



Review on Deep Learning in Feature Selection

Yizhuo Zhang, Yiwei Liu, and Chi-Hua Chen^(✉)

Fuzhou University, Fuzhou 350100, China
chihua0826@gmail.com

Abstract. Feature selection (FS) plays an important role in the machine learning (ML) field. Since FS solves the problem of dimensional explosion in ML very well, more and more people are paying attention to FS. Not only that, but this technique also takes advantage of the computational complexities and time reductions. Inspired by the points mentioned above, more and more FS algorithms solved by deep learning framework are appearing. Due to the importance of FS, it is necessary to conduct further research. However, FS is wide coverage, and the algorithms involved are numerous, which makes researchers need to spend a lot of time searching and reading the literature. In order to provide researchers with dedicated information and enable them to quickly have an overall understanding of the FS field, this article will from three aspects, including the main functions and framework of FS, search strategies of FS, and the evaluation strategy and algorithms in related fields to introduce FS from whole to part. Finally, this article discusses some existing problems and points out some promising research directions.

Keywords: Feature selection · Dimension reduction · Deep learning · Big data

1 Introduction

Nowadays, the Internet of things (IoT) has been developed rapidly, a large amount of data is collected by the enterprises and industries [1]. In the information age, people's daily lives and devices on the Internet also produce a large amount of data. These data meet the needs of ML, but the amount of data is too large to mark every piece of data, and not all features are useful, some data even have errors [2]. Another signification problem is that meaningless data need to be removed which may cause negative impacts on learning tasks. Therefore, there is a need for a method that can extract useful subsets from the original data and reduce the impact of redundant data to improve ML accuracy and stability. FS aims to reduce the data dimensions by removing some irrelevant features, so it is very suitable for solving these problems. Because deep neural networks proved working well in processing massive data, FS based on deep neural networks had been proposed to deal with these problems [3–5]. In recent years, there are many constructive methods integrated with ML top research theories have been proposed, such as Knockoff generative adversarial network (GAN), FS guided auto-encoder.

In order to have a better understanding of the FS, this study will raise three general questions (GQ), which are focus on the overall FS and domain-specific methods.

Table 1 describes the general questions. Answering these questions can give a general understanding of the FS.

Table 1. General questions.

Identifier	General Questions (GQ)
GQ1	What is the main purpose of the FS?
GQ2	What are the search strategies for FS?
GQ3	What are the evaluation methods for FS?

The remainder of the paper is organized as follows. Section 2 presents search strategies and sources. In Sect. 3, review results are discussed for these general questions. The conclusions and future work are given in Sect. 4.

2 Sources

The articles involved in this study are from journals and conferences by searching in online academic searching engines (e.g., Web of Science). The relevant keywords used for searching such as FS, search strategy, filter, wrapper, evaluation methods. When selecting papers, the quality, amount of citations and contribution of algorithms are considered.

3 Results and Discussions

In this section, the relevant papers have been reviewed and discussed to answer the three general questions in Table 1.

3.1 GQ1–What Is the Main Purpose of the FS?

Due to the rapid development of the IoT and information technology, the data from various collection devices usually contain a large amount of redundant information and error information. This kind of data may cause an adverse impact on big data analysis. To solve this problem, the data is usually pre-processed which is dimension reduction. There are two popular dimension reduction methods which are feature extraction (FE) and FS. The result of FS is a subset of the original data, and the result of FE is a mapping of the original data. Figure. 1 shows the process of FS. FS aims to approximate the original data by selecting a set of essential features. So, FS method can make a model more reliable and be furnished with better interpretability. In FS, search strategies and evaluation methods are critical to the quality of selection results.

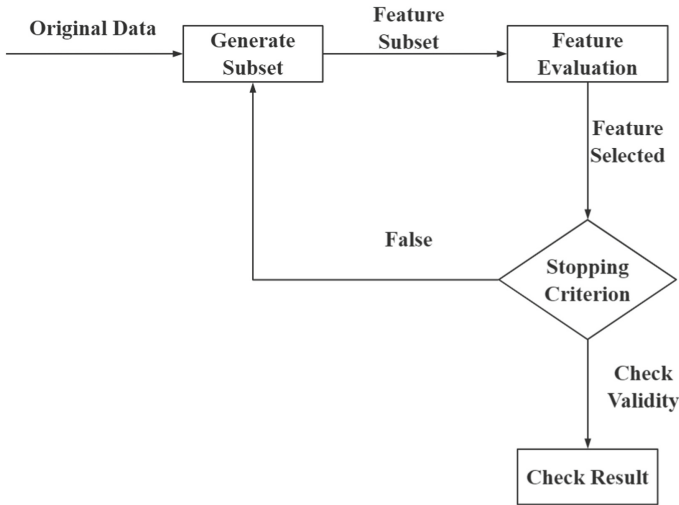


Fig. 1. The process of FS.

3.2 GQ2-What Are the Search Strategies for FS?

The current research generally divides the search strategy of FS into three parts: (1) global search strategy, (2) random search strategy, and (3) heuristic search strategy [6].

The global search strategy can analyze and evaluate each subset of data to decide whether to select this feature. Generally, there are enumeration method and branch-and-bound method [7]. The enumeration method is easy to achieve but they will consume a lot of time. The branch-and-bound method is preferred if the amount of data is large. The branch-and-bound method will define an evaluation function and a number of features that would be selected before the algorithm runs [8].

The random search strategy can weight each piece of data according to the validity of its feature, and select depend on a defined threshold. The random search strategy usually combines FS problems with other artificial intelligence algorithms such as simulated annealing algorithm (SAA) and genetic algorithm (GA) [9]. Therefore, if the number of epochs in these algorithms is too small, the accuracy of the results may be unstable. When the number of epochs is large enough and the parameter settings are appropriate, the random search strategy methods would show a better result.

The heuristic search strategy can create a sequence for the original data. In this sequence, a forward selection method and a backward selection method are used to add and delete a feature, until the optimal feature subset is reached [10]. Four types of methods for the heuristic search strategy include sequence forward selection method (SFS), generalized sequence forward selection method (GSFS), sequence backward selection method (SBS), generalized sequence backward selection method (GSBS).

3.3 GQ3-What Are the Evaluation Methods for FS?

The evaluation methods of FS are generally divided into filter methods, wrapper methods, and embedded methods [11, 12].

Filter Methods. The filter method generally uses the statistical performance of all training data directly to evaluate features, so it is independent of the subsequence learning algorithm. It can quickly remove redundant features in large data sets [13]. A variance threshold method (VTM) and a mutual information method (MIM) will be introduced in this subsection.

Variance Threshold Method. The VTM can calculate the variance of each feature and select the features whose variance is greater than the defined threshold [14].

The variance of feature is defined as Eq. (1).

$$V(t) = \frac{1}{m} \sum_{i=1}^m (f_i(t) - \bar{f}(t))^2 \tag{1}$$

In Eq. (1), m is the number of categories, and the $f_i(t)$ means the frequency of feature t in the i -th category. $\bar{f}(t)$ notes the average of the frequency of feature t in Eq. (2).

$$\bar{f}(t) = \frac{1}{m} \sum_{i=1}^m f_i(t) \tag{2}$$

The variance of feature $V(t)$ denotes the discrete trend of feature t . The distribution of feature t in m categories is more concentrated with a larger variance $V(t)$. Equation. (3) could be used to normalize variances $V(t)$ for avoiding the suppression of low frequencies by high frequencies.

$$N(t) = \frac{\frac{1}{m} \sum_{i=1}^m (f_i(t) - \bar{f}(t))^2}{\frac{1}{m} \sum_{i=1}^m (f_i(t))^2} = \frac{V(t)}{\frac{1}{m} \sum_{i=1}^m (f_i(t))^2} \tag{3}$$

The distribution probability of feature t in category c_j is defined as Eq. (4).

$$\theta_j = \frac{f_{c_j}(t)}{|c_j|} \tag{4}$$

In Eq. (4), $f_{c_j}(t)$ denotes the frequency of feature t in category c_j , and $|c_j|$ denotes the sum number of category c_j occurred in data. Therefore, the feature evaluation function is defined as Eq. (5).

$$T(t) = V(t) \times \max_{1 \leq j \leq m} \theta_j \tag{5}$$

After calculating the all features evaluation score $T(t)$, the features whose score is greater than the defined threshold can be selected.

Mutual Information Method. The MIM is a way of representing the correlation between features. The information entropy of \mathbf{X} can be estimated by Eq. (6), and the conditional information entropy of \mathbf{X} after the known variable \mathbf{Y} is calculated by Eq. (7) [15].

$$H(\mathbf{X}) = - \sum_{x \in \mathbf{X}} \Pr(x) \times \log \Pr(x) \quad (6)$$

$$H(\mathbf{X}|\mathbf{Y}) = - \sum_{y \in \mathbf{Y}} \Pr(y) \times \sum_{x \in \mathbf{X}} \Pr(x|y) \times \log \Pr(x|y) \quad (7)$$

In Eqs. (6) and (7), $\Pr(x)$ means the probability that feature \mathbf{X} takes x , and $\Pr(x|y)$ notes the probability that feature \mathbf{X} takes x when feature \mathbf{Y} takes y . The mutual information $I(\mathbf{X}; \mathbf{Y})$ between feature \mathbf{X} and feature \mathbf{Y} can be calculated by Eq. (8).

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}) &= H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) \\ &= \sum_{y \in \mathbf{Y}} \sum_{x \in \mathbf{X}} \Pr(x, y) \times \log \frac{\Pr(x, y)}{\Pr(x) \Pr(y)} \end{aligned} \quad (8)$$

Furthermore, the correlation between feature \mathbf{X} and feature \mathbf{Y} can be calculated by Eq. (9).

$$R(\mathbf{X}; \mathbf{Y}) = \frac{I(\mathbf{X}; \mathbf{Y})}{H(\mathbf{X}) + H(\mathbf{Y})} \quad (9)$$

The value range of $\text{Sim}(x, y)$ is between $[0, 1]$. If the value of $R(\mathbf{X}; \mathbf{Y})$ is 0, it means that two features are not related; if the value of $R(\mathbf{X}; \mathbf{Y})$ is 1, it means that two features are completely related.

Wrapper Methods. Because the evaluation of the wrapper methods depends on the training accuracy of the subsequent learning algorithm, they have fewer errors than the filter methods. However, the computation cost of wrapper methods is large, it is not suitable for large data set operations.

In related research, Guyon et al. used the support vector machine (SVM) classification algorithm to measure the importance levels of features, and constructed a classifier with better classification performance [16]. Gui et al. proposed a method based on attention mechanism with high accuracy and stability for noisy and small data sets [17]. Ye et al. proposed a fast wrapper FS method called fast feature subset ranking (FFSR) [18].

Embedded Methods. The embedded methods which are different from the filter methods and the wrapper methods combine the variable selection and training process with the advantages of high efficiency and better integration ability. The embedded

methods also consider the relationship between each two-features and usually incorporate the relationship into the learning phase of another algorithm.

Several embedded unsupervised FS methods use regularization for selecting discrete features. However, by using regularization, the embedded methods usually require higher computation cost to get the final results. An embedded end-to-end FS method brings a good solution to this problem. The proposed method introduces a concrete auto-encoder for differentiable FS and reconstruction [19]. And the concrete auto-encoder contains a concrete selector layer encoder and a standard neural network decoder.

Han et al. proposed an auto-encoder feature selector (AEFS) method which combines group lasso tasks and auto-encoder [20]. By excavating the linear and nonlinear information among features, the AEFS can select the optimal features. Most traditional embedded based methods (e.g., least absolute shrinkage and selection operator (LASSO) [21]) only work in the linear relationship and ignore the nonlinear relationship between each two-features. Therefore, the neural networks are used to learn the flexible nonlinear relationship between features, which can flexibly handle various tasks (e.g., Image segmentation [22] and image recognition [23, 24]). The studies used an auto-encoder to improve the effect on unsupervised learning and selected features with high recognition.

Auto-Encoder Feature Selector. The goal of AEFS is to select t optimal features from \mathbf{X} . The matrix $\mathbf{X} = [x_1, x_2, \dots, x_m]^T$ is the unlabeled sample matrix from data; d is the sample dimension, and m is the number of features.

The auto-encoder in AEFS has two fully connected layers and uses a typical h -dimensional hidden layer including a coding function $g(\mathbf{X})$ and a decoding function $h(g(\mathbf{X}))$ as Eqs. (10) and (11).

$$f(\mathbf{X}) = \sigma_1(\mathbf{X}\mathbf{W}^{(1)}) \quad (10)$$

$$\hat{\mathbf{X}} = h(g(\mathbf{X})) = \sigma_2(g(\mathbf{X})\mathbf{W}^{(2)}) \quad (11)$$

In Eqs. (10) and (11), σ_1 and σ_2 represent the activation functions in the hidden layer and the output layer; $\Theta = \{w^1, w^2\}$ is the weight parameters, and $h(g(\mathbf{X}))$ is the overall function of the auto-encoder. In the learning process, the article uses the least square method to describe the loss function of the auto-encoder. The loss function is expressed as Eq. (12).

$$\mathfrak{S}(\Theta) = \frac{1}{2m} \|\mathbf{X} - h(g(\mathbf{X}))\|_F^2 \quad (12)$$

The goal is optimizing loss function. After learning process, the auto-encoder can reduce the dimension of the matrix \mathbf{X} and output the decoding matrix as $\hat{\mathbf{X}} = h(g(\mathbf{X}))$.

In the FS stage, by comparing the reconstructed data result $\hat{\mathbf{X}}$ and original data \mathbf{X} after discarding some features; if the features of these two matrices are similar, redundant features can be removed. The matrix $\mathbf{W}^{(1)} = [w_1, w_2, \dots, w_d]^T$ is used to

represent the weights among the neurons in the input layer and the neurons in the hidden layer. If $\|w_i\|_2 \approx 0$, the i -th feature is very sparsely connected; otherwise, the connection is very closed. In order to be able to select the more important feature from \mathbf{X} , the row-sparse regularization is used for $\mathbf{W}^{(1)}$. The row-sparse regularization can be calculated by Eq. (13) which is used to find the L2 norm and the L1 norm for $\mathbf{W}^{(1)}$ in turn ($\|\mathbf{W}^{(1)}\|_{2,1}$). In Eq. (13), h is the dimension of hidden layer; d is the sample dimension; d represents the weight parameters of the i -th neuron in the l -th layer to the j -th neuron in the $(l+1)$ -th layer.

$$\|\mathbf{W}^{(1)}\|_{2,1} = \sum_i^d \sqrt{\sum_j^h (w_{ij}^{(1)})^2} \quad (13)$$

Therefore, the Eq. (12) can be reformulated as Eq. (14).

$$\mathfrak{S}(\Theta) = \frac{1}{2m} \|\mathbf{X} - h(g(\mathbf{X}))\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} \quad (14)$$

In the training process, in order to avoid the problem of overfitting, a weight decay term is necessary. The final loss function is defined as Eq. (15).

$$\mathfrak{S}(\Theta) = \frac{1}{2m} \|\mathbf{X} - h(g(\mathbf{X}))\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} + \frac{\beta}{2} \sum_{i=1}^2 \|\mathbf{W}^{(i)}\|_F^2 \quad (15)$$

In Eq. (15), β is a penalty parameter. The AEFS [19] can be used to optimize Eq. (15).

Summary and Discussions. In summary, there are three kinds of evaluation methods in FS, which are filter method, wrapper method and embedded method. For the filter method, it mainly focuses on the correlation between the features in the original data. It is responsible for selecting optimal features and sending these features to the subsequent learning algorithm. Therefore, the filter method is more versatile and would not be affected by the subsequent learning process. If the amount of original data set is huge, the filter method is a good solution, and it can quickly remove a large number of redundant features. However, because it is independent of the learning algorithm, the accuracy of FS of the filter method is usually lower than the wrapper method. For the wrapper method, it relies on the learning effect of the subsequent learning algorithm to evaluate the feature subset. Therefore, if you want to perform FS and evaluate FS performance on an original data set, it is necessary to train a classifier and evaluate the feature subset based on the performance of the classifier. The performance of the wrapper method is better than the filter method, and the wrapper method is more accurate and reliable. The disadvantages of the wrapper method are that it requires training and evaluate each generated subset, so it is not suitable for large-scale data set operations. The embedded method is similar to the wrapper method, but the difference from the first two methods is that the embedded method uses the FS algorithm itself as

a part of the learning algorithm and selects features during each learning process. Embedded needs to redesign the learning algorithm to combine FS, so the complexity and difficulty of implementation will increase. But the embedded method would save a lot of time in the model training stage. Embedded as a hotspot of current research, it has the advantages of high efficiency and integration with machine learning.

4 Conclusions and Future Work

This paper first analyzes the main goals and general architecture of FS, then classifies FS according to search strategies and evaluation strategies. To solve high-dimensional problems, FS can remove redundant features according to global search strategy, random search strategy, or heuristic search strategy. It can choose the filter method to deal with massive data or the wrapper method to improve the quality of generated subsets. It also can use the embedded method and combine other ML algorithms to make better performance. However, there are still some shortcomings such as a single algorithm that cannot support both high-dimensional and low-dimensional data. In the future, the development direction of FS would focus on combining with other ML algorithms, such as adversarial neural networks, convolutional neural networks, etc., to be more suitable for practical applications. And the multi-view FS would also be a hot topic.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (Nos. 61906043, 61877010, 11501114 and 11901100), Science Foundation of the Fujian Province, China (No. 2019J01243), Funds of Education Department of Fujian Province (No. JAT190026), and Fuzhou University (Nos. 510730/XRC-18075, 510809/GXRC-19037, 510649/XRC-18049, and 510650/XRC-18050).

References

1. Yin, S., Ding, S.X., Xie, X., Luo, H.: A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans. Ind. Electron.* **61**(11), 6418–6428 (2014)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
3. Wang, Q., Zhang, J., Song, S., Zhang, Z.: Attentional neural network: feature selection using cognitive feedback. In: *NIPS 2014: Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2033–2041. ACM, NY, USA (2014)
4. Li, Y., Chen, C.Y., Wasserman, W.W.: Deep feature selection: theory and application to identify enhancers and promoters. *Lecture Notes in Computer Science*, vol. 9029, pp. 205–217 (2015)
5. Roy, D., Murty, K.S.R., Mohan, C.K.: Feature selection using deep neural networks. In: *Proceedings of the 2015 International Joint Conference on Neural Networks*, pp. 1–6. IEEE, NJ, USA (2015)
6. Belkin, M.: *Problems of Learning on Manifolds*. The University of Chicago, IL, USA (2003)
7. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.* **26**(9), 917–922 (1977)

8. Li, M., Kamil, M.: Research on feature selection methods and algorithms. *Comput. Technol. Dev.* **2013**(12), 16–21 (2013)
9. Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., Zhao, H.: Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* **19**(13), 1636–1643 (2003)
10. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. *Pattern Recogn. Lett.* **15**(11), 1119–1125 (1994)
11. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: a data perspective. arXiv: 1601.07996 (2016)
12. Wang, Y., Xu, C., Xu, C., Tao, D.: Beyond RPCA: flattening complex noise in the frequency domain. In: *NIPS 2014: Proceedings of AAAI 2017: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2761–2766. ACM, NY, USA (2017)
13. Yao, X., Wang, X., Zhang, Y., Quan, W.: Summary of feature selection algorithms. *Control Decis.* **27**(2), 161–313 (2012)
14. Yuan, Y., Wang, X.: Text feature selection algorithm based on variance. *Comput. Eng.* **2012** (12), 155–157 (2012)
15. Jiang, S., Wang, L.: Feature selection based on feature similarity measure. *Comput. Eng. Appl.* **46**(20), 153–156 (2010)
16. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**, 389–422 (2002)
17. Gui, N., Ge, D., Hu, Z.: AFS: An attention-based mechanism for supervised feature selection. In: *Proceedings of AAAI 2019: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 3705–3713. AAAI, CA, USA, (2019)
18. Ye, J., Gong, X.: A novel fast wrapper for feature subset selection. *J. Changsha Univ. Sci. Technol. (Nat. Sci.)* **2010**(4), 69–73 (2010)
19. Balin, M.F., Abid, A., Zou, J.Y.: Concrete autoencoders: differentiable feature selection and reconstruction. In: *Proceedings of ICML 2019: Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:, pp. 444–453. JMLR, Inc. and Microtome Publishing, USA (2019)
20. Han, K., Wang, Y., Zhang, C., Li, C., Xu, C.: Autoencoder inspired unsupervised feature selection. In: *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2941–2945. IEEE, NJ, USA (2018)
21. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.* **58**(1), 267–288 (1996)
22. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440. IEEE, NJ, USA (2015)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, NJ, USA (2016)
24. Wang, Y., Xu, C., You, S., Tao, D., Xu, C.: Cnnpack: packing convolutional neural networks in the frequency domain. In: *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pp. 253–261 (2016)