Variable Selection with Metaheuristic Methods

Myung Soon Song[1], Francis Vasko[2], Yun Lu[3], and
Kyle Callaghan[4]
[1234]Kutztown University of Pennsylvania, 15200 Kutztown Rd, Kutztown, PA 19530

Abstract
Variable selection (or feature selection) has been one of old topics in regression models. Beside many classical approaches, some metaheuristic approaches from the optimization research such as GA (Genetic Algorithm) or SA (Simulated Annealing) have been developed so far. These methods have a considerable advantage to deal with high dimensional problems over the conventional methods, but they must control associated fine-tuning parameters, which is very hard in practice. In this article, JAYA, one of the parameter-free approaches will be suggested and explored. Many methods such as GA, TBO, and JAYA will be compared to one another with the results from real-world datasets.

Key Words: variable selection, metaheuristics, optimization, Jaya

1. Introduction

Variable selection (or feature selection) is a classical topic in regression models which has many applications in many areas including, but not limited to, engineering, medicine, psychology, or business.

Among numerous variable selection methods developed, some classical sequential methods such as stepwise selection methods have been widely used because they are simple and work very well if there are not too many variables and they have low prediction error. But there are some drawbacks in these methods. Two most serious issues among them are (1) they tend to converge to local optima (Hans et al. (2012), Hocking (1976), Kiezun et al. (2009), Meiri & Zahavi (2006), Paterini & Minerva (2010)) and (2) they do not work very well in high dimensional spaces. (Hand et al. (2012), Kapetanios (2007)). Later in this section, it will be explained how these problems can be resolved with 'metaheuristics' in optimization research.

The selection of the most adequate variables in regression models can be stated as a combinatorial optimization problem with the objective to select explanatory variables that maximize the adequacy of the model according to statistical criteria (objective function). (Meiri (2006), Paterlini & Minerva (2010)) Some methods or algorithms from optimization research have been used for variable selection, including but not limited to, genetic algorithm (Broadhurst et al.(1997), Kapetanios (2007), Kiezun et al. (2009), Jirapech-Umpai & Aitken (2005), Mohan et al. (2018), Paterini & Minerva (2010), Peng et al. (2005),Sinha et al. (2015)), simulated annealing (Kiezun et al. (2009), Meiri & Zahavi (2006)), iterated local search (Hans et al., 2012). These methods are characterized as metaheuristics, a stochastic search strategy dedicated to solve difficult problems (NP-hard problems) in optimization research.

In particular, genetic algorithms (GA hereafter) and simulated annealing (SA hereafter) are known to be very effective to resolve the two issues mentioned above – (1) convergence to local optima (Kapatenios (2007), Kiezun et al. (2006), Meiri (2006), Paterini & Minerva (2010),) and (2) handling high dimensional spaces. (Kapatenious (2007), Meiri (2006))

Even if these metaheuristics (GA or SA) have good properties such as tending to reach the global optima and capability to deal with many variables, their performance heavily depend on the choice of 'tuning parameters', which is very experimental and time consuming in practice. For example, the GA and SA need to fine tune four parameters (crossover type, crossover rate, mutation type, and mutation rate) and five parameters (initial temperature, final temperature, cooling ratio, temperature function, and accept function), respectively.

To resolve these obvious and practical problems, this paper will suggest using 'parameter-free metaheuristics' for variable selection in regression – Jaya (Rao, 2016) and Teaching Based Optimization (TBO hereafter) (Rao et al., 2011).
In the next section, TBO and Jaya will be briefly described.

2. Approach

2.1 What is Teaching Based Optimization?

The Teaching-learning-based optimization (TLBO) metaheuristic is a two-phase population-based metaheuristic designed to solve continuous nonlinear optimization problems. It was proposed by Rao et al. (2011) as a method for solving large constrained mechanical design optimization problems which involve no specific parameters to tune. Since the tuning of parameters in other metaheuristics can often be time consuming and largely experimental, Rao et al. (2011) describe a procedure in which the only parameters that need to be specified are those common to all other metaheuristics--population size and termination criterion.

TLBO consists of two phases referred to by Rao et al. (2011) as the teaching phase and the learning phase. The first phase of TLBO, the teaching phase, utilizes a global search procedure which really uses intensification-focused moves as discussed in Hill and Pohl (2019). The "difference mean" is created by subtracting the quality of the best solution with the current mean solution. The objective here is to improve all solutions by this difference. The operator creating a new solution in the teaching phase is given as the following: $X_{new} = X_{old} + r\left(X_{teacher} - T_f \times X_{mean}\right)$, where X_{old} is a current solution of a population being modified, r is a random number in the range [0,1], $X_{teacher}$ is the best solution of a population, T_f = round(1+rand(0.1)) implying that T_f takes on the values 1 or 2 with equal probability and X_{mean} is the mean solution of a population (Rao et al, 2011). Here, two variables r and T_f could have been used as parameters; however, they are defined as being random numbers and therefore their values are **not** specified as input parameters. The teaching phase is completed by checking if the new solution is better than the current.

The second phase of TLBO adjusts each solution relative to a randomly selected solution (another learner). The learning phase involves diversification-focused moves as discussed in Hill and Pohl (2019). The operator is given by the following (for a minimization problem):

$$X_{i,new} = \begin{cases} X_i + r\left(X_i - X_j\right), & \text{if } f\left(X_i\right) < f\left(X_j\right) \\ X_i + r\left(X_j - X_i\right), & \text{otherwise} \end{cases} \quad (1)$$

where, similar to the teaching phase, r is randomly chosen in the range of [0,1], X_i is the current solution and X_j is a randomly chosen solution where $i \neq j$. For both phases of TLBO, since X_j is a vector of real numbers, the actual implementation of TLBO requires the use of these update formulas on each component of X_j. For more information on TLBO, we suggest reading Rao et al. (2011).

In this paper, Teaching-based Optimization (TBO), which is a special case of TLBO with only teaching phase will be used because the learning phase will be replaced with a local search which will also be incorporated into Jaya.

2.2 What is Jaya?

The Jaya metaheuristic by Rao (2016) is a single phase population-based metaheuristic designed to solve continuous nonlinear optimization problems. It is very similar to the teaching phase of TLBO except that a different transformation formula is used to update each solution in the current population. Specifically, if $X_{j,k,i}$ is the value of the jth variable for the kth candidate solution during the ith iteration, then this value is modified based on the equation $X^{new}_{j,k,i} = X_{j,k,i} + R1_{j,i}(X_{j,best,i} - X_{j,k,i}) - R2_{j,i}(X_{j,worst,i} - X_{j,k,i})$, where $X_{j,best,i}$ is the value of the variable j for the best candidate solution in the current population and $X_{j,worst,i}$ is the value of the variable j for the worst candidate solution in the current population. $X^{new}_{j,k,i}$ is the updated value of $X_{j,k,i}$ and $R1_{j,i}$ and $R2_{j,i}$ are two random numbers for the jth variable during the ith iteration in the range [0,1]. This transformation equation is trying to move the current solution toward the best solution and away from the worst solution. We suggest reading Rao (2016) for more details on Jaya.

2.3 Binarization of TLBO and Jaya

Both TLBO and Jaya are designed to solve continuous nonlinear optimization problems; whereas, variable selection is a zero-one constrained optimization problem (either a variable is in the model or not). The solutions in the population of a problem using the original versions of TLBO or Jaya will be vectors of real (rational) numbers. The solutions in the population for the variable selection problem are bit strings (zeros and ones). To adapt TLBO and Jaya to deal with bit strings, we used the approach that Lu and Vasko (2015) used successfully for the Set Covering Problem. In any of the transformation formulas (teaching, learning, or Jaya), the variables are now bits. The random numbers that took on any values between 0 and 1 now take on only 0 or 1 with equal probability. As in the original TLBO, the teaching factor in TLBO takes on the values 1 or 2 with equal probability. Also, in the teaching phase, the mean solution is replaced by the median solution. If, after a transformation formula is performed, a variable value is less than 0, it is set to 0. If it is greater than 1, it is set to 1. Intuitively, if the result of a transformation formula produces a variable that "wants" to have a value less than 0, we simply set it to 0. In a like manner, variables that "want" to have a value greater than 1 are set to 1. The empirical results will demonstrate that this simple binarization approach yields good results. Additionally, it is important to note that there are other (more complicated) approaches in the literature for binarization of metaheuristics originally designed to solve continuous nonlinear optimization problems (Lanza-Gutierrez, 2016). However, Vasko and Lu (2017) reported that the simple approach outlined above performed the best for the set covering problem.

3. Data and Application

3.1 Real-world data set – Crime data

3.1.1 Background

This dataset is generated from Communities and Crime Unnormalized Data Set on UCI Machine Learning Repository. (Redmond, 2011)

The original dataset combines socio-economic data from the '90 Census, law enforcement data from the 1990 Law Enforcement Management and Admin Stats survey, and crime data from the 1995 FBI UCR. This dataset includes 2215 cases and 147 variables, but the ‘crime dataset’ used in this section consists of 760 randomly selected cases (communities) with the population size between around 14,000 and 43000 and 31 variables. One variable (the number of burglaries) is used as the response variable and the 30 remaining variables, including per capita income and median gross rent, are used as explanatory variables for a linear regression model.

Among the 760 cases, 380 randomly selected cases are used for the training set to fit the model and the remaining 380 cases are used for the validation set to evaluate the model selected from the training set. These two sets are used for analysis and comparisons in the next section..

3.1.2 Analysis and Result

In this section, a multiple linear regression model is used to find a relationship between the response variable and the explanatory variables described in the previous section. The programming language C++ was used for analysis on the computer with Windows 10 Pro edition (64 bit) and Intel core i5-6300U.

The weighted average of the Akaike Information Criterion (AIC) (Akaike, 1974) is used as the (*ad hoc*) objective function for optimization:

$$AIC_w = pAIC_t + (1-p)AIC_v \qquad (2)$$

where *p* and *1-p* are the proportions of the training set and the validation set from the whole dataset, respectively. In the crime dataset, *p*=0.5 because the training set and the validation set have the same size of 380. AIC_t and AIC_v are the AICs calculated from the training set and the validation set, for each.

AIC_t is used to estimate the coefficients in multiple regression with the training set and then AIC_v is used to evaluate a model derived in the previous step with the cases in the validation set.

Now, it is explained briefly how to conduct variable selection process (for getting relevant variables and the corresponding coefficients) step by step.

1. Generate a population of a fixed size of bit strings for a given metaheuristic method.
2. With a selected bit string from the population in step 1, use the data points in the training set, estimate coefficients and calculate the corresponding AIC_t.

3. Use the data points in the validation set and the coefficients (or model) from step 2, calculate the corresponding AIC_v and then calculate the AIC_w.
4. Repeat steps 2-3 as needed and update population as desired until the stopping criteria are satisfied (either the limit of 3600 second of running time or no observed improvement in terms of AIC_w, whichever comes first.)

Basic comparison of three metaheuristics. GA, TBO, and Jaya are used as our main metaheuristics in this section, but specific procedures for each method are not given in the steps above due to their complexity. One can easily check them in the references mentioned in section 1 for GAs, if needed. For TBO and Jaya, one can check section 2.

These three methods are used for analysis and compared with one another for their performance and efficiency in terms of the magnitude of objective function (AIC_w) and running time, for each. Four cases for GA and one case for TBO and Jaya, respectively, are used as described:

1) Case 1: GA with random selection of parents.
2) Case 2: GA with random selection of parents plus mutation.
3) Case 3: GA with crossover.
4) Case 4: GA with crossover plus mutation.
5) Case 5: Jaya.
6) Case 6: TBO.

Each case used 5 lists with the population size of 300 for calculation.

Table 1: Comparison of metaheuristic methods I – Crime.

Case	Best AIC_w[1]	Explanatory variables selected		Time[3]
		Number	Index[2]	
1	3555.72	13	1,2,8,10,12,16,23,24,25,26,27,28,29	36.22
2	3555.72	13	1,2,8,10,12,16,23,24,25,26,27,28,29	64.05
3	3555.72	13	1,2,8,10,12,16,23,24,25,26,27,28,29	63.32
4	3555.72	13	1,2,8,10,12,16,23,24,25,26,27,28,29	103.32
5	3555.72	13	1,2,8,10,12,16,23,24,25,26,27,28,29	16.07
6	3558.65	12	1,2,8,9,10,12,24,25,26,27,28,29	19.50

[1] Best (smallest) AIC_w from 5 lists in each case.
[2] If the index i is shown, it implies the i^{th} explanatory variable is selected. (i=1,···, 30)
[3] The unit of time is minute.

Table 1 shows the summary from the 5th list. (other lists show similar results.) It can be checked the global optimum (minimum) of AIC_w is 3555.72 from different methods including exhaustive methods or sequential methods.

All GAs (Cases 1 to 4) and Jaya (Case 5) attained the global optimum, but TBO did not. Jaya was much faster than GAs (16.07 vs. 36.22 or 64.04 or 63.32 or 103.32) and TBO was fast (19.50) comparing with GAs but wound up with a sub-optimum.

More comparisons with different Jaya population sizes. Based on the results above, Jaya looks superior to other metaheuristic methods. In this section, it will be explored how Jaya can be improved with controlling population size, which is one of only two parameters (population size and stopping criteria) used in Jaya. Four scenarios are defined as J1, J2, J3, and J4 with the different population size of 300, 200, 100, and 50, respectively. Each scenario used five lists with the same population size for calculation.

Table 2: Comparison of Jaya with different population sizes – Crime.

Scenari o	Number of lists detecting the optimum[1]	Time[2]
J1	2	14.10
J2	3	11.71
J3	0	13.44
J4	0	2.91

[1] Number of trials in which the minimum AIC_w is detected among 5 lists.
[2] The unit of time is minute.

It provides a clue that Jaya can be improved by a certain amount of population size reduction. It also suggests a trade-off between the quality of results and the size of population. If a small-sized population is used, running time will be reduced at the cost of more likelihood of getting sub-optima.

4. Summary

In this paper, variable selection, one of the classical topics in regression, was dealt with using metaheuristic methods. It can be stated as a combinatorial optimization problem with the goal to select variables that maximize (or minimize) the given objective function. Even if some metaheuristics such as Genetic Algorithm (GA) or Simulated Annealing (SA) have shown better performance in many problems over conventional methods, it turned out that 'fine tuning of parameters' is very challenging.

This paper explored some 'parameter-free' metaheuristics like Teaching-Based Optimization (TBO) and Jaya and compared them to GA. The previous sections illustrated that Jaya is superior to other metaheuristic methods in terms of performance and efficiency when it is properly used with relatively small population and neighborhood search.

It must be noted that one of the main purposes of this paper is not to develop a complete package to solve many different problems but to suggest how parameter-free metaheuristics such as Jaya can be used for variable selection.

Also, it must be admitted that the algorithms used for the datasets serve as an initial trial for the development of better parameter-free metaheuristic algorithms to come. Even if some simulations in high dimensional, say 100, space were conducted, their results were not included in the paper, due to issues arising from complexity and too much 'noise', which implies that there is a lot of room for improvement.

Lastly, a possible direction for future research may include, but is not limited to, handling highly correlated variables and developing stronger computing methods to manage 'the curse of dimensionality' to some extent.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. https://doi.org/10.1109/tac.1974.1100705

Akaike, H. (1978). A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics*, *30*(1), 9–14. https://doi.org/10.1007/bf02480194

Broadhurst, D., Goodacre, R., Jones, A., Rowland, J. J., & Kell, D. B. (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*(1–3), 71–86. https://doi.org/10.1016/s0003-2670(97)00065-2

Fan, J., & Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360. https://doi.org/10.1198/016214501753382273

Hans, C., Dobra, A., & West, M. (2012) Shotgun Stochastic Search for "Large p" Regression. *Journal of the American Statistical Association. 102:478,* 507-516, https://www.tandfonline.com/doi/abs/10.1198/016214507000000121

Hill, R., & Pohl, E., (2019) A structural taxonomy for metaheuristic optimization search methods. *International Journal of Metaheuristics.7*(2), 127-151. https://www.inderscienceonline.com/doi/abs/10.1504/IJMHEUR.2019.098261

Hocking, R. (1976), The Analysis and Selection of Variables in Linear Regression. Biometrics, 32, 1–49.

Kapetanios, G. (2007). Variable selection in regression models using nonstandard optimization of information criteria. *Computational Statistics & Data Analysis*, *52*(1), 4–15. https://doi.org/10.1016/j.csda.2007.04.006

Kiezun, A., Lee, I.-T. A., & Shomron, N. (2009). Evaluation of optimization techniques for variable selection in logistic regression applied to diagnosis of myocardial infarction. *Bioinformation*, *3*(7), 311–313. https://doi.org/10.6026/97320630003311

Jirapech-Umpai, T., & Aitken, S. (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, *6*(1), 148. https://doi.org/10.1186/1471-2105-6-148

Lanza-Gutierrez, J. M., Crawford, B., Soto, R., Berrios, N., Gomez-Pulido, J. A., & Paredes, F. (2017). Analyzing the effects of binarization techniques when solving the set covering problem through swarm optimization. *Expert Systems with Applications*, *70*, 67–82. https://doi.org/10.1016/j.eswa.2016.10.054

Lu, Y., & Vasko, F. J. (2015). An OR Practitioner's Solution Approach for the Set Covering Problem. *International Journal of Applied Metaheuristic Computing*, *6*(4), 1–13. https://doi.org/10.4018/ijamc.2015100101

Meiri, R., & Zahavi, J. (2006). Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, *171*(3), 842–858. https://doi.org/10.1016/j.ejor.2004.09.010

Mohan, S., Buchanan, B. R., Wollenberg, G. D., Igne, B., Drennen, J. K., & Anderson, C. A. (2018). Variable selection optimization for multivariate models with Polar Qualification System. *Chemometrics and Intelligent Laboratory Systems*, *180*, 1–14. https://doi.org/10.1016/j.chemolab.2018.06.002

Paterini, S., & Minerva, T. (2010). Regression Model Selection Using Genetic Algorithms. *Recent Advances in neural networks, fuzzy systems & evolutionary computing*, ed. D. Monteanu et al, 19-27. Iasi, Romania: Wseas.us.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions* *on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/tpami.2005.159

Rao, R., & Kalyankar, V. (2011). *Parameters optimization of advanced machining processes using TLBO algorithm* http://www.ppml.url.tw/EPPM/conferences/2011/download/SESSION1/21_32.pdf

Redmond, M. (2011). Communities and Crime Unnormalized Data Set.UCI Machine Learing Repository Uci.Edu http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized

Sinha, A., Malo, P., & Kuosmanen, T. (2015). A Multiobjective Exploratory Procedure for Regression Model Selection. *Journal of Computational and Graphical Statistics*, *24*(1), 154–182. https://doi.org/10.1080/10618600.2014.899236

Vasko, F.J., & Y. Lu, Y. (2017). Binarization of Continuous Metaheuristics to Solve the Set Covering Problem: Simpler is Better. *invited talk, 21st Triennial Conference of The International Federation of Operational Research Societies (IFORS),* Quebec, Canada.

Venkata Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 19–34. https://doi.org/10.5267/j.ijiec.2015.8.004