The CSI Journal on
Computer Science and Engineering
Vol. 19, No. 1, 2025
Pages 61-69
doi.org/10.22034/jcse.2025.181903
Regular Paper

Feature Selection in Multi-label Classification based on Binary Quantum Gravitational Search Algorithm

Hojat Noormohammadi[1] Mohammad Bagher Dowlatshahi[2]

[1]Department of Engineering, Lorestan University, Khorramabad, Iran.
[2] Department of Engineering, Lorestan University, Khorramabad, Iran.

Abstract

Unlike a single-label supervisor dataset where each instance is assigned to one class label, in multi-label datasets, several class labels are assigned to each instance, which makes it difficult to build an accurate and comprehensive model from this dataset. In this study, a memetic algorithm for feature selection in a multi-label dataset is proposed. The principal innovation of this study is the offer of a novel local search algorithm which, in collaboration with binary quantum-inspired gravitational search algorithm (BQIGSA), forms the main framework of the proposed memetic algorithm. The main invention of the proposed local search algorithm is to build a number of neighbors for a solution using the prior knowledge vector and the posterior knowledge vector to select effective features and remove useless and irrelevant features. The results of implementing the proposed algorithm and comparing these results with similar works show that the proposed method in most cases leads to better results.

Keywords: Multi-label feature selection, Memetic algorithm, Local search algorithm, Prior knowledge vector, Posterior knowledge vector, Gravitational search algorithm.

1. Introduction

Feature selection is one of the most significant topics in the enhancement the efficiency of machine learning algorithms. In many applications, there are many features, many of which are either unused or have little distinction power and are actually misleading. Therefore, identifying and removing them from the dataset will increase the speed and accuracy of classification algorithms. Feature selection seeks to remove these features from the dataset [1], [2]. Multi-Label datasets are used in various fields such as bioinformatics, text classification, image processing, gene function classification, etc. Unlike single-label datasets in which one label is assigned to each instance, multi-label datasets belong a subset of a limited set of labels to each sample, therefore, feature selection and classification in multi-label datasets will be challenging. A set of labels in this case is a binary set that the presence of zero value in each of the labels for each sample means that the label is not assigned to the sample and the value of one indicates the assigned of the label to the sample. In this study, a memetic algorithm for feature selection in a multi-label dataset is proposed. The main innovation of this paper is the presentation of a new local search algorithm which, in combination with the binary quantum-inspired gravitational search (BQIGSA) algorithm [13], forms the main framework of the proposed memetic algorithm. The main invention of the proposed local search algorithm is to construct a number of neighbors for a solution using a combination of prior knowledge vector and posterior knowledge vector with dynamic weighting to select effective features and remove useless features.

At the beginning of the algorithm, the decision in the local search algorithm is based only on the prior knowledge vector, and gradually, by completing the posterior knowledge from the knowledge received from the performance of the features during execution, the weight of the prior knowledge vector decreases and the weight of the posterior knowledge vector increased. How to update the posterior vector is somewhat similar to updating the pheromone vector in ACO algorithm, in which features that will perform well during execution are amplified, and features that do not perform well due to evaporation switch to not selection state. The results of implementing the proposed algorithm and comparing these results with similar tasks show that the proposed method in most cases leads to better results. In each iteration of the algorithm, the Multi-Label K Nearest Neighbors (MLKNN)

classification algorithm is used to evaluate the desirability of the created population as well as the neighbors produced from the best member of the population.

The rest of the paper is organized in this way: Section 2 will description a review of related works. Section 3 describes the proposed algorithm for the multi-label feature selection. The results of the implementation of the proposed algorithm will be present in Section 4. Eventually, Section 5 presents the conclusion.

2. Related works

Common multi-label feature selection algorithms first convert a set of labels to a label with a convert function (problem conversion process) and then solve the feature selection problem with a label using traditional feature selection methods. For example, Young and Pederson compared five methods for multi-label feature selection based on five score functions for text classification [4]. In this work, the relationship between the labels is not considered because each label is considered separately. Chen et al. suggested an entropy-based method for each label that gives weights to a multi-label template based on the entropy of the label [5]. Patterns with many labels are lost during the training phase because of the low weights assigned to them. Although the problem-conversion approach simplifies the feature selection process, it can lead to other problems. For example, because problem-conversion methods convert label set into one label that contains several classes, if the labels in the original set are very different, a single label from multiple classes will be formed. As a result, due to the lack of samples for each class label, the performance of the learning algorithm may be severely decreased.

To overcome this problem, approaches to adapting classification algorithms to multi-label problems are proposed that solve multi-label problems directly. Zhang et al. have proposed a method for selecting multi-label features based on genetic algorithm that assessments the usefulness of a subset of selected features using the accuracy of a multi-label classifier [6]. These methods also have weaknesses such as early solutions and low convergence speed. Ji and Ye provide a comprehensive learning algorithm in which selecting features and multi-label classification are performed simultaneously [7]. This framework considers linear connections between features and labels. Therefore, the goal of this framework is to discover a combination of linear functions for each label.

Gu et al. have developed a method for selecting multi-label features that minimizes label ranking errors [8]. Although this method is highly generalizable due to the use of SVM algorithm, but it also has a high computational cost. Lee et al. have proposed a multi-label feature selection method for text categorization using label frequency difference [9]. Li et al. have proposed a multi-label feature selection method based on mutual information [10]. In this work, by granulating the label space, it tries to select the features that have the highest correlation with the labels of each granule and the least increase with them. Barani et al. has proposed a feature selection method based on the BQIGSA algorithm, which is used in combination with the k-nearest neighbor classifier as a wrapper method to evaluate selected subsets of features.

Paniri et al. has proposed a multi-label relevance–redundancy feature selection approach based on ACO (Ant colony optimization), called MLACO [11]. By offer two heuristic functions (unsupervised and supervised), MLACO seek in the features space to discover the most superior features with the lowest redundancy (unsupervised) and highest relevancy with class labels (supervised) over several iterations. Gonzalez-Lopez et al. suggested a distributed method to calculate a score that evaluate the quality of each feature corresponding to multiple labels on Apache Spark [12]. This method suggests two different methods that evaluate how to accumulate the mutual information of multiple labels: Euclidean Norm Maximization (ENM) and Geometric Mean Maximization (GMM). The preceding selects the features with the biggest L^2 norm whereas the recent selects the features with the biggest geometric mean.

Hashemi et al. has proposed a multi-label relevance–redundancy feature selection approach based on graph by PageRank centrality [3]. In this method, a graph is created in which each vertex is one of the features and the weight between the two vertices indicates their Euclidean distance. The importance of each feature is calculated using the PageRank algorithm. Dowlatshahi et al. have developed a discrete gravitational search method that solves combinatorial optimization problems [20]. This method uses a Path Re-linking approach instead of the classic approach in which the agents of GSA usually move from their current location to the location of other agents.

Lee et al. suggested a memetic feature selection method for multi-label feature selection which is created by combining genetic algorithm with a local search and purification algorithm [21]. The local search algorithm tries to find a subset of better-fitting features by applying a set of operators. Kashef and Nezamabadi-pour propose a method to find label-specific features using multi-objective domain dominance concepts [22]. In this method, the features are mapped in a multi-dimensional space and the superior features are selected using the concepts of Pareto-dominance. In pruned problem transformation (PPT) method, the set of examples with labels that seldom appear in a multi-label dataset are pruned by a threshold amount or designated to another class [23]. Momeni et al. have developed a neural network method based on gravitational search algorithm in anticipate the deformity of geogrid-reinforced soil construction [24]. Hashemi et al. have proposed a multi-label feature selection method based on a multi-criteria decision making (MCDM) model [25]. In this method, first calculate a decision-making matrix by the ridge regression algorithm and then compute the weight of each column of this matrix based on the entropy of each label. Later, the TOPSIS method is used to assign an amount to each feature based on the weighted decision-making matrix.

Hashemi et al. have modeled the problem of multi-label feature selection to a bipartite graph matching process. The suggested method constructs a bipartite graph of features (as the left vertices) and labels (as the right vertices), called Feature-Label Graph (FLG), where each feature is connected to the set of labels, where the weight of the edge between each feature and label is equal to their correlation. Then, the Hungarian algorithm estimates the best matching in FLG. The selected features in each matching are sorted by weighted

correlation distance and added to the ranking vector. To select the discriminative features, the proposed method considers both the redundancy of features and the relevancy of each feature to the class labels [26]. Dowlatshahi et al. have proposed a novel hybrid filter-wrapper algorithm, called Ensemble of Filter-based Rankers to guide an Epsilon-greedy Swarm Optimizer (EFR-ESO), for solving high-dimensional feature subset selection. The Epsilon-greedy Swarm Optimizer (ESO) is a novel binary swarm intelligence algorithm introduced in this paper as a novel wrapper [27]. In the proposed EFR-ESO, they extract the knowledge about the feature importance by the ensemble of filter-based rankers and then use this knowledge to weight the feature probabilities in the ESO. Kou et al. have developed a novel feature selection method to select a more relevant and compact feature subset by considering the label distribution and inter-label correlations. First, the concept of label distribution was defined to reflect the significance of each label. Second, a new algorithm for mining association rules was designed to obtain the correlation between labels by improving the existing association rules algorithm. Thus, a new information system was designed by combining the label distribution and correlation between labels [28]. Zhang et al. propose a method that first identify two underlying assumptions based on high-order label distribution: Label Independence Assumption (LIA) and Paired label Independence Assumption (PIA). Second, we systematically analyze the strengths and weaknesses of two assumptions and introduce joint mutual information to satisfy more realistic label distribution. Furthermore, by decomposing joint mutual information, an interaction weight is proposed to consider multiple label correlations. Finally, a new method considering join mutual information and interaction weight is proposed [29].

3. The proposed memetic feature selection

In this section, first, a brief explanation of the BQIGSA algorithm is given, and then the general principles of the proposed algorithm are stated, and at the end, the pseudo-code of the proposed method will be explained.

3.1. Binary quantum-inspired gravitational search algorithm

GSA algorithm is one of the population-based optimization algorithms which its performance is based on the gravitational effects of particles on each other [13]. The original version of this algorithm was created to solve continuous optimization problems that have proven their efficiency in various applications. In various works, the binary version of this algorithm (BGSA) has been used to solve binary coded problems and the discrete version (DGSA) has been used to solve hybrid problems [13], [17].

Most of optimization methods are implemented in binary space, such as feature selection. Rashedi et al. has presented a binary form of the GSA algorithm [13]. In the basic form of the GSA algorithm, gravitational strengths precisely change the location of particles in a continuous multidimensional space. In the binary form of the GSA (BGSA) algorithm, the effects of these strengths become a probabilistic value for each bit of the binary vector, so each element of the vector is now in only one of the 0 or 1 states.

The BQIGSA contains all the concepts and equations in BGSA. In BQIGSA, the equations of position and velocity are redefined with the equations in quantum computing. In quantum computing, a qubit (Q-bit) is the smallest unit of information on which quantum computers perform a series of quantum operations. Each Q-bit can be in "0" or "1" states, or a combination of both states. The third state, which combines two states of "0" and "1" at the same time, is called superposition. Each Q-bit is represented as a pair of α and β numbers (αβ), so that the $|\alpha|^2$ and $|\beta|^2$ values indicate the probability of being zero or one states, respectively.

Nezamabadi-pour [17] suggested a binary version of the GSA (BGSA) algorithm in which the BGSA algorithm is combined with quantum computing, called Binary Quantum-Inspired Gravitational Search Algorithm (BQIGSA). In this version of the algorithm, a new rotation Q-gates is integrated into the framework of the GSA algorithm. The purpose of implementing this strategy was to solve binary problems and improve the exploration operations of the GSA algorithm. In the BQIGSA algorithm, each member of the population is represented as a string of Q-bits of length *n*. Equation (1) shows the encoding of a member of the population.

$$q_i(t) = [q_i^1(t), q_i^2, \dots, q_i^n(t)] = \begin{bmatrix} \alpha_i^1(t) & \alpha_i^2(t) & \cdots & \alpha_i^n(t) \\ \beta_i^1(t) & \beta_i^2(t) & & \beta_i^n(t) \end{bmatrix} \quad (1)$$

For each Q-bit the following equation must be established:

$$\left|\alpha_i^d\right|^2 + \left|\beta_i^d\right|^2 = 1 \quad (2)$$

In each iteration of the BQIGSA algorithm, the binary members of the population $SW(t) = \{X_1(t), X_2(t), \dots, X_s(t)\}$ are recreated by observing Q-bit objects according to Equation (3), where $X_i(t) = (x_i^1(t)\ x_i^2(t), \dots, x_i^n(t))$.

$$\begin{aligned} &\boldsymbol{If} \quad rand[0,1] < \left(\alpha_i^d(t)\right)^2 \\ &\boldsymbol{Then} \quad x_i^d(t) = 0 \\ &\boldsymbol{Else} \quad x_i^d(t) = 1 \end{aligned} \quad (3)$$

In each iteration of the algorithm, the best solution found by each population member is stored in set $SB(t) = \{B_1(t), B_2(t), \dots, B_s(t)\}$, where $B_i(t) = (b_i^1(t), b_i^2(t), \dots, b_i^n)$.

In the first step of the algorithm, the initial population Q(t) consisting of s Q-bits is randomly constructed according to Equation (3). Most of the binary quantum inspired evolutionary algorithms direct members of the population to the object with the best value of fitness. This position change is implemented via a Rotation Q-gate (RQ-gate) which spin the quantum state of members of the population towards the quantum state of the best solution. Figure (1) shows how to update Q-bits objects in the Q (t) population using the RQ-gate.

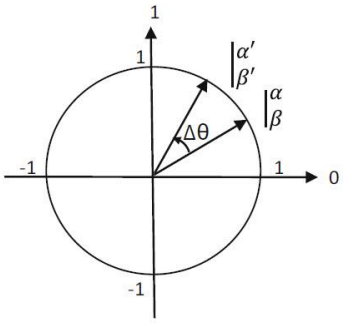

Fig. 1. The operation of RQ-gate

The RQ-gate operation is also shown by Equation (4).

$$\begin{bmatrix}\alpha_i^d(t+1)\\ \beta_i^d(t+1)\end{bmatrix} = \begin{bmatrix}\cos(\Delta\theta_i^d) & -\sin(\Delta\theta_i^d)\\ \sin(\Delta\theta_i^d) & \cos(\Delta\theta_i^d)\end{bmatrix} \times \begin{bmatrix}\alpha_i^d(t)\\ \beta_i^d(t)\end{bmatrix} \quad d = 1,2,\dots,n \tag{4}$$

In quantum computing, angular velocity regulates the amount of motion toward "0" or "1", and is used in RQ-gate operations on Q-bits. This value is calculated for each Q-bit using equation (5).

$$\alpha_i^d(t) = \sum_{j\in kbest} G(t)\frac{M_j(t)}{R_{ij}(t)+\varepsilon}\left(b_j^d(t) - x_i^d(t)\right) \tag{5}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{s} m_j(t)} \tag{6}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{7}$$

$$best(t) = \max_{j\in\{1,\dots,s\}} fit_j(t) \tag{8}$$

$$worst(t) = \min_{j\in\{1,\dots,s\}} fit_j(t) \tag{9}$$

kbest refers to k members with the highest amount of fitness in the SB(t) set. $R_{ij}(t)$ represents the distance between two vectors, which is calculated using the normalized Hamming distance. Equation (10) shows how to calculate this distance.

$$R_{ij}(t) = \frac{\sum_{l=1}^{n}\left|x_i^l(t) - b_j^l(t)\right|}{n} \tag{10}$$

3.2. Main structure of proposed method

In this study, we seek to enhance the performance of the binary quantum-inspired gravitational search algorithm in selecting multi-label features using a new local search algorithm in population refinement. The method is that from the initial population that is randomly developed, the best member of the population is selected for refinement based on the amount of mutual information of its selected features with each of the labels. In the refinement phase, a new local search algorithm is used to decide to remove some useless features and add some useful features. In the proposed local search algorithm, a combination of two vectors of prior knowledge and posterior knowledge is used. Prior knowledge is the amount of mutual information of each feature with the class labels that is initially calculated and remains constant until the end of the algorithm. So, to define the prior knowledge vector, we have a vector match to the total of features, each element of which represents the amount of mutual information of a feature with the class label. In the initial steps of the local search algorithm, due to the unavailability of the values of the posterior knowledge vector, the knowledge contained in this vector is mainly used to decide on the selection and removal of features.

Over time, after each iteration of the algorithm, feedbacks on the performance of various features are received and stored in another vector called the posterior knowledge vector. The idea of updating this vector is inspired of updating the pheromone vector in the Ant Colony Optimization (ACO) algorithm. In such a way that in each iteration, the selected features will have a positive effect on their corresponding knowledge in this vector. Therefore, after the first few steps of the algorithm, we will have knowledge in this vector that will be more useful for deciding on the selection of features than the prior knowledge vector, because it contains knowledge gained from the actual performance of the features during algorithm execution. Therefore, as we get closer to the final steps of the BQIGSA algorithm, we need to increase the weight of the posterior knowledge vector in improving the solutions over the weight of the prior knowledge vector. Therefore, in each step of the algorithm, the knowledge contained in the two vectors of prior knowledge and posterior knowledge are combined and a total knowledge vector is obtained.

To determine the importance of each of these two vectors in the formation of the total knowledge vector, an α coefficient is used, which initially has a value of 1 and is multiplied by 0.95 in each step. The reason for choosing this value for the knowledge vectors coefficient is that after completing the initial steps, and gradually completing the posterior knowledge vector, we intend to reduce the importance of the prior knowledge vector in decision making and increase the importance of the posterior knowledge vector so that in the final steps of the algorithm, decisions will be made mainly on the basis of a posteriori knowledge vector. Figure (2) shows the process of reducing the α coefficient in 50 iterations of the algorithm.

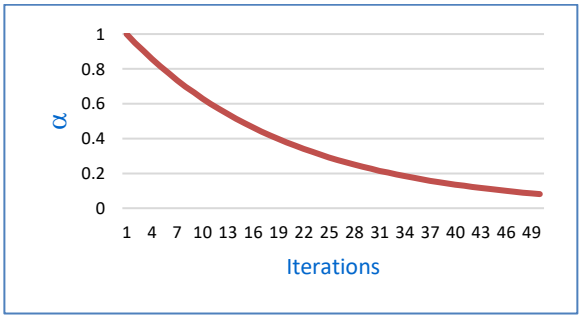

Fig. 2. α coefficient update process

The formation of the total knowledge vector at each step of the algorithm is performed using equation (11) in which preK(t) is the prior knowledge vector, postK(t) is the posterior knowledge vector, and TK(t) is the total knowledge vector:

$$TK(t) = \alpha \times preK(t) + (1-\alpha)\times posK(t) \tag{11}$$

Figure (3) shows how to generate a total knowledge vector using the prior knowledge vector and posterior knowledge vectors.

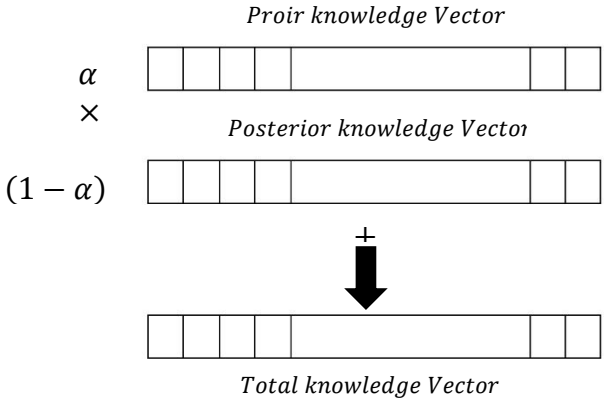

Fig. 3. Process of total knowledge vector generation

In the local search algorithm, eight neighbors are generated for the best member of the population according to the total knowledge vector. First, four neighbor solutions are created by switches four bit from zero to one of the best members of the population, and then the other four neighbors are created by switches four bit from one to zero of the best members of the population. In the first four neighbors, bits change to the value of one that has the highest value in their corresponding value in the total knowledge vector relative to the other zero bits. In the other four neighbors, bits change to the value of zero that has the lowest value in their corresponding value in the total knowledge vector relative to the other one bit. In fact, with this action, we select four features that are of high importance but have not yet been selected in the best member of the population, and remove the four features that are of low importance but have been selected in the best member of the population.

In the next step, the eight generated neighbors are evaluated in terms of utility and the best of them (If better than the best member of the population) replaces the best member of the population. If there is a better neighbor, we should apply its effects to the posterior knowledge vector. This effect is achieved by reinforcing the corresponding elements of the four features of the first four neighbors that were selected in the posterior knowledge vector. This amplification is done by adding a value of (1-α) to these elements. Selecting this value causes weaker reinforcements in the initial steps where posterior knowledge is still being formed, but in the final steps where posterior knowledge is almost complete, reinforcements are made more intensively. In fact, the more complete the general knowledge of features, impact of features on the posterior knowledge vector is greater. Evaporation is performed on all posterior knowledge vector elements by subtracting a fixed value from them. It should be noted that the steps mentioned above in the form of a local search algorithm within the framework of the BQIGSA algorithm, perform the action of population refinement.

3.3. Pseudo-code of the proposed memetic algorithm

Pseudocode (1) shows the general process of implementing the proposed algorithm.

Pseudocode (1): Pseudocode of the proposed memetic algorithm.

```
1: Procedure MMLFSBQIGSA (u, m)
2:    t=0
3:    Initializing P(t)
4:    Calculate prior knowledge vector
5:    While t ≤ u do
6:       Evaluating P(t)
7:       Apply Local Search to P(t)
8:       Update P(t) using BQIGSA operators
9:       t=t+1
10:   End while
11: End procedure
```

In the third line of the proposed algorithm, the initial population is generated as random binary vectors. The length of each vector is equal to the number of current dataset features represented by the variable m. In the fourth line, the prior knowledge vector is calculated. This vector contains the amount of mutual information of each attribute with the set of labels. In the fifth to tenth lines in a loop that runs u times, the following steps are performed in order. First, the generated population is evaluated using an evaluation function, which in this article is the MLKNN classifier, and the entire population is sorted by quality. Then, the local search operation is performed on the best member of the population. this refinement is done only if a better neighbor is found for the best member of the population, in which case the found neighbor will replace the best member of the population, and will have its effects on the posterior knowledge vector. After the local search algorithm, the BQIGSA algorithm is executed on the population and the steps are repeated to the number of iterations mentioned by the user(u). In BQIGSA, the number of population members and the number of iterations of the algorithm are considered to be 50. Pseudocode (2) shows the operation steps of the local search algorithm on the best member of the population.

Pseudocode (2): Pseudocode of the local search algorithm.

```
1: Procedure LOCAL SEARCH ()
2:    c=best member in P(t)
2:    Calculate total_knowledge vector by Eq. (1)
3:    Generate 4 neighboors by flip 0 to 1 bit
4:    Generate 4 neighboors by flip 1 to 0 bit
5:    Evaluate 8 neighboors
6:    b=best of neighboors
7:    If fitness b is superior than fitness c then
8:       Substitution c with b
9:       Update post_knowledge vector by reinforcement
10:      Evaporation all elements in post_knowledge vector
11:   End if
12:   Update α
13: End procedure
```

4. Experimental results

4.1. Datasets

In this study, eight widely used datasets in other studies have been selected to evaluate the performance of the present method and also to compare its results with some similar feature selection methods in filter and wrapper methods. Details of each of the listed datasets can be seen in Table I.

Table I
The multi-label data set used in this paper

Datasets	Domain	Instances	Features	Features Type	Labels
Scene	Image	2407	294	Continuous	6
Image	Image	2000	294	Continuous	5
Corel5k	Image	5000	499	Binary	374
cs	Image	9270	909	Binary	274
Enron	Text	1702	948	Binary	53
Yeast	Biology	2417	130	Continuous	14
Medical	Text	978	1494	Binary	45
Genbase	Biology	662	1185	Binary	27

4.2. Classifier and Evaluation Metrics

For classification, the Multi-label K-nearest neighbor classification algorithm is used as a fitness function. In MLKNN, for each invisible sample x, its k-nearest neighbor is selected in the training set, then, based on the statistical information achieved from the sample class labels of the selected neighbors, the number of neighbors associated to each class and the maximum a posterior (MAP) class label set for the unseen sample is determined [18]. The appropriate value for the k parameter is considered 10 in most experiments as well as this study.

In this paper, four criteria of Hamming loss, multi-label accuracy, Ranking Loss and Average Precision have been used to evaluate the performance of the proposed method. Assume that $T = \{(x_i, Y_i) \mid i = 1, \dots, n\}$ is a test set and h is a multi-label classifier. Y⊆L is also a true subset of labels and $L = \{l_j : j = 1, \dots, q\}$ is a set of all labels. Upon receipt of sample xi, the set of tags predicted by MLKNN is shown as Z_i and the evaluation meters are calculated as follows [19]:

Multi-label accuracy displays the percentage of correctly anticipated labels among all anticipated and actual labels.

$$Accuracy(h,T) = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

Hamming loss estimates the total of times a sample-label couple has been misclassified.

$$Hamming - Loss(h,T) = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \Delta Z_i|}{|L|}$$

while Δ is the symmetric difference between the two sets.

Ranking Loss evaluates the number of reversely ordered label pairs; an irrelevant label is ranked higher than a relevant label.

$$rloss(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\overline{Y_i}|}|\{(y', y'') | f(x_i, y') \leq f(x_i, y''), (y', y'') \in Y_i \times \overline{Y_i}\}|$$

Average Precision evaluates the average percentage of relevant labels ranked higher than an actual label $y \in Y_i$.

$$Avg - Pre(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y \in Y_i}\frac{\{y' \mid rank_f(x_i, y) \leq rank_f(x_i, y), y \in Y_i\}}{rank_f(x_i, y)}$$

4.3. Results

The results of the suggested method are compared with the seven feature selection methods with filter and wrapper approaches. Table II shows the performance of the seven existing methods as well as the proposed method based on the Hamming Loss criteria. The values highlighted in the results section of the proposed method indicate the superiority of this method in classifying test data sets. As can be seen, the proposed algorithm has shown a significant dominance over the compared methods.

Table II
Comparison of the results of the proposed method and previous methods with the criterion of Hamming Loss

Datasets	Scene	Image	Corel5k	cs	Enron	Yeast	Medical	Genbase
ParetoFS	0.1225	0.9130	**0.0093**	0.0078	0.0532	0.1976	0.0159	0.0053
Ppt-chi	0.1565	0.2227	0.0094	0.0090	0.0632	0.1996	0.0155	0.0059
MLACO	0.1161	0.1935	0.0094	0.0089	0.0567	0.2033	0.0226	0.0073
MGFS	0.1080	0.1861	0.0094	0.0076	0.0510	0.1955	0.0164	0.0052
MAMFS	0.0939	0.1535	0.0094	0.0047	0.0486	0.2150	**0.0021**	0.0044
MLDRS	0.1253	0.2105	0.0095	**0.0067**	0.0490	**0.1802**	0.0152	0.0037
MFSJMI	**0.0624**	**0.1504**	0.0094	0.0069	0.0481	0.1922	0.0123	0.0038
Proposed Method	0.0828	**0.**1643	**0.0093**	0.0089	**0.0477**	0.1905	0.0143	**0.0034**

Table III shows the performance of the seven existing methods as well as the proposed method based on the multi-label accuracy evaluation criteria. As the results show, in this evaluation criterion, the proposed method has a significant dominance over the compared methods. The results also show that the proposed method can work well in both discrete and continuous data.

Table III
Comparison of the results of the proposed method with previous methods in the field of multi-label accuracy

Datasets	Scene	Image	Corel5k	cs	Enron	Yeast	Medical	Genbase
ParetoFS	0.4869	0.3637	0.0118	0.0840	0.3069	0.5076	0.5465	0.9359
Ppt-chi	0.2609	0.2177	0.0049	0.0782	0.0403	0.4952	0.5611	0.9309
MLACO	0.5115	0.3881	0.0070	0.1031	0.1513	0.5213	0.6147	0.9631
MGFS	0.5874	0.4465	0.0064	0.0702	0.3774	0.5086	0.6079	0.9465
MAMFS	0.4026	0.3252	0.0044	0.0156	0.0592	0.4816	0.7096	0.9564
MLDRS	0.6712	0.5411	**0.0116**	0.1044	0.2503	0.5128	**0.7405**	0.9511
MFSJMI	0.6500	0.5396	0.0106	**0.1053**	**0.3861**	0.5004	0.7170	0.9403
Proposed Method	**0.6918**	**0.5446**	0.0092	0.1070	0.1867	**0.5321**	0.7229	**0.9683**

Table IV shows the performance of the seven existing methods as well as the proposed method based on the Ranking Loss criteria. The values highlighted in the results section of the proposed method indicate the superiority of this method in classifying test data sets. As can be seen, the proposed algorithm has shown a significant dominance over the compared methods.

Table IV
Comparison of the results of the proposed method with previous methods in the field of ranking loss

Datasets	Scene	Image	Corel5k	cs	Enron	Yeast	Medical	Genbase
ParetoFS	0.1607	0.2066	0.1833	0.1389	0.1846	0.1904	0.0960	0.0367
Ppt-chi	0.2229	0.2876	0.1419	0.1485	0.2096	0.2025	0.1100	0.1040
MLACO	0.1189	0.2107	0.2081	0.1397	0.2081	0.1801	0.0685	0.0134
MGFS	0.1232	0.1812	0.1405	0.1476	0.1916	0.1937	0.0841	0.0203
MAMFS	0.0205	**0.1303**	0.1527	**0.1084**	0.1808	0.1783	0.0725	0.0148
MLDRS	0.1478	0.2294	0.1511	0.1233	0.1889	**0.1485**	0.0641	0.0108
MFSJMI	**0.0200**	**0.1303**	0.1402	0.1100	0.1690	0.1567	0.0513	0.0099
Proposed Method	0.0744	0.1702	**0.1339**	0.1347	**0.1670**	0.1705	**0.0471**	**0.0078**

Table V shows the performance of the seven existing methods as well as the proposed method based on the average precision evaluation criteria. As the results show, in this evaluation criterion, the proposed method has a significant dominance over the compared methods. The results also show that the proposed method can work well in both discrete and continuous data.

Table V
Comparison of the results of the proposed method with previous methods in the field of average precision

Datasets	Scene	Image	Corel5k	cs	Enron	Yeast	Medical	Genbase
ParetoFS	0.7942	0.7402	0.2430	0.3566	0.6211	0.7584	0.7900	0.9841
Ppt-chi	0.6819	0.6714	0.2191	0.3340	0.6186	0.7529	0.7942	0.9807
MLACO	0.8017	0.7529	0.2288	0.3546	0.4501	0.7782	0.7703	0.9950
MGFS	0.8221	0.7540	0.2416	0.3502	0.4833	0.7425	0.7004	0.9767
MAMFS	0.7801	0.7611	0.2302	0.3497	0.5517	0.7444	0.8014	0.9076
MLDRS	0.7669	0.7301	**0.2621**	0.3314	0.6015	**0.7883**	0.8134	0.9654
MFSJMI	0.8332	**0.7984**	0.2506	0.3591	**0.6304**	0.7725	0.8308	0.9607

Proposed Method	**0.8659**	0.7880	0.2542	**0.3609**	0.5010	0.7750	**0.8437**	**0.9998**

Figures (4-7) demonstrate the comparison of the proposed method with other algorithm in term of hamming loss, multi-label accuracy, ranking loss and average precision, respectively. According the obtained result, it can be seen that the proposed method has a significant advantage over other methods in terms of four classification criteria.

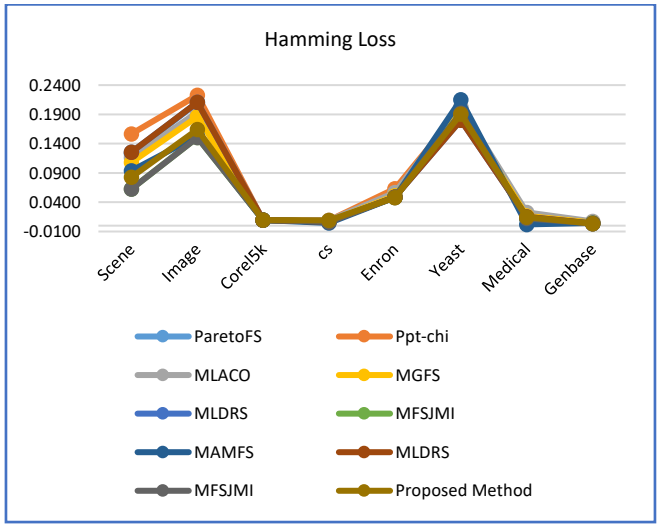

Fig. 4. Comparison of the proposed method with other algorithm in term of hamming loss

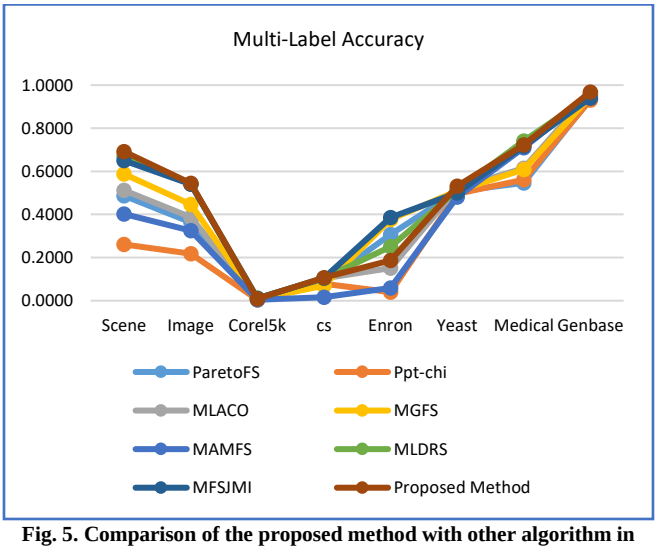

Fig. 5. Comparison of the proposed method with other algorithm in term of multi-label accuracy

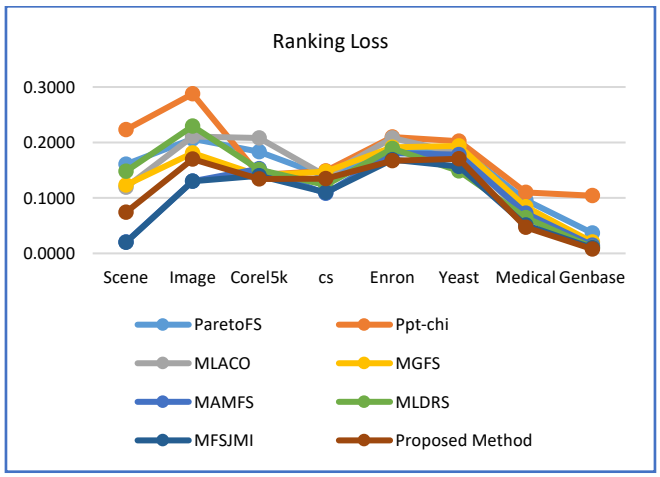

Fig. 6. Comparison of the proposed method with other algorithm in term of ranking loss

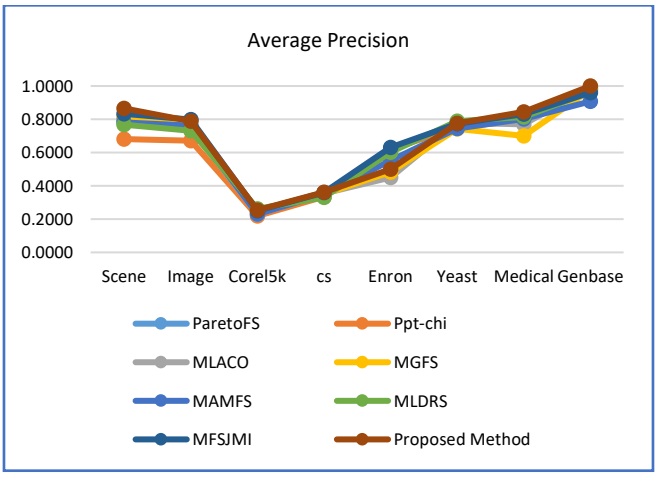

Fig. 7. Comparison of the proposed method with other algorithm in term of average precision

5. Conclusion

Feature selection in multi-label datasets is a challenging issue in many new applications. Like single-label classification, multi-label classification performance may result in inaccurate prediction of output labels due to the large number of input features. Therefore, multi-label feature selection is one of the effective techniques to overcome this problem in multi-label datasets. In the present paper, a new local search algorithm is proposed which, by combining it with the BQIGSA algorithm and presenting a new memetic algorithm, tries to find a high-quality subset of features with high differentiation. In the proposed local search algorithm, the best member of the population is subjected to a local refinement and by choosing neighbors for this member, tries to find a more desirable neighbor and an alternative in the present population. The production of neighbors in this case is based on the combination of prior and posterior knowledge. Therefore, in the production of neighbors, features are considered that have had a good performance both at the beginning of the algorithm and during the implementation of the algorithm have been able to prove their goodness. Experiments have shown that the proposed algorithm will often perform better than similar methods.

References

[1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

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

[3] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality," *Expert Systems with Applications*, vol. 142, p. 113024, 2020.

[4] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. 14th Int. Conf. Machine Learning*, Nashville, USA, 1997, pp. 412–420.

[5] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang, "Document transformation for multi-label feature selection in text categorization," in *Proc. 7th IEEE Int. Conf.*, 2007, pp. 451–456.

[6] M. Zhang, J. Pena, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Information Sciences*, vol. 179, no. 19, pp. 3218–3229, 2009.

[7] S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proc. 21st Int. Joint Conf. Artificial Intelligence*, Pasadena, USA, 2009, pp. 1077–1082.

[8] Q. Gu, Z. Li, and J. Han, "Correlated multi-label feature selection," in *Proc. 20th ACM Int. Conf. Information and Knowledge Management*, Glasgow, UK, 2011, pp. 1087–1096.

[9] J. Lee, I. Yu, J. Park, and D. Kim, "Memetic feature selection for multilabel text categorization using label frequency difference," *Information Sciences*, vol. 485, pp. 263–280, 2019.

[10] F. Li, D. Miao, and W. Pedrycz, "Granular multi-label feature selection based on mutual information," *Pattern Recognition*, vol. 67, pp. 410–423, Jul. 2017.

[11] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MLACO: A multi-label feature selection algorithm based on ant colony optimization," *Knowledge-Based Systems*, vol. 192, pp. 263–280, 2020.

[12] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed multi-label feature selection using individual mutual information measures," *Knowledge-Based Systems*, vol. 188, pp. 263–280, 2020.

[13] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Computing*, vol. 9, pp. 727–745, 2010.

[14] M. B. Dowlatshahi and H. Nezamabadi-pour, "GGSA: A grouping gravitational search algorithm for data clustering," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 114–121, 2014.

[15] C. Li and J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm," *Energy Conversion and Management*, vol. 52, no. 1, pp. 374–381, 2011.

[16] S. Sarafrazi and H. Nezamabadi-pour, "Facing the classification of binary problems with a GSA–SVM hybrid system," *Mathematical and Computer Modelling*, vol. 57, no. 1–2, pp. 270–278, 2013.

[17] H. Nezamabadi-pour, "A quantum-inspired gravitational search algorithm for binary encoded optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 62–75, 2015.

[18] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, pp. 2038–2048, Jul. 2007.

[19] 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, MA: Springer US, 2010, pp. 667–685.

[20] M. B. Dowlatshahi, H. Nezamabadi-pour, and M. Mashinchi, "A discrete gravitational search algorithm for solving combinatorial optimization problems," *Information Sciences*, vol. 258, pp. 94–107, 2014.

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

[22] S. Kashef and H. Nezamabadi-pour, "A label-specific multi-label feature selection algorithm based on the Pareto dominance concept," *Pattern Recognition*, vol. 88, pp. 654–667, 2019.

[23] J. Read, "A pruned problem transformation method for multi-label classification," in *Proc. New Zealand Computer Science Research Student Conference*, 2008.

[24] E. Momeni, A. Yarivand, M. B. Dowlatshahi, and D. J. Armaghani, "An efficient optimal neural network based on gravitational search algorithm in predicting the deformation of geogrid-reinforced soil structures," *Transportation Geotechnics*, vol. 26, p. 100446, 2020.

[25] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MCDM: Multi-label feature selection using multi-criteria decision-making," *Knowledge-Based Systems*, vol. 206, p. 106365, 2020.

[26] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, "A bipartite matching-based feature selection for multi-label learning," *International Journal of Machine Learning and Cybernetics*, pp. 1–17, 2020.

[27] M. B. Dowlatshahi, V. Derhami, and H. Nezamabadi-pour, "Ensemble of filter-based rankers to guide an epsilon-greedy swarm optimizer for high-dimensional feature subset selection," *Information*, vol. 8, no. 4, p. 152, 2017.

[28] Y. Kou, G. Lin, Y. Qian, and S. Liao, "A novel multi-label feature selection method with association rules and rough set," *Information Sciences*, vol. 624, pp. 299–323, 2023.

[29] P. Zhang, G. Liu, and J. Song, "MFSJMI: Multi-label feature selection considering joint mutual information and interaction weight," *Information Sciences*, vol. 138, p. 109378, 2023.

Hojat Noormohammadi
Email: nourmohammadi.ho@fe.lu.ac.ir

Mohammad Bagher Dowlatshahi received his BSc and MSc degrees in software engineering from Sharif University of Technology, in 2010 and 2012 respectively.
Email: dowlatshahi.mb@lu.ac.ir