**Label Clustering for PSO-based Multi-Label Feature Selection**

**Abstract.** Feature selection is an important classification pre-processing step for multi-label classification, which selects a small subset of complementary features, while still improve classification performance. Recently Particle Swarm Optimization (PSO) has been widely applied to achieve multi-label feature selection. However, existing PSO-based multi-label approaches are based on a large number of labels which makes it more challenging to select features due to many complex interactions between features and labels. Therefore, this work proposes a novel PSO fitness evaluation algorithm, which investigates the relationships between different labels in multi-label classification. Based on the investigation, the original label set can be decomposed into multiple label clusters. In the proposed fitness function, each label cluster is considered one label, and thus the number of labels is significantly reduced. Feature selection can be performed on a new and small label set, which is less challenging than performing feature selection on the original and large label set. Compared with using all features and standard PSO feature selection on the original label set, the proposed mechanisms achieve better performance on real world datasets.

**Keywords:** Particle Swarm Optimization · Feature selection ·

Multi-label classiﬁcation

**1.Introduction**

Goals:

**2.Background**

**2.1 Multi-label Classiﬁcation**

* 1. **Multi-label Feature Selection**

**2.3 Particle Swarm Optimization**

Particle swarm optimization (PSO) [8] is a population-based optimization tech-

nique, where each particle represents a candidate solution. The particles move

in the search space with their own position and velocity vectors, denoted by

x and v, respectively. Each particle records its best position, called pbest, and

the best position discovered by all particles, called gbest. Based on the two best

positions, the position and velocity of the ith particle can be updated according

to the following equations:

where t represents the tth iteration, d represents the dth dimension in the search

space, c1 and c2 are acceleration constants, r1 and r2 are random constants in [0,

1], w is the inertia weight. PSO has been widely applied to feature selection [7,28,

29], but very few researches in the literature have focused on PSO for multi-label

feature selection. To the best of our knowledge, Zhang et al. [11] proposed the

ﬁrst PSO-based multi-label feature selection, but it considers feature selection

as a multi-objective problem. This paper focuses on designing initialization for

a PSO-based multi-label feature selection algorithm where feature selection is

considered as a single-objective problem.

**3. Proposed Approach**

This section includes an overall structure of proposed PSO-based feature selection algorithm, and detailed description of label clustering, which is the proposed wrapped-in fitness evaluation.

**3.1 Overall Structure**

In the proposed PSO-based feature selection algorithms, a particle’s position is represented by a vector of real numbers, where each vector element corresponds to an original feature. The element value at the dth dimension, xid, is in [0, 1], which shows

whether the dth feature is selected or not. Particularly, a threshold θ is used to compare with the position value xid. If xid > θ, the dth feature is selected. Otherwise, the dth feature is not selected. Hence, we can consider each particle a feature subset candidate.

We can gain a number of feature subsets based on each particle’s vector, where the number of subsets equals to swarm population size. The quality of all these feature subsets need to be evaluated. The particle with the best quality is the final solution, which represents the final selected features from original full features.

The way to evaluate a particle’s quality, is applying its corresponding selected features and target labels for validation classification, where the classifier is MLKNN from scikit-multilearn library. MLkNN builds uses k-NearestNeighbors find nearest examples to a test class and uses Bayesian inference to select assigned labels. Finally calculate the hamming loss with true label set and predicted label set as its fitness value.

In a standard PSO-based feature selection approach, the evaluation classification uses the selected features plus the complementary label set for classification, which probably could be challenging and expensive if the target label set is large. Meanwhile, the standard PSO-based multi-label approach does not take the relationships and interactions between labels into account.

The proposed label clustering approach performs kmeans clustering on target label set, all the labels with similarities will be assigned to a same subgroup. This step can decompose original label set and transform it into a much smaller super-label set. The pseudo-code of the proposed algorithm can be seen in Algorithm 1, the label clustering which is from line 10 to line 18. Our main contribution is to reduce the computing cost when evaluating the quality of each particle in the process of feature selection, meanwhile, through investing the interaction between target labels, select better and more essential feature subsets thereby improve the final classification accuracy. More details about the two versions pf label clustering are presented in the following subsection.

**3.2 Label Clustering**

***Version 1 --original version***

*Algorithm 1: Pseudo-code of the proposed algorithm*

*1* ***Input:*** *Training set, Test set and labels*

*2* ***begin***

*3 initialize the position of each particle in the swarm;*

*4 initialize the velocity of each particle in the swarm;*

*5* ***while*** *Maximum iterations has not been met do*

*6 evaluate the ﬁtness of each particle according to Eq. (4);*

*7* ***for*** *i = 1 to Swarm Size do*

*8 update pbest and gbest of particle i;*

*9*  ***end***

*10*  ***if*** *gbest is not improved for m iterations then*

*11* ***while*** *the maximum number of ﬂipping operators is not*

*reached* ***do***

*12 perform the ﬂipping operator on gbest;*

*13 if better solution is found then*

*14 replace the current gbest by the found subset ;*

*15 end the local seach ;*

*16 end*

*17 end*

*18 end*

*19 for i = 1 to Swarm Size do*

*20 update vi of particle i according to Eq. (2);*

*21 update xi of particle i according to Eq. (1);*

*22 end*

*23 end*

*24 calculate the training and testing Hamming loss of the selected*

*feature subset;*

*25 return the position of gbest, the training and testing loss;*

*26 end14 end*

*15 calculate the training and testing hamming loss of gbest;*

*16 return gbest and its training/testing loss;*

*17 end*

***Version 2 –improved version***

*Algorithm 1: Pseudo-code of the proposed algorithm*

*1* ***Input:*** *Training set, Test set and labels*

*2* ***begin***

*3 initialize the position of each particle in the swarm;*

*4 initialize the velocity of each particle in the swarm;*

*5* ***while*** *Maximum iterations has not been met do*

*6 evaluate the ﬁtness of each particle according to Eq. (4);*

*7* ***for*** *i = 1 to Swarm Size do*

*8 update pbest and gbest of particle i;*

*9*  ***end***

*10*  ***if*** *gbest is not improved for m iterations then*

*11* ***while*** *the maximum number of ﬂipping operators is not*

*reached* ***do***

*12 perform the ﬂipping operator on gbest;*

*13 if better solution is found then*

*14 replace the current gbest by the found subset ;*

*15 end the local seach ;*

*16 end*

*17 end*

*18 end*

*19 for i = 1 to Swarm Size do*

*20 update vi of particle i according to Eq. (2);*

*21 update xi of particle i according to Eq. (1);*

*22 end*

*23 end*

*24 calculate the training and testing Hamming loss of the selected*

*feature subset;*

*25 return the position of gbest, the training and testing loss;*

*26 end14 end*

*15 calculate the training and testing hamming loss of gbest;*

*16 return gbest and its training/testing loss;*

*17 end*

**4. Experiment design**

**5. Results and discussion**

**6. Conclusions and Future Work**

**References**