Communications for Statistical Applications and Methods
2022, Vol. 29, No. 6, 629–640

https://doi.org/10.29220/CSAM.2022.29.6.629
Print ISSN 2287-7843 / Online ISSN 2383-4757

Ensemble variable selection using genetic algorithm

Seogyoung Lee[a], Martin Seunghwan Yang[a], Jongkyeong Kang[b], Seung Jun Shin[1,a]

[a]Department of Statistics, Korea University, Korea;
[b]Department of Information Statistics, Kangwon National University, Korea

Abstract

Variable selection is one of the most crucial tasks in supervised learning, such as regression and classification. The best subset selection is straightforward and optimal but not practically applicable unless the number of predictors is small. In this article, we propose directly solving the best subset selection via the genetic algorithm (GA), a popular stochastic optimization algorithm based on the principle of Darwinian evolution. To further improve the variable selection performance, we propose to run multiple GA to solve the best subset selection and then synthesize the results, which we call ensemble GA (EGA). The EGA significantly improves variable selection performance. In addition, the proposed method is essentially the best subset selection and hence applicable to a variety of models with different selection criteria. We compare the proposed EGA to existing variable selection methods under various models, including linear regression, Poisson regression, and Cox regression for survival data. Both simulation and real data analysis demonstrate the promising performance of the proposed method.

Keywords: genetic algorithm, ensemble learning, generalized linear model, cox regression

1. Introduction

Variable selection has become of great interest in supervised learning, such as regression and classification in various applications. Suppose we are given a pair of random variables $(Y, \mathbf{X}) \sim F_{Y,\mathbf{X}}$ where Y is a univariate response, $\mathbf{X} = (X_1, \ldots, X_p)^T$ is a p-dimensional predictor, and $F_{Y,\mathbf{X}}$ denotes their joint distribution. The goal of variable selection is to identify an index set $\mathcal{S}$ defined as

$$\mathcal{S} = \left\{ j \mid F_{Y|\mathbf{X}} \text{ functionally depends on } X_j,\ j = 1, \ldots, p \right\}, \tag{1.1}$$

where $F_{Y|\mathbf{X}}$ is conditional distribution of Y given $\mathbf{X}$.

Variable selection often assumes the relationship between Y and $\mathbf{X}$ is completely determined through a linear function of $\mathbf{X}$, i.e., $\boldsymbol{\beta}^T\mathbf{X}$ where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is a parameter of interest. To be more precise, we assume

$$Y \perp \mathbf{X} \mid \boldsymbol{\beta}^T\mathbf{X}, \tag{1.2}$$

where $\perp$ denotes the statistical independence. The assumption (1.2) covers a wide variety of models, including linear regression, generalized linear model, Cox regression, and single index model, to name a few. Under (1.2), $\mathcal{S}$ is equivalently written as

$$\mathcal{S} = \left\{ j \mid \beta_j \neq 0,\ j = 1, \ldots, p \right\}.$$

[1] Corresponding author: Department of Statistics, Korea University, 145 Anam-Ro, Sungbuk-Gu, Seoul 02841, Korea. E-mail: sjshin@korea.ac.kr

Published 30 November 2022 / journal homepage: http://csam.or.kr

In this article, we focus on the parametric models under (1.2), and let $\ell(\boldsymbol{\beta}; Y, \mathbf{X})$ denotes a log-likelihood function of the model. In population level, it is natural to assume that the true model with informative variables in $\mathcal{S}$ achieves the optimal efficiency. That is, we have

$$\mathcal{S} = \underset{\mathcal{S}\subseteq\{1,\ldots,p\}}{\operatorname{argmax}} \sup_{\boldsymbol{\beta}_{\mathcal{S}}} E\left\{\ell\left(\boldsymbol{\beta}_{\mathcal{S}}; Y, \mathbf{X}_{\mathcal{S}}\right)\right\}, \tag{1.3}$$

where $\mathbf{v}_{\mathcal{U}} = \{v_j \mid j \in \mathcal{U}\}^T$ for a given vector $\mathbf{v} = (v_1, \ldots, v_p) \in \mathbb{R}^p$ and an index set $\mathcal{U} \subseteq \{1, \ldots, p\}$. Given a set of data $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, the best subset selection $\hat{\mathcal{S}}$ solves an empirical version of (1.3) given as follows.

$$\hat{\mathcal{S}} = \underset{\mathcal{S}\subseteq\{1,\ldots,p\}}{\operatorname{argmax}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(\hat{\boldsymbol{\beta}}_{\mathcal{S}}; y_i, \mathbf{x}_{i,\mathcal{S}}\right) - C \cdot |\mathcal{S}|, \tag{1.4}$$

where $\hat{\boldsymbol{\beta}}_{\mathcal{S}}$ denotes the maximum likelihood estimator (MLE) of $\boldsymbol{\beta}_{\mathcal{S}}$, i.e., $\hat{\boldsymbol{\beta}}_{\mathcal{S}} = \operatorname{argmax}_{\boldsymbol{\beta}_{\mathcal{S}}} 1/n \sum_{i=1}^{n} \ell(\boldsymbol{\beta}_{\mathcal{S}}; y_i, \mathbf{x}_{i,\mathcal{S}})$. We remark that the second term is necessary to prevent over-fitting in the sample level. Here, C is a non-negative constant, and the popular choice of C include $\log n$ and 2 which are known as Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC), respectively.

Although it seems straightforward, (1.4) is a combinatorial problem and thus infeasible unless p is restrictively small. To tackle this, one can solve (1.4) sequentially. Sequential variable selection methods include forward selection and backward elimination and have become greatly popular in practice due to their simplicity. However, the sequential methods are inevitably sub-optimal since they cannot search the entire space of the model.

In the meantime, the penalization has been another canonical approach of the variable selection since the seminal least absolute shrinkage and selection operator (LASSO) proposed by Tibshirani (1996). The penalized variable selection solves the following problem to seek a sparse estimator $\hat{\boldsymbol{\beta}}$.

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(\boldsymbol{\beta}; y_i, \mathbf{x}_i\right) - \sum_{j} p_\lambda\left(|\beta_j|\right),$$

where $p_\lambda(\beta)$ denotes the sparsity-pursuing penalty function with λ being a non-negative tuning parameter that controls the degree of sparsity. Popular choices of the penalty function include, but not are not limited to LASSO, elastic net (Zou and Hastie, 2005), group lasso (Yuan and Lin, 2006), smoothly clipped absolute deviance (SCAD) (Fan and Li, 2001) penalty, and minimax concave penalty (MCP) (Zhang, 2010) penalty. The penalized approach estimate $\hat{\mathcal{S}}$ as $\{j \mid \hat{\beta}_j \neq 0, j = 1, \ldots, p\}$ where $\hat{\beta}_j$ denotes the j^{th} element of $\hat{\boldsymbol{\beta}}$. The penalized variable selection is also closely connected to the best subset selection. It is well known that L_0-penalized variable selection is equivalent to the best subset selection, and all the aforementioned penalties can be viewed as a continuous approximation of the L_0-penalty.

Our work is motivated by the following question: Can we solve (1.4) directly? In this paper, we propose to solve (1.4) via genetic algorithm (GA), a popular stochastic search algorithm motivated by the principle of Darwinian evolution. GA is an effective tool for solving NP–hard optimization problems such as the best subset selection. See for example, Leardi and Gonzalez (1998); David *et al.* (1997) and Niazi and Leardi (2012).

In the meantime, Zhu and Chipman (2006) proposed to parallel version of GA-based variable selection to enhance the variable selection by running multiple GA with several initial values and then

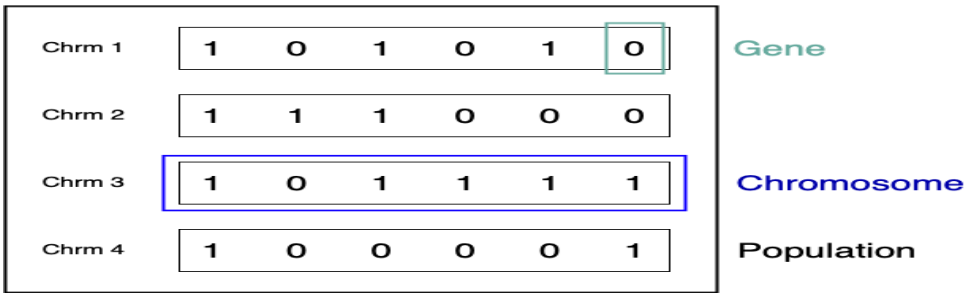


Figure 1: *Illustration of genetic representation of GA: gene, chromosome, and population.*

synthesizing the multiple results. One can notice that their idea can be viewed as an ensemble variable selection.

The success of the ensemble variable selection depends on the diversity of base selectors to be aggregated for the final result. There are two ways to introduce the diversity in the ensemble variable selection. First type is generate randomness from the data by using resampling method like bootstrap or subsampling. See Zhu and Fan (2011); Wang *et al.* (2011), and Meinshausen and Bühlmann (2010). Another way to introduce the diversity is to use a stochastic base selector. The parallel version of GA-based variable selection proposed by Zhu and Chipman (2006) is a canonical example for this type. See also Xin and Zhu (2012) and Zhang *et al.* (2016).

In this article, we extend the idea of Zhu and Chipman (2006), which focuses only on the linear regression model, to more general models satisfying (1.2) that include the generalized linear model and cox-regression for censored responses as a special case. Although our proposal is technically identical to Zhu and Chipman (2006), except for extension beyond the linear regression, we focus more on the ensemble aspect of the method instead of the GA aspect. Namely, Zhu and Chipman (2006) understand their algorithm as a variant of GA, but we regard it as a version of ensemble variable selection. From our point of view, GA is not essential, and any stochastic variable selection method can be used as a base selector, which brings a natural extension of the method. For this reason, we call the method ensemble variable selection based on the genetic algorithm (to be referred to as EGA hereafter) instead of the parallel GA algorithm named by Zhu and Chipman (2006).

The proposed EGA has a clear advantage in practice due to its conceptual simplicity. We demonstrate its promising performance through a simulation study and real data analysis under various models beyond the classical linear regression.

The remainder of the paper is organized as follows. In Section 2, we introduce GA and describe how to solve the best subset selection problem using GA. In Section 3, the ensemble extension of the GA-based variable selection is described in great detail. Simulation studies to evaluate the finite sample performance of the EGA are conducted in Section 4, and illustration to real data examples is given in Section 5. Concluding remarks follow in Section 6.

2. Variable selection via genetic algorithm

We start by briefly reviewing GA. Three components consist of GA: genetic representation, fitness function, and evolutionary operators. First, *genetic representation* is defined as a coded representation of input variable of objective function to be optimized. In the variable selection context, $\mathcal{S}$ can be equivalently encoded as a binary array of length p, denoted as $\mathbb{1}_{\mathcal{S}}$ whose j^{th} element is 1 if $j \in \mathcal{S}$ and 0 otherwise, where $j = 1, \ldots, p$. We call this binary array a *chromosome*, and each of its digits

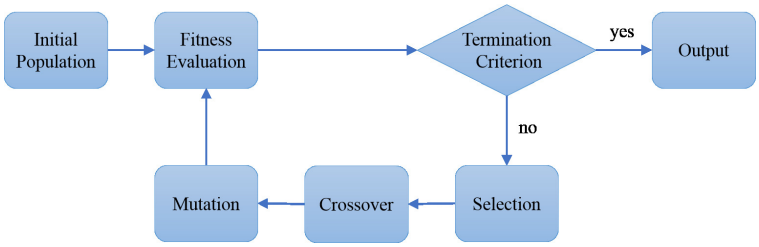


Figure 2: *A flowchart of the genetic algorithm.*

a *gene*. To run GA, we need a multiple number of chromosomes. A set of chromosomes is called a *population*. See Figure 1 for the graphical description of the genetic representation.

Second, the *fitness function* measures the quality of a current solution and is often an objective function to be optimized. Without loss of generality, we use Bayesian Information Criterion (BIC) that solves the best subset selection (1.4) with $C = \log n$.

Once we decide our genetic representation and fitness function, we only need to iteratively apply the *genetic operators*, the last component of GA. There are three operators, *selection*, *crossover*, and *mutation*. Having a population of chromosomes in hand, we evaluate the fitness of each chromosome. The *selection* operator chooses those chromosomes with high fitness, for instance, the top 50% of the population. The crossover operator randomly selects two chromosomes among the survived chromosomes. Then flip a coin p times to select indices of the genes to be swapped, where p denotes the length of a chromosome. Finally, genes that correspond to randomly selected indices of selected chromosomes are swapped. This operation gives two new chromosomes with mixed genes of survived chromosomes, possibly having above-average fitness. Applying the crossover operators sufficiently many times, we construct a new generation of the population having the same size as the previous one. The *mutation* operator is then applied as follows. Once we have a new population, we flip each digit in the chromosomes with a small probability. In this way, we may maintain our search space moderately large.

The main idea of GA is that if we iterate this process repeatedly, the average fitness of chromosomes in the population will gradually increase. One can allow the best organisms of the present generation to move to the next generation without change, which is called *Elitism*. This strategy ensures that the quality of the solution obtained by GA does not degrade from one generation to the next. The iteration is over when the termination criterion is met. We may either set the upper bound in the number of generations or iterate it until some homogeneity is achieved in the population. When the iteration is over, the chromosome with the highest fitness is our desired output. Finally, Figure 2 summarizes GA.

3. Ensemble GA for variable selection

Since the initial population is arbitrary, running GA essentially independently results in slightly different outcomes. Most GA grasp the informative variables well, while erroneously selected noise variables differ among each GA. This motivates to repeat the GA run multiple times to reduce the false positives and ensemble multiple GA runs by taking a digit-wise average of the outputs. We call this ensemble GA (EGA). Suppose we run GA K-times independently to solve (1.4). Let $\mathbb{1}_{\hat{\mathcal{S}}_k}$ be the solution (i.e., *Chromosome* that yields the highest fitness) of the k^{th} GA run for $k = 1, \ldots, K$ and c_{kj} denotes the j^{th} gene of the $\mathbb{1}_{\hat{\mathcal{S}}_k}$. Let us define $m_j, j = 1, \ldots, p$ as the relative frequency of the j

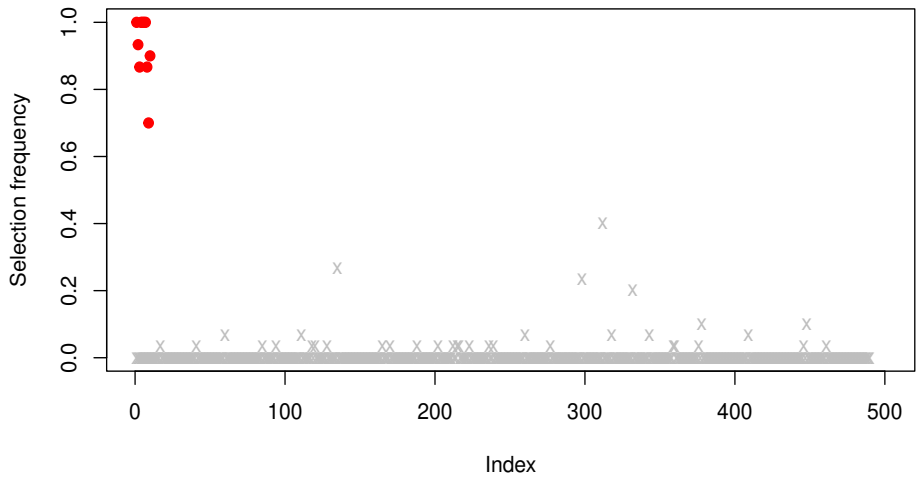


Figure 3: *Selection frequency $m_1, \ldots, m_p$ for a simulated example. The first 10 informative variables (red dots) are all close to 1 while others (gray 'x' symbols) are not. If we set $\delta = 0.5$, the EGA successively selects all informative variables.*

predictor selected during the K independent GA runs.

$$m_j = \frac{1}{K} \sum_{k=1}^{K} c_{kj}, \quad j = 1, \ldots, p.$$

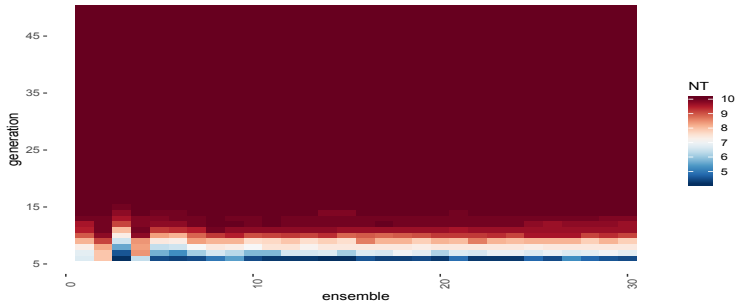
Since single GA performs reasonably well, m_j is significantly larger than 0 if j^{th} predictor is informative one, and not otherwise. Figure 3 illustrates m_j, $j = 1, \ldots, p$ computed from 30(= K) independent GA runs for an illustrative data generated as example 1 with $p = 500$ given in Section 4, but different $\boldsymbol{\beta}$ structure. In this example, the first ten variables are informative and the rest are noises. One can observe a clear distinction between the informative and noise variables. Therefore we can conclude the j^{th} variable is informative when $m_j > \delta$.

Now, the next question is how to choose the cutoff value δ. We propose δ to be

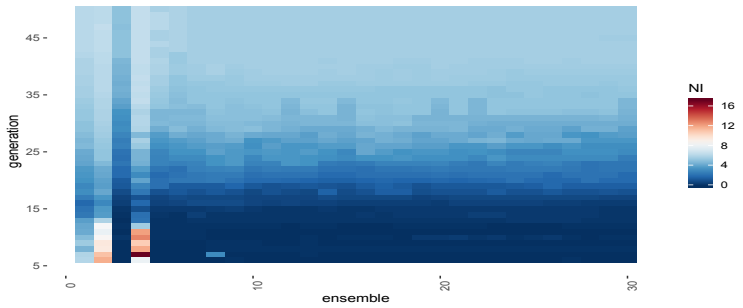
$$\delta = \frac{(m_{(i^*+1)} + m_{(i^*)})}{2},$$

where $i^* = \underset{i}{\text{argmax}} \ (m_{(i+1)} - m_{(i)})$ with $m_{(i)}$ being the i^{th} ordered value of m. That is, δ is the half of the the largest gap among the sorted $m_{(1)} \leq \cdots \leq m_{(p)}$.

Finally, we investigate how sensitive the EGA is to the number of independent GA to the ensemble (K), and the number of generations for each GA, we denote n_g. Figure 4 depicts the heat-map of the number of (a) informative variables and (b) noise variables selected by the EGA for the previous illustrative example for different combinations of (K, n_g). One crucial observation is that n_g needs not be very large in EGA unless K is too small. This is practically attractive in practice since each run of EGA can be easily parallelized, which drastically reduces the computational burden of EGA.



(a) Number of selected informative variables



(b) Number of selected uninformative variables

Figure 4: *Effect of K and n_g for EGA: n_g needs not to be large as long as K is not very small.*

4. Simulation studies

To assess the performance of the proposed methods based on the GA, we conduct simulation studies under three popular models satisfying the linearity assumption (1.2): linear regression, Poisson regression for counting response, and Cox proportional hazard (PH) regression for censored response. We try the logistic regression, one of the most popular classification methods, yet, `glm()` function often fails to converge during the GA. The results are omitted. As competing methods, we consider LASSO- and SCAD-penalized regressions, both of which are implemented in `R` (resp. `glmnet` and `ncvreg` pakages). We use BIC for example 1 and 2, and a modified BIC (Volinsky and Raftery, 2000) for the Cox PH regression example 3 to select variables.

In all scenarios, we have eight predictors to be informative variables out of $p \in \{20, 50, 100\}$ variables, and set $n = 200$. We generate all predictors from p-variate normal distribution with mean 0 and three different covariance variances $\mathbf{\Sigma}$, i) independent $\{\mathbf{\Sigma}\}_{ij} = 0, \forall i \neq j$, ii) autoregressive $\{\mathbf{\Sigma}\}_{ij} = 0.5^{|i-j|}, \forall i \neq j$, and iii) compound symmetry $\{\mathbf{\Sigma}\}_{ij} = 0.5, \forall i \neq j$. We find that the patterns of simulation results are similar for all covariance structures, and report independent case only to avoid redundancy. See Online Supplementary Materials for the results for the autoregressive and the compound symmetry covariance cases.

We set $n_g = 200$ for GA, and $n_g = 10$ for EGA with $K = 100$. For both versions of GA, the population size is set as $40, 60$, and 80 for $p = 20, 50$, and 100, respectively. We set twenty percent of genes in each chromosome to have a value of 1 at an initial population and allow the top ten-

Table 1: Linear regression for independent predictors

p	Methods	NT	NI	NC	ER
20	LASSO	8.00 (0.00)	0.00 (0.00)	100	0.293 (0.177)
	SCAD	8.00 (0.00)	0.00 (0.00)	100	0.044 (0.023)
	GA	8.00 (0.00)	0.30 (0.56)	75	0.054 (0.033)
	EGA	8.00 (0.00)	0.15 (0.36)	85	0.049 (0.027)
50	LASSO	8.00 (0.00)	3.93 (2.40)	3	0.165 (0.062)
	SCAD	8.00 (0.00)	0.13 (0.53)	92	0.045 (0.021)
	GA	8.00 (0.00)	1.33 (1.26)	29	0.092 (0.054)
	EGA	8.00 (0.00)	0.09 (0.32)	92	0.049 (0.028)
100	LASSO	8.00 (0.00)	4.43 (3.06)	7	0.220 (0.074)
	SCAD	8.00 (0.00)	0.11 (0.57)	94	0.041 (0.022)
	GA	8.00 (0.00)	2.80 (1.93)	8	0.143 (0.081)
	EGA	8.00 (0.00)	0.03 (0.17)	97	0.042 (0.025)

Four performance measures averaged over 100 independent repetitions are reported.
The numbers in parentheses are corresponding standard errors.

Table 2: Poisson regression for independent predictors

p	Methods	NT	NI	NC	ER
20	LASSO	7.82 (0.39)	0.13 (0.34)	82	0.388 (0.289)
	SCAD	7.91 (0.38)	0.04 (0.20)	93	0.134 (0.286)
	GA	8.00 (0.00)	0.29 (0.50)	73	0.024 (0.014)
	EGA	8.00 (0.00)	0.07 (0.26)	93	0.020 (0.013)
50	LASSO	7.99 (0.10)	4.18 (2.07)	0	0.154 (0.079)
	SCAD	7.96 (0.40)	0.12 (0.38)	89	0.086 (0.188)
	GA	8.00 (0.00)	0.83 (0.90)	44	0.042 (0.030)
	EGA	8.00 (0.00)	0.07 (0.26)	93	0.024 (0.015)
100	LASSO	7.99 (0.10)	5.77 (2.67)	0	0.204 (0.095)
	SCAD	7.98 (0.14)	0.20 (0.45)	80	0.081 (0.180)
	GA	8.00 (0.00)	1.69 (1.32)	20	0.059 (0.037)
	EGA	8.00 (0.00)	0.02 (0.20)	99	0.020 (0.014)

Four performance measures averaged over 100 independent repetitions are reported.
The numbers in parentheses are corresponding standard errors.

percent best elite (i.e., highest BIC) chromosomes in the current generation to survive (i.e., move on to the next generation without changing). To compare the performance, we consider the following four measures. We record the average selection frequency of informative and non-informative predictors denoted by NT and NI, respectively, to evaluate the variable selection performance. We also count the cases identifying the true model perfectly and denote it by NC. In addition, we also compare the estimation error by computing ER $= (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \Sigma (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$. Note that the proposed methods give the variable selection result $\hat{S}$ only. To compute the estimation error, we fit a model best on the selected predictors only. On the other hand, the penalized methods directly give $\hat{\boldsymbol{\beta}}$.

Finally, We repeat the simulation 100 times independently and report the averaged values over the repetitions.

4.1. Linear regression

For the linear regression model $y_i \mid \mathbf{x}_i \sim N(\boldsymbol{\beta}^T \mathbf{x}_i, \sigma^2)$ with $\sigma^2 = 1$. The regression parameter $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ is

$$\beta_j = \begin{cases} \beta_j = 1, & j \in \{1, 2, 3, 4, 6, 7, 9, 10\}, \\ \beta_j = 0, & \text{otherwise.} \end{cases}$$

Table 1 shows the comparison results under the linear model with independent predictors. One can see that all methods are comparable in terms of NT. However, GA and LASSO have too many false positives (i.e., high NI). EGA and SCAD perform similarly and show nearly perfect performance, while EGA is slightly but consistently better than SCAD. We also like to point out that both EGA and SCAD are stable even for large p.

4.2. Poisson regression

We consider Poisson regression model $y_i \mid \mathbf{x}_i \overset{iid}{\sim} \text{Poisson}\{\exp(\boldsymbol{\beta}^T \mathbf{x}_i)\}$. The regression coefficient is set to be

$$\beta_j = \begin{cases} \beta_j = 0.5, & j \in \{1, 2, 3, 4, 6, 7, 9, 10\}, \\ \beta_j = 0, & \text{otherwise.} \end{cases}$$

Table 2 contains the simulation result under Poisson regression. Under this example, the penalized variable selection method does not perform very well. However, the proposed EGA still performs perfectly, as in the previous example. The single GA looks okay but identifies too many noise predictors. In terms of the model error, EGA outperforms all others.

4.3. Cox proportional hazard regression

We consider the hazard function $\lambda(t|\mathbf{x}_i) = \lambda_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i)$ with constant baseline hazard $\lambda_0(t) = 1$ in order to generate true survival time t_i. We then simulate censoring time from Uniform$(0, M)$ where M is determined so as to have censoring rate 30%. We can obtain a pair of censored responses (y_i, Δ_i) where $y_i = \min\{t_i, c_i\}$ and $\Delta_i = I\{c_i \geq t_i\}$ with $I(\cdot)$ being an indicator function. The regression coefficients are set to be

$$\beta_j = \begin{cases} \beta_j = 0.7, & j \in \{1, 2, 3, 4, 6, 7, 9, 10\}, \\ \beta_j = 0, & \text{otherwise.} \end{cases}$$

The simulation results for the Cox PH regression are given in Table 3, like other examples, EGA method seems to outperform all the other methods in all the simulation settings.

We remark that the effect of the ensemble becomes clearer as p increases for all models under consideration. One possible explanation is that the GA tends to have more false positives as p increases, yet these falsely selected noise variables are removed in the EGA.

5. Real data analysis

5.1. Slump flow data

Slump flow data (Yeh, 2007) includes 103 observations with seven predictors (*cement*, *ash*, *slag*, *water*, *sp*, *ca*, and *fa*) associated with three response variables (*slump*, *flow*, and *compressive strength*). Slumps and flows measure the viscosity of concrete. Slumps are vertical height, the horizontal distance through which the wet concrete cone is sagging and flowing. The slump and flow of concrete are affected by its moisture content and its relationship with other concrete components. The goal is to model the slump and flow with seven concrete components. Yeh (2007) adopted the second-order regression $Y = \sum_{j=1}^{p} \beta_j X_j + \sum_{j<k}^{p} \beta_{jk} X_j X_k$. Following Yeh (2007), we also employed the second-order regression model in our analysis, and thus have twenty eight predictors in the model.

We randomly partitioned the data into training and test sets with 80 and 23, respectively. Table 4 summarizes the mean squared error (MSE) and the selection frequencies for leading variables for

Table 3: Cox regression for independent predictors

p	Methods	NT	NI	NC	ER
20	LASSO	7.91 (0.81)	0.01 (0.10)	98	1.154 (0.606)
	SCAD	7.91 (0.81)	0.02 (0.14)	97	0.460 (0.719)
	GA	8.00 (0.00)	0.43 (0.61)	63	0.124 (0.084)
	EGA	8.00 (0.00)	0.24 (0.45)	77	0.112 (0.079)
50	LASSO	8.00 (0.00)	1.84 (1.45)	22	0.901 (0.297)
	SCAD	8.00 (0.00)	0.74 (1.04)	54	0.154 (0.104)
	GA	8.00 (0.00)	1.65 (1.50)	26	0.264 (0.209)
	EGA	8.00 (0.00)	0.18 (0.46)	85	0.123 (0.093)
100	LASSO	7.92 (0.80)	1.86 (1.61)	22	1.198 (0.416)
	SCAD	8.00 (0.00)	1.29 (1.24)	31	0.209 (0.155)
	GA	8.00 (0.00)	3.79 (2.37)	4	0.531 (0.450)
	EGA	8.00 (0.00)	0.07 (0.29)	94	0.100 (0.077)

Four performance measures averaged over 100 independent repetitions are reported.
The numbers in parentheses are corresponding standard errors.

Table 4: Mean squared error and the number of selected variables for Slump flow data

Response	Method	MSE	NV	*slag*	*water*	*cement*×*sp*	*slag*×*water*	*slag*×*ca*	*slag*×*fa*	others
slump	GA	15.10 (6.19)	5.90 (1.23)	0.96	1.00	0.76	0.99	0.84	0.89	0.46
	EGA	15.31 (5.26)	5.02 (1.36)	0.93	1.00	0.57	0.98	0.65	0.75	0.13
	LASSO	19.74 (6.10)	3.12 (2.10)	0.53	0.98	0.46	0.52	0.00	0.29	0.34
	SCAD	16.23 (5.55)	5.28 (2.07)	0.88	1.00	0.75	0.88	0.49	0.65	0.63
flow	GA	13.34 (3.58)	5.44 (1.79)	0.99	1.00	0.49	0.70	0.53	0.71	1.02
	EGA	13.23 (3.28)	3.77 (1.44)	0.99	1.00	0.29	0.51	0.26	0.51	0.21
	LASSO	14.41 (3.40)	2.33 (0.83)	0.91	1.00	0.20	0.02	0.00	0.03	0.17
	SCAD	13.81 (3.35)	2.21 (0.86)	0.90	1.00	0.13	0.07	0.02	0.04	0.05

A×B indicates the interaction term between A and B.

each method, based on 100 independent replications. Given that the number of predictors is only 28(= p), we reduce the number of generations for GA to twenty-five. For EGA methods, the number of generations and ensemble repetition are set same as simulation studies, i.e., (n_g = 10 and K = 100). All methods select *water* almost always for both responses, *slag* is also frequently selected by all methods. GA based method seems to have slightly better performance in terms of test MSE.

5.2. Medical care data

Medical care data (Partha and Pravin, 1997) intend for 4,406(= n) individuals aged 66 and over who are covered by the public insurance program. The goal is to model the demand for medical care by features available to patients, and we adopt the number of physician office visits as the response variable. As predictors, four continuous variables(*nchr*: the number of chronic conditions; *nsch*: the number of years of education; *age*: a person's age; *income*: family income) and seven binomial variables(*adl*: condition that limits activities of daily living; and *race*: black or other; *sex*: male or female; *married*: married or not; *employed*: employed or not; *privins*: covered by private insurance; *pubins*: covered by public insurance) and two categorical variables(*health*: poor, average, excellent; *region*: northeastern, midwestern, western, other) are used. Finally, we assume the second-order Poisson regression in the similar manner of the slump flow data set, and compare the variable selection performance.

We randomly partitioned the data into training and test sets with 3,000 and 1,406, respectively. Table 5 summarizes the mean absolute error (MAE) and the number of selected variables for each

Table 5: Mean absolute error and the number of selected variables for Medical care data

	GA	EGA	LASSO	SCAD
MAE	4.22 (0.09)	4.23 (0.10)	4.17 (0.08)	4.19 (0.09)
NV	44.92 (3.85)	19.95 (8.83)	34.52 (8.73)	26.18 (5.36)

Table 6: Variable selection result for the PBC data

Methods	*age*	*asc*	*oed*	*bili*	*alb*	*cop*	*ast*	*prot*	*stage*
LASSO	0.016(0.005)	0.102(0.061)	0.658(0.186)	0.084(0.013)	−0.557(0.140)	0.003(0.001)	0.001(0.001)	0.130(0.043)	0.273(0.068)
SCAD	0.011(0.003)	0(-)	0.396(0.122)	0.114(0.015)	−0.461(0.111)	0.004(0.001)	0(-)	0.074(0.025)	0.225(0.053)
GA	0.029(0.010)	0(-)	0.978(0.338)	0.103(0.018)	−0.763(0.273)	0.003(0.001)	0(-)	0(-)	0.442(0.144)
EGA	0.029(0.010)	0(-)	0.978(0.338)	0.103(0.018)	−0.763(0.273)	0.003(0.001)	0(-)	0(-)	0.442(0.144)

method, based on 100 independent replications. Penalized methods seem to have better performance compared to GA-based methods in terms of MAE. The EGA method tended to select the smallest number of variables with comparable predictive performance.

5.3. Primary biliary cirrhosis (PBC) data

Primary biliary cirrhosis (PBC) data was gathered in the Mayo Clinic trial. PBC data is available through the R package 'survival' and is widely used in the survival analysis i.e.,Tibshirani (1997). PBC data consists of 418 patients with 17 covariates. Covariates are as follows:

trt : *D*-penicillamin treatment code, *age*, *sex*, *asc* : presence of ascite, *hep* : presence of hepatomegaly, *spid* : presence of spiders, *oed* : presence of oedema, *bili* : serum bilirubin, *chol* : serum cholesterol, *alb* : albumin, *cop* : urine copper, *alk* : alkaline phosphatase, *ast* : aspartate aminotransferase = SGOT, *trig* : triglycerides, *plat* : platelet counts, *prot* : prothrombin time, *stage* : histologic stage of disease.

Observations with missing values are omitted, and 276 complete cases with 17 covariates are used for analysis. The censoring rate is nearly 60%. Table 6 presents the estimated coefficients and corresponding standard errors by four methods. LASSO selects the largest number (9) of variables. And both GA and EGA select the least (6). Sets of variables chosen by LASSO, SCAD, GA, and EGA methods were somewhat similar. 6 number of covariates (*'age', 'oed', 'bili', 'alb', 'cop', and 'stage'*) are commonly selected.

6. Concluding remarks

In this paper, we study a novel variable selection method based on GA. The idea of ensemble significantly improves the variable performance, as demonstrated by a simulation study and real data analysis. Although the EGA method is computationally intensive, it is easily parallelizable. Since the proposed idea directly tackles the best subset selection, it is straightforward to understand and applicable to various statistical models with different model selection criteria.

The GA has a lot of tuning parameters that might affect the variable selection performance. This article explores a limited number of scenarios, and further study should be done to more clearly understand the proposed method.

The idea of the ensemble has been extensively applied to various statistical learning problems but not to variable selection. In the EGA method, a single run of GA plays the role of a weak learner. However, we can extend this idea by replacing other algorithms instead of GA, such as particle swarm optimization and differential evolution. This can be a future research direction.

Supplementary materials

The online Supplementary Materials contain the complete results of the simulation.

Acknowledgments

This work is funded by the National Research Foundation of Korea (NRF) grants (2018R1D1A1B070 43034, 2019R1A4A1028134) and Korea University (K2000461).

References

David B, Royston G, Alun J, Jem JR, and Douglas BK (1997). Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry, *Analytica Chimica Acta*, **348**, 71–86.

Partha D and Pravin KT (1997). Demand for medical care by the elderly: A finite mixture approach, *Journal of applied Econometrics*, **12**, 313–336.

Fan J and Li R (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association*, **96**, 1348–1360.

Leardi R and Gonzalez AL (1998). Genetic algorithms applied to feature selection in PLS regression: how and when to use them, *Chemometrics and Intelligent Laboratory Systems*, **41**, 195–207.

Meinshausen N and Bühlmann P (2010). Stability selection, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**, 417–473.

Niazi A and Leardi R (2012). Genetic algorithms in chemometrics, *Journal of Chemometrics*, **26**, 345–351.

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

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

Volinsky C and Raftery A(2000). Bayesian information criterion for censored survival models, *Biometrics*, **56**, 256–262.

Wang S, Nan B, Rosset S, and Zhu J (2011). Random lasso, *The Annals of Applied Statistics*, **5**, 468.

Xin L and Zhu M (2012). Stochastic stepwise ensembles for variable selection, *Journal of Computational and Graphical Statistics*, **21**, 275–294.

Yeh IC (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks, *Cement and Concrete Composites*, **29**, 474–480, Available from: https://doi.org/10.10-16/j.cemconcomp.2007.02.001

Yuan M and Lin Y (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**, 49–67.

Zhang CH (2010). Nearly unbiased variable selection under minimax concave penalty, *The Annals of Statistics*, **38**, 894–942.

Zhang CX, Zhang JS, and Kim SW (2016). PBoostGA: Pseudo-boosting genetic algorithm for variable ranking and selection, *Computational Statistics*, **31**, 1237–1262.

Zhu M and Chipman HA (2006). Darwinian evolution in parallel universes: A parallel genetic algorithm for variable selection, *Technometrics*, **48**, 491–502.

Zhu M and Fan G (2011). Variable selection by ensembles for the Cox model, *Journal of Statistical Computation and Simulation*, **81**,1983–1992.

Zou H and Hastie T (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**, 301–320.

Received March 08, 2022; Revised April 21, 2022; Accepted April 29, 2022