Evolutionary Computation for Feature Manipulation: Key Challenges and Future Directions

Bing Xue
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Bing.Xue@ecs.vuw.ac.nz

Mengjie Zhang
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Mengjie.Zhang@ecs.vuw.ac.nz

***Abstract*—In machine learning and data mining, feature manipulation is a data pre-processing step to increase the quality of a feature space, which can significantly improve the performance of a learning algorithm in terms of the accuracy, the learning speed, and the complexity and the interpretability of the learnt models. However, feature manipulation is a difficult task and facing more challenges along with the trend that more and more data is collected in many domains. Evolutionary computation (EC) techniques have recently attracted much attention for dealing with complex feature manipulation problems. Current work has demonstrated some strengths of EC for feature manipulation, but also shown some limitations and issues that need to be addressed. More importantly, there are some highly interesting research topics in the EC for feature manipulation area, which could potentially result in promising approaches to data analysis in a variety of real-world applications. This position paper describes and discusses the main issues and key challenges of feature manipulation, and also provides a number of directions for further consideration in future research.**

I. Introduction

GIGO (garbage in, garbage out) is a common long-standing principle in computer science, mathematics, and many other fields, which expresses the idea that the quality of the output is determined by the quality of the input. In machine learning and data mining, the output of an algorithm could be in different forms, e.g. a classifier for classification, a function for regression, a set of cluster centres for clustering, or a model for prediction [1]. However, the input of all algorithms is the same, i.e. the *data* describing the problem to be solved, where the data is almost always expressed by a number of features (attributes or variables) showing different properties of the problem. So the quality of the *feature space* (i.e. the input) very much determines the performance (i.e. the output) of every machine learning or data mining technique.

Although recent advanced machine learning or data mining algorithms are very powerful in handling different kinds of tasks, their performance is still limited or influenced when the feature space is of poor quality. Feature manipulation, mainly including feature (subset) selection, feature construction (or feature extraction), and feature weighting (or ranking), can improve the feature space in order to improve the learning performance (e.g. classification accuracy), reduce the dimensionality, speed up both the training and test processes, simplify the learnt model, help visualisation and interpretability, reduce the memory/storage space, and/or reduce the data collection cost [2], [3], [4], [5], [6], [7], [8]. Therefore, feature manipulation is a multi-disciplinary research topic heavily studied in many areas, such as computer science, statistics, mathematics, biology, engineering, and business [2], [3], [4], [5], [9].

However, feature manipulation, especially feature selection and construction, is a challenging task because of the large search space growing exponentially with the total number of features, and the complex interactions between features. They are typically NP-hard problems with a large and complex search space containing many local optima, and there are often multiple conflicting objectives involved. Research on feature selection has been for more than 50 years, but the feature space becomes increasingly large and complicated in recent years, which pushes the capability limits of current algorithms, and new efforts are required for dealing with the emerging challenges.

Evolutionary computation (EC) includes a group of nature-inspired approaches [10], such as genetic algorithms (GAs), evolutionary strategy (ES), genetic programming (GP), particle swarm optimisation (PSO), ant colony optimisation (ACO), and differential evolution (DE). EC techniques have been proven to be highly successful across a wide range of tasks in the past 20 years [10], [11], [12], [13]. There also have been a lot of evidence showing the success of EC in machine learning and data mining [14], including feature manipulation [9], [6], [15], [16].

The position of this paper is that EC techniques have great potential to address feature manipulation tasks in machine learning and data mining. They are still facing some issues and challenges, but there are some encouraging directions that should be explored in the future. These points will be discussed in the following sections.

II. State-of-the-Art in EC for feature manipulation

A poor feature space may be that the dimensionality is too high (curse of dimensionality), features are not equally important, some features are irrelevant, redundant or even

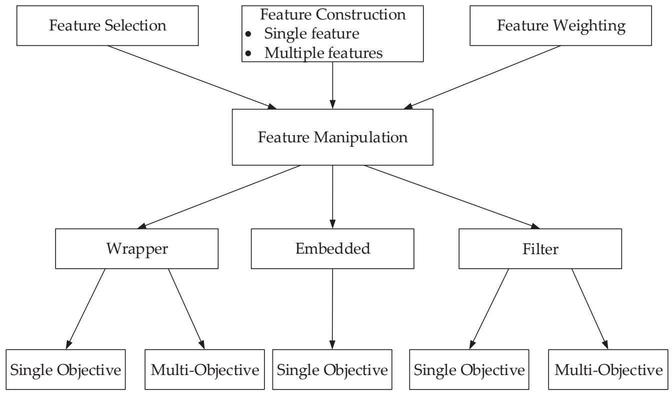

Fig. 1. Overall categories of feature manipulation methods.

noisy, the original features are not informative enough, the features are not linearly separatable, and so on [2]. These will lead to various performance limitations. For example in classification, these will lead to low classification accuracy, a long training time, a complex classifier [2], [3], etc. Feature manipulation can address these problems by feature (subset) selection that aims to select a small subset of relevant features, feature construction (or feature extraction) that aims to construct new high-level features from the original features using some function operations, or feature weighting (or ranking) that aims to give each feature a weight based on their relative importance. Most existing feature manipulation approaches focus on feature selection or feature construction for classification [5], although there are some works on clustering and other tasks [4].

Two key components in feature manipulation are the search strategy that finds the optimal feature set(s) and the evaluation criterion that evaluates the quality/fitness of the feature subsets. Based on the evaluation criteria, feature manipulation methods can be categorised to wrapper, filter and embedded approaches depending whether and how a learning algorithm is involved in the evaluation procedure [4]. Since feature manipulation often involves multiple conflicting objectives, such as maximising the accuracy, miminising the dimensionality, and minimising the model complexity, the existing approaches can also categorised into single objective approaches and multi-objective approaches. Fig.1 shows the overall categories of exiting feature manipulation methods, where feature construction approaches may construct a single new feature or multiple new features, and there are very few multi-objective methods in the embedded category.

Although the earliest work on EC for feature manipulation was published around 1990 when problems with over 20 features were referred as large-scale [17], this area started booming since 2007 when the number of features in many areas reaches a few hundreds or more. Now, EC based feature manipulation methods have covered all the above categories shown in Fig 1. Among them, almost all methods focus on EC for feature selection or construction in classification [9], although there are a few papers on EC for feature weighting. This is mainly because EC technique were firstly introduced to feature manipulation due to their search ability while feature weighting usually does not require a powerful search algorithm. For both feature selection and construction, there are a small number of embedded methods compared with filter and wrapper approaches since it is hard to for most EC techniques to perform learning and feature selection simultaneously except for GP and learning classifier systems (LCSs). On EC for feature selection, the most widely used EC techniques are GAs, PSO, ACO and GP, and most of these approaches focus on wrapper approaches [9], although there are a number of filter and embedded approaches recently developed [18], [9]. On EC for feature construction, although there are some initial works on GAs and PSO for feature construction, GP is the most widely used algorithm due to its flexible tree-based representation that is able to deal with function operations easily and directly. There is much more work on EC for feature selection than for feature construction, since most EC techniques cannot cope with function operations easily.

EC based feature manipulation approaches have made a big progress over the last ten years, and shown their superior performance over commonly used traditional (non-EC) methods, especially on the wrapper based approaches. The most significant contribution of EC for feature manipulation relies on the multi-objective category, where all the current multi-objective feature manipulation approaches are based on EC techniques. The targeted problems mainly have hundreds of features, and there are also some targeted problems having thousands and tens of thousands of features. Furthermore, EC based feature manipulation approaches have also been applied to a variety of areas, such as image analysis, face recognition, handwritten digit recognition, EEG brain-computer-interface, speaker recognition, gene analysis, disease diagnosis, financial problems, customer churn prediction, text mining, web service, network security, power system, music, agriculture, chemistry, geoscience, and weather forecast [19], [20], [9].

III. EC FOR FEATURE MANIPULATION: STRENGTHS

Feature manipulation is typically an NP-hard problem with a huge search space especially when the number of available features is big, which easily reaches thousands and even millions [21]. The search space also has many local optima, especially features often have interactions with each other. These make the search space of an feature manipulation task very complex and poorly understood. Therefore, many mathematical or statistical approaches that often have assumptions about the problem, such as the data points being normally distributed, or linearly separable, fail to achieve good performance because the assumptions often cannot be satisfied in real-world data [11], [12], [22], [23].

EC techniques have gained attention in feature manipulation and obtained good results in recent years. The main reasons can be summarised as follows. Firstly, EC algorithms do not make any assumption about the problem, such as whether it is linearly or non-linearly separable, and differentiable, so that they are widely applicable and easily transferable at a low cost. This is an advantage that attracts researchers to use EC for addressing many complex problems, and feature manipulation is one of such problems.

Secondly, EC techniques do not require domain knowledge, but they are also flexible and can be easily incorporated within, or make use of, domain-specific methods or existing methods such as local search, which often leads to a better hybrid approach. There are a large number of exiting feature manipulation approaches including different filter measures and search methods. The filter measures, such as fuzzy and rough sets, information measures, and correlation measures, can be easily used as the fitness function of an EC algorithm to address feature manipulation problems [9]. The search methods, such as sequential forward/backward search, floating search, and Tabu search, can be incorporated with EC in different ways to build a good approach, e.g. as a local search to fine tune the solutions found by the EC method [24], [25], [26]. Also, traditional feature weighting methods can also be easily used during the EC search to improve the performance [5].

Thirdly, EC algorithms maintain a population of initially randomised choices/solutions, which makes them robust, particularly critical for problems with many local optima. More importantly, they can produce multiple solutions in a single run, which makes them particularly suitable for multi-objective problems where multiple trade-off solutions of the problem are needed so that users can make an informed decision according to different requirements. The relationship between the multiple objectives in feature manipulation problems is very complicated, e.g. the number of features is conflicting with the classification accuracy in some regions of the search space, but not some others. This makes it very hard to combine them into a single fitness function by predefining their weights to reflect the user preference. Therefore, EC for multi-objective feature manipulation, especially feature selection, has been very popular in recent few years [27], [28], [29], [30], [31].

In addition, EC techniques are capable of producing unexpected solutions because they are blind to human preconceptions and so can find effective, but non-intuitive solutions, which are often valuable in design domains [12]. In feature manipulation, because of the highly complex feature interaction issues, it is extremely challenging to predict which features working together can achieve the best performance, even for domain experts. EC techniques have the potential to find solutions that are even better than the best solution designed by human experts. One example is that our recent work on GP for biomarker detection [32], where domain experts predefined 9 biomarkers to achieve perfect classification accuracy for a cancer task, but GP is able to find a feature set with only 5 features to achieve perfect performance. This is due probably to the very underlying interactions between features. Although feature interaction is an important problem recognised by the feature manipulation community for decades, there are only very few papers explicitly working on feature interaction [33] because they are very complex and vary in different datasets. Therefore, using EC techniques to automatically and implicitly handle feature interaction is not a bad choice before an effective feature interaction detector is proposed.

IV. EC FOR FEATURE MANIPULATION: WEAKNESS AND ISSUES

Despite the achievement of EC for feature manipulation, there are also some issues and challenges in current EC techniques for feature manipulation. Four major ones are the search space, the long computational time, the poor scalability, the feature selection or construction bias issue, and the generalisation issue.

The size search space highly depends on the representation in the EC methods. GAs and PSO are the most popular algorithms, and the most commonly used representation is bit-string/vector with a length equal to the total number of features, which leads to a large search space. The fitness landscape is another important factor influencing the quality of the obtained solution. Most of the existing approaches used the classification accuracy or existing filter measures in the fitness function, which often cannot lead to a smooth fitness landscape or with low locality because changes in the solutions (feature sets) often cannot lead to corresponding changes in the fitness values. This is especially the case when a complicated classification algorithm, such as support vector machines, is used to calculate the accuracy, since such algorithms perform extensive optimisation to process the input features during the learning stage.

Since EC approaches have a large number of evaluations, being computationally intensive is one of the major issues in EC for feature manipulation [34], especially when a wrapper approach is used, i.e. each evaluation involves a learning process of a machine learning or data mining algorithm. This makes EC techniques much more expensive than most traditional feature manipulation methods, which also limits their applications to many areas. Furthermore, many conclusions in traditional feature manipulation methods do not stand in EC based approaches, such as filters are computationally cheaper than wrappers [35].

Scalability in EC for feature manipulation refers to two aspects, the number of features/dimensions and the number of instances. EC based approaches have achieved promising results on feature manipulation problems with hundreds of features (the dimensionality of the search space often equals to the total number of features). When the number of features reaches thousands, or even millions, EC approaches face difficulty. The large-scale global function optimisation research in EC currently has only reached to tens of thousands, but researchers from other fields have started their work on feature manipulation problems with millions of features [21]. When the number of instances is large, EC based approaches are often not applicable because of the long computational time. This will limit the use of EC for feature manipulation on big data, which is a trend in current real-world problems.

Feature selection or construction bias is another common issue in many EC for feature manipulation approaches, which happens if the whole set of data is used during the feature selection or construction process [36]. This is very common in wrapper based approaches, where each evaluation involves a training and a test processes of an algorithm, e.g. a classification algorithm. Although a separate training set and a test set are often used for classification during the feature selection or construction process, the selected or constructed features are not tested on any unseen data that has never been involved in the feature selection (i.e. training) process. The reported classification accuracy is actually a "training" performance, or biased performance. The experiment design is even more complicated in the case k-fold cross-validation is used since the experiments need to repeat k times for the cross validation, and the stochastic EC process needs to perform a number of independent runs. Although this has been discussed in [3], many papers on EC for feature manipulation still have feature selection or construction bias, especially for gene datasets. This definitely should be avoided since the conclusions could be very different in the situations with and without feature selection bias [37].

Generalisation is a common issue in many machine learning and data mining problems. It is also an issue in EC for feature manipulation, especially when a wrapper approach is used. The selected or constructed features can easily overfit the wrapped learning algorithm and the training data, leading to poor performance on unseen test data. Filter approaches suffer less than wrappers in terms of generalisation since they often aim to capture the patterns of the data itself, rather than fitting to any learning algorithms. Compared with feature selection, the poor generalisation happens more in feature construction because the constructed high-level features often do not maintain the original low-level information, and the way they are created might fit the training data only. When a wrapper approach is used in feature construction, the performance of the algorithm on unseen test set could be very poor, which strongly limit its use in real-world problems.

V. Future Directions

Nowadays, feature manipulation is becoming increasingly important in a variety of areas. Future research should focus on utilising the strengths of EC for feature manipulation, overcoming the weakness to discover their great potential, and address the current issues.

Developing a good *filter measure* for the fitness function could solve a number of problems in EC for feature manipulation. An efficient and effective measure could reduce the computational cost, smooth the landscape of the search space, improve the learning and generalisation performance, and increase the interpretability/understandability of the obtained feature set. Many simple traditional filter measures from the machine learning field have been used in EC for feature manipulation without or with minor modifications. They are effective, but definitely not optimal or even near-optimal because most of them only consider relationship between a pair of features (or a feature and the class label), rather than a group of features that is the only way to capture underlying complementary (or interactive) information to find the optimal feature sets with the maximum useful information but minimum redundancy. A good filter measure in the EC scenario should consider the search behaviours of the EC method and also be able to discover complex feature interactions to obtain good feature subsets. This is of course a very challenging task, but with great potential. A good starting point would be rather than looking at the traditional measures, the advanced or state-of-the-art measures, such as the sparse learning based methods [38], [39], [40], can be introduced and investigated with modifications according to the characteristics of EC since such measures also require a powerful search method.

Current *representation* scheme is one of the main factors that limit the scalability of feature manipulation methods, since the most common bit-string or vector representation leads to a large search space. It only reflects whether a feature is selected in feature selection and cannot be easily used for feature construction except for the tree based representation in GP. A good representation should be able to avoid the huge search space, and can incorporate more information of about the features, such as the relative importance of features, feature interactions or feature similarity [41], [42], which will significantly improve the performance of EC for feature manipulation.

Search mechanism is a key factor in developing a feature manipulation approach, which is also the primary advantage of EC techniques. In most current work, EC techniques are directly applied to feature manipulation without specifically considering the characteristics of the tasks. An advantage of EC over other search methods is that it is flexible to combine with domain-specific method or other existing methods, which should be taken into consideration when designing a new approach in feature manipulation. In some early work which considered the characteristics of feature selection, or hybridised EC with local search (memetic algorithms) or traditional feature selection methods, has shown a significant improvement on the performance [25], [24], [43]. In recent years, there have been great achievements in different EC streams that are closely related to feature manipulation, especially the following ones:

- Evolutionary multi-objective optimisation [44], [45], [46], [47]: feature manipulation is essentially a complex multi-objective task with the two main objectives are to maximise the learning performance (e.g. classification accuracy) and minimise the number of selected or constructed features. Some other objectives such as the complexity of the learnt models could also be considered to better solve the problems. It is worthy mentioning that feature selection could also be useful for the currently hot topic of evolutionary many objective optimisation by effectively reducing the objective space [48].
- Combinatorial optimisation [49], [50]: both feature selection and feature construction are combinatorial problems, where feature selection aims to choose the best combination of the existing features while feature construction needs to find the best combination of features and func-

tion operators to create new informative features.

- Memetic computing [51], [52]: memetic computing has shown its superior performance in feature selection [24] by using a relatively cheap local search to fine tune the solution found by EC search. This could be further investigated to solve the new challenges in feature selection. Furthermore, feature construction is more challenging than feature selection, but capability of memetic computing has not been investigated in feature construction.
- Large-scale optimisation [53], [54]: most current EC for large-scale global (function) optimisation is based on decomposition by analysing the interaction between variables, which is very similar to feature interactions in feature manipulation. Although the tasks are quite different, there are a lot of overlap between these two types of problems. To achieve, large-scale feature manipulation for problems with thousands or even millions of features, the advances in large-scale optimisation will be very helpful, and the study of decomposition could also benefit both fields.
- Surrogate models [55], [56], [57], [58]: filter approaches were initially developed as a surrogate model of the wrapper approaches, i.e. using a simpler and more efficient measure to approximate the classification accuracy rather than directly training a classifier in each evaluation. Therefore, the advances in surrogate models can also be utilised in feature manipulation problems.
- Adaptive parameter control techniques [59], [60]: choosing the right parameters can significantly improve the performance of an EC algorithm. This is particularly important for feature manipulation since different datasets often have need different parameters, so it is almost impossible to find the best fixed parameter values for all datasets, but adaptive parameter control based the data itself could overcome this issue.

Feature construction is extremely important when the original features (or data) are not informative enough, such as the raw pixels values in image data. However, feature construction is more challenging than feature selection, which is probably why there is much more work on feature selection than on feature construction. To achieve automatic feature construction/extraction, an algorithm is required to be able to deal with both the features and function operators, which is not what most EC techniques (and many other non-EC algorithms) are good at, except for GP, which has a flexible representation [61]. Most current feature construction approaches focus on constructing a single high-level feature, which might not be sufficient when the problem is complex. Multi-feature construction is a promising approach for complex tasks, but not heavily investigated. Meanwhile, GP can automatically perform feature selection, construction and classification (or symbolic regression) in a single process [62], [63], [64], [65]. In most existing work on EC for image analysis, the features are often drawn by using an independent feature construction/extraction method, and the extracted features are then fed to an EC method for performing image analysis. An effective way to combine all these steps in a single system could reduce the complexity, decrease the computational time, and improve the overall performance [66]. Furthermore, since both feature selection and feature construction have their own advantages, a system that can take the advantages of both feature selection and construction will be needed for some complex problems.

Instance selection and construction [67] can also improve the quality of the input data by removing some noisy data and selecting or constructing only representative instances, to improve the learning performance. Combining feature manipulation with instance selection/construction could effectively and efficiently further improve the quality of the data over using either of them. When dealing with large-scale (big data) problems, where both the number of instances and the number of features are large, each evaluation for feature manipulation is expensive since it involves a large number of instances. So it will be of significant help by combining feature manipulation and instance selection/construction to reduce the size and improve the quality of the data. There has been some initial work on this direction and shown encouraging results, but more advanced work will be needed for large-scale (big data) tasks in the future.

Combining EC with *machine learning approaches* for implicitly feature manipulation is also very interesting. From the landmark paper by Yao [68] using GA to evolve artificial neural networks, to EC with deep neural networks [69], [70], there have been a rich body of evidence showing the potential benefit of this direction. Furthermore, in transfer learning, one of the four major types of approaches is feature-based transfer learning [71], where a key part is to miminise the feature distribution difference between the source domain and the target domain. Hellinger distance is a promising feature distribution difference measure, but it can not be used in many cases since it is non-differentiable. However, we can use EC techniques to solve this problem and use the idea of transfer learning to improve the EC performance [72].

There are many *machine learning tasks* with various real-world applications, but most existing work on EC for feature manipulation is for classification. There are many other important fields that require feature manipulation to improve the learning performance, such as clustering and symbolic regression. Especially, many EC techniques have been directly used as a learning algorithm in such fields, for example, GP can be directly used for symbolic regression, and PSO or ABC has been directly used for clustering [73]. So embedding feature manipulation into the learning system of GP for symbolic regression or PSO/ABC for clustering could be an effective way to improve the learning performance, reduce the computational time, simplify the models, etc. In addition, to develop a specific algorithm for a certain problem domain will also be important for users, for example developing a feature selection method for financial analysis or for biomarker detection in biology. It will be very interesting by deeply analysing the solutions in given background to see exactly which features are useful for the task and investigate why they are important. This will be inter-disciplinary research and require experts from the problem domain.

VI. Conclusions

This paper discussed the current work, the strengths, issues, and potential future directions in EC for feature manipulation. Although EC techniques for feature selection have achieved some success, they still face challenges and their potential has not been fully investigated. The computational cost, scalability and feature selection or construction bias are the most serious limitations in EC for feature manipulation. How to deal with feature interactions is probably the most challenging issue in this area. Researchers should focus on utilising the advantages of EC over other approaches and combining EC with other approaches if necessary. Developing good filter measures can solve a number of problems currently faced by EC for feature manipulation and a better representation is also needed. Taking the advantages of the powerful ability of EC for optimisation, such as multi-objective optimisation, combinatorial optimisation and surrogate models, conducting feature construction, and combining EC with machine learning techniques are also interesting to be investigated in the future.

References

[1] E. Alpaydin, *Introduction to machine learning*. The MIT Press, 2004.

[2] H. Liu and H. Motada, Eds., *Feature extraction, construction and selection: A data mining perspective*. Norwell, MA: Kluwer Academic Publishers, 1998.

[3] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[4] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[5] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 4, pp. 131–156, 1997.

[6] K. Neshatian, M. Zhang, and P. Andreae, "A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 645–661, 2012.

[7] K. Krawiec and B. Bhanu, "Visual learning by evolutionary and coevolutionary feature synthesis," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 635–650, 2007.

[8] H.-L. Wei, S. Billings *et al.*, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 162–166, 2007.

[9] B. Xue, M. Zhang, W. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, p. doi: 10.1109/TEVC.2015.2504420, 2015.

[10] D. B. Fogel, *Introduction to Evolutionary Computation*. John Wiley Sons, Inc., 2007.

[11] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.

[12] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, 2015.

[13] J. R. Koza, "Human-competitive results produced by genetic programming," *Genetic Programming and Evolvable Machines*, vol. 11, no. 3-4, pp. 251–284, 2010.

[14] J. Zhang, Z.-H. Zhan, Y. Lin, N. Chen, Y.-J. Gong, J. hui Zhong, H. Chung, Y. Li, and Y. hui Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, 2011.

[15] K. Krawiec, "Genetic programming-based construction of features for machine learning and knowledge discovery tasks," *Genetic Programming and Evolvable Machines*, vol. 3, no. 4, pp. 329–343, 2002.

[16] B. Tran, B. Xue, and M. Zhang, "Genetic programming for feature construction and selection in classification on high-dimensional data," *Memetic Computing*, vol. 8, no. 1, pp. 3–15, 2015.

[17] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[18] S. Ahmed, M. Zhang, and L. Peng, "Prediction of detectable peptides in ms data using genetic programming," in *the 2014 Conference Companion on Genetic and Evolutionary Computation Companion (GECCO)*, 2014, pp. 37–38.

[19] M. Mohamad, S. Omatu, S. Deris, and M. Yoshioka, "A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 6, pp. 813–822, 2011.

[20] P. Ghamisi, M. Couceiro, and J. Benediktsson, "A novel feature selection approach based on FODPSO and SVM," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2935 –2947, 2015.

[21] Y. Zhai, Y.-S. Ong, and I. Tsang, "The emerging "big dimensionality"," *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 14–26, 2014.

[22] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, 1st ed. Bristol, UK, UK: IOP Publishing Ltd., 1997.

[23] A. E. Eiben and G. Rudolph, "Theory of evolutionary algorithms: a bird's eye view," *Theoretical Computer Science*, vol. 229, no. 1, pp. 3–9, 1999.

[24] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 70–76, 2007.

[25] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261–276, 2014.

[26] B. Tran, B. Xue, and M. Zhang, "Improved pso for feature selection on high-dimensional datasets," in *Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, 2014, vol. 8886, pp. 503–515.

[27] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[28] T. N. Hisao Ishibuchi, "Multi-objective pattern and feature selection by a genetic algorithm," in *Proceeding of the 2nd Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*, 2000, pp. 1069–1076.

[29] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, "Multi-objective evolutionary algorithms for filter based feature selection in classification," *International Journal on Artificial Intelligence Tools*, vol. 22, no. 04, pp. 1 350 024:1–31, 2013.

[30] W. Albukhanajer, J. Briffa, and Y. Jin, "Evolutionary multiobjective image feature extraction in the presence of noise," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1757–1768, 2015.

[31] H. B. Bguyen, B. Xue, and M. Zhang, "A subset similarity guided method for multi-objective feature selection," in *Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2016)*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2016, vol. 9592, pp. 198–310.

[32] S. Ahmed, M. Zhang, and L. Peng, "Enhanced feature selection for biomarker discovery in LC-MS data using GP," in *IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 584–591.

[33] Y. Zhai, M. Tan, I. Tsang, and Y. S. Ong, "Discovering support and affiliated features from very high dimensions," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012, pp. 1 – 8.

[34] Y. Yu, C. Qian, and Z.-H. Zhou, "Switch analysis for running time analysis of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 777–792, 2015.

[35] B. Xue, M. Zhang, and W. N. Browne, "A comprehensive comparison on evolutionary feature selection approaches to classification," *International Journal of Computational Intelligence and Applications*, vol. 14, no. 02, p. 1550008, 2015.

[36] S. K. Singhi and H. Liu, "Feature subset selection bias for classification learning," in *23rd International Conference on Machine Learning (ICML)*. ACM, 2006, pp. 849–856.

[37] B. Tran, B. Xue, and M. Zhang, "Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias," *Connection Science*, p. to appear, 2016.

[38] M. Tan, I. Tsang, and L. Wang, "Minimax sparse logistic regression for very high-dimensional feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1609–1622, 2013.

[39] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognition*, vol. 48, no. 2, pp. 438–446, 2015.

[40] Q. Mao and I.-H. Tsang, "A feature selection method for multivariate performance measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2051–2063, 2013.

[41] P. Mitra, C. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[42] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–14, 2013.

[43] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424 –1437, 2004.

[44] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2014.

[45] ——, "Survey of multiobjective evolutionary algorithms for data mining: Part ii," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[46] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[47] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[48] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO?08)*. ACM, 2008, pp. 673–680.

[49] M.-H. Tayarani-N and A. Prugel-Bennett, "On the landscape of combinatorial optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 420–434, 2014.

[50] S.-C. Horng, S.-Y. Lin, L. H. Lee, and C.-H. Chen, "Memetic algorithm for real-time combinatorial stochastic simulation optimization problems with performance analysis," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1495–1509, 2013.

[51] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.

[52] E. Munoz, J. Cadenas, Y. Ong, and G. Acampora, "Memetic music composition," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 1–15, 2016.

[53] W. Dong, T. Chen, P. Tino, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 797–822, 2013.

[54] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[55] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[56] B. Liu, Q. Zhang, and G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[57] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[58] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evolutionary computation*, vol. 21, no. 2, pp. 313–340, 2013.

[59] G. Karafotias, M. Hoogendoorn, and A. Eiben, "Parameter control in evolutionary algorithms: Trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 19, pp. 167 – 187, 2015.

[60] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[61] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[62] P. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 2, pp. 121–144, 2010.

[63] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, "Evolving diverse ensembles using genetic programming for classification with unbalanced data," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 368–386, 2013.

[64] J. Zhong, Y. Ong, and W. Cai, "Self-learning gene expression programming," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 65–80, 2016.

[65] E. J. Vladislavleva, G. F. Smits, and D. Den Hertog, "Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 333–349, 2009.

[66] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Two-tier genetic programming: Towards raw pixel-based image classification," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 291–12 301, 2012.

[67] H. Liu and H. Motoda, *Instance selection and construction for data mining*. Springer Science & Business Media, 2013, vol. 608.

[68] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[69] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Transactions o Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3263–3277, 2015.

[70] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR '15)*, 2015.

[71] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[72] A. Gupta, Y. Ong, and L. Feng, "Multifactorial evolution: towards evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 10, p. Doi:10.1109/TEVC.2015.2458037, 2015.

[73] M. OMRAN, A. P. ENGELBRECHT, and A. SALMAN, "Particle swarm optimization method for image clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 03, pp. 297–321, 2005.