JOURNAL OF THE ROYAL SOCIETY OF NEW ZEALAND
2019, VOL. 49, NO. 2, 205–228
https://doi.org/10.1080/03036758.2019.1609052

Taylor & Francis
Taylor & Francis Group

REVIEW ARTICLE

A survey on evolutionary machine learning

Harith Al-Sahaf, Ying Bi, Qi Chen, Andrew Lensen, Yi Mei, Yanan Sun, Binh Tran, Bing Xue and Mengjie Zhang

School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

ABSTRACT

Artificial intelligence (AI) emphasises the creation of intelligent machines/systems that function like humans. AI has been applied to many real-world applications. Machine learning is a branch of AI based on the idea that systems can learn from data, identify hidden patterns, and make decisions with little/minimal human intervention. Evolutionary computation is an umbrella of population-based intelligent/learning algorithms inspired by nature, where New Zealand has a good international reputation. This paper provides a review on evolutionary machine learning, i.e. evolutionary computation techniques for major machine learning tasks such as classification, regression and clustering, and emerging topics including combinatorial optimisation, computer vision, deep learning, transfer learning, and ensemble learning. The paper also provides a brief review of evolutionary learning applications, such as supply chain and manufacturing for milk/dairy, wine and seafood industries, which are important to New Zealand. Finally, the paper presents current issues with future perspectives in evolutionary machine learning.

ARTICLE HISTORY
Received 30 September 2018
Accepted 15 April 2019

KEYWORDS
Artificial intelligence; machine learning; evolutionary computation; classification; regression; clustering; combinatorial optimisation; deep learning; transfer learning; ensemble learning

Introduction

Artificial intelligence (AI) is a broad umbrella covering a wide range of techniques for building systems that can simulate human intelligence including thinking, behaviours, perception in computers. Although AI was first coined in the 1950s, its applications have flourished in just the last several decades within its core sub-area of machine learning (ML), where computers exhibit the ability to automatically learn and improve without being explicitly programmed.

ML has been applied to many applications in different domains, such as in manufacturing industry, finance, and biomedical problems. Its main tasks include classification, regression, and clustering. The first two tasks are supervised learning in which a model is learnt from a set of labelled data, while the last is unsupervised learning that does not have labelled data. Classification is a task where each example/instance is classified into one of the predefined categories, whereas regression predict numeric outputs for instances. However, both aim to build a model that can correctly predict the output of an unseen instance by observing a set of labelled instances. On the other hand, clustering algorithms aim to learn a model that can group instances into separate clusters based on the intrinsic

CONTACT Mengjie Zhang mengjie.zhang@ecs.vuw.ac.nz; Bing Xue bing.xue@ecs.vuw.ac.nz

characteristics of the unlabelled data. Solving scheduling and combinatorial optimisation problems such as determining product delivery routes and flight scheduling, as well as analysing patterns and recognising objects in computer vision are also important research areas in ML.

Although these ML tasks have been studied for decades, challenges still arise when massive datasets are collected due to advanced technologies and a rapidly growing user market. Firstly, the number of features has increased over time in different domains such as images, gene expression, text, and web mining (Zhang et al. 2016). The search space of recent ML tasks continues to rise. This space may be 'infinite' in some domains such as materials design and drug discovery (Le and Winkler 2016). Secondly, more and more complex applications need to be solved without domain expertise. Therefore, more powerful search techniques are needed to find better solutions. Evolutionary computation (EC) is a sub-field of AI that contains a family of nature-inspired algorithms. These are population-based algorithms, which maintain a population of candidate solutions (individuals) and evolve towards good/optimal solutions. By evolving multiple solutions simultaneously, EC techniques are well known for their good global search ability.

EC techniques can be broadly divided into two main categories: evolutionary algorithms and swarm intelligence (Bäck et al. 1997). Evolutionary algorithms apply Darwinian natural selection principles to search for optimal solutions. Genetic algorithms (GAs) and genetic programming (GP) are widely-used algorithms in this category. Both methods use genetic operators such as crossover and mutation to evolve new individuals. While GAs use a fixed-length bit string representation, GP can work with more flexible structures such as trees and graphs with variable sizes. In contrast, swarm intelligence techniques are inspired by the social behaviours of animals. Typical techniques of this branch are particle swarm optimisation (PSO) and ant colony optimisation (ACO), which mimic birds and ants, respectively. While PSO uses information about the best-found solutions shared among particles to guide the search towards more fruitful areas, ACO works by simulating a communication system based on pheromones between ants about favourable paths to food. There are also other popular EC algorithms such as differential evolution (DE), learning classifier systems (LCS), artificial immune systems (AIS), and artificial bee colony (ABC) algorithms (Bäck et al. 1997).

With the ability to evolve multiple solutions simultaneously, EC techniques have shown significant promise in solving multi-objective problems where optimal solutions need to be considered in the presence of two or more conflicting objectives, e.g. minimising both cost and travel time in flight bookings. Because it is unlikely to have an optimal solution that satisfies both conflicting objectives, a multi-objective method returns a set of nondominated (Pareto optimal) solutions that cannot be improved in one objective without another objective suffering (Zhou et al. 2011). Evolutionary multi-objective optimisation (EMO) is one of the most-studied EC topics recently, with a dramatic increase in publications over the last ten years.

Although a number of surveys exist on the use of EC for machine learning tasks (EML), they focus on a particular task/aspect such as feature selection (Xue et al. 2016), classification using GP (Espejo et al. 2010), a particular EC technique (Neri and Tirronen 2010), technical orientation such as EC and ML (Zhang et al. 2011), or EMO (Zhou et al. 2011). There is no survey that covers EML techniques for different tasks with a non-technical presentation for a broader range of readers. Given the rapid development and growth of this field and its role in facilitating more ML applications, this paper aims to provide a comprehensive survey on using EC techniques for major ML tasks. The remainder of this section briefly summarises the current EML applications in different domains.

Current evolutionary machine learning applications

EML methods have been widely applied to real-world problems in various fields, including agriculture, manufacturing, power and energy, internet/wifi/networking, finance, and healthcare.

In agriculture, EML techniques are used to plan land use (Kaim et al. 2018). The decision making in crop farming (Pal et al. 2016) and fishing (Cobo et al. 2018) have also been addressed by EML.

EML techniques have been widely applied to manufacturing in different industries such as dairy production (Notte et al. 2016), wine production (Mohais et al. 2012), wood production (Zhao et al. 2017), mineral processing (Yu et al. 2011), and transportation scheduling for seafood and milk products (Sethanan and Pitakaso 2016). EML methods can find solutions that help to reduce time and cost for both production and transportation. Supply chain optimisation has been performed using EML methods to reduce held inventory and cost in supply chains of different industries such as food (Cheraghalipour et al. 2018) and fisheries (Tabrizi et al. 2018).

EML techniques have been applied to the energy industry, including load forecasting in power systems (Liao and Tsao 2006) and wind farm design (Hou et al. 2015).

Finance is another important application area of EML. Financial data are often time series data, which are difficult to analyse due to their temporal nature. EML methods have been widely employed for financial data analysis (Wagner et al. 2007), market price prediction (Bagheri et al. 2014), bankrupt ratio analysis (Lakshmi et al. 2016), and credit risk management (Srinivasan and Kamalakannan 2018).

In healthcare and biomedical applications, EML techniques are used for gene sequence analysis, gene mapping, structure prediction and analysis of DNA (Pal et al. 2006), and biomarker identification (Ahmed et al. 2014). Computation of 3D protein structure has been addressed by many EML methods (Correa et al. 2018). EML also shows promising results in important applications such as drug discovery (Le and Winkler 2015) and materials design (Le and Winkler 2016), where the search space is effectively infinite.

In addition, EML techniques have been applied to earthquake prediction (Asim et al. 2018), web service composition (da Silva et al. 2016), cloud computing (Guzek et al. 2015), cyber security (Buczak and Guven 2016), and video games (Yannakakis and Togelius 2018). Readers are referred to (Chiong et al. 2012) for more EML real-world applications.

Organisation

This survey is presented mainly in a task-based manner. The first three sections present EML algorithms for classification, regression, and clustering tasks. Sections 5 and 6 discuss two large EML application areas: computer vision, and scheduling and computational optimisation, respectively. Section 7 is dedicated to evolutionary deep learning, a hot topic in ML. Emerging topics and current issues/challenges are described in

Section 8. Section 9 concludes this paper. Due to the page limit, we cite only representative works. Figure 1 shows the taxonomy and structure of the paper.

Evolutionary computation for classification

EML techniques have been widely used for classification. The aim of classification algorithms is to learn a model/classifier that can correctly classify unseen instances (test data) by observing a set of given instances (training data). Each instance is represented by a set of attributes/variables/features, and a label. The quantity, quality, and representation of the data are important factors influencing the performance of the learned classifiers (Tran et al. 2016b).

Although classifier construction is the main task in classification, some other tasks related to data preprocessing are also crucial. The existence of irrelevant and redundant features negatively affects the performance of learning algorithms. Therefore, feature selection/dimensionality reduction is widely used to remove irrelevant and redundant features, which effectively reduces the search space of a problem and hence improves the learning ability and running time. Feature construction is typically used to create high-level features that can better represent the problem.

The following subsections discuss the main EML techniques that have been proposed for these tasks and related tasks such as unbalanced or missing data.

EC for classifier construction

Many EML techniques are used for classification such as GAs, GP, and PSO. GAs were the earliest EC technique used to evolve classification rules. Many methods have been proposed for general classification problems (Vivekanandan et al. 2013) as well as for specific domains such as text classification (Khaleel et al. 2016) and medical diagnosis (Fidelis et al. 2000). Chernbumroong et al. (2015) proposed a GA-based method for activity classification using data from multiple sensors to recognise a person's activity.

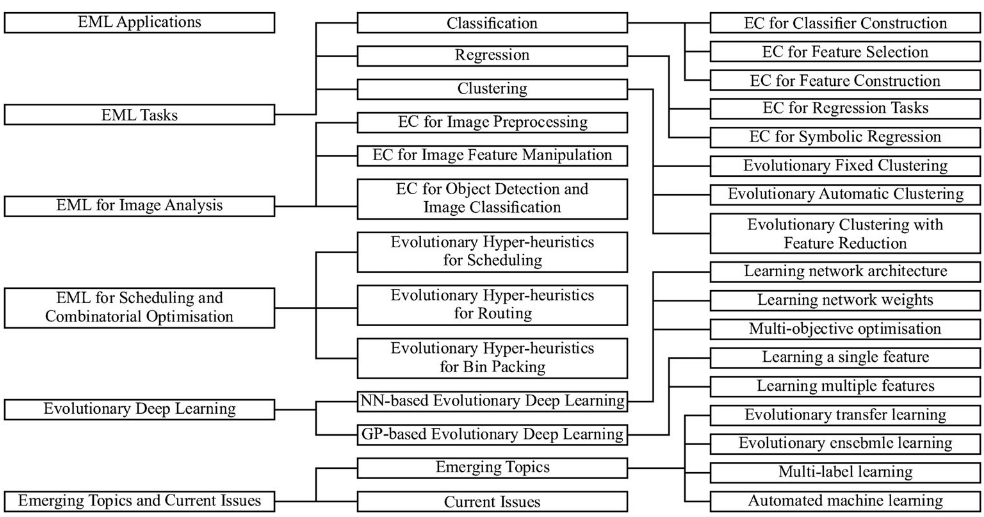

Figure 1. Taxonomy and structure of the paper.

Using a similar vector representation to GAs, PSO has also been proposed for rule induction (Zheng et al. 2014). PSO has been shown to be more flexible than GAs and other classification algorithms.

Unlike PSO and GAs where individuals are represented using vectors, GP has more flexible representations. Discriminant functions are a form of mathematical expression that classify an instance into a class based on different thresholds (Espejo et al. 2010). With a tree-based representation, GP is well suited to developing discriminant functions (Nag and Pal 2016). GP also has a long history in evolving decision trees (Zhao 2007) which are more effective than traditional approaches. GP is also used in inducing classification rules (Luna et al. 2014) which are more representative and use fewer conditions. Rivero et al. (2010) developed a GP-based method to automatically evolve artificial neural networks. LCS are also strong at evolving rules, having been applied in boolean function/classifier learning for multiplexer and even-parity problems (Iqbal et al. 2014).

In addition to evolving classifiers, GP can also deal with common problems in classification. Bhowan et al. (2014) proposed a GP-based classification method that can effectively cope with unbalanced data: a common problem where the number of instances in one class is much smaller than in other class(es), e.g. malignant traffic is infrequent compared to normal traffic in network intrusion detection. Missing data is another common problem in real-world applications, which negatively affects the learning performance or makes some classification algorithms unusable. Tran et al. (2018b) developed a GP-based imputation method that can effectively impute/predict missing values based on the other features.

Instance selection is used to reduce learning time by selecting a good subset of instances that give maximum classification performance. EC techniques have been used for this task (Derrac et al. 2010).

By using a population-based search, EML techniques can evolve better classifiers than using greedy search methods, which use a heuristic for making locally optimal choices in classifier construction. Readers are referred to (Espejo et al. 2010) for further information.

EC for feature selection

Feature selection (FS) is a complex problem. With N original features, there are 2^N different feature subsets, which is impractical for exhaustive search on high-dimensional datasets (with thousands or more features). FS is challenging due to the large search space and possible interactions between features, which makes traditional greedy search prone to local optima. Many non-EC FS methods have been proposed, however, these tend to be limited by these issues. EC techniques are effective at FS as they search globally and can evaluate the whole feature subset while considering possible interactions between features.

GAs were the first EC technique widely applied to FS. GAs use an N-dimension binary vector to represent a feature subset where '1' means the corresponding feature is selected and '0' means it is not (Zhu et al. 2010). Many strategies have been proposed to improve GAs' performance such as improving the crossover and mutation operators (Jeong et al. 2015), enhancing the initial population (Oreski and Oreski 2014) and using different feature subset evaluation methods (Xue et al. 2012) to better guide the search during the evolutionary process.

Although PSO was proposed much later than GAs, a larger number of PSO-based FS methods have been developed during the last decade (Xue et al. 2016). To represent a feature subset, PSO can use bit-strings (binary PSO) as GAs or continuous vectors (continuous PSO) where a threshold is used to determine if the corresponding feature is selected. Although binary PSO naturally suits FS, Engelbrecht (2007) observed limitations leading to inferior performance compared to continuous PSO. Researchers continue to improve PSO's performance for FS in a number of ways including initialisation strategies (Xue et al. 2014), representation (Tran et al. 2018a), updating mechanisms (Gu et al. 2018), integrating local search (Tran et al. 2016a), and evaluating features based on their intrinsic characteristics (aka filter approach) (Nguyen et al. 2016) or performance of learning algorithms (aka wrapper approach) (Xue et al. 2014).

Thanks to the implicit FS process in building GP trees, GP has been proposed for FS implicitly or explicitly. Implicit FS happens in all GP-based classification algorithms (Nag and Pal 2016). Explicit FS methods using GP have also been proposed for feature subset selection (Sandin et al. 2012), feature subset ranking (Neshatian and Zhang 2009a), and feature space exploration (Neshatian and Zhang 2009b). More information about FS using EC can be found in (Xue et al. 2016).

EC for feature construction

Feature construction (FC) is another technique used to enhance the representation of the feature space. FC combines the original features to construct new high-level features with better discriminating ability. The created features are used to augment the original ones (Muharram and Smith 2005) or replace them as a dimensionality reduction solution (Neshatian et al. 2012).

Compared with FS, FC is more challenging due to a larger search space as it must choose not only a good subset of features but also an appropriate set of operators to combine them. The optimal model to combine the original features is unknown in practice. With a flexible representation, GP can automatically evolve models without assuming any model structure. Constructed features can be represented with tree-based GP, where leaf nodes are features/constants and internal nodes are operators.

Many GP-based FC methods have been proposed using single-tree (Neshatian et al. 2012) or multiple-tree representations (Tran et al. 2017). GP is used to construct features that are generally good for all classes (class-independent) (Krawiec 2002) or for a specific class (class-dependent) (Neshatian et al. 2012). Different approaches are also used to evaluate the constructed features during the evolutionary process such as filter (Tran et al. 2017), wrapper (Smith and Bull 2005), or a combination (Tran et al. 2016b).

In addition to GP, GAs (Alfred 2008) and PSO (Dai et al. 2014) have also been proposed and shown promise for FC.

Evolutionary computation for regression

Regression is a major ML task that attempts to identify and express the underlying relationship between input features/variables and the target variable(s). Regression analysis is utilised for forecasting in widespread areas, such as finance, traffic, medicine, and biology (Glaeser and Nathanson 2017). To build a regression model, two main tasks need to be solved: model identification and parameter estimation. Many EC techniques have been proposed for these two tasks.

EC for regression tasks

Model identification

Paterlini and Minerva (2010) developed a new GA method which not only selects the input features but also determines the most appropriate mathematical transformations on these features. A multi-objective GA method (Sinha et al. 2015) was proposed to identify regression models with a good balance between empirical error and model complexity. An evolutionary algorithm was proposed for fuzzy regression by choosing the best fuzzy function within a predefined library (Buckley and Feuring 2000). PSO was used to generate the structures of fuzzy models in a nonlinear polynomial form (Chan et al. 2011).

Parameter estimation

A large number of EC methods have been proposed for parameter estimation for complicated or non-differentiable regression models. A GA method (Zhou and Wang 2005) was used for least-squares estimation of parameters in linear regression models. A GA with seven different crossover operators for parameter estimation has been proposed (Kapanoglu et al. 2007). The convergence of the GA method was explored by analysing the convergence of parameters in regression models with different levels of difficulty. Chen et al. (2010) employed PSO to optimise the parameters for orthogonal forward regression.

EC for symbolic regression

Some EC techniques for regression are able to learn directly from the data and evolve both the structure and parameters of the regression models simultaneously. This task is known as symbolic regression. The distinguishing characteristic of symbolic regression is its interpretability, which can provide domain experts with meaningful insight into the underlying data generating process and highlight the most relevant features.

The symbolic nature of GP solutions and its flexible representation make GP a very suitable approach for symbolic regression (Vyas et al. 2018).

Interpretability is a distinct property of symbolic models, with which the models are able to distil novel knowledge (Schmidt and Lipson 2009). Many studies have improved the interpretability of models evolved by GP. A typical approach is introducing parsimony pressure into GP, which considers the size of the solutions in their fitness evaluation. Parsimony pressure was added to the fitness function as an adaptive penalty based on growth metrics in individuals and the population (Poli and McPhee 2008). A new FS method based on a permutating test was developed, producing GP regression models with good interpretability (Chen et al. 2017).

Prediction/generalisation ability is another important metric for regression techniques. The validation set, which is the most widespread mechanism for improving generalisability in ML, is also used for symbolic regression (Schmidt and Lipson 2009). Geometric semantic GP, which drives the search in GP using the semantic information, has been shown to have good prediction performance (Chen et al. 2018b).

Controlling the functional complexity of models is an effective way to improve prediction performance. Tikhonov regularisation was introduced into GP to control model complexity (Ni and Rockett 2015). Vladislavleva et al. (2009) used the order of nonlinearity (based on the minimum degree of Chebyshev polynomials) to approximate the complexity of GP models. Chen et al. (2018) introduced the Vapnik-Chervonenkis dimension to directly measure and control the complexity of GP solutions.

Some other EC techniques are also used for symbolic regression. However, most of them are in their infancy. LCS was recently used for symbolic regression for the first time (Naqvi and Browne 2016). AIS (Johnson 2003) was applied to solve symbolic regression tasks. Grammatical evolution (O'Neill and Ryan 2001), which evolves binary strings to select production rules in a grammar definition to generate any kind of programmes, can also be used for symbolic regression.

Evolutionary computation for clustering

Often, data may have no labels and so the previously discussed supervised ML methods cannot be used. ML algorithms developed for this scenario are called unsupervised learning algorithms, which discover underlying patterns within the data (Nanda and Panda 2014). There are several approaches to this problem, but the most studied is clustering. Clustering algorithms split a dataset into a set of clusters, so that data within a cluster are similar, while data in distinct clusters are different. A good clustering result (partition) gives insight into a dataset by splitting it into 'natural' groups. Clustering is widely used in real-world tasks such as text mining, bioinformatics, and image categorisation (Nanda and Panda 2014).

EML has been widely applied to clustering problems (Hruschka et al. 2009) due to its ability to find good partitions in reasonable computational time on 'big data' or when the number of clusters (K) is not known in advance (Garcı´a and Gómez-Flores 2016). The field of evolutionary clustering algorithms can be split into two categories: fixed algorithms that require that K is known, and automatic algorithms which discover K themselves. Fixed clustering algorithms are prevalent historically, whereas most recent work tackles the more difficult automatic clustering problem. The third category of algorithms which has emerged recently uses feature reduction to improve clustering performance. Traditional clustering approaches assume all features of a dataset to be equally useful. This is often untrue: for example, clustering weather records by 'day of the week' is clearly less useful than by 'daily rainfall'. This becomes even more problematic in high-dimensional datasets. EC has also recently been used to reduce the dimensionality of data in clustering (Alelyani et al. 2013).

Evolutionary fixed clustering

The first EC methods used for clustering were GAs, and this continues to be the most popular approach. Initial work (Krovi 1992) used primitive encodings (representations) on small datasets, with two or three clusters and at most 150 instances. Since then, substantial progress has been made on extending GAs to much more difficult problems, with over 40 clusters, and thousands of instances. Several new encoding schemes have been proposed that are suited to different clustering problems (Hruschka et al. 2009).

The label-based encoding scheme represents a partition as a vector of length N for N instances, where each instance has a label of the cluster it is in. This encoding was first proposed in the binary form ($K = 2$) (Krovi 1992), but more general forms have been explored since, such as bioinformatics with 16 clusters (Lu et al. 2004). In recent years, as clustering has been applied to larger datasets, this encoding is seldom used due to its inefficient representation.

The centroid-based encoding scheme is the most popular in recent EC clustering work, with an encoding length of $K \times D$ for D dimensions in the data. This encoding represents each cluster by a set of D features which form the cluster centre (centroid). Each instance is assigned to the cluster whose centroid is closest by distance. One of the pioneering works in this field proposed a hybrid approach of a GA and k-means clustering to balance global and local search (Bandyopadhyay and Maulik 2002). PSO is also often used with this encoding as it can efficiently optimise real-valued problems. The first approach was proposed for image clustering (Omran et al. 2005), with good results compared to GA and traditional methods. Other swarm intelligence methods such as ACO have also seen some use (Handl and Meyer 2007).

GP has also been briefly investigated for fixed clustering. Multi-tree GP was proposed for clustering, where each tree represents a cluster, and an instance is assigned to the tree producing the maximum output (Boric and Estévez 2007). GP is also used to build ensembles of clustering algorithms to produce more robust and accurate partitions (Coelho et al. 2011).

Evolutionary automatic clustering

One of the seminal works in evolutionary automatic clustering is MOCK (Handl and Knowles 2007). MOCK uses a graph-inspired label (locus) GA representation, where each instance's label indicates an instance it has an edge to. The set of graphs in this encoding represents the set of clusters. This encoding is shape-invariant, i.e. clusters are not assumed to be a certain shape (e.g. hyper-spherical) as in many clustering methods. The use of a multi-objective fitness function was also very novel. Recently, many EMO clustering methods have been proposed (Garcı´a and Gómez-Flores 2016), including a number of extensions to MOCK (Garza-Fabre et al. 2018). Other graph-inspired techniques have been proposed, including GPGC, which uses GP to evolve tailored similarity measures for clustering problems (Lensen et al. 2017a).

A flexible-length centroid encoding (Sheng et al. 2016) and a medoid-based encoding have also been used for automatic clustering, primarily with GAs or PSO. A medoid-based encoding is a binary encoding of length N, where an instance is coded as a '1' if it is a medoid and '0' if it is not. A medoid indicates that an instance is the centre of a cluster. This has the advantages of a fixed-length encoding, while also allowing K to be discovered automatically (Lensen et al. 2016).

Many other EC algorithms such as DE, ABC, and GP have also seen some use for automatic clustering (Garcı´a and Gómez-Flores 2016).

Evolutionary clustering with feature reduction

NMA_CFS (Sheng et al. 2008) was a pioneering GA method that simultaneously performs feature selection and clustering, selecting features tailored to the clusters found. Recently, PSO-based approaches have been investigated using sophisticated initialisation and local search methods (Lensen et al. 2017c). Feature weighting for clustering has also been proposed (O'Neill et al. 2018).

FC methods are very effective at improving performance in classification tasks (Espejo et al. 2010) but have seen little use in clustering. An initial wrapper approach was proposed using GP for FC to improve k-means clustering (Lensen et al. 2017b), and embedded approaches have also been proposed (Nanda and Panda 2014). Given the upsurge of high-dimensional data, it is expected that future work will focus on new ways of incorporating feature manipulation techniques into clustering.

Evolutionary computer vision

Utilising EML to tackle a variety of problems in different computer vision tasks such as image classification, image segmentation, object detection, feature extraction, image compression, image registration, image restoration, and image enhancement has received significant attention over the last few decades. Generally, EML for computer vision problems and applications can be categorised based on the application domain (e.g. medical, military, and environment), task (e.g. classification, segmentation, and feature manipulation), and the solution representation (e.g. tree structure, and chromosomes or strings of bits). A brief review of EML methods in computer vision is provided in the following subsections, and interested readers can check (Olague 2016).

EC techniques for image preprocessing

Designing a method to handle tasks such as noise cancellation, image segmentation and image enhancement often requires human intervention and sufficient domain knowledge. EC techniques have been successfully utilised to automatically handle such tasks and such methods do not only remove/reduce the human intervention requirement but also evolve potentially better models compared to the domain-expert designed ones.

Image segmentation aims divides an image into different regions based on some criteria such as the connectivity of the pixels. GP has been applied to image segmentation by automatically evolving a similarity measure (Vojodi et al. 2013) or using object segmentation (Liang et al. 2015). PSO was utilised for road sign segmentation (Mussi et al. 2010), and region identification (Dhanalakshmi et al. 2016). Defining the threshold values for image segmentation is a challenging task that has been tackled using AIS (Cuevas et al. 2012). Other EC techniques such as DE (Maulik and Saha 2009) and ACO (Tao et al. 2007) show significant promise in improving fuzzy clustering for image segmentation by grouping pixels into different clusters.

Edge detection is a very important task that finds the edges between different regions in an image, which helps in finding the boundaries of an object of interest. Lu and Chen (2008) utilised ACO to improve the performance of edge detection; GP has been used to automatically evolve an edge detector in (Fu et al. 2015).

Salient object detection (SOD) identifies the most attention-grabbing regions in an image, as a form of preprocessing that focuses the search into a specific part of the image. Finding an optimal set of weights for different features to improve SOD has been achieved using PSO (Afzali et al. 2017).

EC techniques for image feature manipulation

Traditionally, building or training an image classifier requires a set of features, as operating directly on the raw pixel values is very challenging due to the large search space. Feature manipulation, including feature extraction, feature construction and feature selection, is very important in computer vision and pattern recognition.

GP has been utilised for automatically evolving models that improve existing image descriptors such as speeded-up robust features (Perez and Olague 2013); EMO has been adopted for extracting image features (Albukhanajer et al. 2015).

Image descriptors are used to identify different image keypoints, e.g. lines, corners and spots, in an image and generating the corresponding feature vector. In (Al-Sahaf et al. 2017), GP was utilised to automatically evolve image descriptors that automatically detect keypoints for multi-class texture image classification. This method mimics the well-known and largely utilised local binary pattern (LBP) image descriptor. While both methods operate in a similar fashion, i.e. using a sliding window, they differ in being manually designed by domain-experts (LBP) or automatically designed by the evolutionary process. Furthermore, LBP is designed to detect a specific set of keypoints whereas GP-based descriptors automatically design the keypoints to be detected.

PSO has also been used in conjunction with SIFT for face recognition (Lanzarini et al. 2010). Furthermore, selecting optimal image features by utilising accelerated binary PSO is investigated in (Aneesh et al. 2011). In (Valarmathy and Vanitha 2017), AIS was used for image feature selection in MRI images.

EC techniques for object detection and image classification

Object detection aims to localise the different objects in an image. Bhanu and Lin (2004) used GP for object detection, with promising results.

Image classification is the task of categorising images into different groups (classes) based on their visual content. In order to detect breast cancer in images, GP has been used to classify different cut-outs of medical images into malignant and benign classes (Ryan et al. 2015), whereas Ain et al. (2017) tackled the problem of skin cancer classification in images by utilising GP with a mix of biomedical and LBP features. A GP-based classification method for identifying active tuberculosis in X-ray images was proposed by Burks and Punch (2018). Motivated by the promising results achieved in (Li and Ciesielski 2004), Abdulhamid et al. (2011) further investigated the potential of utilising loops with GP for binary image classification, revealing a number of interesting observations.

Feature extraction is a crucial task that identifies/generates informative features to discriminate the different classes/objects. GP has been shown to perform very well in this regard (Al-Sahaf et al. 2012), even with the presence of noise (Albukhanajer et al. 2015). Perez et al. (2010) utilised PSO to extract features for face and iris localisation. PSO has been used for object (Perlin et al. 2008) and face recognition (Ramadan and Abdel-Kader 2009).

Template matching is a well-known approach for object detection and recognition. An ACO-based method for fingerprint matching was proposed in (Cao et al. 2012), and the results were shown to outperform the state-of-art methods.

Other EC techniques such as LCSs (Kukenys et al. 2011), and AISs (Wang et al. 2008) have been proposed for image classification, and an AIS based identity recognition system was proposed by Silva et al. (2015).

Evolutionary computation for scheduling and combinatorial optimisation

Scheduling and combinatorial optimisation is an important research area with many real-world applications such as manufacturing and cloud computing. These areas have been studied extensively as pure optimisation problems. Recently, more research regards them as machine learning tasks due to the following two main motivations.

First, the environment is often dynamic in reality. For example, in manufacturing, job orders arrive in real time and need to be scheduled immediately. Traditional optimisation approaches such as mathematical programming are not fast enough to respond, and it is necessary to find a heuristic/rule that can generate/adjust the solution in real time effectively.

Second, manually designing an effective optimisation algorithm for a complex problem requires substantial domain expertise and time. Using ML techniques to automatically design algorithms/heuristics saves significant human effort.

ML approaches that search for promising heuristics are called hyper-heuristics (Burke et al. 2013). EC methods have been successfully applied as hyper-heuristics by modelling each individual as a heuristic. In contrast to conventional optimisation, the fitness evaluation is key when EC methods are used as hyper-heuristics. A heuristic is evaluated by applying it to a set of training instances, generating solutions. The fitness of a heuristic is set as the average quality of the solutions it generates.

In the rest of this section, we will provide a brief review on evolutionary hyper-heuristics for classic problems including scheduling, routing and bin packing.

Evolutionary hyper-heuristics for scheduling

Scheduling aims to design a schedule to process a set of jobs by a set of machines at minimal cost and time. Dispatching rules are commonly used to generate schedules in an online fashion. GP-based Hyper-Heuristics (GPHHs) have achieved great success in automatically designing dispatching rules.

In a standard job shop scheduling problem, a dispatching rule is invoked whenever a machine becomes idle. It uses a priority function to prioritise the jobs in the machine's queue and decides the job to be processed next. There have been a number of studies on developing GPHHs to evolve such priority functions for the standard job shop scheduling problem. Branke et al. (2016) and Nguyen et al. (2017) give comprehensive surveys of this area.

In addition to the standard job shop scheduling problem, people have also applied GPHHs for solving other problem variants, such as multi-objective job shop scheduling (Nguyen et al. 2014) and flexible job shop scheduling (Yska et al. 2018).

Evolutionary hyper-heuristics for routing

A routing problem seeks optimal routes subject to some constraints, e.g. delivering to all customers, or visiting all the attractions on a trip.

Oltean and Dumitrescu (2004) employed GPHHs to evolve heuristics which decide the next node to add into the current partial tour in the travelling salesman problem. Weise et al. (2012) developed a GPHH algorithm to evolve heuristics for the arc routing problem, which uses a vehicle to serve the streets on a road network, where each street has an uncertain request. Whenever the vehicle becomes idle, it calculates the priority of the remaining streets to decide the street to be served next. Here, GPHH evolves the priority function. Liu et al. (2017) considered a more realistic problem model and improved the performance of GPHH by designing more features. Jacobsen-Grocott et al. (2017) developed a GPHH approach to the vehicle routing problem with time windows, which serves the nodes rather than edges.

Evolutionary hyper-heuristics for bin packing

Bin packing aims to minimise the number of bins needed to store a set of items. A typical heuristic for bin packing starts with empty bins. Then, for each item, the heuristic calculates a priority value for each bin based on the current situation and places the item into the bin with the best priority. GPHHs have been used to evolve the priority function. Burke et al. (2006) developed a GPHH for one-dimensional bin packing. Burke et al. (2010) and Allen et al. (2009) extended the problem to two-dimensional and 3-dimensional packing, respectively.

Evolutionary deep learning

Deep learning (DL) is a class of ML algorithms that use multiple layers of nonlinear processing units to solve a problem (LeCun et al. 2015). DL has achieved remarkable performance in addressing increasingly complex data with large feature sizes from different domains such as images, gene expression, text, and web mining (Zhang et al. 2016), due to it automatic feature generation and selection capabilities (Bengio 2009). Evolutionary DL (EDL) aims at using EC approaches to improve the usability or improve the performance of DL algorithms. Existing EDL algorithms are mainly composed of neural network-based (NN-EDL) and GP-based (GP-EDL) algorithms.

Neural network-based evolutionary deep learning

NN-EDL algorithms mainly focus on designing network architectures, optimising the weights, and solving multi-objective optimisation problems.

Existing approaches for designing architectures can be divided into two different categories: supervised NN-EDL and unsupervised NN-EDL. One typical work on NN-EDL for unsupervised deep learning is the EUDNN method (Sun et al. 2018). Existing supervised NN-EDL algorithms includes Large-scale Evolution (Real et al. 2017), EvoCNN (Sun et al. 2017), and so on.

There are two main different weight optimisation strategies. The first directly encodes the weights (Lehman et al. 2018), whereas the second searches for the best weights indirectly (Sun et al. 2018).

An NN-based deep learning algorithm with promising performance usually has a large number of parameters, necessitating a large amount of computational resources. However, computational resources are often limited, such as on mobile devices. Thus, maximising performance and minimising computational resources are two conflicting objectives, i.e. a multi-objective (MO) optimisation problem. NN-EDL for MO was highlighted by Sun et al. (2017) and specifically investigated by Dong et al. (2018).

Because NN-based DL often has a large number of parameters, high-performance hardware is used to accelerate their performance, such as the graphics processing unit (GPU), field-programmable gate array and tensor processing unit.

GP-based evolutionary deep learning

GP has been used to achieve DL without NNs. The flexible structure of GP allows it to learn abstract and compact representations with suitable model complexity in a layer-by-layer feature transformation manner, which meet the key characteristics of DL.

GP-EDL has been used to integrate multiple steps to learn a single high-level feature for image classification in a single GP tree. The first method was a multi-tier GP using image filtering, aggregation, and classification tiers to perform region detection, feature extraction, feature construction, and image classification simultaneously (Atkins et al. 2011). Bi et al. (2018) proposed a multi-layer GP method with utilisation of image-related operators to learn high-level features for image classification.

GP-EDL has also been proposed to learn multiple features. Shao et al. (2014) proposed a multi-objective GP with a multi-layer structure to learn features for difficult image classification tasks. Rodriguez-Coayahuitl et al. (2018) defined a structured layered GP for representation learning and introduced deep GP. A GP auto-encoder was designed with an encoding forest and a decoding forest to transform an original representation into a new representation of fewer features using arithmetic operators.

Emerging topics and current issues

This section provides a number of emerging topics and summarises the major issues/challenges in EML while providing future perspective.

Emerging topics

Evolutionary transfer learning

Transfer learning has become increasingly popular in ML in recent years. It aims to improve the performance of learning algorithms in a target task/domain by using useful knowledge extracted from a source task/domain (Pan and Yang 2010). In transfer learning, it is important to address three questions: what to transfer, when to transfer, and how to transfer (Pan and Yang 2010).

Recently, EC methods have been used with transfer learning. Iqbal et al. (2017) transferred subtrees learnt by GP on the source domain to improve the performance of GP on related target tasks. Jiang et al. (2018) transferred the probability distributions of solutions to population generation in a dynamic MO algorithm to reduce the computation cost. The parameters of DE learnt from the source problems were transferred to the target problems by Gong et al. (2015).

More approaches can be investigated in EC using instances transfer, feature representation transfer, parameter transfer, and rational-knowledge transfer (Pan and Yang 2010).

Evolutionary ensemble learning

Ensemble learning algorithms learn multiple learners/models from the training data. An ensemble consists of multiple base learners, which are learnt using traditional learning algorithms. Commonly used ensemble methods include bagging, boosting and stacking (Zhou 2012). Generally, to construct a strong ensemble, the base learner needs to be accurate and diverse (Zhou 2012).

EC methods are also beneficial in ensemble learning in different ways. EC has been combined with learning algorithms to obtain strong SVM ensembles (de Araújo Padilha et al. 2016) and NN ensembles (Pulido et al. 2014). EC has also been used to evolve ensembles using bagging and boosting (Folino et al. 2006). Finally, EMO has been used to improve the diversity of ensembles for difficult problems (Bhowan et al. 2013). Further development of EC in ensemble learning is expected to address the diversity of base learners and the interpretability of ensembles.

Automated machine learning (AutoML)

AutoML aims to automate ML techniques, allowing people without ML domain knowledge to use them for problem-solving. An AutoML method optimises the integration of different methods and their hyper-parameters for data preprocessing, feature engineering, and learning. Well-known AutoML methods include Auto-WEKA, Auto-Sklearn and Auto-Keras, which are based on existing ML libraries.

EC methods have also been used for AutoML. The well-known tree-based pipeline optimisation tool (TPOT) uses GP to evolve a set of data transformations and ML models (Olson and Moore 2016). Chen et al. (2018a) developed an Autostacker method, using an EC algorithm to find the optimal hyper-parameters for ML pipelines. There are still many unexplored opportunities in this topic, such as EMO for AutoML, which need to be investigated in the future.

Current issues

Despite its successes, EML remains an active research area with challenges and opportunities. This section discusses some of its major issues: its theoretical foundation, computational cost, scalability, generalisability, and interpretability.

There has been some theoretical analysis of EML methods on running time, convergence guarantee, and parameter settings (Auger and Doerr 2011). However, current EML methods still lack mathematical foundation, which may prevent scientists and practitioners from using EML methods.

Computational cost is another major issue in existing EML methods. EML methods evaluate a population of individuals at each generation, which often makes them more expensive than many traditional ML methods.

Scalability is a common problem in EML where learning methods do not scale well when datasets increase in size. An increase in the number of features and the number of instances often requires larger memory and longer computation time. This may limit the viability of EML methods in large-scale problems.

Like most ML techniques, EML methods also face the challenge of poor generalisability, due to insufficient data, overfitting, and poor feature choice. For EML methods, poor generalisability is often due to overfitting, where the learnt model perfectly fits the training data, but works poorly on unseen data. The issue of overfitting in EML warrants further investigation in the future.

Interpretability is another important issue in EML and ML. Good interpretability of a learnt model not only provides insights into why it obtains a result but also encourages experts to accept and reuse the model. The use of arcane features and complex models can often lead to poor interpretability. Among EML methods, several methods such as tree-based GP have good interpretability of solutions, which can be further investigated in the future.

Conclusions

This paper provided a comprehensive review of major EC techniques for ML tasks, covering both supervised and unsupervised tasks, and applications. A number of emergent techniques such as evolutionary deep learning and transfer learning were studied. This paper also discussed major current issues and challenges in this area, including scalability, generalisability, and interpretability/comprehensibility of evolved models.

The fast development of hardware such as GPU devices and cloud computing facilities has allowed previously impossible EML tasks to become reality. Involvement and investment from large corporations such as Google, Microsoft, Uber, Huawei, and IBM have made EML methods more practically useful. It is anticipated that EML methods will play a significant role in AI and ML in the next ten years. EML is expected to be applied to most real-world data mining and big data tasks and applications in our daily life. In the future, the AI and EC/ML group at Victoria University of Wellington will seek research collaborations with colleagues who are interested in using AI in science, engineering, commerce/business, humanities and social sciences, education, law, and in the primary industries of New Zealand.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported in part by the Marsden Fund of the New Zealand Government under Contract VUW1509, VUW1614, VUW1615 through the Royal Society of New Zealand, in part by the Novel Transfer Learning Techniques in Genetic Programming for Big Data Mining under Grant 216378/3764 through the Victoria University of Wellington, and Industry Grant E2880/3663 Huawei Technologies.

ORCID

Harith Al-Sahaf http://orcid.org/0000-0003-4633-6135
Ying Bi http://orcid.org/0000-0003-2758-6067
Qi Chen http://orcid.org/0000-0001-9367-4757
Andrew Lensen http://orcid.org/0000-0003-1269-4751
Yi Mei http://orcid.org/0000-0003-0682-1363
Yanan Sun http://orcid.org/0000-0001-6374-1429
Binh Tran http://orcid.org/0000-0002-2445-1231
Bing Xue http://orcid.org/0000-0002-4865-8026
Mengjie Zhang http://orcid.org/0000-0003-4463-9538

References

Abdulhamid F, Neshatian K, Zhang M. 2011. Image recognition using genetic programming with loop structures. In: Proceedings of the 26th International Conference on Image and Vision Computing New Zealand; vol. 29. p. 553–558.

Afzali S, Xue B, Al-Sahaf H, Zhang M. 2017. A supervised feature weighting method for salient object detection using PSO. In: Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence. IEEE. p. 1–8.

Ahmed S, Zhang M, Peng L, Xue B. 2014. Multiple feature construction for effective biomarker identification and classification using genetic programming. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM. p. 249–256.

Ain QU, Xue B, Al-Sahaf H, Zhang M. 2017. Genetic programming for skin cancer detection in dermoscopic images. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE. p. 2420–2427.

Al-Sahaf H, Al-Sahaf A, Xue B, Johnston M, Zhang M. 2017. Automatically evolving rotationinvariant texture image descriptors by genetic programming. IEEE Transactions on Evolutionary Computation. 21(1):83–101.

Al-Sahaf H, Song A, Neshatian K, Zhang M. 2012. Extracting image features for classification by two-tier genetic programming. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE.

Albukhanajer W, Briffa J, Jin Y. 2015. Evolutionary multiobjective image feature extraction in the presence of noise. IEEE Transactions on Cybernetics. 45(9):1757–1768.

Alelyani S, Tang J, Liu H. 2013. Feature selection for clustering: a review. In: Data clustering: algorithms and applications. Milton Park: CRC Press; p. 29–60.

Alfred R. 2008. A genetic-based feature construction method for data summarisation. Berlin, Heidelberg: Springer. p. 39–50.

Allen S, Burke EK, Hyde M, Kendall G. 2009. Evolving reusable 3D packing heuristics with genetic programming. In: Proceedings of Genetic and Evolutionary Computation Conference. ACM. p. 931–938.

Aneesh M, Masand AA, Manikantan K. 2011. Optimal feature selection based on image preprocessing using accelerated binary particle swarm optimization for enhanced face recognition. In: Proceedings of the 2011 International Conference on Communication Technology and System Design. Elsevier. p. 750–758.

Asim KM, Idris A, Iqbal T, Martı´nez-Álvarez F. 2018. Seismic indicators based earthquake predictor system using genetic programming and adaboost classification. Soil Dynamics and Earthquake Engineering. 111:1–7.

Atkins DL, Neshatian K, Zhang M. 2011. A domain independent genetic programming approach to automatic feature extraction for image classification. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE. p. 238–245.

Auger A, Doerr B. 2011. Theory of randomized search heuristics: foundations and recent developments. vol. 1. Singapore: World Scientific.

Bäck T, Fogel DB, Michalewicz Z. 1997. Handbook of evolutionary computation. Oxford: CRC Press.

Bagheri A, Peyhani HM, Akbari M. 2014. Financial forecasting using anfis networks with quantum-behaved particle swarm optimization. Expert Systems with Applications. 41(14):6235–6250.

Bandyopadhyay S, Maulik U. 2002. An evolutionary technique based on k-means algorithm for optimal clustering in R^N. Information Science. 146(1–4):221–237.

Bengio Y. 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning. 2(1):1–127.

Bhanu B, Lin Y. 2004. Object detection in multi-modal images using genetic programming. Applied Soft Computing. 4(2):175–201.

Bhowan U, Johnston M, Zhang M, Yao X. 2013. Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Transactions on Evolutionary Computation. 17(3):368–386.

Bhowan U, Johnston M, Zhang M, Yao X. 2014. Reusing genetic programming for ensemble selection in classification of unbalanced data. IEEE Transactions on Evolutionary Computation. 18 (6):893–908.
Bi Y, Xue B, Zhang M. 2018. An automatic feature extraction approach to image classification using genetic programming. In: Proceedings of the International Conference on the Applications of Evolutionary Computation. Springer. p. 421–438.
Boric N, Estévez PA. 2007. Genetic programming-based clustering using an information theoretic fitness measure. In: Proceedings of the IEEE Congress on Evolutionary Computation. p. 31–38.
Branke J, Nguyen S, Pickardt CW, Zhang M. 2016. Automated design of production scheduling heuristics: A review. IEEE Transactions on Evolutionary Computation. 20(1):110–124.
Buckley JJ, Feuring T. 2000. Linear and non-linear fuzzy regression: evolutionary algorithm solutions. Fuzzy Sets and Systems. 112(3):381–394.
Buczak AL, Guven E. 2016. A survey of data mining and machine learning methods for cybersecurity intrusion detection. IEEE Communications Surveys & Tutorials. 18(2):1153–1176.
Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E, Qu R. 2013. Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society. 64(12):1695–1724.
Burke EK, Hyde M, Kendall G, Woodward J. 2010. A genetic programming hyper-heuristic approach for evolving 2-D strip packing heuristics. IEEE Transactions on Evolutionary Computation. 14(6):942–958.
Burke EK, Hyde MR, Kendall G. 2006. Evolving bin packing heuristics with genetic programming. In: Parallel problem solving from nature. Berlin: Springer; p. 860–869.
Burks AR, Punch WF. 2018. Genetic programming for tuberculosis screening from raw X-ray images. In: Proceedings of the 2018 Genetic and Evolutionary Computation Conference. ACM. p. 1214–1221.
Cao K, Yang X, Chen X, Zang Y, Liang J, Tian J. 2012. A novel ant colony optimization algorithm for large-distorted fingerprint matching. Pattern Recognition. 45(1):151–161.
Chan KY, Dillon TS, Kwong CK. 2011. Modeling of a liquid epoxy molding process using aparticle swarm optimization-based fuzzy regression approach. IEEE Transactions on Industrial Informatics. 7(1):148–158.
Chen B, Wu H, Mo W, Chattopadhyay I, Lipson H. 2018a. Autostacker: a compositional evolutionary learning system. arXiv preprint arXiv:180300684.
Chen Q, Xue B, Zhang M. 2018b. Improving generalisation of genetic programming for symbolic regression with angle-driven geometric semantic operators. IEEE Transactions on Evolutionary Computation. doi:10.1109/TEVC.2018.2869621.
Chen Q, Zhang M, Xue B. 2017. Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Transactions on Evolutionary Computation. 21(5):792–806.
Chen Q, Zhang M, Xue B. 2018. Structural risk minimisation-driven genetic programming for enhancing generalisation in symbolic regression. IEEE Transactions on Evolutionary Computation. doi:10.1109/TEVC.2018.2881392.
Chen S, Hong X, Harris CJ. 2010. Particle swarm optimization aided orthogonal forward regression for unified data modeling. IEEE Transactions on Evolutionary Computation. 14(4):477–499.
Cheraghalipour A, Paydar MM, Hajiaghaei-Keshteli M. 2018. A bi-objective optimization for citrus closed-loop supply chain using Pareto-based algorithms. Applied Soft Computing. 69:33–59.
Chernbumroong S, Cang S, Yu H. 2015. Genetic algorithm-based classifiers fusion for multisensor activity recognition of elderly people. IEEE Journal of Biomedical and Health Informatics. 19 (1):282–289.
Chiong R, Weise T, Michalewicz Z. 2012. Variants of evolutionary algorithms for real-world applications. Berlin: Springer.
Cobo Á, Llorente I, Luna L, Luna M. 2018. A decision support system for fish farming using particle swarm optimization. Computers and Electronics in Agriculture. doi:10.1016/j.compag.2018.03.036
Coelho ALV, Fernandes E, Faceli K. 2011. Multi-objective design of hierarchical consensus functions for clustering ensembles via genetic programming. Decision Support Systems. 51 (4):794–809.

Correa L, Borguesan B, Farfan C, Inostroza-Ponta M, Dorn M. 2018. A memetic algorithm for 3D protein structure prediction problem. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 15(3):690–704.
Cuevas E, Osuna-Enciso V, Zaldivar D, Pérez-Cisneros M, Sossa H. 2012. Multithreshold segmentation based on artificial immune systems. Mathematical Problems in Engineering. 2012 (2012):1–20.
Dai Y, Xue B, Zhang M. 2014. New Representations in PSO for Feature Construction in Classification. In: Proceedings of the applications of evolutionary computation. Springer; p. 476–488.
da Silva AS, Ma H, Zhang M. 2016. Genetic programming for QoS-aware web service composition and selection. Soft Computing. 20(10):3851–3867.
de Araújo Padilha CA, Barone DAC, Neto ADD. 2016. A multi-level approach using genetic algorithms in an ensemble of least squares support vector machines. Knowledge-Based Systems. 106:85–95.
Derrac J, Garcı´a S, Herrera F. 2010. A survey on evolutionary instance selection and generation. International Journal of Applied Metaheuristic Computing. 1(1):60–92.
Dhanalakshmi L, Ranjitha S, Suresh HN. 2016. A novel method for image processing using particle swarm optimization technique. In: Proceedings of 2016 International Conference on Electrical, Electronics, and Optimization Techniques. IEEE. p. 3357–3363.
Dong JD, Cheng AC, Juan DC, Wei W, Sun M. 2018. Dpp-net: device-aware progressive search for pareto-optimal neural architectures. arXiv preprint arXiv:180608198.
Engelbrecht AP. 2007. Computational intelligence: An introduction, 2nd ed. Hoboken (NJ): Wiley.
Espejo P, Ventura S, Herrera F. 2010. A survey on the application of genetic programming to classification. IEEE Transactions on systems, Man, and Cybernetics. Part C: Applications and Reviews. 40(2):121–144.
Fidelis MV, Lopes HS, Freitas AA. 2000. Discovering comprehensible classification rules with a genetic algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation; vol. 1; July. p. 805–810.
Folino G, Pizzuti C, Spezzano G. 2006. GP ensembles for large-scale data classification. IEEE Transactions on Evolutionary Computation. 10(5):604–616.
Fu W, Johnston M, Zhang M. 2015. Distribution-based invariant feature construction using genetic programming for edge detection. Soft Computing. 19(8):2371–2389.
Garcı´a AJ, Gómez-Flores W. 2016. Automatic clustering using nature-inspired metaheuristics: a survey. Applied Soft Computing. 41:192–213.
Garza-Fabre M, Handl J, Knowles JD. 2018. An improved and more scalable evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation. 22(4):515–535.
Glaeser EL, Nathanson CG. 2017. An extrapolative model of house price dynamics. Journal of Financial Economics. 126(1):147–170.
Gong W, Yan X, Liu X, Cai Z. 2015. Parameter extraction of different fuel cell models with transferred adaptive differential evolution. Energy. 86:139–151.
Gu S, Cheng R, Jin Y. 2018. Feature selection for high-dimensional classification using a competitive swarm optimizer. Soft Computing. 22(3):811–822.
Guzek M, Bouvry P, Talbi EG. 2015. A survey of evolutionary computation for resource management of processing in cloud computing. IEEE Computational Intelligence Magazine. 10(2):53–67.
Handl J, Knowles JD. 2007. An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation. 11(1):56–76.
Handl J, Meyer B. 2007. Ant-based and swarm-based clustering. Swarm Intelligence. 1(2):95–113.
Hou P, Hu W, Soltani M, Chen Z. 2015. Optimized placement of wind turbines in large-scale offshore wind farm using particle swarm optimization algorithm. IEEE Transactions on Sustainable Energy. 6(4):1272–1282.
Hruschka ER, Campello RJGB, Freitas AA, de Carvalho ACPLF. 2009. A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C. 39 (2):133–155.

Iqbal M, Browne WN, Zhang M. 2014. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. IEEE Transactions on Evolutionary Computation. 18 (4):465–480.
Iqbal M, Xue B, Al-Sahaf H, Zhang M. 2017. Cross-domain reuse of extracted knowledge in genetic programming for image classification. IEEE Transactions on Evolutionary Computation. 21 (4):569–587.
Jacobsen-Grocott J, Mei Y, Chen G, Zhang M. 2017. Evolving heuristics for dynamic vehicle routing with time windows using genetic programming. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE. p. 1948–1955.
Jeong YS, Shin KS, Jeong MK. 2015. An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems. Journal of the Operational Research Society. 66(4):529–538.
Jiang M, Huang Z, Qiu L, Huang W, Yen GG. 2018. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Transactions on Evolutionary Computation. 22(4):501–514.
Johnson CG. 2003. Artificial immune system programming for symbolic regression. In: Proceedings of the European Conference on Genetic Programming. Springer. p. 345–353.
Kaim A, Cord AF, Volk M. 2018. A review of multi-criteria optimization techniques for agricultural land use allocation. Environmental Modelling & Software. 105:79–93.
Kapanoglu M, Ozan Koc I, Erdogmus S. 2007. Genetic algorithms in parameter estimation for nonlinear regression models: an experimental approach. Journal of Statistical Computation and Simulation. 77(10):851–867.
Khaleel MI, Hmeidi II, Najadat HM. 2016. An automatic text classification system based on genetic algorithm. In: Proceedings of the 3rd Multidisciplinary International Social Networks Conference on Social Informatics. ACM. p. 31:1–31:7.
Krawiec K. 2002. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. Genetic Programming and Evolvable Machines. 3:329–343.
Krovi R. 1992. Genetic algorithms for clustering: a preliminary investigation. In: Proceedings of the 25th Hawaii International Conference on System Sciences; vol. 4. IEEE. p. 540–544.
Kukenys I, Browne WN, Zhang M. 2011. Transparent, online image pattern classification using a learning classifier system. In: Proceedings of the 2011 European Conference on the Applications of Evolutionary Computation. Springer. p. 183–193.
Lakshmi TM, Martin A, Venkatesan VP. 2016. A genetic bankrupt ratio analysis tool using a genetic algorithm to identify influencing financial ratios. IEEE Transactions on Evolutionary Computation. 20(1):38–51.
Lanzarini L, Battaglia JL, Maulini J, Hasperué W. 2010. Face recognition using SIFT and binary PSO descriptors. In: Proceedings of the 32nd International Conference on Information Technology Interfaces. IEEE. p. 557–562.
Le TC, Winkler DA. 2015. A bright future for evolutionary methods in drug design. ChemMedChem. 10(8):1296–1300.
Le TC, Winkler DA. 2016. Discovery and optimization of materials using evolutionary approaches. Chemical Reviews. 116(10):6107–6132.
LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. Nature. 521(7553):436–444.
Lehman J, Chen J, Clune J, Stanley KO. 2018. ES is more than just a traditional finitedifference approximator. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM. p. 450–457.
Lensen A, Xue B, Zhang M. 2016. Particle swarm optimisation representations for simultaneous clustering and feature selection. In: Proceedings of the Symposium Series on Computational Intelligence. IEEE. p. 1–8.
Lensen A, Xue B, Zhang M. 2017a. GPGC: genetic programming for automatic clustering using a flexible non-hyper-spherical graph-based approach. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM. p. 449–456.
Lensen A, Xue B, Zhang M. 2017b. New representations in genetic programming for feature construction in k-means clustering. In: Simulated Evolution and Learning. Springer. p. 543–555.

Lensen A, Xue B, Zhang M. 2017c. Using particle swarm optimisation and the silhouette metric to estimate the number of clusters, select features, and perform clustering. In: Proceedings of the Applications of Evolutionary Computation. Springer. p. 538–554.
Li X, Ciesielski V. 2004. Using loops in genetic programming for a two class binary image classification problem. In: Proceedings of the 17th Australian Joint Conference on Artificial Intelligence; vol. 3339. Springer. p. 898–909.
Liang Y, Zhang M, Browne W. 2015. A supervised figure-ground segmentation method using genetic programming. In: Proceedings of the 18th European Conference on the Applications of Evolutionary Computation; vol. 9028. Springer. p. 491–503.
Liao GC, Tsao TP. 2006. Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. IEEE Transactions on Evolutionary Computation. 10(3):330–340.
Liu Y, Mei Y, Zhang M, Zhang Z. 2017. Automated heuristic design using genetic programming hyper-heuristic for uncertain capacitated arc routing problem. In: Proceedings of Genetic and Evolutionary Computation Conference. ACM. p. 290–297.
Lu DS, Chen CC. 2008. Edge detection improvement by ant colony optimization. Pattern Recognition Letters. 29(4):416–425.
Lu Y, Lu S, Fotouhi F, Deng Y, Brown SJ. 2004. Incremental genetic k-means algorithm and its application in gene expression data analysis. BMC Bioinformatics. 5:172–181.
Luna JM, Romero JR, Romero C, Ventura S. 2014. On the use of genetic programming for mining comprehensible rules in subgroup discovery. IEEE Transactions on Cybernetics. 44(12):2329–2341.
Maulik U, Saha I. 2009. Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognition. 42(9):2135–2149.
Mohais A, Schellenberg S, Ibrahimov M, Wagner N, Michalewicz Z. 2012. An evolutionary approach to practical constraints in scheduling: a case-study of the wine bottling problem. Berlin, Heidelberg: Springer. p. 31–58.
Muharram M, Smith G. 2005. Evolutionary constructive induction. IEEE Transactions on Knowledge and Data Engineering. 17:1518–1528.
Mussi L, Cagnoni S, Cardarelli E, Daolio F, Medici P, Porta PP. 2010. GPU implementation of a road sign detector based on particle swarm optimization. Evolutionary Intelligence. 3(3):155–169.
Nag K, Pal N. 2016. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. IEEE Transactions on Cybernetics. 46(2):499–510.
Nanda SJ, Panda G. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm and Evolutionary Computation. 16:1–18.
Naqvi SS, Browne WN. 2016. Adapting learning classifier systems to symbolic regression. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC). p. 2209–2216.
Neri F, Tirronen V. 2010. Recent advances in differential evolution: a survey and experimental analysis. Artificial Intelligence Review. 33(1–2):61–106.
Neshatian K, Zhang M. 2009a. Genetic programming for feature subset ranking in binary classification problems. In: Vanneschi L, Gustafson S, Moraglio A, De Falco I, Ebner M, editor. Berlin, Heidelberg: Springer Genetic Programming; p. 121–132.
Neshatian K, Zhang M. 2009b. Pareto front feature selection: using genetic programming to explore feature space. In: Proceedings of the Conference on Genetic and Evolutionary Computation. p. 1027–1034.
Neshatian K, Zhang M, Andreae P. 2012. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. IEEE Transactions on Evolutionary Computation. 16(5):645–661.
Nguyen BH, Xue B, Andreae P. 2016. Mutual information for feature selection: estimation or counting? Evolutionary Intelligence. 9(3):95–110.
Nguyen S, Mei Y, Zhang M. 2017. Genetic programming for production scheduling: a survey with a unified framework. Complex & Intelligent Systems. 3(1):41–66.
Nguyen S, Zhang M, Johnston M, Tan KC. 2014. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming. IEEE Transactions on Evolutionary Computation. 18(2):193–208.

Ni J, Rockett P. 2015. Tikhonov regularization as a complexity measure in multiobjective genetic programming. IEEE Transactions on Evolutionary Computation. 19(2):157–166.
Notte G, Pedemonte M, Cancela H, Chilibroste P. 2016. Resource allocation in pastoral dairy production systems: evaluating exact and genetic algorithms approaches. Agricultural Systems. 148:114–123.
Olague G. 2016. Evolutionary computer vision: the first footprints. Berlin: Springer.
Olson RS, Moore JH. 2016. Tpot: a tree-based pipeline optimization tool for automating machine learning. In: Proceedings of the Workshop on Automatic Machine Learning. p. 66–74.
Oltean M, Dumitrescu D. 2004. Evolving TSP heuristics using multi expression programming. In: Proceedings of the International Conference on Computational Science. Springer. p. 670–673.
Omran MGH, Engelbrecht AP, Salman AA. 2005. Particle swarm optimization method for image clustering. International Journal of Pattern Recognition and Artificial Intelligence. 19(3):297–321.
O'Neill D, Lensen A, Xue B, Zhang M. 2018. Particle swarm optimisation for feature selection and weighting in high-dimensional clustering. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE. p. 1–8.
O'Neill M, Ryan C. 2001. Grammatical evolution. IEEE Transactions on Evolutionary Computation. 5(4):349–358.
Oreski S, Oreski G. 2014. Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert Systems with Applications. 41(4):2052–2064.
Pal BB, Roy S, Kumar M. 2016. A genetic algorithm to goal programming model for crop production with interval data uncertainty. In: Handbook of research on natural computing for optimization problems. IGI Global; p. 30–65.
Pal SK, Bandyopadhyay S, Ray SS. 2006. Evolutionary computation in bioinformatics: A review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 36 (5):601–615.
Pan SJ, Yang Q. 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. 22(10):1345–1359.
Paterlini S, Minerva T. 2010. Regression model selection using genetic algorithms. In: Proceedings of the 11th WSEAS international conference on evolutionary computing. p. 19–27.
Perez CA, Aravena CM, Vallejos JI, Estevez PA, Held CM. 2010. Face and iris localization using templates designed by particle swarm optimization. Pattern Recognit Lett. 31(9):857–868.
Perez CB, Olague G. 2013. Genetic programming as strategy for learning image descriptor operators. Intelligent Data Analysis. 17(4):561–583.
Perlin HA, Lopes HS, Centeno TM. 2008. Particle swarm optimization for object recognition in computer vision. In: Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer. p. 11–21.
Poli R, McPhee NF. 2008. Parsimony pressure made easy. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM. p. 1267–1274.
Pulido M, Melin P, Castillo O. 2014. Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican stock exchange. Information Sciences. 280:188–204.
Ramadan RM, Abdel-Kader RF. 2009. Particle swarm optimization for human face recognition. In: Proceedings of the 2009 IEEE International Symposium on Signal Processing and Information Technology. IEEE. p. 579–584.
Real E, Moore S, Selle A, Saxena S, Suematsu YL, Tan J, Le Q, Kurakin A. 2017. Large-scale evolution of image classifiers. In: Proceedings of Machine Learning Research. p. 2902–2911.
Rivero D, Dorado J, Rabunal J, Pazos A. 2010. Generation and simplification of artificial neural networks by means of genetic programming. Neurocomputing. 73(16):3200–3223.
Rodriguez-Coayahuitl L, Morales-Reyes A, Escalante HJ. 2018. Structurally layered representation learning: Towards deep learning through genetic programming. In: Proceedings of the European Conference on Genetic Programming. Springer. p. 271–288.
Ryan C, Fitzgerald J, Krawiec K, Medernach D. 2015. Image classification with genetic programming: building a stage 1 computer aided detector for breast cancer. In: Handbook of Genetic Programming Applications. Springer. p. 245–287.

Sandin I, Andrade G, Viegas F, Madeira D, Rocha L, Salles T, Gonçalves M. 2012. Aggressive and effective feature selection using genetic programming. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE. p. 1–8.
Schmidt M, Lipson H. 2009. Distilling free-form natural laws from experimental data. science. 324 (5923):81–85.
Sethanan K, Pitakaso R. 2016. Differential evolution algorithms for scheduling raw milk transportation. Computers and Electronics in Agriculture. 121:245–259.
Shao L, Liu L, Li X. 2014. Feature learning for image classification via multiobjective genetic programming. IEEE Transactions on Neural Networks and Learning Systems. 25(7):1359–1371.
Sheng W, Chen S, Sheng M, Xiao G, Mao J, Zheng Y. 2016. Adaptive multisubpopulation competition and multiniche crowding-based memetic algorithm for automatic data clustering. IEEE Trans Evolutionary Computation. 20(6):838–858.
Sheng W, Liu X, Fairhurst MC. 2008. A niching memetic algorithm for simultaneous clustering and feature selection. IEEE Transactions on Knowledge Data Engineering. 20(7):868–879.
Silva JCD, Lima FPDA, Lotufo ADP, Batista JMMDCP. 2015. Identity recognition using an artificial intelligence based on artificial immune system. In: Proceedings of the 2015 Asia-Pacific Conference on Computer Aided System Engineering. IEEE. p. 158–162.
Sinha A, Malo P, Kuosmanen T. 2015. A multiobjective exploratory procedure for regression model selection. Journal of Computational and Graphical Statistics. 24(1):154–182.
Smith M, Bull L. 2005. Genetic programming with a genetic algorithm for feature construction and selection. Genetic Programming and Evolvable Machines. 6:265–281.
Srinivasan S, Kamalakannan T. 2018. Multi criteria decision making in financial risk management with a multi-objective genetic algorithm. Computational Economics. 52(2):443–457.
Sun Y, Xue B, Zhang M. 2017. Evolving deep convolutional neural networks for image classification. arXiv preprint arXiv:171010741.
Sun Y, Yen GG, Yi Z. 2018. Evolving unsupervised deep neural networks for learning meaningful representations. IEEE Transactions on Evolutionary Computation. doi:10.1109/TEVC.2018.2808689.
Tabrizi S, Ghodsypour SH, Ahmadi A. 2018. Modelling three-echelon warm-water fish supply chain: a bi-level optimization approach under nash–cournot equilibrium. Applied Soft Computing. 71:1035–1053.
Tao W, Jin H, Liu L. 2007. Object segmentation using ant colony optimization algorithm and fuzzy entropy. Pattern Recognition Letters. 28(7):788–796.
Tran B, Xue B, Zhang M. 2017. Class dependent multiple feature construction using genetic programming for high-dimensional data. In: Proceedings of the Australasian Joint Conference on Artificial Intelligence; vol. 10400. Springer International Publishing. p. 182–194.
Tran B, Xue B, Zhang M. 2018a. Variable-length particle swarm optimisation for feature selection on high-dimensional classification. IEEE Transactions on Evolutionary Computation. doi:10.1109/TEVC.2018.2869405.
Tran B, Xue B, Zhang M, Nguyen S. 2016a. Investigation on particle swarm optimisation for feature selection on high-dimensional data: local search and selection bias. Connection Science. 28 (3):270–294.
Tran B, Zhang M, Xue B. 2016b. Multiple feature construction in classification on high-dimensional data using GP. In: Proceedings of the IEEE Symposium Series on Computational Intelligence. doi:10.1109/SSCI.2016.7850130.
Tran CT, Zhang M, Andreae P, Xue B, Bui LT. 2018b. An effective and efficient approach to classification with incomplete data. Knowledge-Based Systems. 154:1–16.
Valarmathy S, Vanitha NS. 2017. Feature selection optimization using artificial immune system algorithm for identifying dementia in MRI images. Journal of Medical Imaging and Health Informatics. 7(1):73–78.
Vivekanandan P, Rajalakshmi M, Nedunchezhian R. 2013. An intelligent genetic algorithm for mining classification rules in large datasets. Computing & Informatics. 32(1):1–22.
Vladislavleva EJ, Smits GF, Den Hertog D. 2009. Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming. IEEE Transactions on Evolutionary Computation. 13(2):333–349.

Vojodi H, Fakhari A, Moghadam AME. 2013. A new evaluation measure for color image segmentation based on genetic programming approach. Image and Vision Computing. 31(11):877–886.
Vyas R, Bapat S, Goel P, Karthikeyan M, Tambe SS, Kulkarni BD. 2018. Application of genetic programming GP Formalism for building Disease Predictive models from protein-protein interactions PPI data. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 15 (1):27–37.
Wagner N, Michalewicz Z, Khouja M, McGregor RR. 2007. Time series forecasting for dynamic environments: The dyfor genetic program model. IEEE Transactions on Evolutionary Computation. 11(4):433–452.
Wang H, Peng D, Wang W, Sharif H, Wegiel J, Nguyen D, Bowne R, Backhaus C. 2008. Artificial immune system based image pattern recognition in energy efficient wireless multimedia sensor networks. In: Proceedings of the IEEE Military Communications Conference. IEEE. doi:10.1109/MILCOM.2008.4753651.
Weise T, Devert A, Tang K. 2012. A developmental solution to (dynamic) capacitated arc routing problems using genetic programming. In: Proceedings of Genetic and Evolutionary Computation Conference. ACM. p. 831–838.
Xue B, Cervante L, Shang L, Browne WN, Zhang M. 2012. A multi-objective particle swarm optimisation for filter-based feature selection in classification problems. Connection Science. 24(2–3):91–116.
Xue B, Zhang M, Browne WN. 2014. Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Applied Soft Computing. 18:261–276.
Xue B, Zhang M, Browne WN, Yao X. 2016. A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation. 20(4):606–626.
Yannakakis GN, Togelius J. 2018. Artificial intelligence and games. Vol. 2. New York: Springer.
Yska D, Mei Y, Zhang M. 2018. Genetic programming hyper-heuristic with cooperative coevolution for dynamic flexible job shop scheduling. In: Proceedings of the European Conference on Genetic Programming. Springer. p. 306–321.
Yu G, Chai T, Luo X. 2011. Multiobjective production planning optimization using hybrid evolutionary algorithms for mineral processing. IEEE Transactions on Evolutionary Computation. 15(4):487–514.
Zhang J, Wang S, Chen L, Gallinari P. 2016. Multiple Bayesian discriminant functions for high-dimensional massive data classification. Data Mining and Knowledge Discovery. 6:1–37.
Zhang J, Zhan Z, Lin Y, Chen N, Gong Y, Zhong J, Chung HS, Li Y, Shi Y. 2011. Evolutionary computation meets machine learning: a survey. IEEE Computational Intelligence Magazine. 6(4):68–75.
Zhao H. 2007. A multi-objective genetic programming approach to developing pareto optimal decision trees. Decision Support Systems. 43(3):809–826.
Zhao Y, Feng X, Shi L, Zhang J, Wang L, Xu Y. 2017. The study on reconfigurable algorithm of the wood flexible manufacturing system based on ootcpn-gasa. In: Proceedings of the 2nd International Conference on Control and Robotics Engineering. IEEE. p. 68–72.
Zheng YJ, Ling HF, Xue JY, Chen SY. 2014. Population classification in fire evacuation: a multiobjective particle swarm optimization approach. IEEE Transactions on Evolutionary Computation. 18(1):70–81.
Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q. 2011. Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm and Evolutionary Computation. 1(1):32–49.
Zhou X, Wang J. 2005. A genetic method of LAD estimation for models with censored data. Computational statistics & Data Analysis. 48(3):451–466.
Zhou ZH. 2012. Ensemble methods: foundations and algorithms. London: Chapman and Hall/CRC.
Zhu Z, Ong YS, Zurada JM. 2010. Identification of full and partial class relevant genes. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 7(2):263–277.