Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220019

Efficient Feature Selection via Joint Neural Network and Pruning

Jie He[a], Zhiqiang Zeng[a,1]
[a] *College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China*

Abstract. With the development of machine learning, artificial intelligence and other fields, the processing of data mining has become more and more complex. As a data preprocessing step, feature selection is very important in many tasks, like classification, clustering and regression, etc. However, traditional feature selection methods learns similarity matrix from original data to calculate relevant data. What this method learns is the relationships which are linear between data and their labels, and it cannot deal with complex nonlinear data well in real-world applications. In this article, we proposed a feature selection method based on neural network that can select discriminative feature subsets by neural network pruning. And update all weights by gradient descent. Experimental results of our method on several real-world datasets achieve competitive or superior performance compared to three close related feature selection approaches.

Keywords. Neural network regression, Feature selection, Pruning algorithm

1. Introduction

Feature selection is important in many fields, such as machine learning[1]. Excellent feature selection method can improve the performance of the model while maintaining the internal structure of the data, which is very important in further performance improving of machine learning algorithm. Generally, there are two main functions in feature selection:

- The generalization ability of the model can be improved by reducing irrelevant features and dimension.
- Explore the importance of features and correlation between features and learning tasks.

Feature selection method is used to measure the excellence of a given feature subset from original feature set by a specific evaluation criterion. By feature selection, irrelevant and redundant features can be identified and further be removed, while useful features are preserved. Search technology and evaluation index can be regarded as the basic composition of feature selection algorithm. The purpose of former is provide candidate subsets of new features, while the latter scores different subsets of features.

Feature selection algorithms can be divided into three types by the form of feature selection: wrapper method[2], filter method[3,4] and embedded method [5,6,7]. The wrapper method score a subset of features by a predictive model. It trains a model for each new subset and tests it against the validation data set. After that, filter method scores

[1] Corresponding Author. E-mail: zqzeng@xmut.edu.cn

the feature subset by counting the number of errors on the validation dataset. Filter method is computation-intensive because these methods train a new model for each feature subset. The filter method uses proxy metrics rather than scoring on the error rate of the feature subset. The selected indices can save the calculation and estimate the correlation of the feature set. In filter method, common indicators include Pearson Correlation Coefficient, Distance Correlation Coefficient, Variance Selection, etc. Generally, filter approaches cost less computation than wrapper approaches, but the subset of features found by these methods cannot be tuned for a particular type of prediction model. Embedded method select features by model of machine learning. The feature selection process is integrated and automatically implement in the training process of the learning model.

To cope with non-linear data, traditional feature selection method usually relies on learning a similarity matrix from the original data to judge the relationship among features. However, this kind of method is easily affected by noise and is difficult to deal with complex nonlinear problems[8,9]. In this paper, the goal of our proposed method is to extract nonlinear features by neural networks. In particular, we apply pruning algorithm to network for improve the sparsity in our proposed model. Through pruning, the network can learn the structural sparse features and speed up the process of training.

The main contributions of our paper are:

- We propose a method of feature selection based on neural network with pruning to find out the relationships which are nonlinear between data and its corresponding response value.
- Experiments on several datasets from real-word show off the superior performance of our method compared to related feature selection approaches.

The rest of this paper is composed of these parts. The related work are presents in Section 2. The neural network based feature selection approach is proposed in Section 3. Section 4 shows the results of our experiment. The conclusion and direction for our future work are provided in Section 5.

2. Related work

We will introduce several related approaches of feature selection in this section, and briefly review the support vector regression model, which is a traditional nonlinear regression method.

2.1 linear feature selection model

Given training dataset $X = \{x_1, x_2, \dots x_n\} \in \mathbb{R}^{d \times n}$, which contains n samples with d dimensions. Feature selection is dedicated to choose a feature subset in each sample without causing much information loss of the original data, so that improve model performance. In the past few decades, a lot of feature selection methods have been proposed[10,11,12,13].

Nie et al. proposed a feature selection model which added $\ell_{2,1}$-norm on both regularization term and loss function[14]. They use $\ell_{2,1}$-norm to remove outliers and useless features in data, and effectively avoid the problem of overfitting. The model obtained in this way has strong anti-interference ability. The objective function is formulated as follows:

$$\min_{W} \sum_{i=1}^{n} ||W^T x_i + b - y_i||_2 + \gamma ||W||_{2,1} \tag{1}$$

where the residual $||W^T x_i + b - y_i||$ is not squared to reduce the influence of the outliers. The regularization term based on $\ell_{2,1}$-norm can achieve sparsity in W by removing the outliers. Then, one can select features in all data points by their scores. Besides, Zhong et al. proposed a model that use adaptive discriminant analysis for feature selection, namely, SADA[15]. This method simultaneously learns a projection matrix W and an adaptive similarity matrix S, and constantly updated during the iteration. The objective function is:

$$\min_{W, W^T W = I} \sum_{i=1}^{n} \sum_{j \in M_i^L \cup M_i^U} ||W^T (x_i - x_j)||_{2,p}^p + \gamma ||W||_{2,1} \tag{2}$$

where M_i^L contains k nearest neighbors of x_i in labeled data and M_i^U contains k nearest neighbors of x_i in unlabeled data. They control the sparsity of matrix S by $\ell_{2,p}$-norm and select top m scored features in W as the final result.

However, these methods have a common drawback: they are linear, which means only simple structure information of data can be explored, ignoring the nonlinear relationships among data.

2.2 Support vector regression

Support vector regression(SVR) model, which is an important application branch of support vector machine (SVM). SVR is a popular model which can be used for curve fitting and prediction for both nonlinear and linear regression types, which is to find a regression plane so that all the data in a set can be closest to the plane. The prediction value in conventional regression method are regarded as correct only when the regression $f(x)$ is exactly equal to y. For example, linear regression is often used to calculate its loss $(f(x) - y)^2$. However, SVR holds that the prediction can be considered correct as long as the value of $f(x)$ is not much different from y.

Most existing feature selection methods paid attention on relationships which are linear between data samples and labels[16]. Although the optimization problem is easy to solve in linear feature selection methods, but the nonlinear relationship of data are ignored, and the discriminative features are hard to select. In this article, we will introduce a neural network based feature selection method that can select discriminative feature subsets by pruning for regression task.

3. Proposed method

3.1 Networks architecture

Given training dataset $X = \{x_1, x_2, \dots x_n\}^T \in \mathbb{R}^{n \times d}$ and its corresponding response value is $Y = \{y_1, y_2, \dots y_n\}^T \in \mathbb{R}^{n \times 1}$, where n denote the number of instances and d denote the number of features in each instance. Feature selection is dedicated to select a most effective feature subset from original features in data X according to $P(X) \in \mathbb{R}^{n\times m}$, where $P(\cdot)$ is used for select features. For the convenience of calculation, the network in our approach only contains one hidden layer, and $f(X) = \sigma_1(XW^{(i)} + b^{(i)})$ is the output of hidden layer. Nonlinear activate function in hidden layer is represented by $\sigma_1(\cdot)$, i.e., Tanh, ReLU, Sigmoid, Leaky ReLU function. $W^{(i)} \in \mathbb{R}^{d\times h}$ and $b^{(i)} \in \mathbb{R}^{h}$ denote weight matrix and bias vector of hidden layer, and h is the number of neural cells in hidden layer. The output of our network is $\hat{Y} = g(f(X)) = f(X)W^{(o)} + b^{(o)})$, where $W^{(o)} \in \mathbb{R}^{h\times 1}$ and $b^{(o)}$ means the weight parameters and bias constant of output layer, respectively. We calculate all the parameters during the training process of our network through minimize the following mean square error loss function.

$$\mathcal{L}\left(W^{(i)}, W^{(o)}, b^{(i)}, b^{(o)}\right) = \frac{1}{2n} || \hat{Y} - Y||_2^2 \tag{3}$$

3.2 Pruning

The goal of pruning is to develop smaller and more efficient neural networks. Numerous experiments have shown that neural network has lots of redundant calculations during the training process[18]. Neural network pruning will first filter out unimportant neurons and weights from the large network, and then delete them from the network, while maintaining the performance of the network as much as possible. Moreover, the model size can be reduced by weight pruning. Lin et al. proposed a pruning method that prunes filters jointly as well as other structures in an end-to-end mode, achieving the purpose of pruning with low precision loss[17].

Algorithm 1 Bank-Balanced Pruning Algorithm

```
Input:
  The matrix to be pruned, $W$;
  The number of blocks in each row, BankNumber;
  The level of sparsity, Sparsity;
for each $W_i \in W.rows$ do
  Divide the row of $W_i$ into BankNumber blocks;
  for each block $\in W_i$ do
    Sort the block's elements;
    Calculate the block internal threshold $T$ in line with sparsity;
    for each element $\in$ block do
      prune element if element $< T$ ;
    end for
  end for
end for
Output: The pruned matrix, $W_p$;
```

In this paper, we employ BBS[18] pruning into our method to achieve sparsity at a high level while maintaining model accuracy. BBS divides each row of weight in the hidden layer into banks, while maintaining the model accuracy by embed fine-grained pruning inside each bank. The weights matrix of hidden layer will be row sparsity when the network is in the process of back propagation. There is a simple example for BBS shown in Fig 1. The details of our proposed approach are described in Algorithm 2.

0.5	0.1	1	0.6	0.9
0.9	0.7	0.8	1.2	0.5
0.6	0.8	0.2	0.4	0.7

	0.1		0.6	
	0.7			0.5
		0.2	0.4	

Fig 1. The original data matrix (left) and bank-balanced sparse matrix (right)

Therefore, the loss function of our method can be presented as follow:

$$\min_{W} \mathcal{L}(W;(X,Y)) = \min_{W} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(W^{(i)};(x_i,y_i)\right),$$

$$s.t. \quad W^{(i)} \in \mathbb{R}^{d\times h}, \qquad N\left(W_i^{(i)}\right) = t. \tag{4}$$

Where $W^{(i)}$ denote pruning to it in the network training process, t denotes sparsity level (i.e., the number of remaining elements that are non-zero), $W_i^{(i)}$ denotes i_{th} row vector of weights matrix $W^{(i)}$, and $N\left(W_i^{(i)}\right)$ means the number of elements that are non-zero in $W_i^{(i)}$, Algorithm 1 described the calculation process.

```
**Algorithm 2** The solution of problem Eq.(4).
**Input**: Data: $X \in \mathbb{R}^{n\times d}$, $Y \in \mathbb{R}^{n\times 1}$, the number of selected features: $m$;
**Initialization**: Weight matrix $W^{(i)}$, $b^{(i)}$, $W^{(o)}$, $b^{(o)}$;
**repeat**
  1. Calculate the partial derivative $\triangle W^{(i)}$, $\triangle b^{(i)}$, $\triangle W^{(o)}$, $\triangle b^{(o)}$ through the Backpropagation;
  2. Update all weights according to the gradient descent algorithm;
  3. Proceed the feedforward algorithm with current updated weights.
**until convergence**
**Output**: Regression values: $\hat{Y}$, updated weights.
```

4. Experiments

In this section, we first summarize the information of datasets that we used in the experiment. Then, we have conducted experiments on the several real-world datasets used for evaluate the performance of our proposed method and compared with some related approaches.

4.1 Dataset description

Three real-world UCI regression datasets and one Kaggle dataset were selected in our experiment, and the information of datasets are described here:

- **Energy prediction:** The dataset is at 10 min for about 4.5 months. A ZigBee wireless sensor network is used to monitor the humidity conditions and house temperature. The dataset consists of 19735 samples and 29 features.
- **Concrete Compressive Strength:** Concrete is an indispensable material in civil engineering. The dataset consists of 1030 samples and 9 features.
- **QSAR fish toxicity:** This dataset was used to predict acute aquatic toxicity in the fish Pimephales promelas. The dataset consists of 908 samples and 7 features.
- **Kc_house:** The dataset consists of house prices between May 2014 to May 2015 from an area which is King County in the US State of Washington. The dataset contains 21613 samples and 21 features.

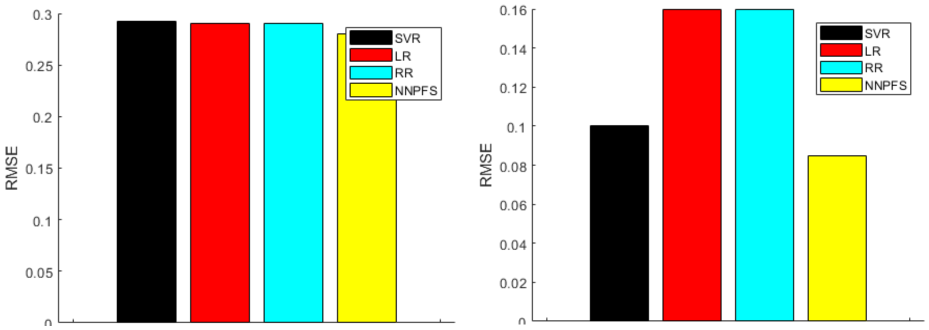
4.2 Experiment setting

For the convenience of data processing, we normalize all data sets and map the data to the range of 0 to 1, which is more convenient and faster. Normalization is essentially a linear transformation and can improve the performance of the data. In addition, we randomly select 70% of the samples for each data set as the training set, and the remaining 30% samples are used for test. We use the value of RMSE as the evaluation standard. All experiments are repeated ten times while the average results are taken to evaluate the performance of our method.

The several related methods which we have compared are a statistical analysis method for regression (LR[19]), a regression method that are more realistic and reliable for regression coefficients can be obtained (RR[20]), and a SVM[22] based regression method (SVR[21]).

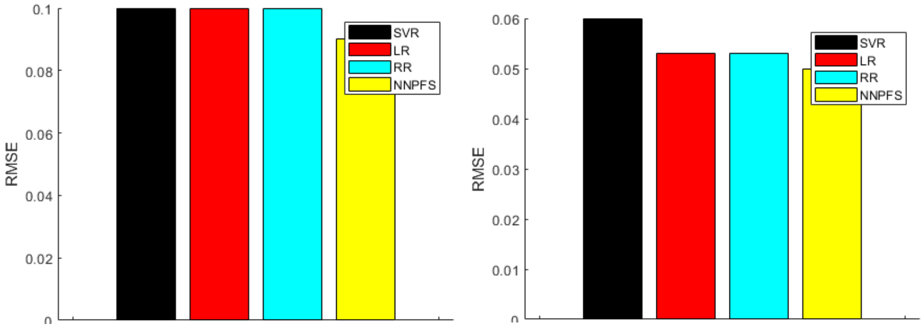
4.3 experiment results

In this section, we selected several real-world datasets which can be used for predict continuous value to validate the effectiveness of our approach. Fig 2. shows the error results, and what we can conclude is our method achieves competitive or superior performance in terms of RMSE values compared to other methods that related to feature selection, especially on the Concrete Compressive Strength datasets. Also, it has provides strong evidence that our method is more effective when dealing with regression tasks.



(a) Energy prediction

(b) Concrete Compressive Strength



(c) QSAR fish toxicity (d) Kc_house

Fig 2. The RMSE value of our method and other related methods on different four real-world datasets

5. Conclusion

In this paper, the feature selection method we proposed based on neural network that can select discriminative feature subsets by pruning, and shows that the features which are structural sparse can be learned to speed up network training. We apply it for regression tasks, and achieve superior performance compared to some relate methods. In our future works, how to achieve effective feature selection on datasets which are high-dimension small-sample-size should be paid more attention.

Acknowledgement

This paper was supported by National Natural Science Foundation of China (Grant Nos.61871464), National Natural Science Foundation of Fujian Province (Grant Nos.2020J01266, 2021J011186), the "Climbing" Program of XMUT (Grant No.XPDKT20031), Scientific Research Fund of Fujian Provincial Education Department (Grant No. JAT200486), Program of XMUT for high-Level talents introduction plan (Grant No.YKJ19003R)

References

[1] G. Chandrashekar, F. Sahin. A survey on feature selection methods. Comput. Electr. Eng 40 (2014), 16-28.

[2] Y. Han, K. Park, Y.K. Lee, Confident wrapper-type semi-supervised feature selection using an ensemble classifier, in: Proceedings of the 2011 2nd Artificial Intelligence, Management Science and Electronic Commerce, AIMSEC 2011, 4581–4586.

[3] L. Yu, H. Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. Icml, vol. 3 (2003), 856-863.

[4] N. Sanchez-Marono, A. Alonso-Betanzos, M. Tombilla-Sanroman. Filter methods for feature selection: a comparative study. Intelligent Data Engineering and Automated Learning-IDEAL, Springer, Berlin, Heidelberg (2007), 178–187.

[5] F. Nie, D. Xu, I. W.-H. Tsang and C. Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction, IEEE Trans. Image Process (2010), 1921-1932.

[6] X. Zhang, G. Wu, Z. Dong, C. Crawford. Embedded feature-selection support vector machine for driving

pattern recognition. J. Frankl. Inst 352 (2015), 669-685.

[7] B. Peralta, A. Soto. Embedded local feature selection within mixture of experts. Inf. Sci 269 (2014), 176-187.

[8] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection, In AAAI (2008), 671--676.

[9] F. Nie, X. Wang and H. Huang. Clustering and projected clustering with adaptive neighbors, Proc. Int. Conf. Knowl. Discovery Data Mining (2014), 977-986.

[10] X. Chen, G. Yuan, W. Wang, F. Nie, X. Chang and J. Z. Huang. Local adaptive projection framework for feature selection of labeled and unlabeled data, IEEE Trans. Neural Netw. Learn. Syst. vol. 29 (2018), 6362-6373.

[11] F. Nie, W. Zhu, X. Li. Structured graph optimization for unsupervised feature selection. IEEE Trans. Knowl. Data Eng. (2019), 1210-1222.

[12] A. Mirzaei, V. Pourahmadi, M. Soltani and H. Sheikhzadeh. Deep feature selection using a teacher-student network, Neurocomputing, vol. 383 (2020), 396-408.

[13] Z. Kang et al. Structured graph learning for clustering and semi-supervised classification, Pattern Recognit, vol. 110 (2021).

[14] F. Nie, H. Huang, X. Cai and C. Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, Proc. Adv. Neural Inf. Process. Syst. (2010), 1813–1821.

[15] W. Zhong, X. Chen, G. Yuan, F. Nie, et al. Adaptive discriminant analysis for semi-supervised feature selection. Information Sciences, vol. 566 (2021),178-194.

[16] Z. Wang, F. Nie, C. Zhang, R. Wang and X. Li. Joint nonlinear feature selection and continuous values regression network. Pattern Recognition Letters, vol. 150 (2021), 197-206.

[17] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, et al. Towards optimal structured cnn pruning via generative adversarial learning, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., (2019), 2790-2799.

[18] C. Shijie et al., Efficient and effective sparse LSTM on FPGA with Bank-Balanced Sparsity, Proc. Int. Symp. Field-Program. Gate Arrays, (2019), 63-72.

[19] D.C. Montgomery, E.A. Peck, G.G. Vining, Introduction to linear regression analysis, 821, John Wiley & Sons, 2012.

[20] Walker. E, Birch, J. B. Influence measures in ridge regression. Technometrics, 30 (1988), 221–227.

[21] D. Parbat, M. Chakraborty. A python based support vector regression model for prediction of COVID19 cases in India. Chaos, Solitons Fractals, 2020.

[22] K. Dai, H.-Y. Yu, Q. Li. A semi-supervised feature selection with support vector machine. J. Appl. Math. 2013 (2013).