A Survey of Genetic Algorithms for Multi-Label Classification

Eduardo Corrêa Gonçalves
Escola Nacional de Ciências Estatísticas
Inst. Brasileiro de Geografia e Estatística
Rio de Janeiro, Brazil
eduardo.correa@ibge.gov.br

Alex A. Freitas
School of Computing
University of Kent
Canterbury, Kent, United Kingdom
a.a.freitas@kent.ac.uk

Alexandre Plastino
Instituto de Computação
Universidade Federal Fluminense
Niterói, Brazil
plastino@ic.uff.br

***Abstract*—In recent years, multi-label classification (MLC) has become an emerging research topic in big data analytics and machine learning. In this problem, each object of a dataset may belong to multiple class labels and the goal is to learn a classification model that can infer the correct labels of new, previously unseen, objects. This paper presents a survey of genetic algorithms (GAs) designed for MLC tasks. The study is organized in three parts. First, we propose a new taxonomy focused on GAs for MLC. In the second part, we provide an up-to-date overview of the work in this area, categorizing the approaches identified in the literature with respect to the taxonomy. In the third and last part, we discuss some new ideas for combining GAs with MLC.**

Keywords—multi-label classification, genetic algorithms, machine learning

I. Introduction

Classification is one of the most active topics of research in the fields of big data analytics and machine learning. It consists in the task of automatically assigning objects to discrete classes (known as class labels or simply labels) based on the features of the objects. In traditional classification problems, each object must be associated to one and only one label within a predetermined set of class labels. These are called single-label classification (SLC) problems [1]–[4]. A well-known example of SLC problem is virus/malware detection [5], where the goal is to determine whether an application has either a “benign” or “malicious” behavior.

However, not all classification problems are single-label. As an example, consider music categorization [6], which consists in associating songs to music genres. For instance, several songs written by Stevie Wonder can be classified as belonging to “Soul”, “Pop”, and “Funk” genres at the same time. In the same way, a number of compositions by the Brazilian composer Tom Jobim are a mixture of both music genres: “Jazz” and “Bossa Nova”. Therefore, music categorization represents a multi-label classification (MLC) problem [7]–[10], since each object can be assigned different labels simultaneously. Over the last few years, a number of other important and modern applications of MLC classification have emerged, such as functional genomics [11]–[13] (determining the multiple biological functions of genes and proteins), drug side-effect prediction [14] (predicting the set of adverse reaction events of new drugs), text categorization [15]–[17] (associating documents to various subjects), software failures classification [18] (associating software failures into one or more fault types), semantic scene classification [19] (categorizing images into semantic classes), just to name a few.

Looking from a database theory angle, we might consider there is a unique difference between SLC and MLC problems: the former corresponds to predicting the state of a single-valued class attribute, whereas the latter the state of a multi-valued class attribute. Although the difference is subtle in theory, in practice MLC problems tend to be much more challenging. This is due to the following reasons:

1. The output space of MLC increases exponentially with respect to the number of labels. More precisely: in a problem involving q distinct labels, the size of the output space in MLC is 2^q (total number of label combinations) whereas it is just q in SLC.
2. MLC applications typically refer to modern big data analytic tasks, where the data under analysis are semi-structured or unstructured: multimedia data, biological sequences, etc. Thus, real-world MLC datasets tend to be huge in terms of both the number of features and instances. It contrasts with traditional SLC applications, which often involve the analysis of ordinary (structured) relational data
3. In a number of MLC problems, labels have correlations with each other. For example, considering the music categorization problem, we may intuitively realize that a song is unlikely to be simultaneously labeled as “Heavy Metal” and “Jazz” because these two music genres have a strong negative correlation. Analogously, the likelihood of a song being labeled as “Pop” becomes stronger if it has been labeled as “Hip Hop” or “R&B”. Thus, the exploitation of label correlations is regarded as an essential step to ensure the effectiveness of several MLC processes [20]–[25].
4. In SLC applications, the classification of a new object can be either correct or wrong. For instance, if an SLC system classifies a “malicious” application as “benign” this clearly corresponds to a wrong output. Nonetheless, in MLC classification results can be partially correct [9], i.e., the classifier may predict some of the correct labels, but it can either miss some of them or include wrong predictions. For example, if

the true labels of a song are “Pop” and “Dance” and it has been labeled as “Pop” and “R&B” by a music categorization system, this corresponds to a partially correct output. Due to this, the performance evaluation of MLC systems employs different metrics than the ones traditionally used in SLC [7]–[10]. Moreover, under certain scenarios, evaluating the quality of an MLC model requires multi-objective evaluation [26].

A considerable body of recent work [11], [14], [18], [27]–[36] has proposed strategies based on genetic algorithms (GAs) [37], [38] to overcome one or more of the above challenges. This paper presents a survey of the literature on GAs for MLC, providing researchers and practitioners with: (i) a taxonomy that highlights the important aspects in the context of evolutionary MLC; (ii) an up-to-date overview of the work in this area, categorizing the GA methods identified in the literature with respect to the taxonomy.

The rest of this work is organized as follows. Section 2 gives an overview of MLC concepts relevant to this paper. Section 3 introduces a new taxonomy of GAs for MLC, where methods are mainly divided into three broad areas: GAs to perform data preprocessing, GAs to perform parameter optimization, and GAs to build classification models. Next, in Section 4, we present a survey of the methods based on the taxonomy. We highlight the contributions of each different method and the key characteristics of the proposed GAs. Finally, we give concluding remarks and suggest new ideas for integrating GAs with MLC in Section 5.

II. Preliminaries: Multi-label Classification

The multi-label classification task can be formally defined as follows. Let $X = \{X_1, ..., X_d\}$ be a set of d predictive (or input) attributes (or features) and $L = \{l_1, ...,l_q\}$ be a set of q possible class labels, where $q \geq 2$. Consider a training dataset D composed of N instances of the form $\{(x_1, Y_1), (x_2, Y_2), ..., (x_N, Y_N)\}$. Each x_i corresponds to a vector $(x_1, ..., x_d)$ that stores values for the d predictive attributes in X and each $Y_i \subseteq L$ corresponds to a subset of labels. The goal of the multi-label classification task is to learn from D a classifier h that, given an unlabeled instance $t = (x, ?)$, is capable of effectively predicting the set of labels (a.k.a. labelset) Y, i.e., $h(t) \rightarrow Y$.

A. Approaches for MLC

According to the literature [7]–[10], existing methods for MLC can be primarily categorized into two fundamental families: problem transformation and algorithm adaptation.

Algorithm adaptation methods extend or adapt an existing SLC algorithm for the task of MLC. E.g., in [39], the authors introduce the Multi-label kNN (ML-kNN) method, which is derived from two SLC algorithms: k-NN and Naïve Bayes [2], [3]. In this approach, the classification process of a new instance t works in two steps. First, the k closest instances to t are identified (i.e., the k instances more similar to t in the training set). Then, for each label l_i present in the set of labels of these k neighbors, Bayes’s rule is employed to estimate if t should be labeled with l_i.

Another example of classic single-label technique adapted for MLC is presented in [40]. This work proposes a method named ML-RBF, which is based on radial basis function neural networks [3]. In an ML-RBF structure, the hidden layer is composed of L sets of prototype vectors (one for each label), where each prototype vector corresponds to a specific point in the input space. In the training process, the set of prototypes for each label l_i is determined by performing k-Means clustering on instances associated to l_i (cluster centers are used as the prototypes). Each output neuron corresponds to a possible class label, whose weights are obtained by minimizing a sum-of-squares error function.

Problem transformation (a.k.a. algorithm independent) methods work by transforming the original multi-label problem into one or more single-label problems. Then, any existing SLC algorithm can be directly applied by simply mapping back its single label predictions into multi-label predictions. The Binary Relevance (BR) method [41] is the most well-known and widely adopted problem transformation method for MLC [10]. In this approach, the original multi-label dataset is decomposed into q binary single-label datasets, one for each label. The induction of a BR model consists in training one single-label classifier for each derived dataset. Once the BR model has been induced, the classification process is quite straightforward: new instances are predicted by simply combining the outputs produced by each binary classifier.

The Classifier Chain (CC) approach [21] is another example of problem transformation method. In the CC approach, q single-label classifiers are inserted in random order into a chain $\{y_1 \rightarrow y_2 \rightarrow ... \rightarrow y_q\}$, where each classifier is responsible for predicting a specific label in L. The chain structure allows each single-label classifier y_j to incorporate the labels inferred by the previous y_1, ..., y_{j-1} classifiers as additional predictive information. Thus, differently from the BR approach, possible correlations among labels can be automatically exploited.

B. Metrics for Evaluating MLC Performance

In SLC problems, a classification result can be either correct or wrong. As a consequence, it is often natural to evaluate the effectiveness of an SLC model by taking into account a single quality measure: the percentage of test instances misclassified by the classification model, known as the error rate [2]–[4]. However, in MLC a new object can be classified as partially correct. Thus, MLC problems require different metrics than the traditional error rate. In [42], authors provide a deep analysis of 16 metrics for the multi-label scenario and help users to better understand them and to choose the most appropriate ones. Two examples of such metrics are Accuracy (ACC) and Hamming Loss (HL), respectively defined in (1) and (2). In both equations, n represents the number of test instances, q is the number of labels, Y_i is the true labelset of the i^{th} test instance, and Z_i is the predicted labelset of the i^{th} test instance. In (2), the expression $|Y_i \Delta Z_i|$ represents the symmetric difference between Y_i and Z_i.

$$ACC = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{1}$$

$$HL = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \Delta Z_i|}{q} \quad (2)$$

The ACC metric provides the user with information about the proportion of correct predictions whereas HL informs the average number of incorrect binary predictions per test instance. For ACC, greater values indicate better performance; whilst for HL, smaller values indicate better performance. As shown in [34], both ACC and HL are important since they provide complimentary information about MLC processes. Actually, the MLC literature [7]–[10] tends to consider that the use of different metrics provides alternative analyses of predictive performance, giving a better understanding about the quality of a classification model. Thus, real-world MLC problems are frequently treated as multi-objective problems, as they involve multiple metrics (objectives) that should be simultaneously optimized [26]. A simple example would be a problem where the user wants the best trade-off between two (sometimes conflicting) objectives: maximize the ACC and minimize the HL of a MLC model.

III. A Taxonomy of Genetic Algorithms for Multi-label Classification

In this section, we first summarize the main motivations behind methods that use GAs to solve MLC problems. Next, we propose a new taxonomy focused on GAs for MLC.

A. *Why to Use GAs?*

In recent years, GAs have gained considerable attention from the MLC community [11], [14], [18], [27]–[36]. The main motivations are closely related to the four major challenges of MLC introduced in Section I. These are described below:

- GAs are a global search method capable of effectively exploring the extremely large search space of 2^q possible solutions associated to the multi-label classification problem.

- As a global method, GAs tend to cope better with attribute interactions than greedy methods [37], [38], [43]. Hence, intuitively, GAs are expected to discover correlations among both labels and predictive attributes that could be missed by greedy approaches.

- As discussed in the previous section, the evaluation of multi-label classifiers often involves the use of several distinct measures. GAs naturally allow the evaluation of a candidate solution by simultaneously considering different quality criteria in the fitness function [26].

- It is also important to observe that over the last decades, GAs have been successfully used to solve a large number of SLC problems in very distinct contexts and application domains. For instance, GAs have been widely employed to perform feature selection [2], [43]–[46], to determine the best set of weights for training neural networks [47] and to discover classification rules [38], [43], [48]. Therefore, it was expected that GAs would also perform well in the MLC context.

B. *Taxonomy of GAs for MLC*

The previous subsection listed the reasons that have led to the development of GA-based solutions for the MLC problem. Nonetheless, how are GAs actually combined with MLC? In Fig. 1, we present a new taxonomy of GAs for MLC, which organizes methods into three different broad categories: one in which GAs are used to perform data preprocessing (divided into wrapper and filter methods), one that employs GAs to perform parameter optimization, and one that uses GAs to build the classification model directly (divided into methods that build part of the model structure and methods that generate the complete MLC model). Although the taxonomy was mainly developed based on the methods found in this survey, it also covers potential applications that can be developed in the future (this will be further discussed in Section V).

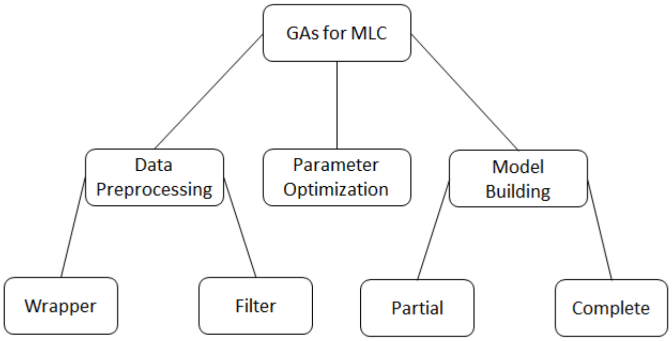

Fig. 1. Taxonomy of GAs for Multi-label Classification

C. *Single-Objective versus Multi-Objective Optimization*

In closing this section, it is important to make a remark concerning the issue of single-objective versus multi-objective optimization. In our survey, we identified that in every category of the taxonomy, there exist methods that cope with each of these two kinds of optimization problem. Thus, we decided not to include the "number of objectives" as a level of the taxonomy. Methods that cope with multi-objective problems in MLC can adopt three distinct approaches, which are briefly described below (for a detailed review on the subject, the reader is referred to [26]).

The first and simplest approach is to transform the multi-objective problem into a single-objective problem by using a weighted formula. An example would be to use the formula ((1-HL) + ACC) / 2 to compute individual fitness values, assuming equal weights for the HL and ACC metrics. The second is the lexicographic approach, where two or more objectives with distinct predetermined priorities (e.g.: model accuracy and model complexity) are taken into consideration to define the quality of each individual. Consider the following example. Let c_i and c_j be two individuals (candidate solutions). In the lexicographic approach, when comparing two individuals, the GA first tries to determine which one is better considering the highest priority objective. If c_i is not better than

c_j, and vice-versa, then both are compared considering the second objective. The third and last approach to address multi-objective optimization is to use the Pareto dominance concept in the fitness evaluation. In this approach, a solution c_i is said to dominate another solution c_j if and only if: (i) c_i is not worse than c_j in any of the objectives; and (ii) c_i is strictly better than c_j in at least one of the objectives.

In the next section, we provide an up-to-date overview of the work on GAs for MLC, categorizing the approaches identified in the literature with respect to the taxonomy presented in Fig 1. This review focuses on the multi-label aspects of the GAs, involving mainly the individual representation, fitness function and selection methods (particularly when a multi-objective approach is used). Genetic operators like crossover and mutation in general are not discussed because they are usually used in the same way as in single-label classification, with no need to adapt them to multi-label classification.

IV. GA-based Methods for Multi-label Classification

A. *GAs for Data Preprocessing*

Several empirical studies have demonstrated that the application of data preprocessing techniques usually lead to the improvement on the effectiveness of both single-label and multi-label classification models [49]–[53]. There are several distinct data preprocessing techniques [2], such as data cleaning, discretization, feature selection and feature construction. In spite of this, current work on GAs for MLC has just addressed one of them: feature selection.

The goal of feature selection is to select the subset of input features most relevant for the classification task, i.e., to identify the subset of features that leads to the best predictive performance [2], [3], [43]–[45]. Since an exhaustive evaluation is computationally infeasible (as there are 2^d possible feature subsets to be taken into consideration, where d is the number of input features), GAs have been commonly used to find an optimized subset of features.

As a data preprocessing task, GA-based feature selection methods can be performed using either the wrapper approach or filter approach [43]–[45]. The distinction between these two approaches refers to the way the fitness function is computed. In the wrapper approach, the quality of an individual is determined by using the target MLC method. On the contrary, in the filter approach, the evaluation function does not use the MLC method, but some metric instead. The advantage of filter methods lies in that they are less computationally expensive, since they do not require several runs of the MLC method. Nonetheless, wrapper approaches tend to be more effective, as they search for an attribute subset that is customized for a given MLC method [44].

In the remainder of this subsection we review the published studies on GA-based feature selection for MLC [14], [27]–[30]. All these methods share one common characteristic: individuals in the population are represented by d-dimensional binary vectors [43]. In this representation, d is the number of original attributes and the i-th bit, $i=1, \dots, d$, can take either the value 1 or 0, respectively indicating whether the i-th attribute is selected or not selected.

The first work that used GAs to perform feature selection in the MLC context was published in [27]. The method works in two steps. First, Principle Component Analysis (PCA) [2] is employed to create an alternative and smaller set of attributes. Next, a wrapper-based genetic algorithm (GA) is applied to select a relevant feature subset for an MLC classifier trained using the BR method with Naïve Bayes as the base classifier. The fitness function is multi-objective, based on a weighted-formula that takes the average of two MLC metrics: Hamming Loss and Ranking Loss [42].

The FS-MLkNN method [14] also takes two steps to perform feature selection. In the first step, a preliminary set of selected attributes is defined according to the value of the mutual information between predictive attributes and labels. In the second, a GA is employed to determine the final set of relevant attributes. FS-MLkNN follows the wrapper approach, using ML-kNN (see Section II-A) to evaluate the quality of a candidate set. A single metric is used as fitness score of chromosomes: the AUPRC (Area Under the Precision-Recall Curve) metric [4].

In [28], a wrapper-based memetic feature selection technique for MLC is proposed, which incorporates a local refinement method into the base GA used for feature selection. At each generation, the individual with the best fitness value is selected to undergo refinement. In this process an attribute that is absent from the selected individual can be added, if it exhibits a high dependency with the set of labels. Conversely, an attribute that is present in the selected individuals can be removed if it is not highly correlated with the labels.

Filter-based GAs were used for feature selection in [29] and [30]. The Multi-Label Correlation-Based Feature Selection method (GA-ML-CFS), introduced in [29], searches for a relevant subset of attributes by employing a single-objective fitness function based on the Merit function of the well-known Correlation-based Feature Selection (CFS) method [54]. In essence, this fitness function assigns higher values for candidate solutions where the selected attributes have little correlation with each other, but at the same time, are highly correlated with the set of labels involved in the MLC problem.

The LexGA-ML-CFS, proposed in [30], corresponds to an extension of the GA-ML-CFS which uses the lexicographic multi-objective approach in the tournament selection procedure, with tournament size of 2. To determine the fitness of candidate solutions, two objectives are taken into consideration: the value of the Merit metric (first priority) and the number of selected attributes (second priority, the fewer the better). Hence, when comparing two individuals in the lexicographic tournament selection, if one of them has a better Merit, it wins the tournament, otherwise the number of selected attributes is used as a tie-breaking criterion

To end this section, we present a final remark about the comprehensibility (or interpretability) [55] of the results produced by each of the discussed methods. With regard to this subject, the only method that should not be applied when the goal is to build interpretable classifiers is the one proposed in

[27], since PCA is used in the first step to transform the original set of attributes – i.e., the new attributes constructed by PCA are not directly interpretable. On the other hand, the methods proposed in [14], [28]–[30] produce interpretable results (assuming the original features are directly interpretable), i.e., at the end of the GA execution, these algorithms will return a single optimized individual, representing an optimized subset of the original features. It is important to consider that in some important application scenarios of MLC, such as drug-side effects prediction, bioinformatics and medical diagnosis, the ability to interpret the classification result might be almost as important as the predictive accuracy itself.

B. *GAs for Parameter Optimization*

Given a classification algorithm and a training dataset, the objective of a GA for parameter optimization is to search for a set of parameters optimized for that classification algorithm and that dataset.

The earliest work on parameter optimization for MLC was published in [31]. This work introduced the EnML method, a multi-objective GA based on the Pareto approach, which is focused on simultaneously maximizing both the accuracy and diversity of an ensemble of ML-RBF neural networks (see Section II-A). An ensemble can be defined as a composite classification model made up of a combination of classifiers (base learners) [2]. In EnML, each individual encodes a set of prototypes. The method introduced two criteria to perform multi-objective optimization: ML-HSIC and ML-NCL. The former is used to evaluate the accuracy of a base learner by employing a dependence evaluation technique proposed in [56]. The later accounts for measuring diversity and works by evaluating the negative correlation of each base learner with the error of the rest of the ensemble. The same multi-objective mechanism was also employed for evolving the prototypes of single ML-RBF models (rather than ensembles) in [32].

The MLL-GA [18] is a GA designed to evolve the weights of a fixed composite classification model formed by twelve distinct base learners. In this composite model, each base learner is trained using a distinct multi-label method (e.g.: the first is an ML-kNN model, the second is a BR model using k-NN as base algorithm, etc.). Each individual is represented by a real-valued vector, which stores the weights associated to each model (the higher the weight the more the base learner will contribute to predict the labelset of a new object). The F-Measure [42] is adopted as fitness function.

A genetic algorithm for automatically and simultaneously selecting an MLC algorithm and configuring its parameters is introduced in [33]. In the proposed GA, each individual is represented by a real-valued vector that encodes a component in a given search space. A search space corresponds to the set of all algorithms (and its parameters) available in a software platform (or tool) for MLC, such as MULAN [57] and MEKA [58]. In turn, a component corresponds to an MLC algorithm with a particular parameter configuration that is available in the search space. The GA adopts a multi-objective fitness function based on a weighted-formula that takes the average of four MLC metrics (Hamming Loss, Ranking Loss, Exact Match, and Macro-F1).

C. *GAs for Classification Model Building*

This family of methods employs GAs to actually build the classification model. The methods proposed in [34]–[36] can build part of the classification model, whilst the one proposed in [11] is able to build the complete MLC model.

The GACC method proposed in [34] performs a genetic search to find the best chain sequence (label ordering) for a Classifier Chain model [21] (described in Section II-A). This is considered a crucial step in the process of training a Classifier Chain model, as it has been empirically demonstrated that the use of distinct label orderings can lead to large differences in the predictive accuracy of the model [59]. In the GACC technique, individuals of the population are represented by *q*-dimensional integer vectors regarding different specific label orderings, where *q* represents the number of labels. For instance, the sequence $\{y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_4\}$ is encoded as the vector [1, 2, 3, 4]. To assess the predictive accuracy, GACC adopts a multi-objective approach, using the Quality (fitness) function, a weighted formula that simultaneously takes into account three multi-label metrics (each assigned the same weight): Exact Match, Accuracy and Hamming Loss [42].

The GA-PartCC method [35] is an extension to the GACC method that is capable of evaluating chain sequences that vary not only in their label ordering but also in their length. More specifically, GA-PartCC performs a global search for an optimized chain (i.e., a label sequence that leads to the best possible predictive accuracy of the CC model), by exploring partial chains with only a subset of labels. In order to accomplish this task, GA-PartCC uses a variable-length integer vector representation and a multi-objective lexicographic fitness function which takes into account two objectives: the model's accuracy (first priority, using the Quality function) and the model's size (second priority, corresponding to the number of labels represented in the individual). The Merit function of the CFS method [54] was also evaluated for training GA-PartCC models in the experiment reported in [36].

The HMC-GA method [11] is the only GA-based approach proposed in the literature that is capable of building a complete MLC model. HMC-GA is a method for discovering classification rules of the form IF (conditions) THEN (labelset) [43]. The focus of the work is on protein function prediction and the generation of comprehensible classification models (i.e., models that can be interpreted by biologists). The method works by evolving the antecedent of classification rules with the goal of optimizing the level of coverage of each antecedent. In HMC-GA, each individual is a vector that mixes integer and real values and contains a sequence of *d* tests in the form $[FLAG|OP|\Delta_1|\Delta_2]$, where *d* corresponds to the number of predictive attributes. For the *i*-th test encoded in the individual, the gene FLAG can be either 1 or 0, indicating whether or not the *i*-th attribute is selected, respectively. The gene OP is an integer that specifies a relational operator ($=$, $\neq$, $>$, $<$, $\geq$, or $\leq$). Finally, the genes Δ_1 and Δ_2 are real values to be used as thresholds within the tests. From a rule induction perspective, HMC-GA follows a sequential covering approach [44], in which instances covered by a rule are removed from the training set, so the new rules generated can fit the remaining uncovered instances. HMC-GA runs a full evolutionary cycle and then saves all rules from the last generation that have the number of covered instances superior to a user-defined threshold. All instances covered by these rules are then removed from the training set and a new evolutionary cycle is performed. The process is repeated until there are no instances in the training set or the number of instances is inferior to a user-specified threshold. To generate the consequent of the rules (i.e., the predicted labelsets), the method computes the probability that instances covered by the rule belong to each of the labels. It is worth mentioning that although HMC-GA has been originally developed for solving hierarchical multi-label classification problems [60] (where the class labels are organized into a generalization-specialization hierarchy), the method can be directly adapted for addressing standard MLC problems.

V. Discussion and Conclusions

A. Summary

There has been significant interest in the use of GAs to perform multi-label classification and there are different ways of introducing GAs into MLC processes. Hence, this article provided a survey on the subject. A new taxonomy of GAs focused on MLC has been proposed and an up-to-date overview of this area has been carried out, categorizing the approaches identified in the literature with respect to the taxonomy.

Tables I and II provide a summary of the characteristics of the methods discussed in this paper. Table I categorizes the algorithms identified in the literature with respect to the taxonomy shown in Fig. 1. A summary of the methods covered in this paper according to their application domain is presented in Table II. It is possible to observe that some of the GA-based MLC methods were proposed for a specific application domain while others were evaluated against datasets containing real-world data from distinct areas.

B. Future Trends

Below, we suggest three topics for future research that seem to deserve special attention.

First, all the GAs performing data preprocessing for MLC proposed in the literature address the feature selection task. However, there are other important data preprocessing tasks that have still not gained attention in the area of GA-based MLC, such as supervised discretization [2], [3], [49] and feature construction [38], [43], [61]. Discretization corresponds to the process of transforming the domain of a numeric predictive attribute into a finite (and usually small) set of adjacent intervals. In some cases such discretization can help to improve the interpretability of the classification model or to improve the efficiency of the classification algorithm. In supervised discretization, this process is performed taking into consideration class information. Since in MLC several class labels must be simultaneously taken into account, supervised discretization can be considering a challenging task. Feature construction (or feature extraction) goes beyond feature selection in the sense that the former has the potential to construct new features with more predictive power than the original features. However, feature construction is a more challenging task than feature selection. In addition, the complexity of supervised feature construction is aggravated in MLC by the need to take multiple class labels into account. Note that feature construction involves applying operators to the original features, and so it is usually performed by Genetic Programming methods [38], [61], rather than GAs.

Second, all the GAs for feature selection proposed in the literature use the traditional binary vector representation. As discussed in [45], for datasets composed by thousands of attributes this can lead to a high computational cost. Therefore, compact representations originally proposed for SLC problems (such as the ones presented in [62], [63]) also need to be evaluated in the context of MLC.

Third, it is noticeable that the proposal of GA-based methods capable of building a complete MLC model is still an open issue, as our survey identified only one method that falls into this category [11].

TABLE I. Summary of GAs For Multi-Label Classification

	Single-objective	**Multi-Objective**
Data Preprocessing – Wrapper	[14], [28]	[27]
Data Preprocessing – Filter	[29]	[30]
Parameter Optimization	[18]	[31]–[33]
Model Building	[11], [36]	[34], [35]

TABLE II. Summary of the Existing Literature on GAs for Multi-label Classification According to the Application Domain

Application Domain	**References**
Audio Classification	[33], [35], [36]
Direct Marketing	[35]
Drug-side Effect Prediction	[14]
Functional Genomics	[11], [27], [28], [31], [32], [34]–[36]
Image Annotation	[34]
Medical Diagnosis	[28], [29], [34]–[36]
Music Categorization	[34]–[36]
Scene Classification	[27], [28], [31]–[34]
Software Failures Classification	[18]
Text Categorization	[28]–[32], [34]–[36]

It should be emphasized that there are a few proposals in the literature that have addressed other kinds of evolutionary algorithms (EAs) for solving MLC problems, such as grammatical evolution [12], particle swarm optimization [64], gene expression programming [65], [66], and learning classifier systems [67]. Nevertheless, due to space constraints, this paper focused on genetic algorithms. Hence, an exam of other kinds of EAs for MLC is left as future work.

References

[1] T. G. Dietterich. "Machine learning," in Encyclopedia of Cognitive Science, vol. II, L. Nadel, Ed. London: Nature Publishing Group, 2003, pp. 971–981.

[2] J. Han, M. Kamber, and J. Pei. Data Mining: Concepts and Techniques. 3rd ed. San Francisco, CA: Morgan Kaufmann, 2011.

[3] I. Witten, E. Frank, M. Hall, and C. Pal. Data Mining: Practical Machine Learning Tools and Techniques. 4th ed. San Francisco, CA: Morgan Kaufmann, 2016.

[4] N. Japkowicz and M. Shah. Evaluating Learning Algorithms: A Classification Perspective. New York, NY: Cambridge University Press, 2011.

[5] A. Martín, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "Evolving deep neural networks architectures for android malware classification," in *Proc. of the 2017 IEEE Congress on Evolutionary Computation (CEC)*. San Sebastian: Spain, IEEE, 2017, pp. 1659–1666.

[6] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in *Proc. of the 18th ISMIR Conference*. Suzhou: China, ISMIR, 2017, pp. 23–27.

[7] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," ACM Computing Surveys (CSUR), vol. 47, no 3, pp. 52:1–52:38, April 2015.

[8] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Boston: Springer, 2010, pp. 667–685.

[9] A. C. P. L. F. de Carvalho and A. A. Freitas, "A tutorial on multi-label classification techniques," in Foundations of Computational Intelligence, vol. 5, Studies in Computational Intelligence, vol 205, A. Abraham, A.-E. Hassanien, and V. Snášel, Eds. Berlin: Springer, 2009, pp. 177–195.

[10] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no 8, pp. 1819–1837, August 2014.

[11] R. Cerri, R. C. Barros, and A. C. P. L. F. de Carvalho, "A genetic algorithm for hierarchical multi-label classification," in *Proc. of the 2012 ACM Symposium on Applied Computing (SAC)*. Riva: Italy, ACM, 2012, pp. 250–255.

[12] R. Cerri, R. C. Barros, A. C. P. L. F. de Carvalho, and A. A. Freitas, "A grammatical evolution algorithm for generation of hierarchical multi-label classification rules," in *Proc. of the 2013 IEEE Congress on Evolutionary Computation (CEC)*. Cancun: Mexico, IEEE, 2013, pp. 454–461.

[13] Y-H. Huang, C-M. Hung, and H.C Jiau, "A multi-label approach using binary relevance and decision trees applied to functional genomics," Journal of Biomedical Informatics, vol. 54, pp. 85–95, April 2015.

[14] W. Zhang, F. Liu, L. Luo, and J. Zhang, "Predicting drug side effect by multi-label learning and ensemble learning," BMC Bioinformatics, vol. 16, no. 365 pp. 1–11, November 2015.

[15] F. Sebastiani "Text categorization," in Encyclopedia of Database Technologies and Applications, L. C. Rivero, J. H. Doorn, and V. E. Ferraggine, Eds. IGI Global, 2005, pp.683–687.

[16] T. Gonçalves and P. Quaresma, "A preliminary approach to the multilabel classification problem of Portuguese juridical documents," in *Proc. of the 11st Portuguese Conference on Artificial Intelligence (EPIA)*. Beja: Portugal, Springer, 2003, pp. 435–444.

[17] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multi-label classification via metalabeler," in *Proc. of the 18th International Conference on World Wide Web*. Madrid: Spain, ACM, 2009, pp. 211–220.

[18] X. Xia, Y. Feng, D. Lo, Z. Chen, and X. Wang, "Towards more accurate multi-label software behavior learning," in *Proc. of the 2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)*. Antwerp: Belgium, IEEE, 2014, pp. 134–143.

[19] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "SUN database: exploring a large collection of scene categories," International Journal of Computer Vision, vol. 116, issue 1, pp. 3–22, August 2016.

[20] M-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proc. of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington D.C.: USA, ACM, 2010, pp.999–1008.

[21] J. Read, B. Pfahringer, G. Holmes, and F. Eibe, "Classifier chains for multi-label classification," Machine Learning, vol. 85, no. 3, pp. 333–359, December 2011.

[22] Y. Guo and S. Gu, "Multi-label classification using conditional dependency networks," in *Proc. of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. Barcelona: Spain, AAAI Press, 2011, pp.1300–1305.

[23] E. Montañes, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," Pattern Recognition, vol. 47, issue 3, pp. 1494–1508, March 2014.

[24] C. Ye, J. Wu, V. S. Sheng, P. Zhao, and Z. Cui, "Multi-label active learning with label correlation for image classification," in *Proc. of the 2015 IEEE International Conference on Image Processing (ICIP)*. Quebec City: Canada, IEEE, 2015, pp.3437–3441.

[25] J. Huang, G. Li, S. Wang, Z. Xue, and Q. Huang, "Multi-label classification by exploiting local positive and negative pairwise label correlation," Neurocomputing, vol. 257, pp. 164–274, September 2017.

[26] A. A. Freitas, "A critical review of multi-objective optimization in data mining: a position paper," SIGKDD Explorations Newsletters, vol. 6, no. 2, pp. 77–86, December 2004.

[27] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," Information Sciences, vol. 179, issue 19, pp. 3218–3229, September 2009.

[28] J. Lee and D.-W. Kim, "Memetic feature selection algorithm for multi-label classification," Information Sciences, vol. 293, issue 19, pp. 80–96, February 2015.

[29] S. Jungjit and A. A. Freitas,, "A new genetic algorithm for mullti-label correlation-based feature selection," in *Proc. of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. Burges: Belgium, ESANN, 2015, pp.285–290.

[30] S. Jungjit and A. A. Freitas, "A lexicographic multi-objective genetic algorithm for multi-label correlation-based feature selection," in *Proc. of the 2015 Conference on Genetic and Evolutionary Computation Conference (GECCO)*. Madrid: Spain, ACM, 2015, pp.989–996.

[31] C. Shi, X. Kong, P. S. Yu, and B. Wang, "Multi-label ensemble learning," in *Proc. of the 2011 ECML/PKDD Conference*. Barcelona: Spain, Springer, 2011, pp.223–239.

[32] C. Shi, X. Kong, D. Fu, P. S. Yu, and B. Wu, "Multi-objective multi-label classification," *Proc. of the 2012 SIAM Conference on Data Mining*. Anaheim: USA, SIAM, 2012, pp.355–366.

[33] A. G. C. de Sá, G. L. Pappa, and A. A. Freitas, "Towards a method for automatically selecting and configuring multi-label classification algorithms," in *Proc. of the 2017 Conference on Genetic and Evolutionary Computation Conference (GECCO)*. Berlin: Germany, ACM, 2017, pp. 1125–1132.

[34] E. C. Gonçalves, A. Plastino, and A. A. Freitas, "A genetic algorithm for optimizing the label ordering in multi-label classifier chains," in *Proc. of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*. Washington D.C.: USA, IEEE, 2013, pp.469–476.

[35] E. C. Gonçalves, A. Plastino, and A. A. Freitas, "Simpler is better: a novel genetic algorithm to induce compact multi-label chain classifiers," in *Proc. of the 2015 Conference on Genetic and Evolutionary Computation Conference (GECCO)*. Madrid: Spain, ACM, 2015, pp.559–566.

[36] H. Manli and W. Zhihai, "Genetic algorithm based on attribute correlation for multi-label classification," in *Proc. of the 2017*

International Conference on Machine Learning and Soft Computing (ICMLSC). Ho Chi Minh City: Vietnam, IEEE, 2017, pp.88–92.

[37] A. E. Eiben and J. E. Smith. Introduction to Evolutionary Computing. New York, NY: Springer, 2003.

[38] A. A. Freitas. Data Mining and Knowledge Discovery with Evolutionary Algorithms. New York, NY: Springer, 2002.

[39] M.-L. Zhang and Z.-H. Zhou, "ML-kNN: A lazy learning approach to multi-label learning," Pattern Recognition, vol. 40, no. 7, pp. 2038–2048, July 2007.

[40] M.-L. Zhang, "ML-RBF: RBF neural networks for multi-label learning," Neural Processing Letters, vol. 29, no. 2, pp. 61–74, April 2009.

[41] T. Joachims, "Text categorization with suport vector machines: learning with many relevant features," in *Proc. of the European Conference on Machine Learning (ECML)*. London: UK, Springer, 1998, pp.137–142.

[42] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H .C. Merschmann, "Correlation analysis of performance measures for multi-label classification," Information Processing & Management, vol. 54, issue 3, pp. 359–369, May 2018.

[43] A. A. Freitas, "A review of evolutionary algorithms for data mining," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Boston: Springer, 2010, pp. 371–400.

[44] G. Pappa, A. A. Freitas and C. A. A. Kaestner, "A multiobjective genetic algorithm for attribute selection," in *Proc. of the 4th Conf. on Recent Advances in Soft Computing (RASC)*. Nottingan Trent University, 2002, pp. 116–121.

[45] B. Xue, M. Zhang, W. N. Browne, and X. Yao "A survey on evolutionary computation approaches to feature selection," IEEE Transactions on Evolutionary Computation, vol. 20, no. 4, pp. 606–625, August 2016.

[46] E. Levy, O. David, and N. S. Netanyahu, "Painter classification using genetic algorithms," in *Proc. of the 2013 IEEE Congress on Evolutionary Computation (CEC)*. Cancun: Mexico, IEEE, 2013, pp. 3027–3034.

[47] M. L. A. Berry and G. Linoff. Data Mining Techniques: For Marketing, Sales, and Customer Support. Indianapolis, IN: John Wiley & Sons, 1997.

[48] M.V. Fidelis, H.S. Lopesm, and A.A. Freitas, "Discovering comprehensible classification rules with a genetic algorithm," in *Proc. of the 2000 IEEE Congress on Evolutionary Computation (CEC)*. La Jolla: USA, IEEE, 2010, pp. 805–810.

[49] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. of the 12nd International Conference on Machine Learning (ICML)*. Tahoe City: USA, Morgan Kaufmann, 1995, pp. 194–202.

[50] P. N. da Silva et al., "Automatic classification of carbonate rocks permeability from 1H-NMR relaxation data," Expert Systems with Applications, vol. 42, no. 9, pp. 4299–4309, June 2015.

[51] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," Artificial Intelligence Review, vol. 49, no. 1, pp. 57–78, January 2018.

[52] A. Cano, J. M. Luna, E. L. Gibaja, and S. Ventura, "LAIM discretization for multi-label data," Information Sciences, vol. 330, pp. 370–384, February 2016.

[53] H. Yin and K. Gai, "An empirical study on preprocessing high-dimensional class-imbalanced data for classification," in *Proc. of the 17th IEEE International Conference on High Performance Computing and Communications; The IEEE International Symposium on Big Data Security on Cloud*. New York: USA, IEEE, 2015, pp. 1314–1319.

[54] M.Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. of the 17th International Conference on Machine Learning (ICML)*. San Francisco: USA, Morgan Kaufmann, 1995, pp. 359–366.

[55] A. A. Freitas, "Comprehensible classification models: a position paper," SIGKDD Explorations Newsletter, vol. 15, no. 1, pp. 1–10, June 2013.

[56] A. Gretton, O. Bousquet, A. Smola, B. Scholkopf, "Measuring statistical dependence with Hibert-Schmidt norms," in *Proc. of the 2005 International Conference on Algorithmic Learning Theory (ALT).* Singapore: Singapore, Springer, 2005, pp.63–77.

[57] G. Tsoumakas, E. Spyromitros, J. Vilcek, and I. Vlahavas, "Mulan: a java library for multi-label learning," Journal of Machine Learning Research, vol. 12, pp. 2411–2114, July 2011.

[58] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes, "Meka: a multi-label/multi-target extension to weka," Journal of Machine Learning Research, vol. 17, no. 21, pp.1–5, February 2016.

[59] P. N. da Silva, E. C. Gonçalves, A. Plastino, and A. A. Freitas, "Distinct chains for different instances: an effective strategy for multi-label classifier chains," in *Proc. of the 2014 ECML/PKDD Conference*. Nancy: France, Springer, 2014, pp.453–468.

[60] C. N. Silla Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," Data Mining and Knowledge Discovery, vol. 22, no. 1-2, pp. 31–72, January 2011.

[61] F. E. B. Otero, M. M. S. Silva, A. A. Freitas, and J. C. Nievola, "Genetic programming for attribute construction in data mining," in *Proc. of the 6th European Conference on Genetic Programming (EuroGP)*. Essex: UK, Springer, 2003, pp.384–393.

[62] J.-H. Hong and S.-B. Cho, "Efficient huge-scale feature selection with speciated genetic algorithm," Pattern Recognition Letters, vol. 27, issue. 2, pp. 143–150, January 2006.

[63] Y.-S. Jeong, K. S. Shin, and M. K. Jeong, "An evolutionary algorithm with the partial sequential forward floating search mutation for largescale feature selection problems," Journal of the Operational Research Society, vol. 66, no. 4, pp. 529–538, April 2015.

[64] Y. Zhang, D.-W. Gong, X.-Y. Sun, and Y.-N. Guo, "A PSO-based multi-objective multilabel feature selection method in classification," Scientific Reports, vol. 7, article 376, March 2017.

[65] J. L. Avila, E. L. Gibaja, A. Zafra and S. Ventura, "A gene expression programming algorithm for multi-label classification," Journal of Multiple-Valued Logic and Soft Computing, vol. 17, no. 2-3, pp. 183–206, 2011.

[66] A. Cano, A. Zafra, E. L. Gibaja, and S. Ventura, "A Grammar-Guided Genetic Programming Algorithm for Multi-Label Classification," in *Proc. of the 16th European Conference on Genetic Programming (EuroGP'13)*. Vienna: Austria, Springer Berlin Heidelberg, 2013, pp.117–228.

[67] F. A. Tzima, M. Allamanis, A. Filotheou, and P. A Mitkas, "Inducing Generalized Multi-Label Rules with Learning Classifier Systems," arXiv preprint arXiv:1512.07982, 2015, December 2015.