Differential Evolution Based Model Selection Approach for Machine Learning

Yi-Chuan Chiu, Hsin-Hung Lin, and Yung-Tsan Jou

Abstract—**As the application of big data becomes more and more popular, machine learning algorithms are changing with each passing day, and the models produced by machine learning are increasingly diversified. The focus of big data applications has gradually shifted to the prediction and inference of models. How to choose the most suitable model for enterprise application scenarios among many machine learning models has become a topic of research that has attracted much attention. Ensemble methods have been proposed to discover best model by multiple training phase. Studies of finding best combination within multiple modes are still few. Configuring different machine learning models with appropriate parameters and looking for parameters is an NP-hard problem, which requires an optimization algorithm. This study proposes to apply differential evolution algorithm to integrate multiple trained machine learning models into an appropriate model. In this paper, the regression model is taken as an example and the differential evolution algorithm is compared with the particles swarm optimization algorithm. The results show that the differential evolution algorithm has better performance.**

***Index Terms*—Big data, differential evolution, machine learning, optimization.**

I. Introduction

Big data applications consist with four major steps: data collection, data preparation, model training and validation, model inference. (Fig. 1) Since more and more machine learning algorithms have developed for model training, the model selection and performance evaluation have become the significant role before model inference for production applications.

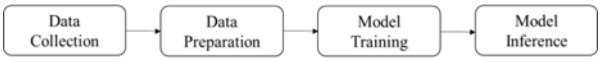

Fig. 1. The major steps of big data applications.

Ensemble learning was originally proposed for classification [1]. The basic concept of ensemble learning is that the integration of a number of experts to make decisions in some specific ways (such as voting method, weighting method), the results will be better than only a single expert. Because each expert's expertise is different, the combined mechanism allows the experts to complement each other and get better results. Ensemble learning focus on training data replacement or not to obtained more efficient model. Sagi and Rokach introduced the concept of ensemble learning, reviews traditional, novel and state- of- the- art ensemble methods [2].

Since model selection is selecting the best performance model from different machine learning models, models selection is different thinking from ensemble methods that is choosing the best result from different approaches (ex. bagging, boosting) with one model.

With the advent of big data, more and more machine learning models will be produced and applied to production, but the past literature mostly focuses on ensemble methods, and the literature of model selection is relatively rare. The purpose of this paper is to explore the best use of the algorithm to find the optimal combination solution of multiple machine learning models.

This study applied different algorithm with same training data, then finding the optimal parameters combination via differential evolution methods. Three different famous datasets were employed to verify the integrated model and the outcomes performed better than any single machine learning model. The rest of this paper is arranged as follows. In section II, related works are reviewed. Research methods are described in section III. Section IV reveals the results of the experiments. At last, conclusion is presented in Section V.

II. Related Works

There have been numerous studies in the literature dealing with regression including generalized linear model, rigid regression and robust regression. In addition, a lot of research works have been conducted in the fields of differential evolution and particle swarm optimization. This section provides review of these works.

A. Regression

In statistic, the Generalized Linear Model (GLM) is a flexible linear regression model. This model allows the distribution of the deviation of the number of strains to have other distributions than the normal distribution. This model assumes that the distribution function of the random variables measured by the experimenter and the systematic effects (ie, non-random effects) in the experiment can establish a function that can explain its correlation via a link function. The representative literature of the generalized linear model is an outline of the principles, calculations (such as the most approximate estimators) and practical applications of generalized linear models [3].

Ridge regression analysis is a technique for the presence of multiple collinear (automatically independent variables) data. In the case of multicollinearity, although the ordinary least

Manuscript received January 4, 2019; revised April 23, 2019.

Yi-Chuan Chiu and Yung-Tsan Jou are with the Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taoyuan, Taiwan (e-mail: angel.chiu957@gmail.com, ytjou@cycu.edu.tw).

Hsing-Hung Lin is with Big Data Lab, Telecom. Laboratory Chunghwa Telecom. Co., Ltd., Taoyuan, Taiwan (e-mail: hsinhung@gmail.com).

DOI: 10.7763/IJMO.2019.V9.698

squares (OLS) is fair to each variable; they vary widely, shifting the observations away from the true values. Ridge regression reduces the standard error by adding a degree of deviation to the regression.

Ridge regression solves the multicollinearity problem by the contraction parameter λ (lambda). The assumption of this regression is similar to the least squares regression, except for the constant term. It shrinks the value of the correlation coefficient but does not reach zero, which indicates that it has no feature selection function. This is a regularization method and uses L2 regularization.

The models for linear regression in the previous article were all based on the least squares method. However, when there are many outliers in the data sample points, the impact of these anomalies on the regression model will be very large, and the traditional regression method based on least squares will not be applicable.

Though, you can consider pre-processing the data and eliminating those abnormal points before doing regression analysis. However, in the actual data, there are two problems: Outliers are not well defined, and there is no good standard for determining which points are outliers.

Even if the abnormal point is determined, are these points that are determined to be abnormal; is it really the wrong data? It is very likely that this seemingly anomalous point is the data of the original model. If this is the case, then the points of these anomalies will carry a large amount of information of the original model, and a large amount of information will be lost after the culling.

Robust regression is an algorithm used to replace the least squares method when the least squares method encounters the above-mentioned data sample points with abnormal points. In addition, robust regression can also be used for outlier detection, or to find those sample points that have the greatest impact on the model.

B. Differential Evolution

Differential evolution (DE) is a stochastic, population-based optimization algorithm that developed to optimize real parameter, real valued functions and was firstly introduced by Storn and Price in [4]. Fig. 2 is the standard flow chart of differential evolution.

DE has the advantage of incorporating a relatively simple and efficient form of self-adapting mutation. The population size does not need to be overly high, and smaller populations can be considerably more efficient. With its ease of implementation and proven efficiency, DE is ideally suited to both novice and experienced users wishing to optimize their simulation models [5].

Through two decades development, DE has improved by different configuration including population, mutation and cross rate to enhance the performance of this state-of-the-art evolution algorithm [6]. An improved version of the differential evolution (DE) based on the orthogonal design (ODE) was proposed that makes the DE faster and robust [7]. Simulations result shoed that the ODE can find the near-optimal solution in all cases and outperforms other state-of-the-art evolutionary algorithms in terms of the quality, stability as well as computational cost. Another enhanced differential evolution optimization algorithm has been developed [8]. The enhancement lies in reducing the number of control parameters from three (NP, F and CR) to two (NP and F), thereby simplifying the tuning process. Comparison is made with the original DE and other famous DE algorithm, and the results demonstrate the superiority of the proposed approach for most of the functions considered. Elsayed and Sarker proposed an adaptive configuration of differential evolution algorithms for solving big data optimization competition problems [9]. The proposed algorithm automatically determines the best variant and shows the superiority compared with the baseline algorithm. Piotrowski briefly reviews the opinions regarding DE population size setting and verifies the impact of the population size on the performance of DE algorithms [10]. Based on the extensive experimental results the use of adaptive population size is highly recommended, especially for higher-dimensional and real-world problems.

The study of Das and Suganthan attempted to provide an overall picture of the DE [11]. It discussed the different schemes of parameter control and adaptation for DE and extended review of the modifications of DE for tackling constrained, multi-objective, uncertain, and large-scale optimization problems.

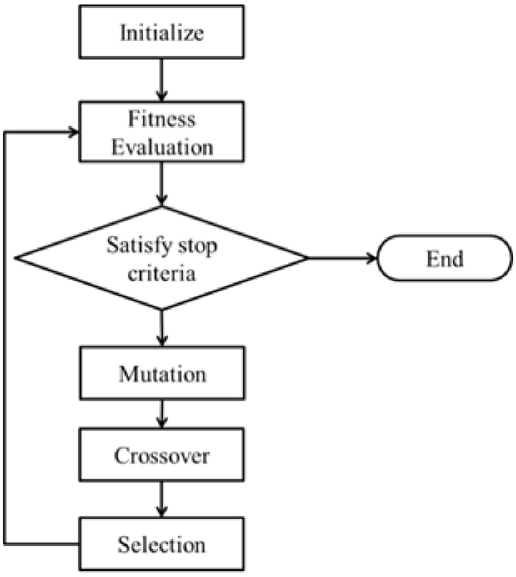

Fig. 2. Flow chart of standard differential evaluation.

C. Particle Swarm Optimization

PSO is an evolutionary technique [12] for solving unconstrained continuous optimization problems. The PSO concept is based on observations of the social behavior of animals. The population consisting of individuals (particles) is assigned a randomized initial velocity according each individual’s own movement experience and that of the rest of the population. The relationship between the swarm and the particles in PSO is similar to the relationship between the population and the chromosomes in the GA.

In PSO, the problem solution space is formulated as a search space. Each position of the particles in the search space is a correlated solution of the problem. Particles cooperate to determine the best position (solution) in the search space (solution space).

Reference [13] proposes the application of particle swarm optimization (PSO) to the problem of full model selection, FMS, for classification tasks. Results obtained in the framework of a model selection challenge show the competitiveness of the models selected with PSO. In predictive maintenances, the estimation of remaining useful life (RUL) of aircrafts engines which affects their maintenance planning has been investigated by support vector regression optimized by PSO [14]. The experimental results show the efficiency of the proposed approach.

All these approaches have demonstrated the advantages of the PSO method: simple structure, immediate applicability to practical problems, ease of implementation, quick solution, and robustness.

Genetic algorithms, particle swarm optimization, and differential evolution algorithms are all branches of evolutionary algorithms. Many scholars have studied these algorithms, and through continuous improvement, the performance of the algorithms has been improved and the application fields have been expanded. Therefore, it is necessary to discuss these algorithms. Features, for different application areas and algorithm adaptability, it is very meaningful to recommend different algorithms for use. In the literature, the author conducted a series of experimental analysis on DE, EA, PSO for the 34 commonly used benchmark functions, and discussed the optimal solution for various algorithms. Through experimental analysis, the DE algorithm obtained the optimal performance. Moreover, the algorithm is relatively stable, and the inverse operation can converge to the same solution; the convergence speed of the PSO algorithm is second, but the algorithm is unstable, and the final convergence result is easily affected by the parameter size and the initial population; the convergence speed of the EA algorithm is relatively slow, but in the In terms of dealing with noise problems, EA can solve it well and DE algorithm is difficult to deal with this noise problem.

III. Methods

There are two stages of experiments in this study. In the first stage, three different regression methods are employed to train models with selected datasets using R software. The dataset is divided into two parts: one is 70% of data as training set and the other is remaining 30% as testing set. The three different regression models are evaluated by three metrics: r squared, mean absolute error (MAE) and root mean squared error (RMSE). In the second stage, the differential evolution algorithm is applied to find the best parameter combination of the three models trained in the first stage. The performance evaluation of parameter combination includes r squared, MAE and RMSE.

The benchmark datasets applied to experiments are from UCI machine learning repository include Combined Cycle Power Plant (CCPP) and Concrete Compressive Strength (Concrete) [15]. Another dataset is Boston Housing (Boston) from Kaggle [16].

The experiment environment ran on PC of i5-6400 CPU 2.7GHz with 4GB RAM.

A. Model Training

Model Training of three different regression methods in the first step was conducted by RStudio. Fig. 3 presented the example code of R to execute data preparation and model training.

```
ccpp <- read.csv("C01.csv")
set.seed(999)
ind = sample(2, nrow(ccpp), replace = TRUE, prob=c(0.7,0.3))
ccpp_train = ccpp[ind ==1, ]
ccpp_test = ccpp[ind ==2, ]
avg_class=mean(ccpp_test$PE)
ccpp_test$sst <- (ccpp_test$PE-avg_class)^2
lm2 <- glm(PE ~., data=ccpp_train)
summary(lm2)
ccpp_test$glmpred <- predict(lm2, ccpp_test)
ccpp_test$glmssr<-(ccpp_test$glmpred-avg_class)^2
sum(ccpp_test$glmssr)/sum(ccpp_test$sst)
ccpp_test$glmabse <- abs(ccpp_test$glmpred-ccpp_test$PE)
mean(ccpp_test$glmabse)
ccpp_test$glmsqre <- (ccpp_test$glmpred-ccpp_test$PE)^2
mean(ccpp_test$glmsqre)
```

Fig. 3. R code of model training CCPP

After running the first stage, we should get the data table for each different dataset. as Table I.

TABLE I: The First Ten Records of Applying Trained Model

	Actual	Glm	Ridge	Robust
1	79.99	53.46346	45.0694	54.84891
2	61.89	53.73476	44.87407	54.94491
3	40.27	56.81259	46.03934	100.8365
4	41.05	67.66368	50.71598	128.4022
5	44.3	60.91206	44.79399	118.7823
6	47.03	26.85992	32.92445	38.89255
7	43.7	68.42076	51.65931	129.5637
8	36.45	29.92792	35.06954	31.77787
9	45.85	19.77815	29.87232	20.90228
10	39.29	31.44208	36.9562	34.10077

B. Exploring Optimal Solution

Based on the outcomes of the first stage, the optimal algorithm is applied to exploring the optimal combination of three different regression models. Differential evolution was conducted to search weights combination for the actual value of each data in the datasets. Parameter setting of differential evolution is also considered in this study. There are three parameters should be set while conduct differential evolution including Population, F and CR. The parameter setting of the experiments is as Table II and the pseudocode of differential evolution is as Fig. 4.

TABLE II: Parameter Setting of DE

Parameter	Value
Iteration	20, 40, 60, 80
Population	50, 100, 150, 200
F	0.2, 0.4, 0.6, 0.8
CR	0.2, 0.4, 0.6, 0.8

TABLE III: Parameter Setting of PSO

Parameter	Value
Iteration	20, 40, 60, 80
Population	50, 100, 150, 200
W	0.2, 0.4, 0.6, 0.8
C1	0.2, 0.4, 0.6, 0.8
C2	0.2, 0.4, 0.6, 0.8

```
Initialize the populating, setting F and CR
Do while not stop criteria
  For each individual j in the population
    Choose three numbers $r_1, r_2, r_3$ where $1 \leqq r_1, r_2, r_3 \leqq N$
    Generate random integer $i_{rand} \in (1, N)$
    For each parameter i
      $y_{i,g} = x_{r_1,g} + F(x_{r_2,g} - x_{r_3,g})$
      if rand()≦ CR or j=$i_{rand}$ than $z^j_{i,g} = y^j_{i,g}$
      else $z^j_{i,g} = x^j_{i,g}$
    if $z_{i,g}$ is better than $x_{i,g} = z_{i,g}$
Loop
```

Fig. 4. Pseudocode of differential evolution.

PSO is the algorithm for comparison in this experiment. Table III illustrates the parameter setting of PSO. The pseudocode of PSO is as Fig. 5 and implemented in Python to execute that is the same as DE.

```
Initialize a population of particles with random positions.
for each particle k do
    Evaluate $X^k$ (the position of particle k)
    Save the $pbest^k$ to optimal solution set S
end for
Set gbest solution equals to the best $pbest^k$
repeat
    Updates particles velocities
    for each particle k do
        Move particle k
        Evaluate $X^k$
        Update gbest, pbest and S
    end for
until maximum iteration limit is reached
```

Fig. 5. Pseudocode of differential evolution.

IV. Results

The benchmark dataset of this study are Boston housing, Concrete Compressive Strength and Combined Cycle Power Plant. The Boston data frame has 506 rows and 14 columns. The target field is median value of owner. The Concrete dataset contains 1030 instances and 9 attributes, while the output variable is the concrete compressive strength in MPa. The CCPP dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature, Ambient Pressure, Relative Humidity and Exhaust Vacuum to predict the net hourly electrical energy output of the plant. The datasets information is listed in Table IV.

TABLE IV: Summary of Datasets

Dataset	Instances #	Attributes #
Boston	506	14
Concrete	1030	9
CCPP	9568	4

In the comparison results of Boston dataset (Table V), DE can reach the performance of general linear model. However, PSO outperforms of MAE in average.

DE presented excellent outcomes in the comparison of concrete dataset (Table VI). General linear model got the same result of DE in RMSE.

TABLE V: The Comparison Results of Boston Dataset

Boston	R-squared	MAE	RMSE
GLM	**0.740644**	3.270869	**4.679184**
Robust	0.474351	3.169009	5.110834
Ridge	0.645192	3.252792	4.683165
DE			
Avg.	0.739277	3.132219	4.679254
Min	0.595747	**3.131343**	**4.679184**
Max	**0.740644**	3.134232	4.680204
PSO			
Avg.	0.740570	3.131470	4.679966
Min	0.740247	**3.131343**	**4.679184**
Max	**0.740644**	3.135852	4.692176

TABLE VI: The Comparison Result of Concrete Dataset

Concrete	R-squared	MAE	RMSE
GLM	0.672419	8.214343	**10.353609**
Robust	0.747838	8.941826	15.644075
Ridge	0.653029	10.011450	12.346823
DE			
Avg.	**0.781034**	8.087327	10.360034
Min	0.563289	8.085100	**10.353609**
Max	0.837150	8.125401	10.535270
PSO			
Avg.	0.609891	8.091543	10.39484
Min	0.561476	**8.077216**	10.35361
Max	0.615520	8.206088	10.69349

The comparison of Table VII showed that DE is superior to PSO in two of three metrics, while GLM performs better than DE in R-squared.

TABLE VII: The Comparison Result of CCPP Dataset

CCPP	R-squared	MAE	RMSE
GLM	**0.960641**	3.625216	**4.557126**
Robust	0.951230	3.625299	4.557127
Ridge	0.959462	3.617954	4.560789
DE			
Avg.	0.925531	3.618046	**4.557126**
Min	0.297427	**3.616068**	**4.557126**
Max	0.928696	3.620319	4.557127
PSO			
Avg.	0.928667	3.617887	4.557132
Min	0.928275	3.616109	**4.557126**
Max	0.928696	3.620619	4.557203

V. Conclusion

This study attempts to find the optimal combination of machine learning models by using the optimization algorithm. The experimental method is to use the regression algorithm. Firstly, three regression algorithms are used to calculate the estimated values of each data of three different data sets. Secondly, the weight combination closest to the actual value is computed by the optimization algorithm with three different estimates in each data set. The algorithms for performance measurement are R-squared, MAE and RMSE. The results of the comparison provide the evidence that DE performs better than PSO in general. Future studies can compare other algorithms or find the optimal combination of other machine learning algorithms (ex, classification or clustering).

Author Contributions

Yung-Tsan Jou conducted the research; Hsin-Hung Lin analyzed the data; Yi-Chuan Chiu wrote the paper.

References

[1] N. J. Nilsson, *Learning Machines: Foundations of Trainable Pattern-classifyig Systems*, New York: McgrawHill, 1965.

[2] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wires Data Mining and Knowledge Discovery*, vol. 8, Jul. 2018.

[3] P. McCullagh and J. Nelder, *Generalized Linear Models*, London: Chapman and Hall, 1989.

[4] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol.11, pp.341–359, 1997.

[5] D. G. Mayera, B. P. Kinghornb, and A. A. Archer, "Differential evolution – An easy and efficient evolutionary algorithm for model optimization," *Agricultural Systems*, vol. 83, pp. 315-328, Mar. 2005.

[6] K. R.Oparaa and J. Arabasb, "Differential Evolution: A survey of theoretical analyses," *Swarm and Evolutionary Computation*, in press.

[7] W. Gong and Z. Cai, "Differential evolution made faster and more robust," in *Proc. 2006 IEEE International Conference on Industrial Technology*, Dec. 2006.

[8] M. Arafa, E. A. Sallam, and M. M. Fahmy, "An enhanced differential evolution optimization algorithm," in *Proc. 2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, May 2014.

[9] S. Elsayed and R. Sarker, "An adaptive configuration of differential evolution algorithms for big data," in *Proc. IEEE Congress on Evolutionary Computation*, pp. 695-702, May 2015.

[10] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol.32, pp.1-24, Feb. 2017.

[11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 4-31, Feb. 2011.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95 - International Conference on Neural Networks*, Nov. 1995.

[13] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle swarm model selection," *Journal of Machine Learning Research*, vol. 10, pp. 405-440, Feb. 2009.

[14] A. E. Afia and M. Sarhani, "Particle swarm optimization for model selection of aircraft maintenance predictive models," in *Proc.2nd International Conference on Big Data, Cloud and Applications (BDCA'17)*, Mar. 2017.

[15] Kaggle dataset. [Online]. Available: https://www.kaggle.com/jhchiuh/boston-housing

[16] UCI machine learning repository. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

Yi-Chuan Chiu was born on Oct. 17, 1975, in Taiwan. He received B.E. degree from the Department of Industrial Management at the Chung Hua University in 1998, and the master's degree from the Department of Industrial Management at Chung Yuan Christian University in 2018, respectively. She is currently a doctoral student in Chung Yuan Christian University.

Yung-Tsan Jou (Integrated (ME, IE) Eng Ph.D., Ohio University, 2003) is assistant professor of industrial and systems engineering at Chung Yuan Christian University in Taiwan. His research interests lie in the areas of green design, human–system interface design, and usability or quality evaluation by using virtual reality tools. Prof. Jou worked in Silicon Valley, USA, between 1999 and 2004, where he served as a senior mechanical/manufacturing engineer for systems R&D in hi-tech companies.

Hsing-Hung Lin was born in Taichung, Taiwan in July 1972. He received the B. S. degree from the Department of Computer Science at Tunghai University in 1989, the M. Eng. from the Department of Industrial Engineering and Management of Chung Hua University in 1998, and the Ph.D. from the Department of Industrial Engineering and Management of National Chiao Tung University in 2010. Dr. Lin currently works as a research fellow at Big Data Lab in the Institute of Chunghwa Telecom. His current research interests include big data analytics, soft computing, artificial intelligence and systematic innovation.