Stable Combinatorially-Optimized Features Selection via Sequential Monte Carlo

Jin-Chuan Duan*

(October 25, 2024)

Abstract

Selecting a subset out of many potential explanatory variables in linear regressions has long been the subject of research interest, and the matter becomes critical in the machine-learning era when models are preferably constructed by identifying critical features among fast-increasing variables gathered through digital footprints. In the literature, the l_1-norm penalty on the regression coefficients such as Lasso of Tibshirani (1996) have become very popular. However, the variable selection problem in its natural setting is a zero-norm penalty problem, i.e., a penalty on the number of variables as opposed to the l_1-norm of model coefficients. But selection with the zero-norm penalty is a highly demanding combinatory optimization task. Here we devise a sequential Monte Carlo optimization method for zero-norm variable selection that is practical (tasks typically completed under 10 minutes on a multi-core desktop computer) and reliable (assessable quality by the extreme-value theory-based inference). We demonstrate through a simulation study the method's reliability and superiority vis-a-vis the adaptive Lasso.

*Duan is with Criat, ADBIZA and the National University of Singapore and holds an adjunct professorship at National Chengchi University. E-mail: bizdjc@nus.edu.sg. The author thanks Yu-Hung Chien, Shuping Li, Kaican Kang and Qiqi Zou for their able research assistance.

(Note: The earlier version of this paper was entitled "Variable Selection with Big Data based on Zero Norm and via Sequential Monte Carlo," 2019.)

1 Introduction

Selecting a suitable subset of regressors in the context of linear regressions or other similarly structured variable selection problems plays a critical role for a wide range of practical issues when facing big data. Its importance is completely obvious so that we will skip the vast literature along many lines of research. Variable selection based on the zero-norm penalty (i.e., number of selected regressors) is conceptually more appealing than other criteria such as the l_1-norm penalty because it directly addresses the variable selection problem. Practically speaking, it also works better because regression coefficients are not distorted by the penalty term (i.e., shrinkage toward zero when being selected). Also interesting to note is the fact that the regression model fit, measured in R^2, is in invariant to linearly transforming a group of regressors, but the corresponding l_p $(0 < p < 2)$ penalty term is not.[1] Therefore, multicolinearity which naturally occurs in data will interfere with regressor selection based on an l_p $(0 < p < 2)$ penalty, but not with the zero-norm regressor selection. In this paper, we propose a practical sequential Monte Carlo solution to the zero-norm regressor selection problem, and demonstrate through a simulation study the distortion caused by Lasso.

The main idea of our zero-norm regressor selection method is to cast this variable selection problem as a pure combinatory optimization problem that is solvable with sequential Monte Carlo optimization (see Duan, *et al* (2022)). The main engine is the density-tempered sequential Monte Carlo (SMC) sampling technique by Del Moral, *et al* (2006), Duan and Fulop (2015), among others. Once a random combination of a fixed number of regressors is given, we rely on the closed-form linear regression solution to attach to the combination a likelihood of occurrence, which in turn defines a probability distribution function, up to a missing norming constant, over all possible combinations of the same fixed number of variables. By sequential probability-tempering, one will arrive at a final SMC sample of, say, 1,000 SMC particles, to mimick this target probability function whose maximum in turn provides the best combination. Our variable selection method is by nature of sampling from the entire distribution a global solution approach.

Our SMC sampling technique originates from the Bayesian literature, but our approach differs from a long line of research papers using the Bayesian statistical techniques. Typical approaches rely on a hierarchical structure whether any given regression coefficient is modeled by a spike-and-slab (i.e., Bernoulli and Gaussian) mixture and its variants, for example, Mitchell and Beauchamp (1988), George and McCulloch (1993) and Polson and Sun (2018), or adaptive sampling for variables' inclusion via some parametric distribution for binary variables as in Schafer and Chopin (2013). In a true Bayesian sense, these algorithms typically

[1] l_p for $0 \leq p < 1$ is actually not a norm, and zero-norm so-named by David Donoho, a special case of $p = 0$, obviously lacks homogeneity. However, zero-norm has become a standard way of describing such a penalty form in the data analytics community.

require a strong prior and some further distribution assumptions to work, and essentially alters the original variable selection problem. In contrast, our SMC variable selection method does not make any additional distributional assumption and solves the combinatory problem in its original form. Practically important is our method's ability to scale up for real big-data problems for which the number of potential regressors may run into hundreds of thousands.

We first show that selecting regressors subject to a zero-norm condition is equivalent to finding the maximum value of a distribution function defined over combinations of regressors at some fixed number of elements, say, k. Our target distribution function has no tractable analytical solution, but can be represented by a sample of k-dimensional points with each representing a k-combination out of all potential regressors. Since different orders of regressors for a given combination yields the same regression solution, permutation has no particular meaning above and beyond reducing to a combination. However, a distribution function defined over permutations are easier to sample and the distribution function defined over permutations can be proportionally scaled up to obtain the target distribution function defined over combinations.

Note that the SMC algorithm used in solving this maximization problem is by design free of any scaling constant. Hence, the SMC sample representing the distribution function over combinations is exactly the same as the one over permutations after collapsing permutations into combinations. On the methodological front, we devise a sensible but arbitrary initialization sampler to generate permutations and absorb its initialization distribution into the importance weight. Then, the progression of the algorithm is performed repeatedly through reweighting, resampling, and support-boosting steps to finally arrive at the SMC sample for the target distribution. For the Metropolis-Hastings move used in the support-boosting step, we also engage a proposal sampler defined over permutations for the same reason.

With a typical multi-core desktop computer, one can complete the fixed-number selection task (for example, choosing 12 variables out of 900) in about one minute. When the number of variables needs to be determined, the overall selection task (determining the right number and the optimal variable combination) can also be completed in under 10 minutes.

We conduct a comprehensive simulation study of selecting 9 or 18 variables out of 900 potential variables, and under which the number of potential combinations equals 1.026×10^{21} and 1.986×10^{37}, respectively. In the 500 simulation repetitions, the selected model under each scenario typically has a higher in-sample R^2 than that of the estimated true model, suggesting that the zero-norm SMC method has likely selected the best variable combination in a practical sense. The SMC method's performance improves when the observations are smaller and/or the residual errors are larger because sampling errors become larger and making room for the selected model to outperform the true model based on the in-sample R^2. Naturally, it also performs better when the number of selected variables is smaller,

i.e., 9 versus 18 because the zero-norm SMC method is less likely to miss the best in-sample solution. Finally, we ascertain the SMC method's capability to find the maximum by comparing the solution to a proxy grand maximum generated as the maximum over 500 randomly seeded SMC optimization runs on the same data sample.

We conduct another simulation study to check the relative performance of the zero-norm SMC method versus the Adaptive Lasso of Zou (2006). In this simulation study, the true model has 12 variables and the selection task is to choose the best combination among 900 potential variables with 200 observations while the number of variables is unknown to the analyst. Five-fold cross-validation is deployed to determine the number of variables in the final solution for both the zero-norm SMC method and the adaptive Lasso. We complete 500 simulation runs to document their performances. The results clearly indicate that the adaptive Lasso is prone to picking too many variables with an average of 53 variables when the true model has 12 by design, and the range is quite wide with the solutions from 13 to 142. In contrast, the zero-norm SMC method yields an average of 11 variables with the range from 7 to 14. The F-scores, a standard way of comparing models in machine learning, also suggest a far superior performance in favor of the zero-norm SMC method.

The zero-SMC algorithm can in theory find the right solution with a probability of one when the number of particles increases to infinity. Other than a theoretical interest, that would be no better than the brute-force approach of exhausting all possible combinations. Similar to using the Central Limit Theorem to assess the Monte Carlo solutions in many contexts, our SMC solution's R^2 is the maximum order statistic of the final SMC sample and can thus be appraised with an estimated Weibull distribution, a max-stable distribution limit implied by the Fisher-Tippett-Gnedenko Extreme Value Theorem. We show by simulation that the predictive R^2 distribution can help assess whether more SMC particles are needed to increase the solution's precision.

2 Linear regression subject to a zero-norm penalty

Consider the classical linear regression model of p regressors with n observations:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{y} = (y_1, \cdots, y_n)'$, and $\boldsymbol{X}$ denotes the n observations of p regressors, i.e., $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_p)$ with $\boldsymbol{x}_i = (x_1, \cdots, x_n)'$, of which the first vector may represent the intercept term. $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)'$ is the p-dimensional regression coefficients, and $\boldsymbol{\epsilon}$ is n-dimensional *i.i.d.* normally distributed errors with mean 0 and variance σ^2.

Sometimes, there are more potential regressors than data; that is, p is greater than n. Penalized regression is the only sensible way to estimate this regression. Even if there are

enough data points, in-sample over-fitting is still a general modeling concern and penalized regression can be very useful in dealing with the over-fitting problem. Multicollinearity is another commonly encountered situation in regressions, meaning that some of them are highly correlated. Trimming away some regressors seems to be a wise thing to do, and penalized regression is an obvious way to go. Whatever is the reason, the general issue boils down to selecting a subset of good regressors that delivers a robust and reliable performance.

The penalized regression considered in this paper is the one subject to the zero-norm regularization.

$$\begin{aligned} &\arg\min_{\boldsymbol{\beta}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_{l_2}^2 \\ &s.t. \;\; ||\boldsymbol{\beta}||_{l_0} \leq p_s \leq p \end{aligned} \tag{2}$$

where $||\cdot||_{l_2}$ and $||\cdot||_{l_0}$ stands for the l_2 and zero norms, respectively. Note that $||\boldsymbol{\beta}||_{l_0}$ counts the number of non-zero entries in $\boldsymbol{\beta}$. Note that the above minimization problem is equivalent to $\arg\min_{\boldsymbol{\beta}} \left\{||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_{l_2}^2 + \lambda||\boldsymbol{\beta}||_{l_0}\right\}$ where the solution is a step function of λ with the jumps corresponding to different values of k. This zero-norm penalized regression problem, albeit being natural for regressor selection, is known by Natarajan (1995) to be NP-hard. Thus, the Lasso technique of Tibshirani (1996) (i.e., replacing $\lambda||\boldsymbol{\beta}||_{l_0}$ by $\lambda||\boldsymbol{\beta}||_{l_1}$) has become extremely popular, because the minimization problem is convex and there is an efficient algorithm to find the unique global solution. However, the Lasso technique does not possess the oracle properties as defined by Fan and Li (2001). Alternatives to the Lasso and with the oracle properties are also popular, for example, the SCAD method of Fan (1997)and Fan and Li (2001) and adaptive Lasso of Zou (2006).

The "Irrepresentable Condition" of Zhao and Yu (2006) is an important factor in applying Lasso and its variants. The condition states that "Lasso selects the true model consistently if and (almost) only if the predictors that are not in the true model are 'irrepresentable' by predictors that are in the true model." What it means in practice is that multicollinearity can create problems in applying Lasso or its variants. Many real problems come with many closely related variables. Individually, they offer meaningful explanatory powers, but including them all does not make sense nor actually improves the model's performance after factoring in sampling errors. The practical issue is to find one or two of them work the best along with other variables of interest. Multicolinearity among these variables can be conceptually viewed as a natural result of linearly transforming a set of independent variables. Note that a regression model fit, measured in R^2, is in invariant to such a linear transformation of regressors, but the corresponding l_1 penalty term is not. Such a natural set of competing variables may incur a high l_1 penalty without measurably increasing the R^2, which then leads to their total elimination from the selected model. In short, multicolinearity interferes with regressor selection, and the Lasso or its variants may lead to the entire set of closely related variables not being selected instead of intuitively keeping one variable or two in the

final model. This observation is not purely theoretical because it often occurs in practice. We will demonstrate this later with a simulation study.

In the next section, we devise a SMC-based combinatorial optimization solution that is stable and runs efficiently. Theoretically, this stochastic solution can be made arbitrarily close to the true solution through the argument of the maximum order statistic. By nature, our SMC approach is a global optimization technique because it samples from the entire distribution defined by the optimization objective function.

3 The distribution-tempered SMC solution

Our approach relies on reformulating the zero-norm constrained variable selection problem in (2) into an equivalent maximization problem over a distribution function defined on a p_s-dimensional random vector, $\boldsymbol{U} = (U_1, U_2, \cdots, U_{p_s})$, that takes values from the set of the regressor sequence numbers $\boldsymbol{P} = \{1, 2, \cdots, p\}$ without replacement and its distribution is unknown. Naturally, $p_s < n$ is understood. If otherwise, the regression model based on any combination of p_s regressors would, generally speaking, produce a perfect fit. We will leave the task of identifying its distribution function for a later discussion. For now, the focus is on setting up this equivalent reformulation. Without loss of generality, first note that the inequality constraint in (2) can be turned into an equality, because the solution will obviously be at the boundary. Thus, the minimization problem in (2) is equivalent to:

$$\arg\max_{\{\boldsymbol{U}\in\boldsymbol{P}(p_s)\}} \exp\left\{-\lambda||\boldsymbol{y} - \boldsymbol{X}_{\boldsymbol{U}}\hat{\boldsymbol{\beta}}(\boldsymbol{U})||_{l_2}^2\right\} \tag{3}$$

where $\boldsymbol{P}(p_s)$ denotes $\{\boldsymbol{U} \in \boldsymbol{P}^{p_s} \& U_1 \neq U_2 \neq \cdots \neq U_{p_s}\}$, $\boldsymbol{P}^{p_s}$ stands for the p_s-Cartesian product of $\boldsymbol{P}$, $\boldsymbol{X}_{\boldsymbol{U}}$ denotes the sub-matrix of $\boldsymbol{X}$ whose columns correspond to the regressor sequence numbers in $\boldsymbol{U}$, and $\hat{\boldsymbol{\beta}}(\boldsymbol{U}) = (\boldsymbol{X}_{\boldsymbol{U}}'\boldsymbol{X}_{\boldsymbol{U}})^{-1}\boldsymbol{X}_{\boldsymbol{U}}'\boldsymbol{y}$ is the optimal regression $\boldsymbol{\beta}$ when $\boldsymbol{U}$ is known. Finally, λ is a positive self-adaptive tuning device for numerical accuracy which tunes the totally discrete target function.

Since $\exp\left\{-\lambda||\boldsymbol{y} - \boldsymbol{X}_{\boldsymbol{U}}\hat{\boldsymbol{\beta}}(\boldsymbol{U})||_{l_2}^2\right\}$ is positive, it can obviously be viewed as a discrete distribution function of $\boldsymbol{U}$, up to a norming constant, over $\boldsymbol{P}(p_s)$, i.e, the set of permutations. Since the order of regressors for a given combination yields the same regression solution, the distribution defined over permutations is a proportionally scaled down version of the distribution function defined over combinations where the scaling factor is $p_s!$. Since permutations are easier to generate, we will target the distribution function over permutations where the SMC algorithm is by design free of any proportional constant.

Instead of having the pure in-sample target as in (3), we choose to conduct the following optimization built upon a cross-validated target function and refer to it as the Stable

Combinatorially-Optimized Features Selector (SCOFS):

$$\arg\max_{\{p_l \le p_s \le p_u, \boldsymbol{U} \in \boldsymbol{P}(p_s)\}} \exp\left\{ -\lambda \sum_{j=1}^{k} ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j} \hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||_{l_2}^2 \right\} \tag{4}$$

where $\hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})$ is the optimal beta corresponding to the $(\boldsymbol{U})$ with the j^{th} data subset excluded, and p_l and p_u define the permissible range for the number of regressors.

The solution for the optimal number of regressors has a natural tendency moving towards p_u because of a better fit even for the cross-validated target function. Insignificant β's in $\hat{\boldsymbol{\beta}}(\boldsymbol{U})$ evaluated at the optimally-selected $\boldsymbol{U}$ may occur unless we prevent the model to have any insignificant regression coefficient. The following scheme addresses over-fitting.

1. For the optimally-selected p_s model, check to see whether there is any insignificant variable with an α-level t-test. If so, reduce p_s by 1, find the optimally-selected $p_s - 1$ model and check for insignificant regression coefficients again.

2. If the optimally-selected p_s model has no insignificant variable, find the optimally-selected $p_s + 1$ model and check for any insignificance again.

Therefore, our target discrete distribution function for sampling at a fixed p_s is

$$f(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X}) \propto \exp\left\{ -\lambda \sum_{j=1}^{k} ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j} \hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||_{l_2}^2 \right\}. \tag{5}$$

The task of selecting regressors has now been converted into finding the maximum of the above distribution function.

The basic idea of finding the maximum of (5) is to generate a sample suitably representing this distribution function. Our distribution-tempered SMC approach to finding this maximum has its root in Del Moral, *et al* (2006) and Duan and Fulop (2015). Our approach resembles more that of Duan and Zhang (2016), which devises a way to effectively generate a high-dimensional random object subject to some condition. Generating a sample of non-Gaussian bridge paths is the target in Duan and Zhang (2016), whereas in this paper, the task is to find the maximum value of $f(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})$ through sequentially generating $(p_s, \boldsymbol{U})$ by tempering distribution to arrive at a sample that properly represents $f(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})$. Thus, its maximal value and corresponding maximizer, up to a Monte Carlo error, becomes readily available.

3.1 The algorithm

Our distribution-tempered SMC method can be divided into three key steps – (1) initialization, (2) reweighting and resampling, and (3) support boosting.

Initialization

Assign each regressor with an initial probability $\bar{q}_i$ for $i \in \boldsymbol{P}$. An intuitive and quick way is to set $\bar{q}_i = R_i^2 / \sum_{j=1}^{p} R_j^2$ where R_i^2 is the regression R^2 using a single Variable i. Let $q_i^{(p_s)}(0) = \bar{q}_i$ for all i's, and use $q_i^{(p_s)}(0)$ to sample $\boldsymbol{U}$ from $\boldsymbol{P}$, i.e., simulate p_s regressors without replacement from the pool of p regressors. Since we need to obtain exactly p_s regressors, sampling can be performed sequentially by choosing the first regressor out of the p potential regressors with the probability of $q_i^{(p_s)}(0)$ for $i \in \boldsymbol{P}$. Then, move on to selecting the second one out of the remaining $p-1$ regressors. Assuming that the first one is Variable i, the probability for choosing Variable j as the second becomes $\frac{q_j^{(p_s)}(0)}{1-q_i^{(p_s)}(0)}$ for $j \in \boldsymbol{P} \setminus \{i\}$. The same logic applies to the third regressor and so on until reaching the last one in the permutation, i.e., p_s, for which the probability is $q_k^{(p_s)}(0)$ for sampling Variable k from the remainder. If $q_i^{(p_s)}(0) = 0$ for some Variable i, it will never be sampled and can thus be trimmed from the set of regressors from the start. Thus, $\boldsymbol{P}$ should be understood as the set of regressors with $q_i^{(p_s)}(0) > 0$. Use $I(\boldsymbol{U} \in \boldsymbol{P}(p_s); q_i^{(p_s)}(0), i \in \boldsymbol{P})$ to denote this permutation sampler based on the distribution $q_i^{(p_s)}(0)$.

Note that this sampler's probability distribution, being a product of the probabilities just mentioned, depends on the specific sequence of appearance. In other words, the sampled points represent different permutations, which are easier to sample and evaluate their probabilities. The regression solution, however, only depends on the combination, and hence different permutations yielding the same combination will share the same value of $\exp\left\{-\lambda \sum_{j=1}^{k} ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j} \hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||_{l_2}^2\right\}$. Thus, the maximum is not unique over $\boldsymbol{U} \in \boldsymbol{P}(p_s)$, but it does not matter to the solution.

Reweighting and resampling

Define an intermediate target distribution function as

$$f_\gamma(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X}) \propto \left(\frac{\exp\left\{-\lambda \sum_{j=1}^{k} ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j} \hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||_{l_2}^2\right\}}{I(\boldsymbol{U} \in \boldsymbol{P}(p_s); q_i^{(p_s)}(0), i \in \boldsymbol{P})}\right)^{\gamma} \times I(\boldsymbol{U} \in \boldsymbol{P}(p_s); q_i^{(p_s)}(0), i \in \boldsymbol{P}) \quad (6)$$

Obviously, $f_\gamma(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})$ equals $I(\boldsymbol{U} \in \boldsymbol{P}(p_s); q_i^{(p_s)}(0), i \in \boldsymbol{P})$ when $\gamma = 0$, and $f(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})$ when $\gamma = 1$.

The distribution-tempered SMC technique moves the sample along a self-adapted control bridge by advancing γ from 0 to 1. The self-adapted control rests with choosing γ so that the effective sample size (ESS) implied by the importance weight does not fall below a threshold ηM where M is the intended size of the SMC sample and η equals, say, 1/2. Denote the incremental importance weight by $w_{\gamma,\gamma^{(j)}}(\boldsymbol{U}^{(i)}) = \left(\frac{\exp\left\{-\lambda \sum_{j=1}^{k} ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j}\hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||_{l_2}^2\right\}}{I(\boldsymbol{U}^{(i)} \in \boldsymbol{P}(p_s); q_i^{(p_s)}(0), i \in \boldsymbol{P})} \right)^{\gamma - \gamma^{(j)}}$ and $\text{ESS} = \frac{\left(\sum_{i=1}^{M} w_{\gamma,\gamma^{(j)}}(\boldsymbol{U}^{(i)})\right)^2}{\sum_{i=1}^{M}\left(w_{\gamma,\gamma^{(j)}}(\boldsymbol{U}^{(i)})\right)^2}$.

Set $\gamma^{(0)} = 0$. Find γ^* such that the ESS is no less than ηM. Note that this solution need not be exactly at ηM, because it is just a control device to prevent the quality of sample from deteriorating too much. Use the incremental importance weight to resample in order to obtain an equally-weighted sample of $\boldsymbol{U}$. Then, set $\gamma^{(1)} = \gamma^*$.

Support boosting

After resampling, the sample is likely to contain more duplicate copies of some $\boldsymbol{U}$'s to reflect their relatively high importance weights, which means that empirical support has shrunk.[2] We need to boost the empirical support before advancing γ again. Support boosting can be accomplished by several Metropolis-Hastings (MH) moves until the cumulative realized acceptance rate has reached a target level, say, 100%, which is to ensure that the empirical support has been properly boosted.[3]

The SMC sample provides a natural basis for coming up with a good proposal for executing MH moves. Compute $c_i^{(p_s)}(\gamma) = \sum_{j=1}^{M} \sum_{l=1}^{p_s} \chi_{\{U_l^{(j)}=i\}}$, which is the total count of Variable i appearing in the sample of size M. Define a probability by $q_i^{(p_s)}(\gamma) = c_i^{(p_s)}(\gamma) / \sum_{j=1}^{p} c_j^{(p_s)}(\gamma)$ for $i = 1, 2 \cdots, p$, which reflects the relative importance of Variable i after the SMC algorithm has reached the stage indicated by γ. If p_s regressors attain a probability of $1/p_s$, the sampler based on these probabilities will always generate different permutations of the same p_s regressors. We use $\mathcal{Q}_{\boldsymbol{U}}(\gamma)$ to represent these probabilities inferred from the sample. Instead of proposing a new permutation hoping to improve the solution, a more efficient and

[2]It is worth noting that duplicates can be expected under a discrete distribution even for an ideal sample. Duplicates merely reflect the number of elements in the theoretical support versus the sample size. Support-boosting is meant to remove duplicates due to resampling, but it is impossible to get rid of duplicates inherent to a discrete distribution.

[3]Since the underlying distribution is discrete, complete distinctiveness of particles cannot be expected. To ensure that proper support boosting has been completed, one can attach to each particle a uniform random number whenever a proposal is made. Resampling destroys particle distinctiveness, but accepting new proposal has in effect restored distinctiveness which is revealed in the distinctiveness of these attached uniform random numbers.

realistic approach is to replace a randomly selected subset of the existing permutation when p_s is large. Importantly, this differs from the single best replacement algorithm of Soussen, *et al* (2015).

Denote by $h(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))$ the conditional distribution based on $\mathcal{Q}_{\boldsymbol{U}}(\gamma)$ for proposing regressors to replace $\tilde{A}$, a random subject of elements in permutation $\boldsymbol{U}$, where $\boldsymbol{U}_{-\tilde{A}}$ stands for removing from $\boldsymbol{U}$ those elements in $\tilde{A}$. Given $\tilde{A}$, this probability is the standard permutation result for proposing element $i \in \boldsymbol{P} \setminus \boldsymbol{U}_{-\tilde{A}}$. Coupling with the probability of sampling $\tilde{A}$ gives rise to the overall proposal probability.

One may be tempted to propose a replacement subset in a way just like the initialization step. However, if for some random chance the current M p_s-variable combinations had completely missed the variables in the optimal solution, the support boosting step would forever miss the optimal choice. A more reliable way of proposing new regressor permutations is therefore to mix $h(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))$ with $I(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \bar{q}_i, i \in \boldsymbol{P})$, the initialization sampler applying to $i \in \boldsymbol{P} \setminus \boldsymbol{U}_{-\tilde{A}}$. Specifically, our proposal sampler is based on $h^{(\omega)}(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma)) = \omega h(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma)) + (1-\omega) I(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \bar{q}_i, i \in \boldsymbol{P})$, where ω is set to 50% in our implementation.

The MH acceptance probability for replacing $\tilde{A}$, a random subset of elements in permutation $\boldsymbol{U}$, is

$$
\begin{aligned}
& \alpha^{(j)}_{\gamma}\{\boldsymbol{U} \in \boldsymbol{P}(p_s) \Rightarrow \boldsymbol{U}^* \in \boldsymbol{P}(p_s)\} \\
= \; & \min\left\{1, \frac{f_\gamma(\boldsymbol{U}^* \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})}{f_\gamma(\boldsymbol{U} \in \boldsymbol{P}(p_s); \boldsymbol{y}, \boldsymbol{X})} \frac{h^{(\omega)}(\boldsymbol{U} \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))}{h^{(\omega)}(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))}\right\} \\
= \; & \min\left\{1, \frac{\exp\left\{-\gamma\lambda \sum_{j=1}^k ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U}^*,j}\hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U}^*)||^2_{l_2}\right\}}{\exp\left\{-\gamma\lambda \sum_{j=1}^k ||\boldsymbol{y}_j - \boldsymbol{X}_{\boldsymbol{U},j}\hat{\boldsymbol{\beta}}_{j-}(\boldsymbol{U})||^2_{l_2}\right\}} \left(\frac{I(\boldsymbol{U}^* \in \boldsymbol{P}(p_s); \bar{q}_i, i \in \boldsymbol{P})}{I(\boldsymbol{U} \in \boldsymbol{P}(p_s); \bar{q}_i, i \in \boldsymbol{P})}\right)^{1-\gamma} \right. \\
& \left. \times \frac{h^{(\omega)}(\boldsymbol{U} \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))}{h^{(\omega)}(\boldsymbol{U}^* \in \boldsymbol{P}(p_s) \mid \boldsymbol{U}^*_{-\tilde{A}} = \boldsymbol{U}_{-\tilde{A}}; \mathcal{Q}_{\boldsymbol{U}}(\gamma))}\right\} \qquad (7)
\end{aligned}
$$

The acceptance probability defines a Markov kernel, and the target intermediate distribution in (6) is, by the standard argument, the stationary solution to the Markov kernel. In one round of support boosting, the MH move may replace a random number of elements to increase combination diversity. An average acceptance rate can be computed for a particular round. The support boosting step is considered satisfactorily completed when the cumulative average acceptance rate over rounds has reached the target level, and for which we set it at 500%.

After the empirical support is properly boosted, the system is ready for advancement from $\gamma^{(1)}$ to $\gamma^{(2)}$, and eventually to 1. This is conducted by repeating the reweighting, resampling and support boosting steps. Finally, the tuning parameter λ can be adapted adjusted so that the optimal solution in the final SMC sample reaches an empirical probability between, say, 10% to 20%. This tuning ensures that the final SMC sample is not degenerate due to over-concentration in a few particles and also distinguishes the optimal solution distinguishes from others.

3.2 k-fold duplication to enlarge the SMC sample

When the number of chosen regressors, $p_s \leq p$, is large, the size of the SMC sample, M, may need to be large enough to properly represent the underlying distribution over the discrete and yet very large set, $\boldsymbol{P}(p_s)$. In fact, if M is made arbitrarily large, the SMC solution will be arbitrarily close to the true solution in probability, or may even be the true solution simply because the underlying distribution is discrete with a large but finite number of combinations. Efficiently increasing the SMC sample size will thus make the method practically powerful. Duan and Zhang' (2016) k-fold duplication technique serves this purpose well.

k-fold duplication is to duplicate the sample of size M to kM by making additional $(k-1)$ identical copies of the SMC sample of size M. One then relies on support-boosting to reduce duplicates (i.e., enlarge the empirical support) to turn the sample into a truly representative sample of size kM. The key to k-fold duplication vis-a-vis straightforward SMC with a sample of size kM is to directly leverage the final SMC sample of size M at the stage of $\gamma = 1$ so as to bypass the intermediate steps required for the distribution-tempering bridge for $(k-1)M$ sample points.

3.3 Assessing the quality of solution

Like all Monte Carlo algorithms, SCOFS is a stochastic scheme that will in principle select the optimal p_s and p_s-variable combination out of the potential variables that delivers the maximum R^2 if the SMC sample size, M, approaches infinity. Practically speaking, however, one would need a means to assess how close the final solution's R^2 is to the true maximum under some finite M. In typical Monte Carlo analyses, the Central Limit Theorem serves as the basis for an assessment. In our context, the Fisher-Tippett-Gnedenko Extreme Value Theorem provides the theoretical foundation.

Let $\bar{R}^2(p_s)$ be the regression R^2 at our final selected set of p_s variables. It is important to note that $\bar{R}^2(p_s)$ may be strictly larger than $R^2(p_s; M) = \max\{R_i^2(p_s); i = 1, 2, \cdots, M\}$, the set of R^2 corresponding to different p_s-variable combinations in the final SMC sample of size M, because $\bar{R}^2(p_s)$ is the best result recorded throughout the whole sequential updating

process. Denote by $R^2_{max}(p_s)$ the theoretical true maximum R^2 after exhausting all possible combinations of p_s variables, and naturally, $R^2(p_s; M) \leq \bar{R}^2(p_s) \leq R^2_{max}(p_s)$. Our tasks are (1) to estimate $R^2_{max}(p_s)$ so as to know the potential room for improvement by increasing the SMC sample size, and (2) to estimate the exceedance probability, i.e., the probability of exceeding $\bar{R}^2(p_s)$.

$R^2(p_s)$ can be viewed as a random variable resulting from random p_s-variable combinations. Although its distribution function, denoted by $G(\cdot)$, is an unknown discrete distribution, R^2 is obviously bounded above by its theoretical maximum at $R^2_{max}(p_s) \leq 1$. In the neighborhood of $R^2_{max}(p_s)$, one can show that $G(x)$ can be approximated by $(R^2_{max}(p_s) - x)^{\alpha} L\left(\frac{1}{R^2_{max}(p_s)-x}\right)$ where $L(\cdot)$ is a slowly varying function converging to zero at infinity and α some unknown positive constant, because $G(x)$ is fundamentally a discrete distribution. Thus, the Fisher-Tippett-Gnedenko Extreme Value Theorem implies convergence to a Weibull distribution with the shape parameter α and scale parameter 1; that is, for large $M' < M$

$$Prob\left\{\frac{R^2(p_s, M') - R^2_{max}(p_s)}{R^2_{max}(p_s) - G^{\leftarrow}(1 - 1/M')} \leq x\right\} \cong \exp\left(-|x|^{\alpha}\right) \quad \text{for } x \leq 0 \tag{8}$$

where $G^{\leftarrow}(x) \equiv \inf\{y : G(y) > x\}$, i.e., the left continuous inverse. Consequently, $R^2(p_s; M')$ has an approximate distribution function:

$$F_{R^2(p_s,M')}(z) = \exp\left[-\left(\frac{R^2_{max}(p_s) - z}{\eta}\right)^{\alpha}\right] \quad \text{for } z \leq R^2_{max}(p_s) \tag{9}$$

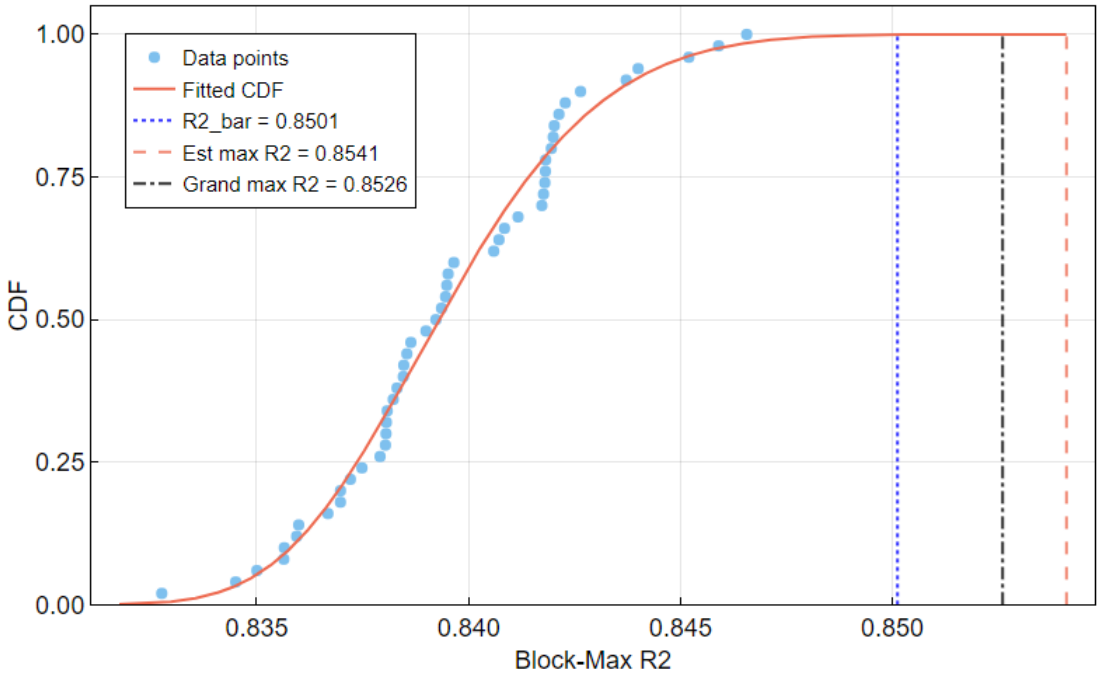
where $\eta = R^2_{max}(p_s) - G^{\leftarrow}(1 - 1/M')$.

The Weibull distribution in (9) can be treated as a two- or three-parameter distribution function. Taking $R^2_{max}(p_s)$, α and η as unknown, the system has three parameters. If instead we view $G^{\leftarrow}(1 - 1/M')$ as known by its empirical distribution derived from the final SMC sample, then the system only has two unknown parameters, i.e., $R^2_{max}(p_s)$ and α. We will estimate the unknown parameters using $\{R^2_i(p_s); i = 1, 2, \cdots, M\}$ by randomly partitioning the SMC sample into k subsamples of size M'; for example, the SMC sample of size 5,000 is partitioned into 50 subsamples of 100 each. We then use these k subsample maximum R^2, i.e., $R^2(p_s; M')$ to find optimal values for $R^2_{max}(p_s)$ (≤ 1 and $\geq \bar{R}^2(p_s)$), $\alpha > 0$ and $\eta > 0$ (if η is treated as a free parameter) by matching closely the k-point empirical distribution to the extreme value distribution by minimizing the l_2 distance.[4]

[4]Note that some of these k subsample maxima may share common values. In which case, the l_2 distance will be computed over fewer than k points. If the number of unique R^2 values equals one, this degenerate case cannot be estimated and the true maximum R^2 is considered already attained by $\bar{R}^2(p_s)$. When the number of unique R^2 is two, we use the two-parameter distribution function whereas for cases with three or more unique R^2 values, we always deploy the three-parameter distribution in estimation. One may want to use the tuning parameter λ to adjust the SMC sample diversity to yield more distinct block-maximum R^2's.

Figure 1 displays a predicted Weibull distribution for R^2 taken from a simulation study to be described in the next section. It is estimated to a 47-point empirical distribution deriving from 50 block-maximum R^2's with a block size of 100 randomly selected SMC particles that are obtained with SCOFS using 5,000 SMC particles. The data sample is generated with a model of 12 variables at a theoretical R^2 of 80%, and the selection task is to choose 12 among 900 potential variables. In this figure, $R^2_{max} = 0.8541$ provides an estimate for how high the R^2 can be potentially increased to from its current value of 0.8501, by increasing the number of SMC particles. We produce a proxy grand maximum R^2 for benchmarking, which is the maximum R^2 over 500 randomly seeded SMC optimization runs on the same data sample. Note that the estimated maximum R^2 (i.e., 0.8541) displayed on the graph only slightly overstates the possibility of exceeding the grand maximum R^2 (i.e., 0.8526).

Figure 1: An example of the extreme value theory predicted R^2 distribution function



4 Simulation studies

To ascertain the reliability of SCOFS, we conduct a simulation study and report the results in Table 1. All SCOFS runs in this simulation study use 1,000 SMC particles. This simulation study intends to address the question as to whether the best combination of explanatory variables can be found. Since the combinatory possibilities cannot be practically exhausted,

Table 1: A simulation study deploys Stable Combinatorially-optimized Feature Selector (SCOFS) with 1,000 SMC particles to select out of 900 potential variables: the selected model vs. the estimated true model over 500 simulated samples. The 900 variables, which are normal random variables of mean 0 and variance 1, are divided into three independent groups of 300 each and at three levels of within-group correlation – 0, 0.4, and 0.8. The magnitude of regression coefficient varies from low to high and the theoretical R^2 of the simulated model is at 40% or 80%.

# of obs=200, # of var=9, and R^2=80%				**# of obs=200, # of var=9, and R^2=40%**			
R^2 of the selected model is strictly greater: 100%				R^2 of the selected model is strictly greater: 100%			
Hit Ratio	Within-group correlation			**Hit Ratio**	Within-group correlation		
Coefficient	0	0.4	0.8	Coefficient	0	0.4	0.8
0.1	0.04	0.02	0.014	0.1	0.024	0.01	0.014
0.5	0.952	0.934	0.4	0.5	0.292	0.156	0.05
1.0	0.992	0.992	0.98	1.0	0.918	0.74	0.256

# of obs=200, # of var=18, and R^2=80%				**# of obs=200, # of var=18, and R^2=40%**			
R^2 of the selected model is strictly greater: 100%				R^2 of the selected model is strictly greater: 100%			
Hit Ratio	Within-group correlation			**Hit Ratio**	Within-group correlation		
Coefficient	0	0.4	0.8	Coefficient	0	0.4	0.8
0.1	0.026	0.028	0.024	0.1	0.024	0.028	0.02
0.2	0.064	0.054	0.03	0.2	0.03	0.016	0.022
0.5	0.444	0.344	0.094	0.5	0.072	0.068	0.026
0.7	0.794	0.656	0.246	0.7	0.194	0.122	0.054
0.85	0.93	0.828	0.36	0.85	0.252	0.164	0.066
1.0	0.976	0.93	0.482	1.0	0.368	0.268	0.094

# of obs=1000, # of var=9, and R^2=80%				**# of obs=1000, # of var=18, and R^2=80%**			
R^2 of the selected model is strictly greater: 97.2%				R^2 of the selected model is strictly greater: 97.4%			
Hit Ratio	Within-group correlation			**Hit Ratio**	Within-group correlation		
Coefficient	0	0.4	0.8	Coefficient	0	0.4	0.8
0.1	0.242	0.218	0.048	0.1	0.064	0.074	0.03
0.5	0.998	0.998	0.996	0.2	0.304	0.356	0.116
1.0	0.998	0.998	0.998	0.5	0.95	0.996	0.734
				0.7	0.996	0.996	0.986
				0.85	0.996	0.996	0.996
				1.0	0.996	0.996	0.996

there is no direct answer to the question, and we need an indirect and sensible way to assess the method's reliability. In this simulation study, we rely on the knowledge that the true model is always a feasible combination, and thus the estimated true model serves as a natural benchmark to assess the quality of SCOFS. Due to sampling errors, the in-sample R^2 of the best combination must be greater than or equal to the benchmark value when the number of explanatory variables is kept the same as that of the true model. If the number of observations increases, it should become increasingly difficult for the best solution to yield an R^2 strictly higher than the benchmark value.

The simulation study reported in Table 1 always involves 900 potential explanatory variables, and we set out to select 9 or 18 variables, depending on which number were used to generate the data. The 900 variables are equally divided into three groups of 300 each. Within each group, the variables have the same correlation. The first group has zero correlation, the second is 0.4, and third is 0.8. Across groups, variables are independent. All 900 variables are normally distributed with mean 0 and variance 1. Among the 300 variables in each group, the regression coefficients are set to 0.1 for 100 variables, 0.5 for the second batch of 100 variables, and finally 1 for the remainder. We consider two simulation setups to ascertain the impact of residual errors in variable selection. We factor in the magnitude of residual errors by considering two levels of theoretical R^2 at 80% and 40%. In addition, we examine the impact of the sample size and study the results under the sample size of 200 and 1,000, respectively. All studies are conducted with 500 repetitions to tally the various rates of occurrence.

We will use the top-left panel in Table 1 to explain the simulation results. This simulation study deploys 9 out of 900 variables to generate the data set as described above. The theoretical R^2 is at 80% and each sample has 200 observations. The results show that SCOFS is able to beat the estimated true model 100% in the 500 simulation repetitions. The results imply that SCOFS has mostly likely found the best in-sample solution. The hit ratios (i.e, success in identifying the variables in the true model) reported in this panel for nine sub-categories suggest that SCOFS yields a higher hit ratio for the lower correlation group and the variables with a larger regression coefficient. Low hit ratios for the high correlation group and the variables with a low regression coefficient are expected, because those variables are not supposed to be easily discernable when the sample size is 200. Moving on to other cases reported in different panels of Table 1 by varying the number of variables to be selected, the level of R^2, and sample size, a consistent pattern emerges to reflect the fact that sampling errors provide room for SCOFS to outperform the estimated true model in-sample.

Next we check how close $\bar{R}^2(p_s)$, the SCOFS-obtained maximum R^2, is to the grand maximum R^2, the true maximum R^2 proxied by the maximum R^2 over 500 randomly-seeded SMC optimization runs with 1,000 particles on the same data sample. The results are

reported in Table 2. It is expected that they are no greater than the grand maximum R^2. As the reported values suggest, $\bar{R}^2(p_s)$ has hit the grand maximum R^2 for smaller p_s. For larger p_s, the room for improvement is practically negligible.

Table 2: Differences between $\bar{R}^2(p_s)$ (R^2 of the SCOFS-selected model using 1,000 SMC particles) and the grand maximum R^2 (a proxy for the true maximum R^2) under different p_s (the number of selected regressors) over 500 random SMC runs on a same data sample of size 200. The data sample is simulated with a generating model that contains 12 variables with a theoretical R^2 of 80% to be selected from 900 potential variables. Four variables are assigned to each of the three correlation groups (i.e., 0, 0.4 and 0.8) with their coefficients equal to 0.1, 0.4, 0.7 and 1, respectively.

p_s	Mean	Std	Min	25%	Median	75%	Max
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	-0.000151	0.000342	-0.001257	0	0	0	0
8	0	0	0	0	0	0	0
9	-0.000013	0.000045	-0.000608	0	0	0	0
10	-0.00005	0.00014	-0.001077	0	0	0	0
11	-0.001405	0.000565	-0.003175	-0.001511	-0.001511	-0.001511	0
12	-0.002114	0.001058	-0.003952	-0.002523	-0.002523	-0.002211	0
13	-0.001389	0.000908	-0.003678	-0.001877	-0.001877	0	0
14	-0.001371	0.001186	-0.005212	-0.002681	-0.001214	-0.00042	0
15	-0.002992	0.001564	-0.006228	-0.004558	-0.003108	-0.001777	0
16	-0.002532	0.001528	-0.007693	-0.003561	-0.00286	-0.001064	0
17	-0.002216	0.001669	-0.00721	-0.002782	-0.001661	-0.001242	0
18	-0.002717	0.001916	-0.009594	-0.003697	-0.002063	-0.001488	0
19	-0.002908	0.002286	-0.010031	-0.003755	-0.002058	-0.001134	0
20	-0.002993	0.00236	-0.010473	-0.004609	-0.001974	-0.001677	0

The comparison study between SCOFS and the Adaptive Lasso of Zou (2006) is based on the true model having 12 variables among 900 potential variables. Each of 500 simulation repetitions generates 200 observations with the 900 variables again equally divided into three groups with different levels of within-group correlation of 0.1, 0.4 and 0.8. Each group is assigned four variables with regression coefficients of 0.1, 0.4, 0.7 and 1, respectively. The theoretical R^2 is always fixed at 80%. Since the number of variables in the true model is treated as unknown, five-fold cross-validation is used to determine the "right" number of variables in the Adaptive Lasso, which matches the five-fold cross-validated SCOFS using 1,000 SMC particles. The results in Table 3 clearly indicate that the adaptive Lasso is prone to choose too many variables with an average of 53 variables when the target number is 12 by design. Moreover, the range is quite wide with the solutions from 13 to 142. In contrast, SCOFS yields an average of 11 variables and covers a much narrower range from 7 to 14.

Table 3: A comparison study of Stable Combinatorially-optimized Feature Selector (SCOFS) using 1,000 SMC particles vs. adaptive Lasso in selecting out of 900 potential variables over 500 simulated samples. The 900 variables, which are normal random variables of mean 0 and variance 1, are divided into three independent groups of 300 each and at three levels of within-group correlation – 0, 0.4, and 0.8. The true number of variables is set at 12 with four assigned to each group, and their coefficients are 0.1, 0.4, 0.7 and 1. The number of observations is fixed at 200 and the theoretical R^2 of the simulated model is set to 80%. Five-fold cross-validation is deployed to determine the number of selected variables for both methods.

Performance	**SCOFS**	**Adaptive Lasso**
# of selected variables (min, mean, max)	(7, 11, 14)	(13, 53, 142)
F-Score	0.57	0.26
Precision	0.60	0.17
Recall	0.54	0.63

Three ratios reported in Table 3 need some explanations. First, the precision measures the number of selected variables among the 12 true variables in relation to the total number of variables being selected. A higher value implies a sort of higher accuracy. However, the precision can be misleading when a method under-selects, and this can be easily understood with an example. Suppose that a method only selects one variable which happens to be among the 12 true variables. The precision will be 100%, but the method has missed all 11 other true variables. The recall computes the number of selected variables among the 12 true variables divided by 12, the number of variables in the true model in this simulation

study. This measure can also be quite misleading if a method over-selects. In the extreme case of selecting all variables, the recall will be by definition 100%.

The F-score strikes a balance of the two, and is the harmonic average of precision and recall, i.e., $\left(\frac{\text{precision}^{-1}+\text{recall}^{-1}}{2}\right)^{-1}$. Due to excessive over-selection by the adaptive Lasso, the recall is high and the precision is low, whereas SCOFS goes in an opposite direction. All in all, the F-score of 0.57 for SCOFS implies its superior performance over the adaptive Lasso's 0.26.

It is interesting to note that the popular Lasso penalty regression of Tibshirani (1996) lacks the oracle properties as defined in Fan and Li (2001) because it generates downward biased regression coefficients (in magnitude) by shrinking all coefficient estimates towards zero. Even though the oracle property can be restored using adaptive Lasso of Zou (2006) or SCAD penalty regression of Fan (1997) and Fan and Li (2001), they are based on asymptotic arguments, which in essence rightly removes in the limit the penalty placed on regressors with non-zero coefficients. Practical usage invariably deals with finite samples, and users often adopt cross-validation as a penalty level selection criterion, which relies on the out-of-sample prediction accuracy. Invariably, more regressors are chosen to counteract coefficient shrinkage. This point has been made by, for example, Leng, *et al* (2006) and Wang, *et al* (2007). Deploying SCOFS removes such a tension between the biased estimates and the right set of regressors.

Our next simulation study is to determine how reliable the prediction based on the extreme value theorem described in Section 3.3 in predicting the true maximum R^2 and generating a probability for further improvement upon the currently obtained $\bar{R}^2(p_s)$ on one sample. For this simulation study, we use one fixed simulated sample of 200 data points based on the 12-variable setup with 900 potential regressors. We conduct 500 times of randomly-seeded SCOFS runs on the same data sample. Since the data set is fixed, the true maximum R^2 is fixed. However, the value predicted by the extreme value theory will vary over the 500 runs. The fact that the true model is based 12 variables is assumed unknown to the analyst. We shall therefore examine the performance under different numbers of selected variables ranging from 1 to 20, and in each case, study how well the extreme value theory performs.

Each SCOFS run deploys 5,000 SMC particles, which are always partitioned into 50 blocks of 100 points each. We compute 50 block maxima and use them to estimate the Weibull distribution as described in Section 3.3. Since each randomly-seeded run produces a SCOFS-obtained $\bar{R}^2(p_s)$, an extreme value theory predicted $R^2_{max}(p_s)$ and an exceedance probability, it gives rise to a sample of 500 points to study performance.

Table 4: The extreme value theory predicted exceedance probability for R^2 improvement under different p_s (the number of selected regressors) over 500 randomly-seeded SCOFS runs on the same data sample of size 200. The data sample is simulated with a generating model of 12 variables at a theoretical R^2 of 80% to be selected from 900 potential variables. Four variables are assigned to each of the three correlation groups (i.e., 0, 0.4 and 0.8) with their coefficients equal to 0.1, 0.4, 0.7 and 1, respectively.

p_s	Mean	Std	Min	25%	Median	75%	Max
2	0.001677	0.004509	0	0	0	1.66E-07	0.045022
3	4.32E-07	0.000006	0	0	0	0	0.000092
4	1.35E-07	0.000003	0	0	0	0	0.000063
5	0.000216	0.00167	0	0	0	0	0.019677
6	0.034897	0.052047	0	3.45E-12	5.29E-09	0.066587	0.185934
7	0.013667	0.023401	0	0	1.39E-15	0.02013	0.147781
8	0.002216	0.006813	0	0	0	0.000986	0.080736
9	0.057673	0.044814	0	0.018746	0.05346	0.089384	0.213498
10	0.009727	0.014128	0	0	0.001944	0.015592	0.072866
11	0.001114	0.00302	0	0	0	0.000602	0.03231
12	0.00185	0.01069	0	0	0	0.000017	0.125854
13	0.012436	0.032315	0	0	2.69E-12	0.001292	0.188275
14	0.010989	0.028548	0	0	0	0.0027	0.220008
15	0.0076	0.019448	0	0	0	0.003098	0.150932
16	0.004698	0.014978	0	0	0	0.001444	0.166613
17	0.001916	0.008102	0	0	0	1.17E-12	0.087696
18	0.001266	0.006225	0	0	0	1.17E-08	0.073881
19	0.000234	0.001841	0	0	0	0	0.028998
20	0.000347	0.003467	0	0	0	0	0.072633

The median values reported in Table 4 indicate that exceedance probabilities are minuscule except when $p_s = 9$. The exceedance probability of 5.346% may suggest a small chance of improving the R^2 by increasing the number of SMC particles, but the expected improvement magnitude in R^2 is actually quite small (around 0.1%) as revealed in the corresponding median value in Table 5 where the estimated magnitudes of potential improvement

under different cases are reported. All in all, the extreme value theorem provides a workable way for one to assess the quality of the current SCOFS solution so as to decide whether to commit more computing resources to improve upon the current solution. When one decides to proceed, the k-fold duplication method described in Section 3.2 becomes handy.

Table 5: Differences between the extreme value theory predicted $R^2_{max}(p_s)$ and SCOFS-obtained $\bar{R}^2(p_s)$ under different p_s (the number of selected regressors) over 500 random SMC runs on the same data sample of size 200. The data sample is simulated with a generating model that contains 12 variables with a theoretical R^2 of 80% to be selected from 900 potential variables. Four variables are assigned to each of the three correlation groups (i.e., 0, 0.4 and 0.8) with their coefficients equal to 0.1, 0.4, 0.7 and 1, respectively.

p_s	Mean	Std	Min	25%	Median	75%	Max
2	0.000903	0.002832	0	0	3.20E-11	0.000041	0.029309
3	0.000312	0.002508	0	7.75E-12	1.10E-11	2.03E-11	0.028416
4	0.000127	0.001728	0	1.00E-10	1.00E-10	1.00E-10	0.028081
5	0.000433	0.003189	0	9.00E-12	1.00E-10	1.00E-10	0.035484
6	0.002846	0.006841	8.40E-11	1.00E-10	1.00E-10	0.001054	0.038931
7	0.000196	0.000897	0	0	0	0.000128	0.015801
8	0.001425	0.005095	0	1.00E-12	1.00E-11	0.000038	0.035166
9	0.002451	0.004003	0	0.00026	0.001031	0.002961	0.0345
10	0.003453	0.005932	0	6.75E-12	0.001181	0.003906	0.039464
11	0.00323	0.006504	0	3.00E-12	1.00E-10	0.003532	0.039332
12	0.00298	0.007178	0	1.00E-12	1.05E-11	0.00118	0.041248
13	0.001207	0.004712	0	0	6.00E-12	0.00003	0.035751
14	0.000889	0.003329	0	0	0	0.000075	0.032541
15	0.001324	0.003917	0	0	4.00E-12	0.000788	0.042377
16	0.002024	0.005014	0	0	3.00E-12	0.001366	0.042143
17	0.00139	0.004119	0	0	3.00E-12	0.000023	0.028418
18	0.002011	0.00579	0	1.00E-12	5.00E-12	0.000174	0.044502
19	0.001922	0.00606	0	1.00E-12	5.50E-12	1.00E-10	0.044396
20	0.002076	0.006959	0	2.00E-12	6.00E-12	1.00E-10	0.043383

References

[1] Del Moral, P., A. Doucet, and A. Jasra, 2006, Sequential monte carlo samplers, *Journal of the Royal Statistical Society: Series B*, 68(3), 411-436.

[2] Duan, J.-C., and A. Fulop, 2015, Density-Tempered Marginalized Sequential Monte Carlo Samplers, *Journal of Business and Economic Statistics*, 33(2), 192-202.

[3] Duan, J.-C., S. Li, and Y. Xu, 2022, Sequential Monte Carlo optimization and statistical inference, *Wiley Integrative Reviews: Computational Statistics*, e1598. https://doi.org/10.1002/wics.1598.

[4] Duan, J.C. and C. Zhang, 2016, Non-Gaussian Bridge Sampling with an Application, National University of Singapore working paper.

[5] Fan, J., 1997, Comments on "Wavelets in Statistics: a Review" by A. Antoniadis, *Journal of the Italian Statistical Society* 6(20, 131-138.

[6] Fan, J. and R. Li, 2001, Variable Selectionvia Nonconcave Penalized Likelihood and its Oracle Properties, *Journal of the American Statistical Association* 96, 1348-1360.

[7] George, E. and R. McCulloch, 1993, Variable Selection via Gibbs Sampling, *Journal of the American Statistical Association*, 88, 881-889.

[8] Mitchell, T. and J. Beauchamp, 1988, Bayesian Variable Selection in Linear Regression (with discussion), *Journal of the American Statistical Association* 83, 1023-1036.

[9] Natarajan, B.K., 1995, Sparse Approximate Solutions to Linear Systems, *SIAM Journal on Computing*, 24(2), 227-234.

[10] Polson, N. and L. Sun, 2018, Bayesian l_0-regularized Least Squares, Applied Stochastic Models in Business and Industry, 1-15.

[11] Schafer, C and N. Chopin, 2013, Sequential Monte Carlo on Large Binary Sampling Spaces, *Statistics and Computing* 23, 163-184.

[12] Soussen C., J. Idier, J. Duan, and D. Brie, 2015, Homotopy Based Algorithms for l_0-regularized Least-squares, *IEEE Trans Signal Process*, 63, 3301-3316.

[13] Tibshirani, R., 1996, Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society, Ser. B*, 58(1), 267-288.

[14] Zhao, P. and B. Yu, 2016, On Model Selection Consistency of Lasso, *Journal of Machine Learning Research* 7, 2541-2563.

[15] Zou, H., 2006, The Adaptive Lasso and Its Oracle Properties, *Journal of the American Statistical Association*, 101(476), 1418-1429.