i = \Theta + \zeta_i + \epsilon_i \tag{4}$$

$$\text{where } \zeta_i \sim N(0, \tau^2) \tag{5}$$

$$\text{and } \epsilon_i \sim N(0, v_i) \tag{6}$$

In this model, Θ is the mean of the distribution of true effect sizes, and τ^2 is its variance, which can be interpreted as the variance between studies.

If the true effect sizes follow a distribution, then even less precise studies (with larger sampling errors) may provide some information about this distribution. Like fixed-effect weights, random effects weights are still influenced by sampling error, but this influence is attenuated by the estimated variance of the true effect sizes. The random-effects weights are thus given by $w_i = \frac{1}{v_i + \hat{\tau}^2}$. It is important to note that, whereas the sampling error for each individual effect size is treated as known, the between-study heterogeneity τ^2 must be estimated. This estimate is represented by $\hat{\tau}^2$.

Meta-regression. The random-effects model assumes that causes of heterogeneity in the true effect sizes are unknown, and that their influence is random. Oftentimes, however, there are systematic sources of heterogeneity in true effect sizes. These between-study differences can be coded as moderators, and their influence can be estimated and controlled for using meta-regression. Meta-regression with p moderators can be expressed with the following equation, where $x_{1 \ldots p}$ represent the moderators, and $\beta_{1 \ldots p}$ the regression coefficients:

$$T_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \zeta_i + \epsilon_i \tag{7}$$

$$\tag{8}$$

Note that β_0 represents the intercept of the distribution of true effect sizes after controlling for the moderators and the error term ζ_i represents residual between-studies heterogeneity. This term is included because unexplained heterogeneity often remains after accounting for the moderators. This is a mixed-effects model; the intercept and effects of moderators are treated as fixed and the residual heterogeneity as random (López-López et al., 2014).

To solve this model, the regression coefficients and residual heterogeneity must be estimated simultaneously. Numerous methods have been proposed to estimate meta-regression models, the most commonly used of which is restricted maximum likelihood (REML). REML is an iterative method, meaning it performs the same calculations repeatedly, updating the estimated regression coefficients and residual heterogeneity until these estimates stabilize. This estimator has low bias, which means that the average value of the estimated regression coefficients and residual heterogeneity is close to their true values (Panityakul, Bumrungsup, & Knapp, 2013). However, this bias comes at the cost of higher variance, which means that the estimated values of a population parameter vary more from one sample to the next. In practice, an estimator with higher variance generalizes less well to new data. This phenomenon is known as the *bias-variance trade-off*. Regularization increases bias to reduce variance. A disadvantage of this trade-off is that model parameters can no longer be interpreted as straightforwardly as OLS regression coefficients. An advantage is that the resulting model is more generalizable and makes better predictions for new data (see Hastie et al., 2009).

Regularized regression. Regularized regression biases parameter estimates towards zero. Before examining the Bayesian case, we will explain the general principle of regularization in frequentist regression. OLS regression estimates the model parameters by minimizing the Residual Sum of Squares (RSS) of the dependent variable, which is given by:

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

The resulting parameter estimates are those that give the best predictions of the dependent variable in the present data set. Penalized regression, by contrast, adds a penalty term to the RSS. One commonly used penalty is the L1-norm of the regression coefficients, or LASSO penalty (Hastie et al., 2009), which corresponds to the sum of their absolute values. This gives the penalized residual sum of squares:

$$PRSS = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

Because the penalty term is a function of the regression coefficients, the optimizer is incentivized to keep the regression coefficients as small as possible. In this equation, λ is a tuning parameter that determines how influential the penalty term will be. If λ is zero, the shrinkage penalty has no impact and the penalized regression will produce the OLS estimates. If $\lambda \to \infty$, all coefficients shrink towards zero, producing the null model. Generally, cross-validation is used to find the optimal value for the penalty parameter λ. Note that the LASSO penalty is but one example of a shrinkage penalty; other penalties exist.

Bayesian estimation. One alternative to the use of a shrinkage penalty is Bayesian estimation with a regularizing prior. Whereas classical, frequentist estimation relies solely on the data at hand, Bayesian estimation combines information from the data with a *prior distribution*. The prior distribution is a probability distribution that reflects expectations about likely parameter values. This prior is updated with the likelihood of the data to form a posterior distribution, which reflects expectations about likely parameter values after having seen the data.

A regularizing prior distribution reflects the expectation that not all regression coefficients are substantial enough to be included in the model. There are many different regularizing prior distributions (Erp, Oberski, & Mulder, 2019). Some of these result in exactly the same solutions as frequentist penalized methods. For example, applying an independent double exponential (i.e., Laplace) prior to regression coefficients results in posterior modes that are equal to the classical LASSO estimates (Park & Casella, 2008). Other distributions have been developed specifically for the purpose of providing good shrinkage properties, meaning that the prior pulls small regression coefficients towards zero, while leaving larger regression coefficients mostly unaffected. A popular prior in this regard is the horseshoe prior (Carvalho, Polson, & Scott, 2010). It has heavier tails than the LASSO prior, which means that it does not shrink (and therefore bias) substantial coefficients as much.

Implementation. Bayesian penalized meta-analysis is implemented in the function `brma()` in the R-package `pema`. For estimation, it depends on Stan, a probabilistic programming language that uses Hamiltonian Monte Carlo to sample from the posterior distribution (Stan Development Team, 2019). Being written in C++, Stan is computationally efficient, but models must be compiled prior to estimation. This results in substantial computational overhead. To avoid this overhead, `pema` uses pre-compiled models corresponding to random-effects and three-level meta-regression, with and without an intercept. Future updates may bring additional models. At the time of writing, `brma()` supports two priors: the LASSO and the regularized horseshoe. The LASSO prior is implemented as follows:

$$\beta_j \sim \text{DE}(0, \frac{s}{\lambda})$$

where DE denotes the double exponential distribution with a location equal to 0 and a scale determined by a global scale parameter s and an inverse-tuning parameter λ. By default in `brma()`, the global scale parameter is set to 1, and the inverse-tuning parameter is given a χ^2 prior with 1 degree of freedom. Its value is thus optimized during model estimation.

The implementation of the horseshoe prior is based on the regularized horseshoe proposed by Piironen and Vehtari (2017b):

$$\begin{aligned}
&\beta_j \sim N(0, \tilde{\tau}_j^2 \lambda), \text{ with } \tilde{\tau}_j^2 = \frac{c^2 \tau_j^2}{c^2 + \lambda^2 \tau_j^2} \\
&\lambda \sim \text{student-}t^+(\nu_1, 0, \lambda_0^2) \\
&\tau_j \sim \text{student-}t^+(\nu_2, 0, 1) \\
&c^2 \sim \Gamma^{-1}(\frac{\nu_3}{2}, \frac{\nu_3 s^2}{2})
\end{aligned}$$

where N denotes the normal distribution, student-t^+ denotes the half-t distribution and Γ^{-1} denotes the inverse Gamma distribution. This extension of the horseshoe is more numerically stable in certain cases. In this formula, λ_0^2 is a global scale parameter that affects the overall shrinkage of the prior, with smaller values resulting in more shrinkage. The default value in `brma()` is 1. However, if prior information regarding the number of relevant moderators is available, it is best to include this information. This is accomplished by setting $\lambda_0^2 = \frac{p_0}{p-p_0}\frac{\sigma}{\sqrt{n}}$, where p_0 represents the number of relevant moderators, p the total number of moderators, σ is the residual standard deviation and n equals the number of observations. An alternative, user-friendly way to accomplish this is by setting the argument `relevant_pars` equal to the expected number of relevant moderators. The thickness of the tails is controlled by two degrees of freedom parameters, ν_1 and ν_2, which default to 1 in `brma()`. Increasing these degrees of freedom parameters results in a prior with lighter tails, which is, strictly speaking, no longer a horseshoe prior. However, in cases where the model is weakly identified, for example when there are more moderators than observations, these lighter tails can aid model convergence. The regularized horseshoe differs from the standard horseshoe in the specification of a finite “slab.” This slab ensures at least some regularization of large coefficients and as a consequence, more stable results. This slab is governed by a degrees of freedom parameter (ν_3, set to 4) and a scale parameter (s, set to 1).

Default settings for these hyperparameters in `brma()` were chosen such that the values are reasonable in most applications. However, it is good practice to perform a prior sensitivity analysis to compare the effect of different hyperparameters on the model results. This is particularly important when the sample is small, as the prior is more influential in this case.

Unlike the frequentist LASSO algorithm, Bayesian regularized estimation does not shrink coefficients to be exactly equal to zero. Therefore, variables must be selected post-estimation. One way to do so is by the use of probability intervals, the Bayesian counterpart of confidence intervals, with a moderator being selected if, for example, a 95% interval excludes zero. The present study considers two types of intervals: The credible interval, which is obtained by taking the 2.5% and 97.5% quantiles of the posterior distribution, and the highest posterior density interval, which is the narrowest possible interval that contains 95% of the probability mass.

Standardizing predictors. Penalized regression analyses typically require the scales of predictors to be equivalent (Tibshirani, 1996). This is because the regularization penalizes coefficients without regard for their scale. If variable scales differ, this can lead to an imbalanced penalization of coefficients that does not reflect differences in variable importance (Lee, 2015). To clarify, a regression parameter β can be interpreted as the expected increase in outcome y for a one unit increase in predictor x. If the scale of predictor x is increased by a factor 10, its regression coefficient is reduced by a factor 10. Standardization is a widely used method for equalizing predictor scales, in which the mean of all predictors is set to 0 and their standard deviation is set to 1 (Gelman, 2008). By default, this type of standardization is used in the `brma()` function. The estimated parameters are restored to their original scales. For the intercept, the transformation is:

$$b_0 = b_{0Z} - \mathbf{b}_Z \frac{\bar{\mathbf{x}}}{\mathbf{s}_X}$$

where b_0 is the intercept, b_{0Z} is the intercept for the standardized predictors, $\bar{\mathbf{x}}$ and $\mathbf{s}_x$ are the vectors of predictor means and variances, and $\mathbf{b_Z}$ is the vector of regression coefficients for the standardized predictors. The regression coefficients are returned to their original scale by applying:

$$\mathbf{b}_x = \frac{\mathbf{b}_z}{\mathbf{s}_x}$$

It is not always necessary or desirable to standardize predictors, however. For example, if predictors are already standardized or on a unified scale for different reasons. In these cases standardization does not make scales more equal, nor the penalization more fair, and the default standardization in `brma()` can be disabled.

There are additional considerations regarding standardization of binary and dummy predictors (Alkharusi, 2012). Some suggest to always standardize binary predictors (Tibshirani, 1997). This makes that, irrespective of initial scaling, the binary predictor will be on the same scale as the continuous standardized predictors (Gelman, 2008). However, standardizing binary predictors may decrease model interpretability (Wissmann, Toutenburg, et al., 2007). To illustrate this point, consider bivariate regression with a single binary predictor x that takes on values 0 and 1 predicting outcome y. The intercept represents the expected value of y when $x == 0$, and the regression coefficient represents the difference in the expected value of y between the two conditions (Alkharusi, 2012). By standardizing this binary predictor, the reference value is no longer zero, and both the intercept and its regression coefficient have no clear interpretation anymore, especially in multivariate cases (Wissmann et al., 2007).

The default in `brma()` is to use dummy coding for categorical predictors and standardize the dummies. Other coding schemes for categorical predictors exist that are equivalent in OLS regression. In penalized regression, by contrast, the choice of coding does affect model fit and interpretation due to the introduction of bias (Chiquet, Grandvalet, & Rigaill, 2016; Detmer, Cebral, & Slawski, 2020). Although the `brma()` function allows users to specify alternative coding schemes and standardization options, be advised that these decisions do affect model accuracy and interpretability in penalized regression (see Chiquet et al., 2016; Detmer et al., 2020).

There are two ways to circumvent the default standardization in `brma()`. The first is to disable standardization entirely, analyzing predictors in their original scale, by setting `standardize = FALSE`. Alternatively, `brma()` allows custom standardization. To use this option, first manually standardize (some of) the predictors. Then, when calling `brma()`, pass a vector of means and a vector of standard deviations to restore the coefficients to the predictors' original scale. This can be accomplished using the argument `standardize = list(center = meanvector, scale = sdvector)`. For predictors that **should not** be standardized, pass a mean of 0 and a standard deviation of 1; this leaves the coefficient in question unaffected.

Intercepts. The standard linear model estimates an intercept, which reflects the expected value of the outcome when all predictors are equal to zero, and regression coefficients for the effect of moderators. In some cases, it may be desirable to omit the intercept. For example, if an analysis contains categorical predictors, these can be encoded as dummy variables, with values $x \in \{0, 1\}$. For a variable with c categories, the number of dummy variables must be equal to $c - 1$; the omitted category functions as a reference category, and its expected value is represented by the model intercept b_0. This so-called *regression specification* of a model may be useful when there is a meaningful reference category. For example, imagine a study on the effectiveness of interventions for specific phobia with two interventions: Treatment as usual, and a novel intervention. In this case, it might make sense to code treatment as usual as the reference category, and dummy-code the new contender. The model will then estimate whether the newly developed intervention has an effect size significantly lower or higher than the industry standard. In other cases, there may not be a straightforward reference category. For example, imagine a study on the effectiveness of one intervention for specific phobia in two continents. In such cases, the average effect in both continents may be estimated by omitting the intercept, and including all c dummy variables. This so-called *ANOVA specification* of a model estimates a mean for all dummy-coded categories. In BRMA, as in other R functions, one can use ANOVA specification by explicitly removing the intercept from the model formula; for example, if `yi` is the effect size and C a categorical moderator, regression specification with $c - 1$ dummies is specified as `yi ~ C`, and ANOVA specification with c dummies is specified as `yi ~ -1 + C`.

Simulation study

The present study set out to validate the BRMA algorithm using a simulation study. As a benchmark for comparison, we used restricted maximum likelihood meta-regression, which is the standard in the field. We evaluated the algorithms' predictive performance in new data, and their ability to recover population parameters. Our research questions are whether BRMA offers a performance advantage over RMA in terms of any of these indicators, and which prior (regularized horseshoe versus LASSO) is to be preferred. All analysis code is available in a version-controlled repository at https://github.com/cjvanlissa/pema.

Performance indicators

Predictive performance reflects how well the algorithm is able to predict data not used to estimate the model parameters, in other words, it indicates the generalizability of the model. To compute it, for each iteration of the simulation both a training dataset and a testing dataset are generated. The model is estimated on the training data, which has a varying number of cases according to the simulation conditions. Predictive performance is then operationalized as the explained variance in the testing data, R^2_{test}. The testing data has 100 cases in all simulation conditions. The R^2_{test} reflects the fraction of variance in the testing data explained by the model, relative to the mean. Note that the mean of the training data, not of the testing data, is used as a benchmark. The resulting metric R^2_{test} is expressed by the following equation:

$$R^2_{test} = 1 - \frac{\sum_{i=1}^{k}(y_{i-test} - \hat{y}_{i-test})^2}{\sum_{i=1}^{k}(y_{i-test} - \bar{y}_{train})^2}$$

With k being the number of studies in the testing dataset, $\hat{y}_{i-test}$ being the predicted effect size for study i, and $\bar{y}_{train}$ being the mean of the training dataset.

The algorithms' ability to perform variable selection was evaluated by sensitivity and specificity. Sensitivity P is the ability to select true positives, or the probability that a variable is selected, $S = 1$, given that it has a non-zero population effect: $P = p(S = 1||\beta| > 0)$. Specificity is the ability to identify true negatives, or the probability that a variable is not selected given that it has a zero population effect: $N = p(S = 0|\beta = 0)$.

The ability to recover population parameters β and τ^2 was examined in terms of bias and variance of these estimates. The bias is given by the mean deviation of the estimate from the population value, and the variance is given by the variance of this deviation.

Design factors

To examine performance in a range of realistic meta-analysis scenarios, several design factors were manipulated: The number of studies in the training data $k \in (20, 40, 100)$, the average within-study sample size $\bar{n} \in (40, 80, 160)$, the population effect size of relevant moderators $\beta \in (0, .2, .5, .8)$, the number of moderators $p \in (2, 3, 6)$, and residual heterogeneity $\tau^2 \in (.01, .04, .1)$. According to a review of 705 published psychological meta-analyses (Van Erp et al., 2017), these values of τ^2 fall within the range observed in practice. Note that both BRMA and RMA assume linear effects. To test the robustness of the algorithms to violations of this assumption, true effect sizes were simulated using two models: one with a linear effect of one moderator, $T_i = \beta x_{1i} + \epsilon_i$, and one with a non-linear (cubic) effect of one moderator, $T_i = \beta x_{1i} + \beta x_{1i}^2 + \beta x_{1i}^3 + \epsilon_i$, where $\epsilon_i \sim N(0, \tau^2)$. The algorithms further assume normality of residuals. To examine robustness of the algorithms to violations of this assumption, moderator variables were simulated as skewed normal moderators, with scale parameter $\omega \in (0, 2, 10)$, where $\omega = 0$ corresponds to the standard normal distribution. The design factors combined to produce 1944 unique conditions. For all simulation conditions, 100 data sets were generated. In each data set, the observed effect size y_i was simulated as a standardized mean difference (SMD), sampled from a non-central t-distribution.

Results

Any iterative algorithm is susceptible to convergence problems. In such cases, the BRMA algorithms provide warning messages, but still return samples from the posterior. We were thus able to use all iterations of the BRMA algorithms, although there may be some that failed to converge, which will likely have poor performance. When the RMA algorithm fails to converge, however, it terminates with an error. To handle this contingency, we automated some of the steps recommended on the `metafor` website. Nevertheless, 10 replications of the RMA algorithm failed to converge. All of these were characterized by low number of cases ($k \leq 40$) and high effect sizes $\beta \geq .5$. These cases were omitted from further analysis.

Predictive performance

Within data sets, the BRMA with a horseshoe prior had the highest predictive performance 50% of the time, followed by RMA, 37%, and finally BRMA with a LASSO prior, 13%. Results indicated that the overall R^2_{test} was highest for BRMA with a horseshoe prior and lowest for RMA, see 1. This difference was driven in part by the fact that explained variance was somewhat higher for the BRMA models when the true effect was non-zero (i.e., in the presence of a population effect), and by the fact that RMA had larger negative explained variance when the true effect was equal to zero (i.e., there was no population effect to detect).

The effect of the design factors on R^2_{test} was evaluated using ANOVAs. Note that p-values are likely not informative due to the large sample size and violation of the assumptions of normality and homoscedasticity. The results should therefore be interpreted as descriptive, not inferential, statistics. Table 2 reports the effect size η^2 of simulation conditions on R^2_{test}.

To test our research questions, we computed interactions of algorithm (HS vs. LASSO, HS vs. RMA and LASSO vs. RMA) with the other design factors. The η^2 of these differences between algorithms are also displayed in Table 2. Note that η^2 for the comparison between HS and LASSO was zero in the second decimal for all conditions; thus, this comparison was omitted from the Table. The effect of design factors by algorithm is displayed in Figure 1; these plots have been ranked from largest difference between BRMA and RMA to smallest. Results indicate that the largest differences between algorithms were due to the effect size β, number of irrelevant moderators M, and the number of cases in the training data k. Evidently, predictive performance increased most for the HS algorithm when the effect size increased above zero. As noted previously, predictive performance of RMA was most negative (negative explained variance) when the effect size was zero. The HS algorithm furthermore had the consistently highest predictive performance regardless of number of irrelevant moderators or number of cases in the training data, and was relatively less affected by increases in the number of irrelevant moderators (panel b) or in the number of training cases (panel c). Conversely, RMA had relatively poor predictive performance on average, and was more responsive to increases in the number of training cases and irrelevant moderators.

Variable selection

To determine the extent to which the algorithms could perform variable selection correctly, the sensitivity to true positives P and specificity to true negatives N were calculated. Only simulation conditions with $\beta > 0$ were used, such that the effect of the first moderator was always positive in the population and could be used to calculate P, and the effect of the second moderator was always zero in the population and could be used to calculate N. Additionally, overall accuracy can be computed, which reflects the trade off between sensitivity and specificity. As the base rate of true positives and true negatives is equal in this simulation, overall accuracy is simply given by $Acc = (P + N)/2$.

As the regularized algorithms shrink all coefficients towards zero, it is unsurprising that sensitivity was highest for the un-regularized algorithm RMA, followed by HS and LASSO, $P_{RMA} = 0.95$, $P_{HS} = 0.91$, $P_{LASSO} = 0.89$. By contrast, specificity was higher for the regularized algorithms, $N_{HS} = 0.98$, $N_{LASSO} = 0.97$, $N_{RMA} = 0.94$. Overall accuracy was approximately equal for RMA and HS, and was lower for LASSO, $Acc_{RMA} = 0.95$, $Acc_{HS} = 0.95$, $Acc_{LASSO} = 0.93$.

Cramer's V, an effect size for categorical variables, was used to examine the effect of design factors on sensitivity (Table 3, Figure 2) and specificity (Table 4, Figure 3). We also computed this effect size for the difference between algorithms in the number of true positives by design factor.

Differences in sensitivity between the algorithms were near-zero for HS and LASSO. The difference between the two BRMA algorithms and RMA were largest for the design factor effect size β, followed by the model and number of studies k. Across all design factors, RMA had the highest sensitivity, followed by HS and then LASSO.

For specificity, differences in sensitivity between HS and LASSO were largest for the number of noise moderators M, followed by the effect size β, number of studies k, and residual heterogeneity τ^2. The difference between the two BRMA algorithms and RMA were largest for the design factor number of studies k, followed by the model, the number of noise moderators M, and the effect size β. Across all design factors, HS had the highest specificity, followed by LASSO and then RMA. Also note that the association between design factors and specificity was not monotonously positive or negative across algorithms. Instead, some design factors had opposite effects for the two BRMA algorithms versus RMA. For instance, a larger number of studies k had a negative effect on specificity for the BRMA algorithms, but a positive effect for RMA - within the context that RMA had lower specificity on average. Conversely, a greater number of noise moderators M had a positive effect on specificity for BRMA, but a negative effect for RMA.

Ability to recover population parameters

The ability to recover population parameters β and τ^2 was examined in terms of bias and variance of these estimates. If the value of the regression coefficient as estimated by one of the algorithms is $\hat{b}$, then the bias B and variance V of this estimate can be computed as the mean and variance of the difference between $\hat{b}$ and β across simulation conditions, respectively. Across all simulation conditions, HS had the lowest bias for τ^2, $B_{HS} = 0.38$, followed by RMA, $B_{RMA} = 0.39$, and then LASSO, $B_{LASSO} = 0.39$. Note that all algorithms yielded positively biased estimates. The LASSO estimates of τ^2 had the lowest variance, $V_{LASSO} = 1.47$, followed by HS, $V_{HS} = 1.50$, and then RMA, $B_{RMA} = 1.71$. The effect of the design factors on the bias in τ^2 was evaluated using ANOVAs. Table 5 reports the effect size η^2 of simulation conditions on $\hat{t^2} - \tau^2$. The design factors β and model had the largest effect on bias in estimated τ^2 for all algorithms. No differences between algorithms in the effect of design factors were observed.

For the estimated regression coefficient, HS had the greatest (negative) bias across simulation conditions, $B_{HS} = -0.07$, followed by LASSO, $B_{LASSO} = -0.06$, and then RMA, $B_{RMA} = -0.01$. Note that all algorithms - including RMA - provided, on average, negatively biased estimates. Across simulation conditions, HS had the lowest variance, $V_{HS} = 0.32$, followed by LASSO, $B_{LASSO} = 0.34$, and then RMA, $B_{RMA} = 0.38$. The effect of the design factors on the bias in estimated β was evaluated using ANOVAs. Table 6 reports the effect size η^2 of simulation conditions on $\hat{b} - \beta$. The skewness of moderator variables had the largest effect on bias in estimated β for all algorithms. Note, however, that this is likely due to the fact that the data simulated with a cubic model are analyzed with a linear model, and thus,

was the estimated model. This was mainly because the algorithms overestimated τ^2 most when the model contained cubic terms. No differences between algorithms in the effect of design factors were observed.

Applied example

In this application, we will work with the `pema::bonapersona` data (Bonapersona et al., 2019). This meta-analysis of over 400 experiments investigated the effects of early life adversity on cognitive performance in rodents. This example uses a small subset of the more than 30 moderators. See the `pema` package documentation (help and vignettes) for further examples.

Our simulation study shows good performance with default hyperparameters. However, experienced users may want to customize the prior. Visualizing the prior can be helpful in this process. This is accomplished using the interactive application visualization application available through `shiny_prior()`. The user can plot the prior distributions resulting from different sets of hyperparameters and compare them. Increasing the values of the scale parameters (`scale_global` and `hs_scale_slab`) results in a more spread out prior, which applies less regularization. Increasing the degrees of freedom (`df_global` and `df_slab`) results in thinner tails, which applies more regularization.

As no prior is specified, this example uses a horseshoe prior with default hyperparameters. To see the default values, open the function documentation using `?brma`.

```r
fit <- brma(yi ~ ., data = df, vi = "vi")
```

By running `summary(fit)`, we obtain the posterior mean, standard deviation, and quantiles of the model parameters (see Table 7). Use the posterior mean or median (50% quantile) and 95% credible interval (2.5% - 97.5%) to perform inference on model parameters. Parameters whose 95% credible interval excludes zero are marked with an asterisk. Note that Bayesian analyses do not use the frequentist notion of significance. Instead, we say that there is a 95% probability that the true population parameter lies within the interval, given the prior and observed data. In this example, however, there are no moderators for which the 95% CI excludes zero.

Many additional convenience functions exist for `rstan` models, which become available by converting a `brma` model object to a `stanfit` object, using the function `as.stan(fit)`. This makes it possible to plot the model parameters instead of tabulating them, using the `plot()` function. For example, one can obtain posterior density plots for parameters using `plot(as.stan(fit), plotfun = "dens", pars = c("Intercept", "year"))`.

It is good practice to assess model convergence. For example, the analysis above returns a warning about "divergent transitions." Converting to a `stanfit` object also facilitates convergence diagnostics; for example, using the function `check_hmc_diagnostics(as.stan(fit))`. Additionally, the MCMC draws can be visualized using `traceplot(as.stan(fit), pars = c("Intercept", "year"))`. The traces of a converged model look like "fat caterpillars," with the different MCMC chains mixing together.

The model summary also offers convergence diagnostics. For example, the column `Rhat` provides information on the split $\hat{R}$, a version of the potential scale reduction factor (PSRF, Gelman & Rubin, 1992). Values close to 1 indicate convergence. In addition, the column `n_eff` provides information on the number of effective (independent) MCMC samples, which should be high relative to the total number of samples (in this case, 4000). In this example, all `Rhat` values are close to 1. The effective number of MCMC samples is relatively small compared to the total number of MCMC samples. An often used heuristic is to consider ratios smaller than 0.1 as problematic. Both statistics indicate convergence in this example.

As mentioned before, this analysis results in a warning message about divergent transitions. Divergent transitions can result in biased estimates. However, the posterior distribution is often good enough to safely interpret the results if the number of divergences is small and there are no further indications of non-convergence. In some cases, divergent transitions may be resolved by increasing the degrees of freedom of the prior. Increasing both `df_global` and `df_slab` to 5 results in fewer divergences for this example, but does not otherwise influence the substantive interpretation of the results. It is prudent to perform similar sensitivity analyses to determine whether results are robust to different priors.

Discussion

This study presented a novel algorithm to select relevant moderators that can explain heterogeneity in meta-analyses, using Bayesian shrinkage priors. The simulation study validated the performance of two versions of the new BRMA algorithm, relative to state-of-the-art meta-regression (RMA). Our analyses examined the algorithms' predictive performance, which is a measure of generalizability, their ability to perform variable selection, and their ability to recover population parameters. Our research questions were whether BRMA offers a performance advantage over RMA in terms of any of these indicators, and which prior (horseshoe versus LASSO) is to be preferred.

Results indicated that the BRMA algorithms had higher predictive performance than RMA in the presence of relevant moderators. In the absence of relevant moderators, RMA produced overfit models; in other words, its models generalized poorly to new data. The predictive performance of the BRMA algorithms also suffered less than that of RMA in the presence of more irrelevant moderators. The BRMA algorithms were also more efficient, in the sense that they achieved greater predictive performance when the number of studies in the training data was low. Across all conditions, BRMA with a horseshoe prior achieved the highest average predictive performance, and within each data set, BRMA with a horseshoe prior most often had the best predictive performance (in 50% of replications). Based on these findings, we would recommend using BRMA with a horseshoe prior when the goal is to obtain findings that generalize to new data.

With regard to variable selection, results indicated that the penalized BRMA algorithms had lower sensitivity: they were less able to select relevant moderators than RMA. Conversely, the BRMA algorithms had better specificity: they were better able to reject irrelevant moderators than RMA. These results are unsurprising because the BRMA algorithms shrink all regression coefficients towards zero. This diminishes their ability to detect true effects and aids their ability to reject irrelevant moderators. Importantly however, the overall accuracy was approximately equal for RMA and BRMA with a horseshoe prior. This means that the total number of Type I and Type II errors will be approximately the same when choosing between these two methods - but there is a tradeoff between sensitivity and specificity. Applied researchers must consider whether sensitivity or specificity is more important in the context of their research. When meta-analyzing a heterogeneous body of literature, with many between-study differences that could be coded as moderators, BRMA may be preferred due to its greater ability to retain only relevant moderators. Conversely, when meta-analyzing a highly curated body of literature with a small number of theoretically relevant moderators, un-penalized RMA might be preferred.

With regard to the algorithms' ability to recover population effect sizes of moderators, we observed that BRMA with a horseshoe prior had the greatest bias towards zero across simulation conditions, followed by LASSO, and then RMA. Note that all algorithms provided, on average, negatively biased estimates. The variance of the estimates followed the opposite pattern. This illustrates the bias-variance trade-off, of which the BRMA algorithms' greater predictive performance is a direct consequence.

With regard to residual heterogeneity, we observed that BRMA with a horseshoe prior had the lowest bias. The BRMA algorithms also had lower variance. This suggests that the penalized regression coefficients do not compromise the estimation of residual heterogeneity. Future research might investigate under what conditions residual heterogeneity is estimated more accurately in a penalized model than in an unpenalized model. Together, these results suggest that BRMA has superior predictive performance and specificity, and provides relatively unbiased estimates of residual heterogeneity, relative to RMA.

We examined the effect of several violations of model assumptions, including simulating data from a cubic model. In applied research, it is often not known what the true shape of the association between a moderator and effect size is. Thus, model misspecification is likely to occur. One advantage of BRMA is that it can accommodate more moderators than RMA and has superior specificity. This allows researchers to specify a more flexible model to account for potential misspecification, with less concern for overfitting and nonconvergence. For example, researchers could add polynomials of continuous variables with suspected non-linear effects, or interactions between predictors. If nothing is known about the shape of the associations between moderators and effect size, non-parametric methods like random forest meta-analysis may be preferable over linear models (Van Lissa, 2020).

Strengths and future directions

The present paper has several strengths. First, we included a wide range of simulation conditions, including conditions that violated the assumptions of linearity and normality. Across all conditions, BRMA displayed superior predictive performance and specificity compared to RMA. Another strength is that the present simulation study used realistic estimates of τ^2, based on data from 705 published psychological meta-analyses (Van Erp et al., 2017). Another strength is that the BRMA algorithms have been made available in a FAIR (Findable, Accessible, Interoperable and Reusable) repository: an R-package published on the "Comprehensive R Archive Network." Thanks to the use of compiled code, the BRMA algorithm is computationally relatively inexpensive.

Several limitations remain to be addressed in future research, however. One limitation is that, by necessity, computational resources and journal space limit the number of conditions that could be considered in the simulation study. To facilitate further exploration and follow-up research, we have made all simulation data and analysis code for the present study available online. This code can also be used to conduct Monte Carlo power analyses for applied research. A second limitation is that the present study did not examine the effect of multicollinear predictors. Regularizing estimators typically have an advantage over OLS regression in the presence of multicollinearity; future research ought to examine whether this advantage extends to BRMA. A third limitation is that the present study did not examine the effect of dependent data (e.g., multiple effect sizes per study). The BRMA algorithm can accommodate dependent data by means of three-level multilevel analysis. To our knowledge, there are no reasons to expect that dependent data would result in a different pattern of findings than we found for independent data, but future research is required to ascertain this. A final limitation of the current implementation is that it relies on 95% credible intervals to select relevant moderators. However, these marginal credible intervals can behave differently compared to the joint credible intervals (Piironen, Betancourt, Simpson, & Vehtari, 2017). A future direction of research is therefore to implement more advanced selection procedures, such as projective predictive variable selection (Piironen & Vehtari, 2017a). Another direction for future research is the specification of different priors, aside from the horseshoe and LASSO priors that were examined in this study. A final disadvantage is that Bayesian estimation is typically more computationally expensive than frequentist estimation. One future direction of research is thus to develop a frequentist estimator for regularized meta-regression.

Recommendations for applied research

BRMA aims to address the challenge that arises when meta-analyzing heterogeneous bodies of literature, with few studies relative to the number of moderators. BRMA can be used to identify relevant moderators when it is not known beforehand which moderators are responsible for between-studies differences in observed effect sizes. To facilitate adoption of this method in applied research, we have published the function `brma()` in the R package `pema`. Here, we offer several recommendations for its use. The first recommendation precedes analysis, and relates to the design of the meta-analysis. When the search for moderators is exploratory, researchers ought to be inclusive, but focus on moderators that are expected to be relevant, including theoretically relevant moderators, as well as moderators pertaining to the sample, methods, instruments, study quality, and publication type. In our experience, many applied researchers code such study characteristics anyway, but omit them from their analyses for lack of statistical power. Moderators can be continuous or categorical. Missing data must be accounted for. The best way to do so is by retrieving the missing information, by contacting authors or comparing different publications on the same data. If missing data remains, users can either use a single imputation method or supply multiple imputed data to the `data` argument (see function documentation). The effect sizes and their variances must be computed using suitable methods; note that many such methods are available in the R package `metafor` (Viechtbauer et al., 2010). With regard to data analysis, we recommend the use of a horseshoe prior by default, because it demonstrated the best predictive performance and most attractive trade-off between sensitivity and specificity in our simulations. When estimating the model, it is important to ascertain that the algorithm has converged before interpreting the results. Stan, the computational back-end of `brma()`, returns warnings and errors if there are any indications of non-convergence. Additionally, users can obtain trace plots as described in the illustrative example.

When reporting results, researchers should substantiate their decision to explore heterogeneity on both subjective and objective grounds. The former can be achieved by simply ascertaining that the body of literature to be meta-analyzed appears to be heterogeneous; the same rationale commonly used to support the use of random-effects meta-analysis (Higgins et al., 2009). The latter can be accomplished by conducting a random-effects meta-analysis without any moderators, and reporting the estimated τ^2. Note that significant heterogeneity does not constitute sufficient grounds, for deciding to explore ignore heterogeneity, for two reasons: Firstly, because data-driven decisions render any analysis (partly) exploratory, and increase the risk of results that generalize poorly (i.e., are overfit). The second reason is that tests for heterogeneity are often underpowered when the number of studies is low, and overpowered when it is high, thus limiting their usefulness (see Higgins & Thompson, 2002). As when conducting RMA meta-analysis, researchers should report both the estimated effect of moderators and residual heterogeneity. Regression coefficients can be interpreted as usual, but it is recommended that researchers acknowledge that they are biased towards zero. If all moderators are centered, the model intercept can be interpreted as the overall effect size at average levels of the moderators. Note that, as BRMA is a Bayesian method, credible intervals or highest posterior density intervals should be used for inference, instead of p-values. The null hypothesis is rejected if such intervals exclude zero. As both types of intervals performed identically in the present study, we suggest using credible intervals, which are computationally less expensive.

Finally, with regard to publication, we highly recommend making the data and code for the meta-analysis publicly available. One way to do this is by creating a reproducible research repository, for example, using the Workflow for Reproducible Code in Science (WORCS, Van Lissa et al., 2020). Transparency allows readers and reviewers to verify that methods were correctly applied, which should inspire greater confidence in the results. Others can easily perform sensitivity analyses by changing the analysis code. Sharing data allows the meta-analysis to be updated in the future, which increases the reuse value of the data. Finally, sharing the model object (or code to reproduce it) allows others to obtain predictions for the expected effect size of a new study on the same topic. This prediction can be used to conduct power analysis for future research. To this end, researchers can simply enter their planned design (or several alternative designs) as new lines of data, using the codebook of the original meta-analysis, and use the published BRMA model to calculate the predicted effect size for a study with these specifications.

BRMA may not be the best solution for every situation. Several trade-offs must be considered to decide what method is most appropriate. Firstly, the fact that BRMA has high predictive performance compared to RMA suggests that it is particularly suitable when a researcher intends to obtain results that will generalize beyond the sample at hand, and is willing to accept some bias in parameter estimates. Conversely, RMA might be more suitable when the goal is to describe the sample at hand in an unbiased manner, with less concern for generalizability to future studies. Secondly, the fact that BRMA has high specificity compared to RMA suggests that it is more suitable when a researcher seeks to eliminate irrelevant moderators at the cost of increasing the Type II error rate. Conversely, RMA might be more suitable when the researcher seeks to identify relevant moderators, at the cost of increasing the Type I error rate. If many moderators have been coded, and many of them are expected to be irrelevant, then BRMA may thus be preferable. Thirdly, there may be pragmatic reasons for preferring BRMA over RMA. For example, if a dataset is small, or the number of moderators is high relative to the number of cases, RMA models may be empirically under-identified. This can be indicated by convergence problems. In such cases, Bayesian estimation may converge on a solution where frequentist estimation does not (Kohli, Hughes, Wang, Zopluoglu, & Davison, 2015). Similarly, BRMA may perform better in the presence of multicollinearity among predictors, which can be examined using the function `vif()` in the R-package `metafor`. Values exceeding 5 are cause for concern. Multicollinearity increases the variance of regression coefficients. BRMA may have an advantage here, because the regularizing priors restrict variance. If multicollinearity is observed, researchers might thus prefer BRMA over RMA.

Conclusion

The present research has demonstrated that BRMA is a powerful tool for exploring heterogeneity in meta-analysis, with a number of advantages over classic RMA. BRMA had better predictive performance than RMA, which indicates that results from BRMA analysis generalize better to new data. This predictive performance advantage was especially pronounced when training data were as small as 20 studies. This is appealing because many meta-analyses have small sample sizes. BRMA further has greater specificity in rejecting irrelevant moderators from a larger set of potential candidates, while maintaining an overall variable selection accuracy equivalent to RMA. Although the estimated regression coefficients are biased towards zero by design, the estimated residual heterogeneity did not show evidence of bias in our simulation. A final advantage of BRMA over other variable selection methods for meta-analysis is that it is an extension of the linear model. Most applied researchers are familiar with the linear model, and it can easily accommodate predictor variables of any measurement level, interaction terms, and non-linear effects. Adoption of this new method may be further facilitated by the availability of the user-friendly R package `pema`.

Highlights

- Many applied meta-analyses concern heterogeneous bodies of literature, with many between-studies differences (moderators).
- Simultaneously, meta-analytic samples are often small. There is thus limited statistical power to account for moderators.
- The present study introduces Bayesian Regularized Meta-Analysis (BRMA), an algorithm that applies regularization to identify relevant moderators from a larger number of candidates.
- The algorithm is made available in a user-friendly R-package, `pema`, which is published on CRAN.
- Readers across fields can use this method to account for between-studies heterogeneity in meta-analysis, without concern that models may be underfit or underpowered.

References

Alkharusi, H. (2012). Categorical variables in regression analysis: A comparison of dummy and effect coding. *International Journal of Education*, *4*(2), 202.

Bonapersona, V., Kentrop, J., Van Lissa, C., Van Der Veen, R., Joëls, M., & Sarabdjitsingh, R. (2019). The behavioral phenotype of early life adversity: A 3-level meta-analysis of rodent studies. *Neuroscience & Biobehavioral Reviews*, *102*, 299–307.

Carvalho, C. M., Polson, N. G., & Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, *97*(2), 465–480. https://doi.org/10.1093/biomet/asq017

Chiquet, J., Grandvalet, Y., & Rigaill, G. (2016). On coding effects in regularized categorical regression. *Statistical Modelling*, *16*(3), 228–237.

Detmer, F. J., Cebral, J., & Slawski, M. (2020). A note on coding and standardization of categorical variables in (sparse) group lasso regression. *Journal of Statistical Planning and Inference*, *206*, 1–11.

Erp, S. van, Oberski, D. L., & Mulder, J. (2019). Shrinkage priors for Bayesian penalized regression. *Journal of Mathematical Psychology*, *89*, 31–50. https://doi.org/10.1016/j.jmp.2018.12.004

Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, *27*(15), 2865–2873.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*(4), 457–472.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Second). New York: Springer.

Hedges, L. V., & Vevea, J. L. (1998). Fixed- and Random-effects Models in Meta-analysis. *Psychological Methods*, *3*(4), 486–504.

Higgins, J. P. T., Thompson, S. G., & Spiegelhalter, D. J. (2009). A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society. Series A, (Statistics in Society)*, *172*(1), 137–159. https://doi.org/10.1111/j.1467-985X.2008.00552.x

Jargowsky, P. A. (2004). The Ecological Fallacy. In K. Kempf-Leonard (Ed.), *The Encyclopedia of Social Measurement* (Vol. 1, pp. 715–722). San Diego, CA: Academic Press.

Kohli, N., Hughes, J., Wang, C., Zopluoglu, C., & Davison, M. (2015). Fitting a Linear-Linear Piecewise Growth Mixture Model With Unknown Knots: A Comparison of Two Common Approaches to Inference. *Psychological Methods*, *20*, 259–275.

Lee, S. (2015). A note on standardization in penalized regressions. *Journal of the Korean Data and Information Science Society*, *26*(2), 505–516.

López-López, J. A., Marín-Martínez, F., Sánchez-Meca, J., Van den Noortgate, W., & Viechtbauer, W. (2014). Estimation of the predictive power of the model in mixed-effects meta-regression: A simulation study. *British Journal of Mathematical and Statistical Psychology*, *67*(1), 30–48. https://doi.org/10.1111/bmsp.12002

Panityakul, T., Bumrungsup, C., & Knapp, G. (2013). On Estimating Residual Heterogeneity in Random-Effects Meta-Regression: A Comparative Study. *Journal of Statistical Theory and Applications*, *12*(3), 253–265. https://doi.org/10.2991/jsta.2013.12.3.4

Park, T., & Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, *103*(482), 681–686. https://doi.org/10.1198/016214508000000337

Piironen, J., Betancourt, M., Simpson, D., & Vehtari, A. (2017). Contributed comment on article by van der Pas, Szabó, and van der Vaart. *Bayesian Analysis*, *12*(4), 1264–1266.

Piironen, J., & Vehtari, A. (2017a). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, *27*(3), 711–735. https://doi.org/10.1007/s11222-016-9649-y

Piironen, J., & Vehtari, A. (2017b). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, *11*(2), 5018–5051. https://doi.org/10.1214/17-ejs1337si

Riley, R. D., Higgins, J. P. T., & Deeks, J. J. (2011). Interpretation of random effects meta-analyses. *BMJ*, *342*, d549. https://doi.org/10.1136/bmj.d549

Schooler, J. (2011). Unpublished results hide the decline effect. *Nature*, *470*(7335), 437–437. https://doi.org/10.1038/470437a

Stan Development Team. (2019). *Stan modeling language users guide and reference manual*, version 2.28.0. Retrieved from http://mc-stan.org

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Tibshirani, R. (1997). The lasso method for variable selection in the cox model. *Statistics in Medicine*, *16*(4), 385–395.

Van Lissa, C. J. (2020). Small sample meta-analyses: Exploring heterogeneity using MetaForest. In R. Van De Schoot & M. Miočević (Eds.), *Small Sample Size Solutions (Open Access): A Guide for Applied Researchers and Practitioners.* CRC Press.

Van Lissa, C. J., Brandmaier, A. M., Brinkman, L., Lamprecht, A.-L., Peikert, A., Struiksma, M. E., & Vreede, B. (2020). WORCS: A Workflow for Open Reproducible Code in Science. https://doi.org/10.17605/OSF.IO/ZCVBS

Viechtbauer, W.others. (2010). Conducting meta-analyses in R with the metafor package. *J Stat Softw*, *36*(3), 1–48.

Wissmann, M., Toutenburg, H.others. (2007). Role of categorical variables in multicollinearity in the linear regression model.

Table 1

Mean and SD of predictive R2 for BRMA with a horseshoe (HS) and LASSO prior, and for RMA, for models with a true effect (ES != 0) and without (ES = 0).

	$\bar{R^2}_{HS}$	CI_{95}	$\bar{R^2}_{LASSO}$	CI_{95}	$\bar{R^2}_{RMA}$	CI_{95}
Overall	0.42	[-0.03, 0.87]	0.42	[-0.01, 0.87]	0.39	[-0.30, 0.87]
ES = 0	0.57	[0.04, 0.89]	0.56	[0.03, 0.88]	0.55	[-0.01, 0.88]
ES != 0	-0.01	[-0.04, -0.00]	-0.01	[-0.02, 0.00]	-0.10	[-0.40, -0.01]

Table 2

Effect size of design factors on predictive R2 of the different algorithms, and of the difference between algorithms. Interpretation indicates whether a main effect was uniformly positive or negative across all algorithms.

Factor	HS	LASSO	RMA	HS vs. LASSO	HS vs. RMA	LASSO vs. RMA	Interpretation
ω	0.02	0.01	0.01	0.00	0.00	0.00	negative
β	0.77	0.76	0.70	0.00	0.01	0.02	positive
k	0.02	0.02	0.06	0.00	0.01	0.01	positive
n	0.05	0.05	0.02	0.00	0.00	0.00	positive
Model	0.17	0.17	0.11	0.00	0.00	0.00	positive
M	0.00	0.00	0.04	0.00	0.01	0.01	negative
τ^2	0.05	0.05	0.03	0.00	0.00	0.00	negative

Table 3

Effect size (Cramer's V) of design factors, and of the difference between algorithms, on sensitivity (P).

Factor	P_{HS}	P_{LASSO}	P_{RMA}	$P_{HSvs.LASSO}$	$P_{HSvs.RMA}$	$P_{LASSOvs.RMA}$	Interpretation
k	0.21	0.23	0.17	0.01	0.02	0.02	positive
n	0.08	0.09	0.07	0.00	0.01	0.01	positive
β	0.36	0.37	0.28	0.01	0.04	0.04	positive
τ^2	0.10	0.10	0.08	0.00	0.01	0.01	negative
ω	0.09	0.10	0.08	0.00	0.01	0.01	negative
M	0.05	0.05	0.02	0.00	0.01	0.01	negative
Model	0.31	0.33	0.22	0.01	0.03	0.03	positive

Table 4

Effect size (Cramer's V) of design factors, and of the difference between algorithms, on specificity (N).

Factor	N_{HS}	N_{LASSO}	N_{RMA}	$N_{HSvs.LASSO}$	$N_{HSvs.RMA}$	$N_{LASSOvs.RMA}$	Interpretation
k	0.02	0.03	0.02	0.03	0.13	0.13	other
n	0.00	0.01	0.00	0.01	0.02	0.02	other
β	0.01	0.02	0.01	0.03	0.06	0.06	other
τ^2	0.02	0.01	0.02	0.03	0.01	0.01	other
ω	0.00	0.01	0.00	0.01	0.02	0.02	other
M	0.04	0.03	0.01	0.11	0.08	0.08	other
Model	0.02	0.03	0.01	0.01	0.08	0.08	positive

Table 5

Effect size of design factors on bias in tau squared for the different algorithms, and of the difference between algorithms.

Factor	HS	LASSO	RMA	HS vs. LASSO	HS vs. RMA	LASSO vs. RMA
ω	0.01	0.01	0.00	0.00	0.00	0.00
β	0.12	0.13	0.11	0.00	0.00	0.00
k	0.00	0.00	0.00	0.00	0.00	0.00
n	0.01	0.01	0.01	0.00	0.00	0.00
Model	0.11	0.12	0.10	0.00	0.00	0.00
M	0.00	0.00	0.00	0.00	0.00	0.00
τ^2	0.00	0.00	0.00	0.00	0.00	0.00

Table 6

Effect size of design factors on bias in beta squared for the different algorithms, and of the difference between algorithms.

Factor	HS	LASSO	RMA	HS vs. LASSO	HS vs. RMA	LASSO vs. RMA
ω	0.16	0.15	0.15	0.00	0.00	0.00
β	0.01	0.00	0.00	0.00	0.00	0.00
k	0.00	0.00	0.00	0.00	0.00	0.00
n	0.02	0.02	0.01	0.00	0.00	0.00
Model	0.01	0.00	0.00	0.00	0.00	0.00
M	0.00	0.00	0.00	0.00	0.00	0.00
τ^2	0.00	0.00	0.00	0.00	0.00	0.00

Table 7

Summary of model parameters for the applied example.

	mean	sd	2.5%	50%	97.5%	n_eff	Rhat
Intercept	-27.64	16.83	-62.15	-27.70	1.13	1,069.48	1.00
mTimeLength	-0.02	0.03	-0.09	-0.01	0.03	861.82	1.00
year	0.06	0.04	0.00	0.06	0.14	1,069.83	1.00
modelLG	0.03	0.03	-0.02	0.02	0.09	623.20	1.01
modelLNB	0.05	0.04	-0.01	0.04	0.14	533.19	1.01
modelM	0.03	0.04	-0.02	0.02	0.11	525.35	1.01
modelMD	0.02	0.03	-0.04	0.01	0.10	428.66	1.01
ageWeek	-0.03	0.03	-0.11	-0.03	0.01	602.54	1.01

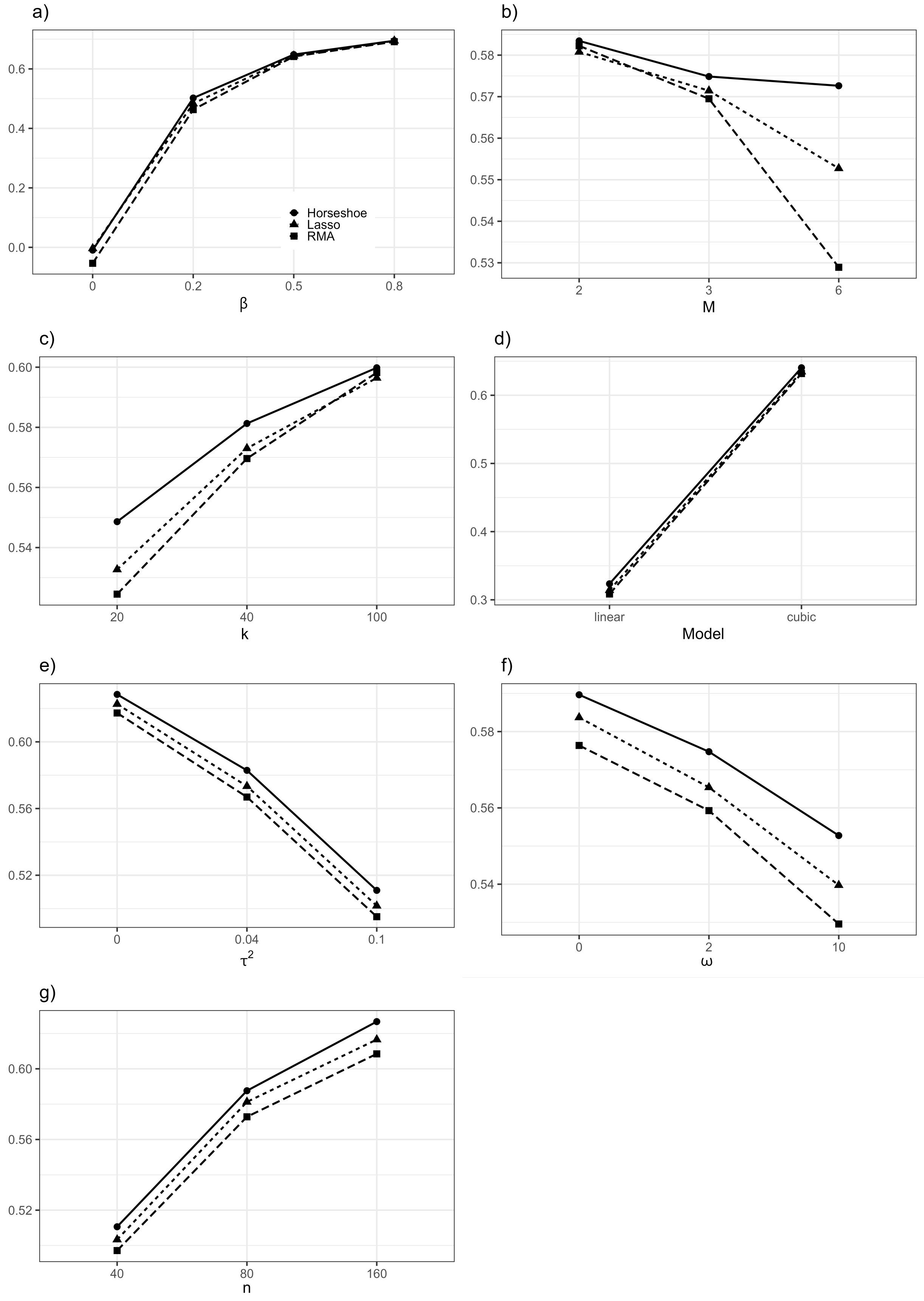

Figure 1. Predictive R2 for BRMA with horseshoe (HS) and LASSO prior, and RMA. Plots are sorted by largest performance difference between BRMA and RMA.

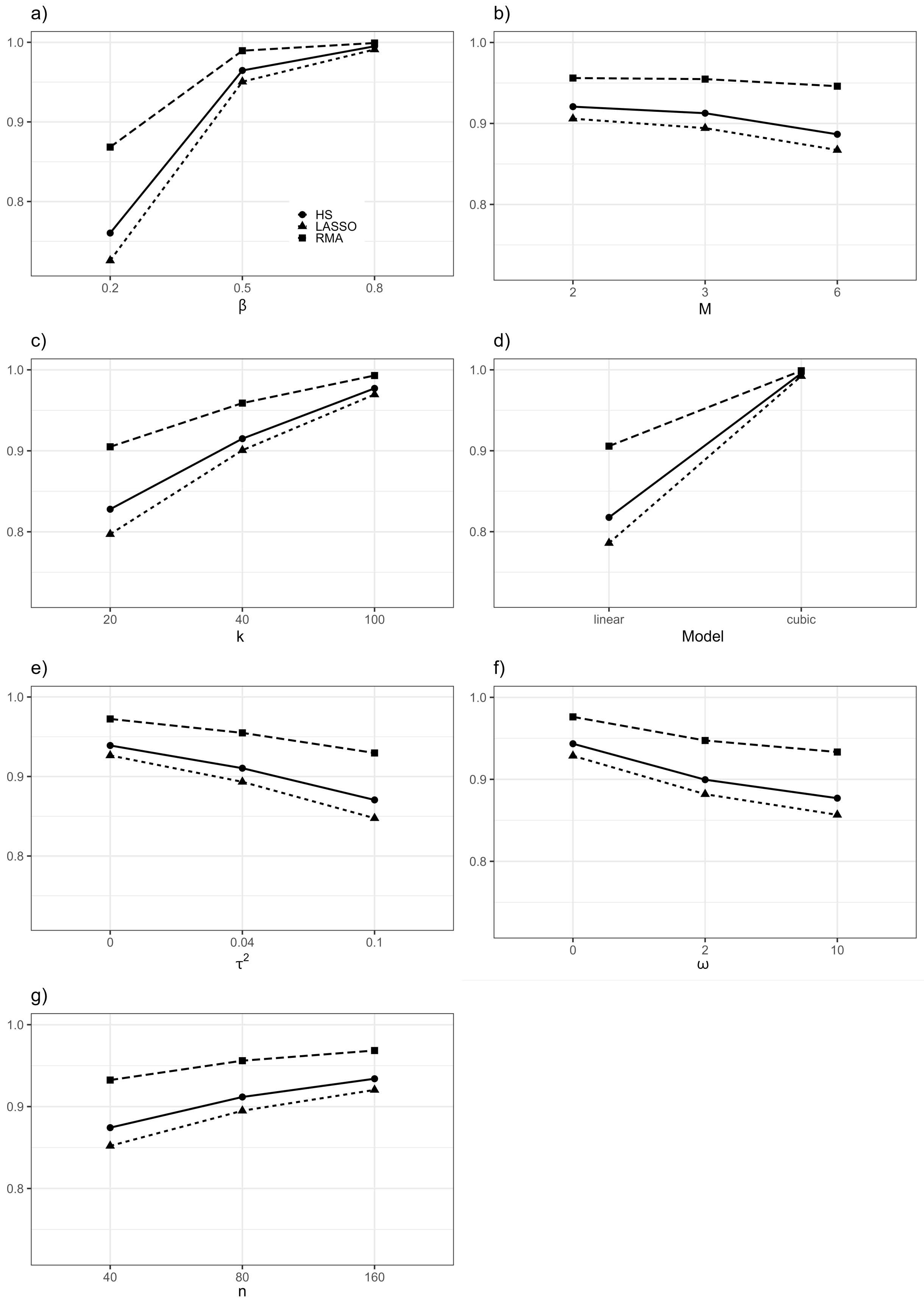

Figure 2. Sensitivity by design factors for the HS (circle, solid line), LASSO(triangle, dotted line) and RMA (square, dashed line) algorithms.

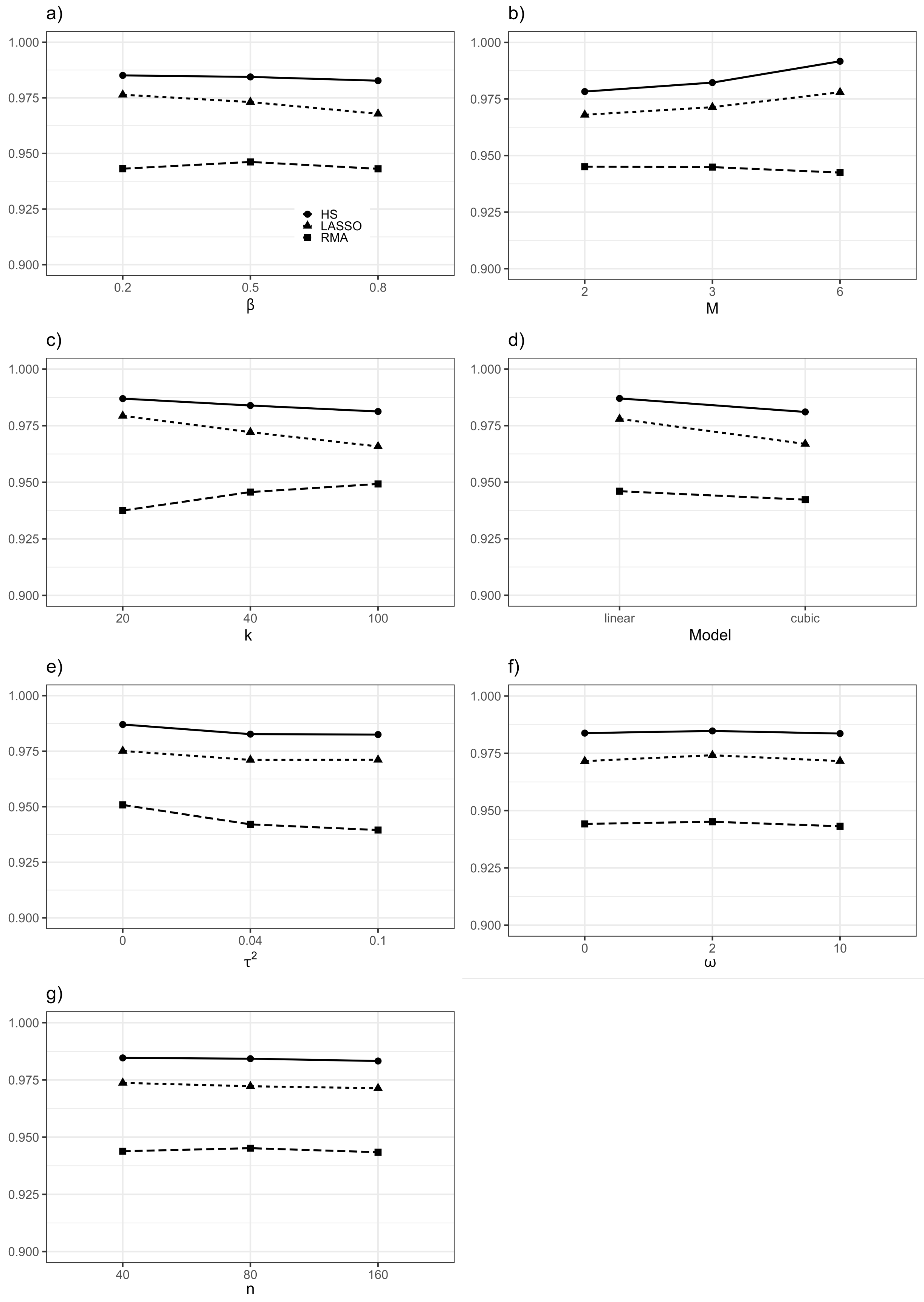

Figure 3. Specificity by design factors for the HS (circle, solid line), LASSO(triangle, dotted line) and RMA (square, dashed line) algorithms.