International Journal of Machine Learning and Cybernetics (2020) 11:217–230
https://doi.org/10.1007/s13042-019-00968-9

ORIGINAL ARTICLE

On selective learning in stochastic stepwise ensembles

Chun-Xia Zhang[1] · **Sang-Woon Kim**[2] · **Jiang-She Zhang**[1]

Received: 11 July 2017 / Accepted: 22 May 2019 / Published online: 6 June 2019

Abstract
Ensemble learning has attracted much attention of researchers studying variable selection due to its great power in improving selection accuracy and stabilizing selection results. In this paper, we present a novel ensemble pruning technique called Pruned-ST2E to obtain more effective variable selection ensembles. The order to aggregate the individuals generated by the ST2E algorithm (Xin and Zhu in J Comput Graph Stat 21(2):275–294, 2012) is rearranged. To estimate the importance of each candidate variable, only some members ranked ahead are remained. Experiments with simulated and real-world data show that the performance of Pruned-ST2E is comparable or superior to several other benchmark methods. Through analyzing the accuracy–diversity pattern in both ST2E and Pruned-ST2E, it is revealed that the inserted pruning step excludes less accurate members. The reserved members also become more concentrated on the true importance vector. Moreover, Pruned-ST2E is easy to implement. Therefore, Pruned-ST2E can be considered as an alternative for tackling variable selection tasks in practice.

Keywords Variable selection ensemble · Ensemble pruning · Variable selection · Selection accuracy · Aggregation order

1 Introduction

Variable selection is an important topic in statistics since it can be used to improve the accuracy and interpretability of the predictions given by the estimated models. With large amount of high-dimensional data emerging in many research and application areas, it has become an indispensable tool for solving various problems. Thus, it is particularly important to perform variable selection effectively and efficiently. Nowadays, the popular methods include subset selection [1, 19], coefficient shrinkage [6, 7, 9, 25, 36], variable screening [8], Bayesian methods [12, 24] and so on. However, to be effective, variable selection generally requires careful tuning of some parameters in the analysis. The correct specification of these parameters is a difficult task, even for expert statisticians. For this reason, variable ranking (i.e., sorting the variables in terms of their importance in the prediction of the outcome) is often carried out first. Once the variables are properly ranked [10, 28, 34], selection can be achieved by using a thresholding rule. In present work, we will follow the latter practice to perform variable selection.

Many scholars [20, 23, 28] have argued that variable selection serves two different objectives depending on whether the modelling purpose is for prediction or for interpretation. The former aims at seeking a parsimonious model so that future data can be well forecast or *prediction accuracy* can be maximized. But for the latter, analysts would like to identify the truly important variables (i.e., having actual influence on an outcome) from the numerous candidate ones, or to maximize *selection accuracy*. Due to the significant difference between predictive models and explanatory models, the corresponding variable selection approaches are also very different. With selection accuracy as the target, we will address variable ranking and selection problems in linear regression models by using ensemble learning techniques.

Ensemble learning, a widely used technique to enhance the performance of single learning machines, has been shown to be effective to address a large range of prediction tasks [13, 18,

✉ Chun-Xia Zhang
cxzhang@mail.xjtu.edu.cn

Sang-Woon Kim
kimsw@mju.ac.kr

Jiang-She Zhang
jszhang@mail.xjtu.edu.cn

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China

[2] Department of Computer Engineering, Myongji University, Yongin 17058, Republic of Korea

22]. Most of the existing ensemble methods, such as bagging and boosting, were developed to handle prediction problems better, and the finally obtained models can be called *prediction ensembles* (PEs). So far, some methods have been specially designed to construct *variable selection ensembles* (VSEs). With some traditional methods such as lasso [25], genetic algorithm [34], more variables than necessary are often picked out (i.e., having high false positive rate). However, if the selection process is repeated using slightly different data for a number of trials, the frequency that the truly important variables are chosen will be high, while that of unimportant ones being falsely considered as important will be low. As a result, the important variables can be easily distinguished from the remaining ones. This explains why ensemble learning is effective in the context of variable selection.

Like PEs, the process of creating a VSE generally consists of two steps, that is, *ensemble generation* and *ensemble integration*. The first step addresses how to generate a series of accurate and diverse members. And the second step aims to fuse them in a suitable way so that selection accuracy is maximized. In generation process, the usual practice is to apply a base learner (i.e., a variable selection method) on multiple different training sets or to inject some randomness into the learner. When fusing the results produced by each member, a simple averaging rule is commonly employed. The existing VSE approaches mainly include parallel genetic algorithm (PGA) [34], stability selection [17], random lasso [27], bagged stepwise search (BSS) [35], stochastic stepwise ensembles (ST2E) [28], PBoostGA [30], AddNoiseGA [31] and stochastic correlation coefficient ensembles (SCCE) [3].

In the study of PEs, it has become well-known that it is usually beneficial to only select some members, instead of keeping all members, to construct a subensemble after Zhou et al. [33] proven the "many-could-be-better-than-all" theorem. Since then, a large variety of ensemble pruning techniques [5, 15, 16, 18, 22] have been proposed in the "overproduce and choose" framework. Compared with full ensembles, the pruned ensembles need less storage, implement a prediction faster, and more importantly, achieve higher prediction accuracy. Borrowing the similar idea to VSEs, we surmise that VSEs can also benefit from selective fusion. As far as we know, however, there is little literature adopting ensemble pruning to constitute a VSE. So far, only Zhang et al. [32] made an attempt to address this issue and put forward one heuristic algorithm. Different from them, we propose a more effective algorithm to get better selection results. Considering that ST2E [28] performs very well in various situations, the novel algorithm is developed to improve it by inserting a pruning step.

The rest of the paper is organized as follows. Section 2 presents the novel ensemble pruning method Pruned-ST2E for variable ranking and selection, in detail. In Sect. 3, a novel manner to analyze the working mechanism of VSEs is provided. Section 4 devotes to examining the performance of Pruned-ST2E and comparing it with several existing techniques by using some simulations. Meanwhile, some real-world examples are analyzed in Sect. 5. Finally, Sect. 6 offers conclusions, together with some future work of the paper.

2 Pruning in ordered stochastic stepwise ensembles for variable ranking and selection

2.1 Brief introduction of stochastic stepwise ensembles (ST2Es)

Since the novel method is developed on the basis of ST2E [28], we first give it a brief introduction here. In the generation phase, ST2E executes the stochastic stepwise (abbreviated as ST2) algorithm multiple times to create its constituent members. Subsequently, a simple averaging rule is used to combine the selection results of each member. Instead of adding (deleting) *one* variable in the forward (backward) step as traditional stepwise selection does, ST2 adds (deletes) a *group* of variables at one time, and the group size is randomly decided. Meanwhile, only a randomly selected few, rather than all possible, groups are assessed and the best one is chosen. The forward and backward selection steps are implemented iteratively until no improvement of the objective function such as AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion) can be made. In doing so, ST2 can always select some important variables. On the other hand, ST2 will produce different selection results even by running it on the same data since it is a stochastic search algorithm. Therefore, it can be easily imagined that ST2E can construct a VSE containing a series of diverse but accurate variable selectors. Due to page limitation, please interested readers refer to Xin and Zhu [28] for more details of the ST2E algorithm.

It is noteworthy that there are two critical tuning parameters in ST2E, i.e., λ and κ, to determine the group size and the number of candidate groups to evaluate when executing ST2 algorithm to produce an ensemble member. Xin and Zhu [28] suggested to take $\lambda = 1/2$ and choose κ ($\kappa > 1$) through looking for the start of an "elbow" in the diversity plot. In later experimental studies, we will carry out some simulations to justify this choice.

2.2 Pruned-ST2E: novel technique to prune ST2E

With regard to the existing VSE approaches [3, 17, 27–29, 34, 35], they usually integrate all the generated members into an ensemble. Some evidence [28, 29, 34, 35] has proven that strength and diversity are two critical ingredients to ensure the good performance of a VSE. Nevertheless, these two terms haven't been simultaneously considered when generating the

ensemble members. Hence, there inevitably exist some individuals whose role is trivial or even negative. Meanwhile, a thresholding rule needs to be adopted to attain final selection results with an unsupervised learning mode once the variables are ranked. In other words, it will be conducive if the difference between the average measure of important variable group and that of unimportant group can be enlarged. In this aspect, it is hence deserved to remove the redundant members of a VSE.

In any existing approach to prune PEs, the essential step is to identify unnecessary members. To achieve this purpose, ranking-based and search-based strategies [15, 16, 18] are two commonly used approaches to select an ensemble subset. Here, we apply the idea of ranking-based strategies to VSEs. The core idea is to first sort all the individuals according to a certain criterion. Then, the top individuals whose rank is above a given threshold are kept to compose a subensemble. Because of the good performance of ST2E [28], we apply selective learning to its ensemble members. To ease presentation, the novel algorithm is denoted by Pruned-ST2E whose detailed steps are listed in the following Algorithm 1.

Algorithm 1. The proposed Pruned-ST2E algorithm.

Input

$\mathbf{y}$: $n \times 1$ vector.

$\mathbf{X}$: $n \times p$ design matrix.

λ: a parameter in ST2 to control group size in a step.

κ: a parameter in ST2 to control number of groups to assess.

B: size of full ensemble.

U: size of pruned subensemble.

Output

Average importance measure computed as

$$R(j) = \frac{1}{U}\sum_{r=1}^{U}\mathbf{E}'(r,j), \quad j = 1, 2, \cdots, p. \tag{1}$$

Main steps of Pruned-ST2E

1. Initialize a matrix $\mathbf{E}$ of order $B \times p$ with all elements being 0.
2. For $b = 1, 2, \cdots, B$
   (a) Provide $\mathbf{y}$ and $\mathbf{X}$ as the input of the ST2 algorithm to perform variable selection, i.e., $\mathcal{S}_b = ST2(\mathbf{X}, \mathbf{y}, \lambda, \kappa)$.
   (b) Let
   $$\mathbf{E}(b,j) = \begin{cases} 1, & \text{if } X_j \in \mathcal{S}_b, \\ 0, & \text{if } X_j \notin \mathcal{S}_b, \end{cases} \quad j = 1, 2, \cdots, p. \tag{2}$$
3. EndFor
4. Sort all the ensemble members in ascending order according to the number of their selected variables, i.e.,
   $$N_b = \sum_{j=1}^{p}\mathbf{E}(b,j), \quad b = 1, 2, \cdots, B. \tag{3}$$
5. Select the U members sorted on top of the ranked list, i.e., arrange the U rows of $\mathbf{E}$ ranked ahead into a new matrix $\mathbf{E}'$.

It is worth mentioning that ST2E differs from Pruned-ST2E only in steps 4–5 of Algorithm 1. Specifically, ST2E computes the average importance measures of each variable j ($j = 1, 2 \ldots, p$) by averaging all the ensemble members, that is, $R(j) = (1/B)\sum_{r=1}^{B} \mathbf{E}(r, j)$. In Pruned-ST2E, the aggregation order of the ensemble members in ST2E is first rearranged and only $U (U < B)$ members that are ranked ahead are incorporated to estimate $R(j)$ by (1). To guide the aggregation process, the number of variables selected in each trial (i.e., N_b defined in (3)) is used. The main reason for choosing this measure is explained below.

Roughly speaking, some ensemble members can correctly choose a part of, or all of the really important variables. But for the other members, they not only deem all the truly important variables as important, but also some additional unimportant ones. It is notable that Pruned-ST2E sorts the ensemble members in accordance with the number of selected variables in ascending order. With the sorted members integrated into the ensemble gradually, the selection frequency of truly important variables will increase quickly while that of the remaining ones will stay at a low level, since they are chosen just by chance. For some medium-sized value of U, the difference between the selection frequencies of important and unimportant groups (i.e., gap value) will reach a maximum. After this point, because the noise variables are chosen by more members, the gap value will decline gradually as more individuals are included into the ensemble. To judge which variables are important based on the ranked list, note that it is critical to choose an appropriate threshold to divide the candidate variables into two groups. By changing the aggregation order of individuals and stopping the fusion early, the gap between the two sets of variables are enlarged. Therefore, it is beneficial to inject an additional selective learning step before the fusion is implemented.

Based on the output of Algorithm 1 (i.e., $R(j), j = 1, 2, \ldots, p$), the p variables can be ranked according to their importance to the response. In order to attain final selection results, a thresholding rule such as the mean rule or searching for the largest gap on the scree plot [26] can be further executed. The former means selecting the variables which satisfy $R(j) > (1/p)\sum_{k=1}^{p} R(k)$. In contrast, the latter can be implemented as follows: sort $R(1), R(2), \ldots, R(p)$ in descending order; search for the largest gap between any consecutive entries; and select the variables which are located above the gap. Similar to Xin and Zhu [28], the former scheme will be adopted in later experiments unless otherwise specified.

Ideally, the number of members (i.e., U) to reserve should be automatically determined to maximize selection accuracy. Unfortunately, however, selection accuracy cannot be computed as in the prediction case, since the truly important variables are unknown in practice. The easiest method is to prescribe a desired number. According to our experiments (refer to Sect. 4) as well as the evidence in the study of PEs [15, 16], it seems to be reasonable to keep 1/3 to 2/3 members which are on top of the ranked list. In comparison with ST2E, Pruned-ST2E simply includes an additional sorting phase before fusion. Therefore, Pruned-ST2E is easy to implement and its time complexity is roughly equal to that of ST2E.

3 Analysis of accuracy-diversity trade-off

In ensemble learning field, it is commonly believed that we can gain more insights for the working mechanism of an ensemble through analyzing the trade-off between the accuracy (also known as strength) and diversity of ensemble members. Due to significant differences between PEs and VSEs, the strategies to investigate the accuracy-diversity pattern in PEs [13] cannot be directly employed in the context of VSEs. Zhu and Fan [35] offered a method to estimate the strength and diversity of a VSE and related these measures to its variable-selection performance. Later on, Xin and Zhu [28] proposed to use the strength-diversity tradeoff to specify an appropriate tuning parameter of ST2E. Here, we would like to provide a new approach to evaluate strength and diversity of a VSE, and then utilize them to analyze the working mechanism of ST2E and Pruned-ST2E.

According to the formula (2), it can be observed that the output of each ensemble member, say, $E(b, \cdot)$ ($b = 1, 2, \ldots, B$), is a binary vector with each entry being 1 or 0. Suppose the true importance vector to be $\mathbf{r}^*$ which takes value 1 on the positions corresponding to truly important variables and 0 otherwise. The accuracy of the bth member can be assessed as

$$\mathrm{Acc}_b = \frac{1}{p}\sum_{j=1}^{p} \mathbb{I}[\mathbf{E}(b, j) = \mathbf{r}^*(j)], \quad b = 1, 2 \ldots, p, \tag{4}$$

in which $\mathbb{I}(\cdot)$ stands for an indicator function taking value 1 if its condition holds and 0 otherwise. Accordingly, the overall accuracy of the VSE is

$$\mathrm{Acc} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{Acc}_b = \frac{1}{pB}\sum_{b=1}^{B}\sum_{j=1}^{p} \mathbb{I}[\mathbf{E}(b, j) = \mathbf{r}^*(j)]. \tag{5}$$

Following the practice of Xin and Zhu [28], the diversity of a VSE can be estimated as

$$\mathrm{Div} = \frac{1}{p}\sum_{j=1}^{p} v(j), \quad \text{with} \quad v(j) = \frac{1}{B-1}\sum_{b=1}^{B}\left[\mathbf{E}(b, j) - \frac{1}{B}\sum_{b=1}^{B}\mathbf{E}(b, j)\right]^2. \tag{6}$$

Actually, the quantity Div is a measure of the average within-ensemble variation. In general, a good VSE is expected to achieve Acc as high as possible. Since Div measures how much each member deviates from the central tendency estimated by the VSE, it will become small when its constituent members become more accurate. It is noteworthy that our defined Acc directly evaluate the selection accuracy of the considered VSE. In contrast, Xin and Zhu [28] made use of the improvement of AIC to implicitly evaluate the mean strength of a VSE. Because our final purpose is variable selection, it can be deemed that Acc defined in (5) is more proper to analyze VSEs.

4 Simulation studies

This section devotes to examining Pruned-ST2E as well as some other procedures with simulated data. First, the effect of modifying the aggregation order of ST2E is first studied. Then, the performance of the novel method Pruned-ST2E is examined and compared with ST2E [28] and some other popular techniques including PGA [34], lasso [25], adaptive lasso [36], SCAD [7], random forest [11] and SCCE [3]. Regarding the variable selection criterion, we utilized AIC in later experiments. There is evidence [19, 34] showing that AIC tends to select more variables than necessary whereas BIC tends to select fewer variables. Notice that the main principle of VSEs is to filter out noise variables through executing a base learner multiple times. If a truly informative variable is missed very often, it will be considered as unimportant by the VSE. In view of this fact, AIC is more suitable than BIC to act as the selection criterion in a VSE. In addition, the following experiments were all conducted in Matlab with version 8.1.

For all the experiments conducted in this section, every simulation was repeated 100 times. The ensemble size B was taken to be 300. The parameters λ and κ in ST2E were chosen to be identical to those adopted by Xin and Zhu [28]. For the examples they didn't considered, λ and κ were set to be 0.5 and exp(1.5), respectively. For Pruned-ST2E, one third of the sorted members were kept. As for the parameter N involved in PGA, i.e., the number of generations for each SGA (single-path genetic algorithm) to evolve, the strategy proposed by Zhu and Chipman [34] were employed to determine it. Regarding lasso, a fivefold cross-validation was adopted to select its optimal regularization parameter. As far as adaptive lasso is concerned, the adaptive weights were first calculated using ordinary least squares estimates and $\gamma = 1$. Subsequently, lasso was applied on the scaled data. In SCAD, we took $a = 3.7$ and generalized cross-validation to estimate its tuning parameter. For random forest, the variables were first sorted by the permutation importance measures and then were selected from the top 1/3 variables so that the smallest OOB (out-of-bag) error was achieved. Moreover, the parameter λ in SCCE was set to be 0.5.

4.1 Simulated data

The data used in the following examples were generated by

$$\begin{aligned} \mathbf{y} =& \mathbf{x}_1\beta_1 + \mathbf{x}_2\beta_2 + \cdots + \mathbf{x}_p\beta_p + \boldsymbol{\epsilon} \\ =& \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}), \end{aligned} \tag{7}$$

where $\boldsymbol{\epsilon}$ is an error term distributed as a normal distribution with mean zero and variance σ^2.

Example 1 There are $p = 20$ variables and $n = 40$ observations [34] in this example. Particularly, only variables 5, 10 and 15 have actual influence on the response $\mathbf{y}$ and their true coefficients are 1, 2, 3, respectively. The rest of the variables are uninformative and their coefficients are all zero. We set $\sigma = 1$ for scenarios 1-3 and $\sigma = 2$ for scenario 4. For the exploratory variables, the following 4 different scenarios were considered, i.e.,

Scenario 1: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{20} \sim N(\mathbf{0}, \mathbf{I})$;

Scenario 2: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{19} \sim N(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_{20} = \mathbf{x}_{10} + 0.25\mathbf{z}$, $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$;

Scenario 3: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{19} \sim N(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_{20} = \mathbf{x}_{15} + 0.25\mathbf{z}$, $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$;

Scenario 4: $\mathbf{x}_j = \mathbf{z} + \boldsymbol{\epsilon}_j, j = 1, 2, \ldots, 20$, $\boldsymbol{\epsilon}_j \sim N(\mathbf{0}, \mathbf{I})$, $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$.

Example 2 This is an experiment [28] specifically designed to examine the capability of a method to correctly detect weak signal ($\mathbf{x}_1$), strong signal ($\mathbf{x}_2, \mathbf{x}_3$) and noise variables ($\mathbf{x}_j, j = 4, \ldots, p$). In this example, there are $p = 20$ variables with each generated from the standard normal distribution $N(0, 1)$. The true coefficient vector is $\boldsymbol{\beta} = (\alpha, 2, 3, 0, \ldots, 0)^T$, where α takes the values starting from 0.1 to 1.5 with an increment 0.1. The three variables to generate $\mathbf{y}$ are correlated, with $corr(\mathbf{x}_i, \mathbf{x}_j) = 0.7$ for $i, j \in \{1, 2, 3\}$ and $i \neq j$. The other variables $\mathbf{x}_4, \ldots, \mathbf{x}_{20}$ and the noise term $\boldsymbol{\epsilon}$ are independent of each other. Here, we considered the situation of $n = 100, \sigma = 3$.

Example 3 The data in this example are modified from a widely used benchmark data set [25, 27, 28]. To increase the problem complexity, we suppose that there are $p = 50$ variables with each generated from $N(0, 1)$. The pairwise correlation between any two variables is $corr(\mathbf{x}_i, \mathbf{x}_j) = 0.5^{|i-j|}$ for all $i \neq j$. The true coefficient vector is $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-5})^T$ and $\mathbf{0}_{p-5}$ indicates a $(p-5)$-dimensional zero vector.

Example 4 Finally, we considered a high dimensional but small sample size (large p, small n) problem which contains $p = 120$ variables and $n = 100$ observations [27, 28]. The first 60 variables are truly important and their true coefficients are drawn from a normal distribution $N(3, 0.5)$, and their values are then fixed for all simulation runs. The exploratory variables are created from a multivariate normal distribution with zero mean and covariance matrix as

$$\begin{bmatrix} \Sigma_0 & 0 & 0 & 0 \\ 0 & \Sigma_0 & 0.2J & 0 \\ 0 & 0.2J & \Sigma_0 & 0 \\ 0 & 0 & 0 & \Sigma_0 \end{bmatrix}. \tag{8}$$

where Σ_0 is a 30×30 matrix with unit diagonal elements and off-diagonal elements taking a value 0.7, and J is a 30×30 matrix with all unit elements. The noise level for ϵ was taken as $\sigma = 50$ similarly to that used by Wang et al. [27], Xin and Zhu [28].

4.2 Evaluation metrics

Because our focus on variable selection is to obtain a parsimonious model for interpretability, selection accuracy is the most natural choice to evaluate an algorithm. Nevertheless, it is infeasible since we have no means to know the ground truth except for simulated data. For this reason, a large variety of metrics have been developed and utilized in the related literature [27, 28, 32, 34, 36] to assess a variable selection method from different perspectives. To extensively study the behavior of each compared method, we adopt several evaluation metrics defined as below.

Let $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$ stand for the true coefficient vector and $\mathcal{T} = \{j : \beta_j \neq 0\}$ indicate the true model. Let IV and UIV represent the index sets for the truly important and unimportant variables, respectively. The symbol $|IV|$ indicates the number of variables in IV and $|UIV|$ is defined accordingly. Given an algorithm, note that each metric is estimated by 100 replications of the simulation. In the tth replication, denote $\hat{r}_t = (\hat{r}_{1,t}, \hat{r}_{2,t}, \ldots, \hat{r}_{p,t})^T$ by the importance measures assigned to each variable, $\hat{\boldsymbol{\beta}}_t = (\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \ldots, \hat{\beta}_{p,t})^T$ by the estimated coefficients, and $\hat{\mathcal{S}}_t = \{j : \hat{\beta}_{j,t} \neq 0\}$ by the identified model. In what follows, $\mathbb{I}(\cdot)$ represents an indicator function. Then, we define

$$\begin{cases} \text{soft metric} = \dfrac{1}{100} \displaystyle\sum_{t=1}^{100} \mathbb{I}\left(\min_{j \in IV} \hat{r}_{j,t} \geq \max_{j \in UIV} \hat{r}_{j,t} \right), \\ \text{hard metric} = \dfrac{1}{100} \displaystyle\sum_{t=1}^{100} \mathbb{I}\left(\hat{\mathcal{S}}_t = \mathcal{T} \right), \end{cases} \tag{9}$$

$$\begin{cases} \%\ \text{of correct zeros} = \dfrac{1}{|UIV|} \left\{ \dfrac{1}{100} \displaystyle\sum_{t=1}^{100} \sum_{j \in UIV} \mathbb{I}\left(\hat{\beta}_{j,t} = 0 \right) \right\} \times 100\%, \\ \%\ \text{of incorrect zeros} = \dfrac{1}{|IV|} \left\{ \dfrac{1}{100} \displaystyle\sum_{t=1}^{100} \sum_{j \in IV} \mathbb{I}\left(\hat{\beta}_{j,t} = 0 \right) \right\} \times 100\%. \end{cases} \tag{10}$$

Let

$$T_j = \sum_{t=1}^{100} \mathbb{I}\left(\hat{\beta}_{j,t} \neq 0 \right),$$

$$\begin{cases} \text{sel. freq. of } IV = \left(\min\limits_{j \in IV} T_j, \text{median}\limits_{j \in IV} T_j, \max\limits_{j \in IV} T_j \right), \\ \text{sel. freq. of } UIV = \left(\min\limits_{j \in UIV} T_j, \text{median}\limits_{j \in UIV} T_j, \max\limits_{j \in UIV} T_j \right), \end{cases} \tag{11}$$

$$\text{model size} = \frac{1}{100} \sum_{j=1}^{p} T_j = \frac{1}{100} \sum_{j=1}^{p} \sum_{t=1}^{100} \mathbb{I}\left(\hat{\beta}_{j,t} \neq 0 \right), \tag{12}$$

$$\text{RPE} = \frac{1}{\sigma^2} E[(\hat{y} - \mathbf{x}^T \boldsymbol{\beta})^2] = \frac{1}{\sigma^2} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T E(\mathbf{x}\mathbf{x}^T)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}). \tag{13}$$

In (9), *soft metric* evaluates the ranking performance of an algorithm, i.e., how well it works to rank variables in line with their importance. On the other hand, *hard metric* assesses how well an algorithm performs to detect the true model (i.e., accurately categorizing all variables into important and unimportant ones). The *percentage (%) of correct and incorrect zeros* defined in (10) are actually equivalent to specificity and false negative rate. In nature, these two metrics provide one way to evaluate the overall capacity of one method to correctly identify IV and UIV. The purpose to compute the *selection frequencies* [27–29] for IV and UIV as defined in (11) is to closely check the selection behavior achieved by an approach. In particular, we recorded the minimum, median and maximum number of times out of 100 simulations among IV and UIV are selected, respectively. Moreover, the *model size (MS)* defined in (12) denotes the estimated sparsity. Since variable selection and prediction are closely related, a good selection approach is also usually expected to exhibit good prediction performance. To evaluate the prediction ability of a method, we utilized the *relative prediction error* (*RPE*) [36] as defined in (13). To estimate this *RPE* term corresponding to an algorithm, a linear regression model was first built by using the selected variables. Then, *RPE* was estimated using a test set composed of 10,000 instances.

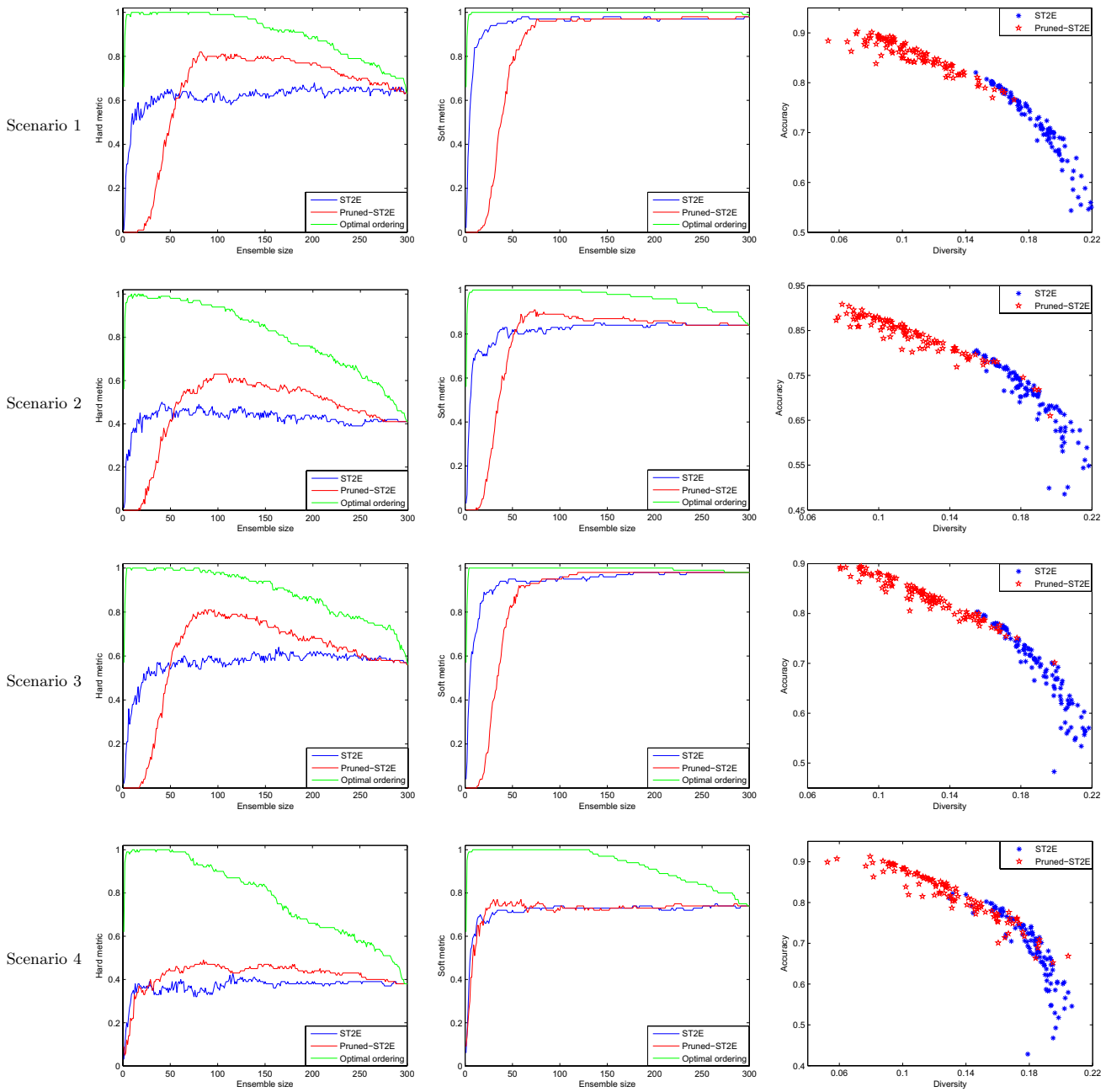


Fig. 1 For scenarios 1-4 in Example 1, the performance of ST2E, Pruned-ST2E and optimal subensembles in terms of hard and soft metrics (i.e., the first two columns) as a function of ensemble size. The subplots shown in the 3rd column illustrate the accuracy-diversity patterns of the full ST2E ensemble and a pruned subensemble

4.3 Effect of rearranging aggregation order

First, Example 1 was used to investigate how the performance of ST2E varies if the aggregation order of its constituent members is rearranged. Both the hard and soft metrics were employed to evaluate a VSE. When calculating the hard metric, searching for the largest gap [26] was applied to the ranked list. For the ST2E ensembles, the members were aggregated gradually in the same random order as they were generated by ST2E. But for pruned ST2E ensembles, the members were first sorted by Algorithm 1 and then fused. We also assessed the performance of the subensembles that were constructed by sorting the ensembles in the optimal order. Since we know that $\mathbf{x}_5$, $\mathbf{x}_{10}$ and $\mathbf{x}_{15}$ are truly important variables, a member was searched for each ensemble size so that the difference between the mean importance measure of the signal variable group and that of the noise variable group was maximized. Fig. 1 depicts the computed hard and soft

metrics for ST2E, Pruned-ST2E and optimal subensembles as a function of ensemble size.

By adopting the accuracy and diversity defined in (5) and (6), respectively, we tried to figure out the difference between ST2E and Pruned-ST2E. In each simulation, we computed the Acc and Div of ST2E by taking into account all its members. Here, $\mathbf{r}^*$ was a 20-dimensional binary column vector with its 5th, 10th and 15th entry being 1 and others being 0. Regarding Pruned-ST2E, the members in ST2E were first sorted with Algorithm 1 and only one third of the top-ranked members was kept to fuse into a VSE. In other words, the Acc and Div of Pruned-ST2E were estimated with the reserved members. The subplots shown in the third column of Fig. 1 illustrate the accuracy-diversity pattern of ST2E and Pruned-ST2E. Note that each point of ST2E (Pruned-ST2E) in these subplots corresponds to the result obtained in one simulation.

Figure 1 shows that the accuracy of ST2E evaluated with both hard and soft metrics sharply increases to a nearly optimal value as the ensemble incorporates more individuals. Then, further improvement resulting from additional members remains small. In terms of hard metric, however, the pruned ensembles have accuracy curves that exhibit a maximum for intermediate numbers of individuals. Regarding the ranking accuracy curve of Pruned-ST2E, it has similar shape with that of ST2E. After the individuals of ST2E are sorted by Algorithm 1, they can be roughly categorized into three types. For the members entering the fusion early, they often identify only some rather than all truly important variables. With respect to those lying in the middle of the ranked list, all signal variables can be deemed by them as important. Regarding the members ranked on the bottom, they often include more noise variables besides all important ones. Therefore, all signal variables can thus be correctly ranked ahead of the noise ones quickly as the sorted members gradually enter the fusion process. But due to the enlarged gap between two groups, the selection accuracy benefits more from this process. Additionally, the subplots in the 3rd column of Fig. 1 illustrate that the pruning phase improves the selection accuracy of ST2E to a large degree. In comparison with the full ST2E ensemble, some less accurate members are filtered out and are thus not fused into pruned subensembles. It is therefore reasonable that Pruned-ST2E has smaller diversity since its members become more concentrated on the true importance vector.

Moreover, it can be observed in Fig. 1 that the selection accuracy of all but very small subensembles lies above that of the full ensemble consisting of all the 300 individuals (i.e, referring to the rightmost point in each subplot). This makes it easy to select a subensemble that outperforms the original ensemble constructed by ST2E. In addition, the plots shown here illustrate that simply keeping the first 1/3 to 2/3 of the sorted individuals to compose a VSE is reasonable. Although the accuracy of the optimal subensembles is much higher than that of ST2E and Pruned-ST2E, it is only an ideal case since we cannot know which variables are truly important in practice.

4.4 Effect of λ and κ

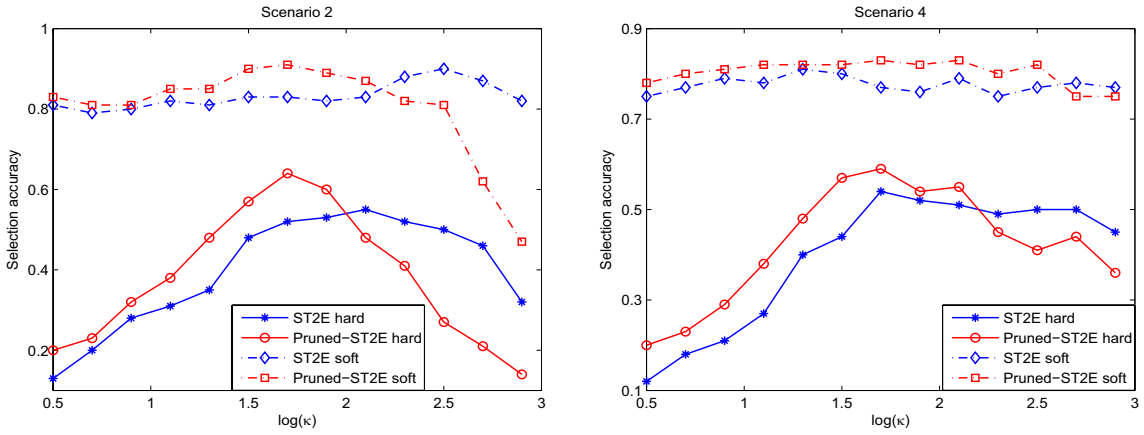
In ST2E, there are two tuning parameters λ and κ involved in its base learner *ST*2 (see, e.g., Algorithm 1). In the process to generate a collection of accurate and diverse members, these two parameters (especially κ) play a key role. Because λ controls how many variables are added (deleted) in one forward (backward) step, its impact is relatively small. As suggested by Xin and Zhu [28], we can simply set $\lambda = 0.5$. With respect to κ, it needs to be set more carefully because it directly affect whether the created members are diverse or not. The larger κ is, the less models are evaluated in each searching step. This reduces the strength of each member on one hand, which is also harmful to encourage the diversity in ST2E on the other hand. But the computational cost of ST2E will be very high if κ is taken to be too small. Due to this fact, we speculate that λ and κ may influence the performance of Pruned-ST2E. In order to clarify this issue, we did the following experiments.

First, we fixed $\lambda = 0.5$ and varied κ from exp(0.5) to exp(3.0) with increment exp(0.2). With each value of κ, we estimated the accuracy of ST2E and Pruned-ST2E with hard and soft metrics through 100 simulations. Aiming at studying the influence of λ, κ was set to exp(1.5) and λ was made to vary from 0.10 to 0.95 with increment 0.05. Figure 2 depicts the representative results obtained on scenarios 2 and 4 in Example 1.

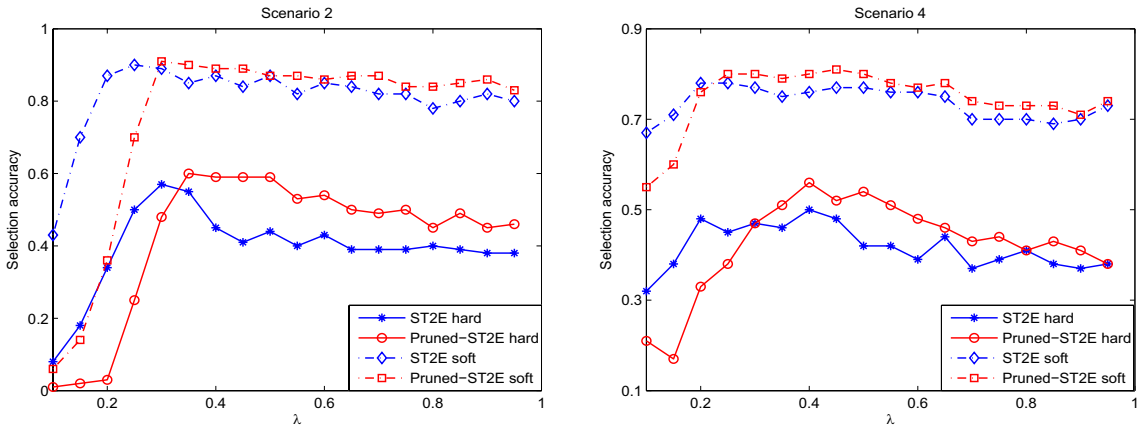
The top two subplots of Fig. 2 indicate that Pruned-ST2E generally improves ST2E when $\kappa \leq \exp(2)$. Note that the smaller κ, the more accurate each individual is. Thus, it is justifiable that changing the aggregation order is helpful in this situation. Checking the bottom two subplots of Fig. 2, i.e., how the selection frequency of ST2E and Pruned-ST2E depends on λ, one can observe that both ST2E and Pruned-ST2E perform well as long as λ is not too small (i.e., $\lambda \geq 0.4$). The behavior of ST2E and Pruned-ST2E, however, deteriorates if λ is too large, which means that adding or deleting too many variables in a single step of ST2 algorithm is unhelpful. This is consistent with the proposal provided by Xin and Zhu [28]. Based on these observations, λ and κ were respectively taken as $\lambda = 0.5$ and $\kappa = \exp(1.5)$ in most experiments.

4.5 Performance comparison

Example 2 was utilized to examine the ability of each algorithm to identify three different types of variables. The subplots (a–c) displayed in Fig. 3 demonstrate how the average



(a) The effect of κ with $\lambda = 0.5$.



(b) The effect of λ with $\kappa = \exp(1.5)$.

Fig. 2 On scenarios 2 and 4 in Example 1, how the selection frequency of ST2E and Pruned-ST2E changes with λ and κ

frequencies that each type of variables is detected to be important change as α varies. In subplots (d, e), we plotted the accuracy of each method assessed by the hard and soft metrics as a function of α. In the meantime, the subplot (f) illustrates the CPU time cost by each method to complete one simulation. The computing environment is a personal computer configured with Intel Core i7-6600 CPU @2.60 GHz, 16.0 GB RAM.

Figure 3a demonstrates that SCCE performs best to identify the weak signal variable $\mathbf{x}_1$. However, this is achieved by losing its ability to exclude noise variables (see Fig. 3c). The subplot (d) also manifests that it cannot accurately discern the right signal variable group even though its ranking performance is excellent. Meanwhile, ST2E and Pruned-ST2E perform equally well to correctly detect $\mathbf{x}_1$, and they significantly outperform the other methods in this situation, especially when the value of α is small. In terms of catching the strong signal variables $\mathbf{x}_2$ and $\mathbf{x}_3$, all the algorithms but SCAD do a good job. As for guarding against the noise variables ($\mathbf{x}_j$ for $j > 3$), the compared methods can be ranked from the best to worst as random forest, Pruned-ST2E, ST2E, PGA, adaptive lasso, lasso, SCCE and SCAD. At the same time, Pruned-ST2E is observed to considerably surpass its rivals except for lasso in terms of selection accuracy as revealed in subplot (d). Although the subplot (e) shows that random forest and SCCE achieve better ranking accuracy than Pruned-ST2E, Pruned-ST2E defeats them in the aspect of excluding noise variables and correctly identifying the signal group, as shown in the subplots (c) and (d).

As far as the time complexity is concerned, Fig. 3f indicates that random forest is most time-consuming. Since the strategy proposed by Genuer et al. [11] was executed, all variables need to be first sorted by permutation importance measures. Based on the 1/3 variables ranked ahead, it selected the subset which achieves the smallest OOB error.

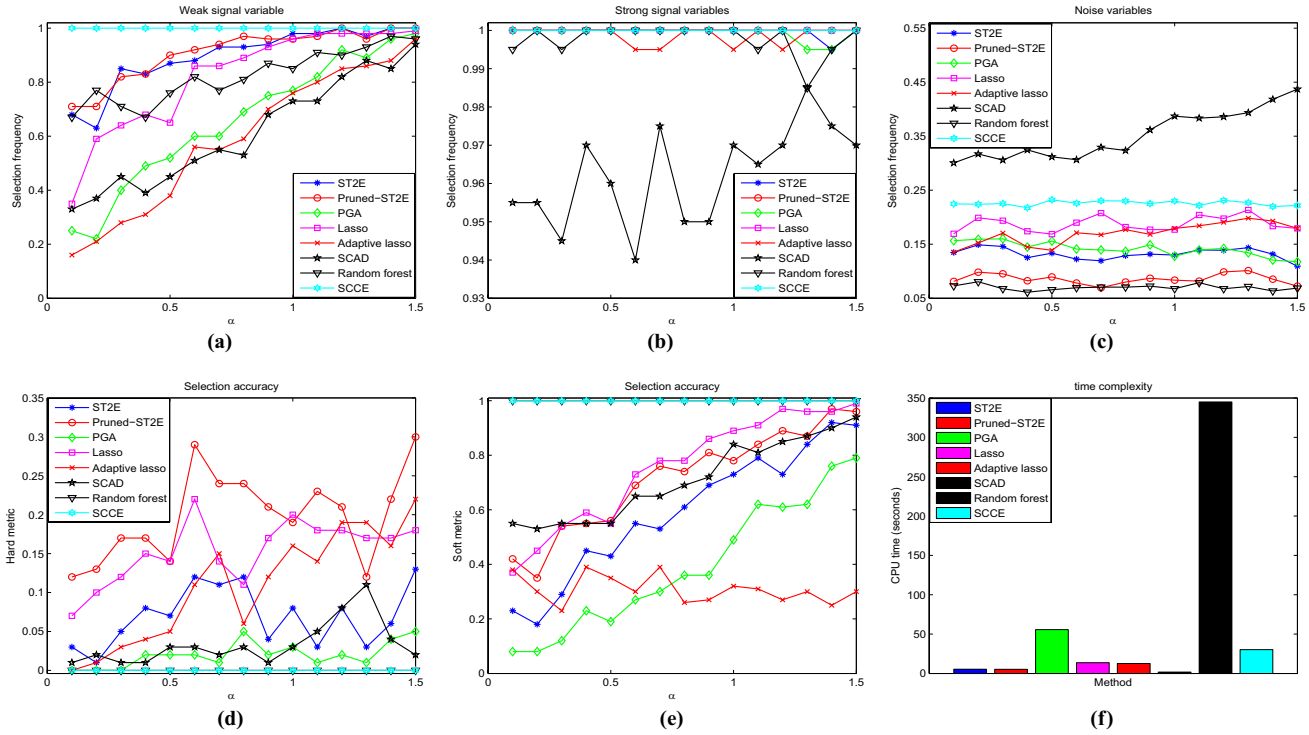


Fig. 3 The experimental results of each method in Example 2. The subplots **a**, **b** and **c** show the average frequencies for the three types of variables (i.e.,) being selected, respectively. The subplots **d** and **e** plot the selection accuracies in terms of hard and soft metrics when α takes different values. And the subplot **f** compares the time complexity of all the methods

In the rest of algorithms, PGA and SCCE cost slightly more time. When compared with ST2E, Pruned-ST2E is a little faster since it only fuses 1/3 top-ranked members to get average importance measures for each variable.

With regard to Example 3, two sets of experiments were conducted. First, we evaluated the performance of all algorithms in handling a relatively easy problem (i.e., only three important variables) and Table 1 reports the obtained results. We can observe that Pruned-ST2E behaves best in almost all cases except that it may sometimes miss some important variables when the noisy level is high. Under this circumstance, PGA, lasso, adaptive lasso and SCCE seem to achieve slightly better performance than ST2E and Pruned-ST2E at catching signal variables. However, they lose their advantage when it comes to exclude noise variables.

Besides the important variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5$, another five important variables having small coefficients (i.e., $\mathbf{x}_6, \ldots, \mathbf{x}_{10} \sim N(\mathbf{0}, 0.5^2\mathbf{I})$) were inserted into Example 3 for the second set of experiments. Table 2 summarizes the results of each algorithm in this more difficult problem. Due to the existence of some weak imponrtant variables (i.e., those having small non-zero coefficients), the performance of each algorithm decreases somewhat, especially in terms of identifying important variables. Nevertheless, Pruned-ST2E still maintains its superiority over the other ones in excluding noise variables.

The results for the problem in Example 4, in which p is large and n is small, are reported in Table 3. Similar to the strategy used by Xin and Zhu [28], we inserted an SIS (sure independence screening [8]) pre-selecting step when generating each ensemble member of ST2E, PGA and SCCE. The value of q (i.e., number of variables to pre-pick) in SIS was taken as 50. For the lasso-type methods and random forest, they were directly applied. It can be observed from Table 3 that ST2E and Pruned-ST2E perform much better than their counterparts. Here, Pruned-ST2E exhibits stronger ability to exclude noise variables but has a higher false negative rate. Notice that there are 60 important variables in this example, and small coefficients for some variables cause ST2E to reserve more members to attain better performance. In the meantime, this example reveals that the advantage of Pruned-ST2E over ST2E is more significant when the true model is sparse.

Table 1 The performance on the widely used benchmark data set with $p = 50$ and 3 important variables in Example 3

Method	% of zero coef.		Selection frequency		Size	RPE
	Correct 0s	Incorrect 0s	$\mathbf{x}_j \in$ UIV	$\mathbf{x}_j \in$ IV		
$n = 100, \sigma = 1$						
ST2E	96.32	0.00	(0, 4, 9)	(100, 100, 100)	4.73	0.189
Pruned-ST2E	97.81	0.00	(0, 2, 8)	(100, 100, 100)	4.03	0.115
PGA	95.64	0.00	(1, 4, 10)	(100, 100, 100)	5.05	0.041
Lasso	85.15	0.00	(7, 15, 32)	(100, 100, 100)	9.58	0.234
Adaptive lasso	86.00	0.00	(8, 14, 22)	(100, 100, 100)	9.98	0.279
SCAD	56.85	0.00	(35, 43, 53)	(100, 100, 100)	23.28	0.771
Random forest	98.38	0.00	(0, 0, 33)	(100, 100, 100)	3.76	0.138
SCCE	73.15	0.00	(13, 21, 100)	(100, 100, 100)	15.62	0.208
$n = 100, \sigma = 3$						
ST2E	96.66	1.67	(0, 3, 11)	(95, 100, 100)	4.52	0.127
Pruned-ST2E	98.13	9.00	(0, 2, 7)	(93, 100, 100)	3.81	0.089
PGA	94.79	0.67	(1, 5, 15)	(95, 100, 100)	5.43	0.075
Lasso	85.06	0.00	(7, 15, 32)	(100, 100, 100)	10.02	0.237
Adaptive lasso	81.68	2.00	(9, 19, 27)	(94, 100, 100)	11.55	0.317
SCAD	92.11	30.33	(3, 8, 13)	(52, 58, 99)	5.80	0.461
Random forest	96.26	1.00	(0, 2, 39)	(98, 99, 100)	4.73	0.102
SCCE	73.13	0.00	(14, 22, 99)	(100, 100, 100)	15.63	0.239
$n = 100, \sigma = 6$						
ST2E	97.55	28.00	(0, 4, 10)	(43, 74, 99)	4.31	0.210
Pruned-ST2E	97.15	31.67	(0, 2, 8)	(36, 70, 99)	3.39	0.175
PGA	92.11	18.00	(2, 7, 15)	(61, 86, 99)	6.17	0.149
Lasso	86.09	8.00	(7, 14, 30)	(82, 95, 99)	9.30	0.231
Adaptive lasso	78.81	15.67	(11, 22, 32)	(66, 88, 99)	12.49	0.363
SCAD	78.83	27.33	(14, 21, 29)	(59, 61, 98)	12.13	0.400
Random forest	88.45	7.67	(4, 10, 47)	(82, 47, 100)	8.20	0.170
SCCE	73.17	0.67	(13, 24, 99)	(98, 100, 100)	15.59	0.267

5 Analysis of real data

5.1 Diabetes data

We first analyzed diabetes data composed of $n = 442$ diabetes patients and $p = 10$ variables. The task was to interpret how some variables (such as age, sex, body mass index and etc.) affect the progression of diabetes disease. In this example, the LARS algorithm [6] was taken as a baseline to compare the different methods. The parameters involved in each algorithm were set up in a way similar to that used in the simulation studies. Table 4 reports the order in which the variables are ranked by each method. Obviously, all methods but SCCE deem that the variables "bmi", "ltg" and "map" are the most important ones while "age" is the least important one. For the intermediate variables, they hold different views. As for the different order of "tc" and "sex" produced by ST2E and Pruned-ST2E, it may be explained that Pruned-ST2E retains more members which puts more emphasis on the estimated coefficient than prediction accuracy. Notice that "tc" has a relatively larger coefficient as shown in the solution path of LARS [28]. The situation for "ldl" and "hdl" can be similarly interpreted.

5.2 Some classification data

Here, we studied the behavior of the compared methods in some classification problems by extending them to logistic regression models. Specifically, the regression coefficients were estimated by maximum likelihood estimation (MLE) method and AIC was computed as $AIC = 2k - 2l(\hat{\boldsymbol{\beta}};\mathbf{y})$ where k indicates the model size and $l(\hat{\boldsymbol{\beta}};\mathbf{y})$ is the estimated log-likelihood for the considered model. Moreover, we considered mRMR [21] instead of SCCE [3] since the the former targets to implement feature selection in classification tasks.

In Table 5, the main characteristics of five UCI [14] binary classification data sets (i.e., sample size of training and test sets, input dimensionality) were first summarized. Given a data set, each method was applied to the training set to detect important variables and a logistic regression model was built using the selected variables. We then estimated its prediction error on the test set. The whole process

Table 2 The performance on the widely used benchmark data set with $p = 50$ and eight important variables in Example 3

Method	% of zero coef.		Selection frequency		Size	RPE
	Correct 0s	Incorrect 0s	$\mathbf{x}_j \in$ UIV	$\mathbf{x}_j \in$ IV		
$n = 100, \sigma = 1$						
ST2E	94.43	13.75	(2, 6, 12)	(53, 94, 100)	9.24	0.329
Pruned-ST2E	95.57	15.13	(1, 4, 10)	(53, 93, 100)	8.65	0.303
PGA	97.40	38.87	(0, 2, 7)	(4, 64, 100)	5.98	1.029
Lasso	71.60	11.00	(21, 28, 36)	(46, 98.5, 100)	19.05	0.458
Adaptive lasso	75.33	6.12	(14, 25, 33)	(77, 98, 100)	17.87	0.547
SCAD	70.02	14.75	(23, 29.5, 37)	(59, 90, 100)	19.41	0.584
Random forest	98.45	61.25	(0, 0, 41)	(0, 3.5, 100)	3.75	1.259
SCCE	69.33	22.88	(16, 27, 100)	(32, 75.5, 100)	19.05	0.627
$n = 100, \sigma = 3$						
ST2E	93.90	50.75	(0, 6, 14)	(8, 28.5, 100)	6.50	0.234
Pruned-ST2E	96.07	52.75	(1, 4, 9)	(7, 26, 100)	5.43	0.206
PGA	95.24	52.75	(0, 5, 13)	(5, 26, 100)	5.78	0.229
Lasso	82.12	36.13	(11, 18, 28)	(15, 65.5, 100)	12.62	0.311
Adaptive lasso	76.90	34.75	(15, 23, 31)	(29, 55.5, 100)	14.92	0.478
SCAD	93.81	68.87	(2, 6, 10)	(7, 11.5, 99)	5.09	0.553
Random forest	95.43	56.63	(0, 3, 43)	(3, 15.5, 100)	5.39	0.224
SCCE	73.38	29.25	(14, 23, 99)	(34, 63, 100)	16.84	0.315
$n = 100, \sigma = 6$						
ST2E	96.29	69.25	(1, 4, 8)	(2, 15, 96)	4.02	0.266
Pruned-ST2E	97.67	70.50	(0, 2, 7)	(0, 13, 97)	3.34	0.218
PGA	92.81	62.50	(2, 7, 15)	(6, 20, 100)	5.96	0.197
Lasso	86.17	52.12	(7, 14, 26)	(15, 31.5, 99)	9.64	0.245
Adaptive lasso	77.14	50.62	(15, 23, 32)	(23, 36.5, 99)	13.55	0.446
SCAD	78.31	58.37	(15, 22, 26)	(15, 28.5, 98)	12.44	0.377
Random forest	88.57	56.50	(2, 10, 47)	(8, 16.5, 100)	8.28	0.181
SCCE	74.62	39.50	(13, 22, 94)	(18, 48.5, 100)	15.50	0.293

Table 3 The performance for the large p small n problem in Example 4

Method	% of zero coef.		Selection frequency		Size	RPE
	correct 0s	Incorrect 0s	$\mathbf{x}_j \in$ UIV	$\mathbf{x}_j \in$ IV		
ST2E	75.87	0.57	(4, 25.5, 52)	(98, 100, 100)	74.14	3.275
Pruned-ST2E	81.42	17.03	(2, 16, 40)	(74, 84, 95)	60.93	1.594
PGA	83.28	48.75	(4, 16, 30)	(40, 51, 65)	40.68	0.961
Lasso	91.88	60.17	(2, 8, 17)	(26, 40, 52)	28.77	0.400
Adaptive lasso	81.12	68.78	(7, 19, 30)	(21, 31, 39)	30.06	1.340
SCAD	70.13	68.15	(21, 29, 41)	(24, 33, 43)	37.03	2.058
Random forest	100.00	62.08	(0, 0, 0)	(29, 37, 48)	22.75	0.441
SCCE	100.00	23.67	(0, 0, 0)	(65, 76.5, 91)	45.80	0.861

was repeated 10 times and Table 5 reports the mean and standard deviation of test errors, and the average number of selected variables for each method. The model including all the variables was also considered. For each set, the result typed in boldface indicates the approach which achieves the best prediction.

Table 5 reveals that the prediction ability of Pruned-ST2E is satisfactory. Across all the used sets, it has lower prediction error than ST2E and PGA. In comparison with other methods that are prediction-oriented, it can also outperform them in some cases. Moreover, the number of selected variables are much lower than the original dimensionality, which facilitates the interpretation. Therefore, Pruned-ST2E also has great potential to cope with variable selection tasks in classification problems.

Table 4 Ranking of variables in diabetes data set

Method	Ranking of variables (top → bottom)									
LARS	bmi	ltg	map	hdl	sex	glu	tc	tch	ldl	age
ST2E	bmi	ltg	map	sex	tc	ldl	hdl	glu	tch	age
Pruned-ST2E	bmi	ltg	map	tc	sex	hdl	ldl	glu	tch	age
PGA	bmi	ltg	map	hdl	sex	tc	ldl	glu	tch	age
Adaptive lasso	bmi	ltg	map	hdl	sex	tc	ldl	tch	glu	age
SCAD	bmi	ltg	map	hdl	sex	tc	ldl	glu	tch	age
Random forest	bmi	ltg	map	tch	hdl	ldl	glu	tc	sex	age
SCCE	bmi	ltg	map	hdl	tch	glu	sex	ldl	age	tc

Table 5 The test error rates and number of selected variables for each method on some classification data sets

Method Tr./Test/# Var.	Ionosphere 251/100/34	German 700/300/24	Sonar 108/100/60	Wdbc 469/100/30	Wpbc 144/50/32
No selection	12.30 ± 3.37	23.20 ± 1.47	31.90 ± 4.01	5.60 ± 2.01	28.98 ± 4.79
	13.90 ± 2.64	23.63 ± 1.60	34.10 ± 4.82	4.70 ± 2.54	25.31 ± 2.19
PGA	(14.80)	(14.60)	(31.90)	(16.50)	(15.60)
	14.20 ± 2.66	23.67 ± 1.47	32.80 ± 5.51	5.10 ± 2.18	24.90 ± 3.44
ST2E	(15.20)	(14.20)	(30.40)	(17.90)	(15.40)
	13.80 ± 2.70	**23.50 ± 1.43**	31.70 ± 3.83	4.80 ± 2.35	**22.81 ± 3.57**
Pruned-ST2E	(16.80)	(14.60)	(30.40)	(16.90)	(16.20)
	14.10 ± 3.93	23.63 ± 1.67	**25.90 ± 3.90**	**4.50 ± 1.90**	24.49 ± 3.19
Lasso	(17.40)	(18.80)	(18.40)	(14.50)	(3.70)
	14.10 ± 3.63	23.50 ± 1.72	26.40 ± 4.43	7.90 ± 3.45	27.55 ± 6.25
Adaptive lasso	(18.00)	(15.90)	(15.10)	(10.10)	(6.90)
	16.00 ± 2.87	24.13 ± 1.52	30.40 ± 2.90	4.90 ± 2.69	26.94 ± 4.69
SCAD	(15.40)	(15.50)	(20.40)	(13.50)	(8.10)
	16.50 ± 3.92	24.83 ± 0.95	27.60 ± 3.86	5.20 ± 2.82	26.53 ± 3.73
Random forest	(14.60)	(11.00)	(16.10)	(17.30)	(8.70)
	14.20 ± 2.53	23.93 ± 1.95	32.60 ± 4.83	5.30 ± 2.00	24.69 ± 4.46
mRMR	(16.10)	(13.30)	(33.10)	(15.70)	(17.50)

Bold indicates the best results

6 Conclusions and future work

In this paper, we proposed a novel method Pruned-ST2E to construct a more effective VSE by using an ensemble pruning strategy. In Pruned-ST2E, the ensemble members are first sorted and only some members which are ranked ahead are included into the fusion process, to produce a subensemble. Based on the average importance measure, the variables are then ranked and further selected by a thresholding rule. Although the idea of Pruned-ST2E is simple, a large batch of experiments conducted with both simulated and real-world data demonstrates its better or competitive performance in comparison with several other counterparts. By the aid of accuracy–diversity analysis of ST2E and Pruned-ST2E, we found that the inserted pruning step excludes less accurate members and makes the remaining members become more concentrated on the true importance vector.

In essence, the pruning technique used in Pruned-ST2E can be easily extended to some other techniques (such as PGA, stability selection) to build VSEs. It would be interesting to study how the corresponding subensembles will perform when coping with different variable selection tasks. On the other hand, Pruned-ST2E retains a prescribed number of ensemble members to integrate into a VSE. The ideal situation is to automatically determine how many members should be kept so that selection accuracy can reach as high as possible. Because the ground-truth information cannot be known in practice and the selection performance of a VSE depends on many factors, this is a very difficult and challenging problem deserved to be studied further. In recent years, data-driven feature selection techniques [2, 4] has received increasing attention due to their good performance and versatile applications. Thus, it is also deserved to explore whether selective ensemble learning can further enhance their performance.

Acknowledgements The authors would like to thank the editor and reviewers for their useful comments which helped to improve the paper. This research was supported by the National Natural Science Foundation of China (Nos. 11671317, 61572393) and the National Research Foundation of Korea (No. NRF-2012R1A1A2041661).

References

1. Breiman L (1996) Heuristics of instability and stabilization in model selection. Ann Stat 24(6):2350–2383
2. Cai J, Luo JW, Wang SL, Yang S (2018) Feature selection in machine learning: a new perspective. Neurocomputing 300:70–79
3. Che JL, Yang YL (2017) Stochastic correlation coefficient ensembles for variable selection. J Appl Stat 44(10):1721–1742
4. Che JL, Yang YL, Li L, Bai XY, Zhang SH, Deng CZ (2017) Maximum relevance minimum common redundancy feature selection for nonlinear data. Inf Sci 409–410:68–86
5. Chung D, Kim H (2015) Accurate ensemble pruning with PL-bagging. Comput Stat Data Anal 83:1–13
6. Efron B, Hastie T, Hohnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499
7. Fan JQ, Li RZ (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96(456):1348–1360
8. Fan JQ, Lv JC (2008) Sure independence screening for ultrahigh dimensional feature space (with discussions). J R Stat Soc (Ser B) 70(5):849–911
9. Fan JQ, Lv JC (2010) A selective overview of variable selection in high dimensional feature space. Stat Sin 20(1):101–148
10. Fakhraei S, Soltanian-Zadeh H, Fotouhi F (2014) Bias and stability of single variable classifiers for feature ranking and selection. Exp Syst Appl 41(15):6945–6958
11. Genuer R, Poggi JM, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Rocognit Lett 31(14):2225–2236
12. Griffin J, Brown P (2017) Hierarchical shrinkage priors for regression models. Bayes Anal 12(1):135–159
13. Kuncheva LI (2014) Combining pattern classifiers: methods and algorithms, 2nd edn. Wiley, Hoboken
14. Dua D, Graff C (2019) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed Dec 2016
15. Martínez-Muñoz G, Suárez A (2007) Using boosting to prune boosting ensembles. Pattern Recognit Lett 28(1):156–165
16. Martínez-Muñoz G, Hernández-Lobato D, Suárez A (2009) An analysis of ensemble pruning techniues based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell 31(2):245–259
17. Meinshausen N, Bühlmann P (2010) Stability selection (with discussion). J R Stat Soc B 72(4):417–473
18. Mendes-Moreira J, Soares C, Jorge AM, de Sousa JF (2012) Ensemble approaches for regression: a survey. ACM Comput Surv 45(1):40 Article 10
19. Miller A (2002) Subset selection in regression, 2nd edn. Chapman & Hall/CRC Press, New Work
20. Nan Y, Yang YH (2014) Variable selection diagnostics measures for high-dimensional regression. J Comput Graph Stat 23(3):636–656
21. Peng HC, Long FH, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intel 27(8):1226–1238
22. Rokach L (2016) Decision forest: twenty years of research. Inf Fus 27:111–125
23. Sauerbrei W, Buchholz A, Boulesteix AL, Binder H (2015) On stability issues in deriving multivariable regression models. Biometrical J 57(4):531–555
24. Subrahmanya N, Shin YC (2013) A variational Bayesian framework for group feature selection. Intern J Mach Learn Cybern 4(6):609–619
25. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc B 58(1):267–288
26. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. J R Stat Soc (Ser B) 63(2):411–423
27. Wang SJ, Nan B, Rosset S, Zhu J (2011) Random lasso. Ann Appl Stat 5(1):468–485
28. Xin L, Zhu M (2012) Stochastic stepwise ensembles for variable selection. J Comput Graph Stat 21(2):275–294
29. Zhang CX, Wang GW, Liu JM (2015) RandGA: injecting randomness into parallel genetic algorithm for variable selection. J Appl Stat 42(3):630–647
30. Zhang CX, Zhang JS, Kim SW (2016a) PBoostGA: pseudo-boosting genetic algorithm for variable ranking and selection. Comput Stat 31(4):1237–1262
31. Zhang CX, Ji NN, Wang GW (2016b) Randomizing outputs to increase variable selection accuracy. Neurocomputing 218:91–102
32. Zhang CX, Zhang JS, Yin QY (2017) A ranking-based strategy to prune variable selection ensembles. Knowl Based Syst 125:13–25
33. Zhou ZH, Wu JX, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intel 137(1–2):239–263
34. Zhu M, Chipman HA (2006) Darwinian evolution in parallel universes: a parallel genetic algorithm for variable selection. Technometrics 48(4):491–502
35. Zhu M, Fan GZ (2011) Variable selection by ensembles for the Cox model. J Stat Comput Simul 81(12):1983–1992
36. Zou H (2006) The adaptive lasso and its oracle properties. J Am Stat Assoc 101(476):1418–1429