International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-024-02098-3

ORIGINAL ARTICLE

Dynamic multi-label feature selection algorithm based on label importance and label correlation

Weiliang Chen[1] · **Xiao Sun**[1]

Received: 4 May 2023 / Accepted: 1 January 2024

Abstract
Multi-label distribution is a popular direction in current machine learning research and is relevant to many practical problems. In multi-label learning, samples are usually described by high-dimensional features, many of which are redundant or invalid. This paper proposes a multi-label static feature selection algorithm to solve the problems caused by high-dimensional features of multi-label learning samples. This algorithm is based on label importance and label relevance, and improves the neighborhood rough set model. One reason for using neighborhood rough sets is that feature selection using neighborhood rough sets does not require any prior knowledge of the feature space structure. Another reason is that it does not destroy the neighborhood and order structure of the data when processing multi-label data. The method of mutual information is used to achieve the extension from single labels to multiple labels in the multi-label neighborhood; through this method, the label importance and label relevance of multi-label data are connected. In addition, in the multi-label task scenario, features may be interdependent and interrelated, and features often arrive incrementally or can be extracted continuously; we call these flow features. Traditional static feature selection algorithms do not handle flow features well. Therefore, this paper proposes a dynamic feature selection algorithm for flow features, which is based on previous static feature selection algorithms. The proposed static and dynamic algorithms have been tested on a multi-label learning task set and the experimental results show the effectiveness of both algorithms.

Keywords Flow feature · Label correlation · Label importance · Multi-label distribution · Neighborhood rough set

1 Introduction

In the traditional machine learning framework, each sample corresponds to only one label; this is called single-label learning. Single-label learning is the most well-studied and widely used machine learning framework [1]. In single-label learning, an instance in the learning framework describes the properties of each real-world object, and the instance is associated with the class label of the semantic object to form a sample. Single-label learning has achieved good results in the single-label learning domain when the target instance has an explicit single class label.

In the real world, however, there is often more than one unique semantics. In fact, most objects are associated with more than one concept at the same time [2–5]. For example, an elderly patient may suffer from several diseases, including diabetes, hypertension, and coronary heart disease; a picture of a tiger in a forest may be associated with multiple keywords, such as "tiger" and "tree." Because polysemantic objects no longer have a single semantic meaning, a single-label learning framework that only considers a single explicit semantics is unlikely to achieve good results. To reflect this problem intuitively for polysemous words, the most obvious approach is to assign multiple category labels to each example of a polysemous word. This set of category labels is referred to as a subset of labels. The learning paradigm that uses the multi-label approach to labeling sample examples is known as multi-label learning [6]. Multiply labeled objects are ubiquitous in all areas of life. The multi-label learning paradigm has been widely used in text classification [7, 8],

✉ Weiliang Chen
2017010113@mail.hfut.edu.cn

[1] AnHui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, Key Laboratory of Knowledge Engineering With Big Data, Ministry of Education, School of Computer Science and Information Engineering, Hefei University of Technology, Feicui road, Hefei 230009, Anhui, China

Published online: 13 March 2024

bioinformatics [9, 10], sentiment recognition [11, 12], and information retrieval [13, 14].

Similarly to traditional single-label learning, multi-label learning has faced many challenges. With respect to the data structure of labeled instances, problems faced by multi-label tasks include large feature dimensionality [15, 16], large numbers of labels [17], label imbalance [18], and flow features [19]. For multi-label learning tasks, the dimensionality of multi-label data is large, often with thousands or tens of thousands of features [20, 21]. For a given learning task, a large proportion of these high-dimensional features may be redundant or invalid. High-dimensional data may cause various problems for learning, including overfitting, longer computation time, and higher memory consumption, compared with single-label data [22–25]. Therefore, reducing the dimensionality of the labeling task is a priority. Multi-label dimensionality reduction is a data preprocessing technique that can be used to remove redundant and irrelevant features and reduce the dimensionality of high-dimensional features. Common methods that have been proposed for multi-label dimensionality reduction include LDA [26], MDDM [27], MLST [28], PUM [29], and PL-ML [30]. Of these multi-label dimensionality techniques, multi-label feature selection methods have received much attention.

There are two main methods for multi-label dimensionality reduction: multi-label feature extraction and multi-label feature selection. Multi-label feature extraction methods, such as LDA, MDDM, and MLST, reduce the dimensionality of the feature space using spatial mapping techniques or spatial transformations, but these destroy the structural information of the original feature space, obscure the physical meaning of the features, and lack a semantic interpretation. Although methods such as PUM and PL-ML can improve learning performance, they all share a common limitation: a complete feature set needs to be collected before feature selection, and no attention is paid to the correlation between labels. In contrast to multi-label feature extraction methods, feature selection methods do not perform any feature space transformation or mapping; instead, they preserve the original spatial structure. A feature selection method selects a subspace that best represents the semantic features of the feature space by ranking the features by importance in the original feature space, and uses this subspace to represent the original feature space to the greatest extent possible [31, 32]. Thus, multi-label selection methods preserve the physical meaning of the feature space well, which is an advantage over feature extraction methods [33]. As the amount of multi-label data has increased, many feature selection methods for multi-label learning have been developed. These methods fall into three main categories: filters, wrappers, and embeddings. A filter first selects the features and then trains the classifier, so the feature selection process is independent of the classifier. This is equivalent to filtering the features first and then training the classifier with a subset of the features [34, 35]. Wrappers directly use the final classifier as the evaluation function for feature selection, to choose the optimal subset of features for a given classifier. Wrapper methods rely on a predetermined classifier to directly select a subset of features; these methods require multiple runs of the classifier to evaluate the quality of the selected features, and they are often computationally expensive [36, 37]. Embedding methods combine the process of feature selection with the process of classifier learning, in which feature selection is performed during the learning process. Embedding methods find a subset of features by the joint minimization of empirical errors and penalties, which can be approximated as a continuous optimization problem. To remove irrelevant and noisy features, the feature selection matrix is usually used for sparse regularization [38]. Filtering methods are independent of the specific learning task and model. Among other advantages, filtering methods are usually more efficient, less computationally expensive, and more general than embedded models; therefore, we focus on filtering methods in this paper. Most existing filtering algorithms for multi-label problems convert the multi-label problem to a single-label problem. Lee et al. [29] proposed a method that converts multiple labels to multiple binary single labels, and then used an evaluation method to evaluate each feature of each label individually to obtain a global feature ranking. However, this approach ignores the inherent correlation in multi-label data and the connection between labels and features. Doquire et al. [39] converted multiple labels to a single label consisting of multiple classes, and then solved the feature selection problem for multi-class single labels. However, this method may dramatically increase the complexity of the feature selection problem.

In contrast to single-label learning, because an instance in multi-label learning corresponds to multiple labels, these labels are often interrelated and interdependent. For example, in a set of instances in which the object is an image, “animal” and “nature” often appear in the same image; in a set of instances in which the object is a document, a document is often associated with multiple topics, such as “politics” and “economics.” However, existing multi-label feature selection methods usually fail to consider label importance or correlation between labels. When we perform feature selection, we can focus on the correlation between tags and use this correlation to better select features [40–42]. Yu et al. [18] constructed a multi-label classification method based on the uncertainty between feature space and label space. Elisseeff and Weston [9] proposed a large-margin ranking system, which shares many properties with support vector machines, to learn the ranks of labels for each instance. This paper proposes

a neighborhood rough set (NRS) model based on label weights and label correlations, which can effectively perform feature selection for multi-label problems.

In most practical applications of multi-label learning, the feature space is usually uncertain, with the same number of samples for each feature arriving in the feature space incrementally, as a flow of feature vectors over time. Such features are known as flow features. For example, on the social networking platform Twitter, trending topics continuously change dynamically over time. When a trending topic appears, it is always accompanied by a fresh set of keywords. These fresh keywords can be used as key features to distinguish trending topics. Multi-tag flow feature selection assumes that features arrive dynamically over time [43] and feature selection is performed as each feature arrives, to maintain an optimal subset of features at all times [44, 45]. Many researchers have attempted to address the challenges posed by flow features. For example, Zhang et al. [46] proposed the use of global features to process flow feature data. Yu et al. [47] conducted a theoretical analysis of the pairwise correlation between features in the currently selected feature subset, and adopted online pairwise comparison techniques to solve the problem of flow features. An online flow feature must satisfy three basic conditions. First, it should not require any prior knowledge to be provided. Second, it should support efficient incremental updates to the selected features. Third, it should be able to make accurate predictions at each update. In this paper, we mainly use NRS to select streaming feature data. The main motivation is that NRS can process mixed types of data without destroying the neighborhood and order structure of the data. In addition, feature selection based on NRS does not require any prior knowledge of the feature space structure, and therefore seems to be an ideal tool for online streaming feature selection.

The main contributions of this paper are the following:

1. A new form of neighborhood granularity is calculated using the average nearest neighbor method and the label weights are calculated using the mutual information method to obtain the label correlation. The neighborhood granularity and label correlation are combined to construct a new NRS relationship and feature importance model.
2. The traditional NRS model is generalized to adapt it to multi-label learning. We propose a static multi-label feature selection algorithm based on the above NRS.
3. We propose a new multi-label flow feature selection algorithm that combines a static multi-label feature selection algorithm with an online importance update framework.

The rest of the paper is organized as follows. Section 2 introduces related concepts, including multi-label learning and NRS. Section 3 presents an NRS model based on label weights and label correlations, including a static algorithm and a dynamic flow feature algorithm. We report our experimental results in Sect. 4 and present our conclusions in Sect. 5.

2 Preliminaries

2.1 Multi-label learning

$NDT = \langle U, F, L \rangle$ is a multi-label decision system. We define $X = R^N$ to represent N-dimensional sample space, which is $U = \{x_1, x_2, \ldots, x_n\} \cdot F = \{f_1, f_2, \ldots, f_m\}$ represents the m-dimensional feature space and $L = \{l_1, l_2, \ldots, l_k\}$ represents the k-dimensional label space. For instance, in the sample space $x_i \in U, x_i = \{F_{i1}, F_{i2}, \ldots, F_{im}\}$ represents a specific m-dimensional feature vector corresponding to the sample x_i, and $y_i \in Y = L$ represents a k-dimensional label vector $y_i = \{y_i^1, y_i^2, \ldots, y_i^k\}$ corresponding to x_i. The task of multi-label learning is to find a mapping $f: X \to Y$: when x_i contains the label l_i, the corresponding value of y_i is 1; otherwise, it is -1. That is, when $y_i^k = 1$, the sample $x_i \in l_k$ is the label category [48].

2.2 Neighborhood rough set

Given a decision system $NDT = \langle U, C, D \rangle, U = \{x_1, x_2, \ldots, x_n\}$ represents a non-empty set of instances, that is, the set composed of all samples, $C = \{a_1, \ldots, a_N\}$ represents the attribute set corresponding to the sample, and D represents the set of decision attributes.

For a given parameter δ and feature set C, the δ-domain relationship on X can be determined. We call the decision system a neighborhood decision system: NDS $=< U, C \cup D, \delta >$.

Definition 1 Given an N-dimensional real space $\Omega, \Delta: R^N \times R^N \to R$, we say that Δ is a metric on R^N if Δ satisfies the following constraints:

(1) $\Delta(x_1, x_2) \geq 0$, if and only when $x_1 = x_2, \forall x_1, x_2 \in R^N$;
(2) $\Delta(x_1, x_2) = \Delta(x_2, x_1), \forall x_1, x_2 \in R^N$;
(3) $\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_2, x_3), \forall x_1, x_2, x_3 \in R^N$

Definition 2 For $\forall x_i \in U$ and a feature subset $B \subseteq C$, we define the δ-neighborhood of x_i based on parameter C as:

$$\delta_B(x_i) = \{x_j \mid x_j \in U, \Delta_B(x_i, x_j) \le \delta\} \tag{1}$$

where $\delta > 0$, by $\Delta_B(x_i, x_j)$, the set of instances of values is granulated. We call $\langle \Omega, \Delta_B \rangle$ the metric space, and $\delta_B(x_i)$ the δ-neighborhood information particle generated by x_i. In this manner, we granulate the neighborhood of all objects in the universal space.

From the neighborhood information particle clusters, $\{\delta(x_i) \mid i = 1, 2, \ldots, n\}$ can lead to a neighborhood relation N on the universal space U. This relation can be represented by a matrix system $M(N) = (r_{ij})_{n \times m}$: if $x_j \in \delta(x_i)$, then $r_{ij} = 1$; otherwise, $r_{ij} = 0$. For neighborhood relations, we have

(1) $\forall x_i \in U : \delta_1(x_i) \subseteq \delta_2(x_i)$
(2) $N_1 \subseteq N_2$

The neighborhood information particle clusters defined in this manner constitute the basic concept system in the universal space.

Definition 3 Given a non-empty finite set $U = \{x_1, x_2, \ldots, x_n\}$ on the actual space and a neighborhood relation N on U, we call the two-tuple $NAS = \langle U, N \rangle$ a neighborhood approximation space.

Definition 4 For a given decision system $NDT = \langle U, C, D \rangle$ and $X \subseteq N$, the lower approximation and upper approximation of X in the neighborhood approximation space $NAS = \langle U, N \rangle$ are defined as [49]

(1) $\underline{N}X = \{x_i \mid \delta(x_i) \subseteq X, x_i \in U\}$
(2) $\overline{N}X = \{x_i \mid \delta(x_i) \cap X \neq \emptyset, x_i \in U\}$

respectively, where $\underline{N}X$ is also referred to as the positive domain of X in the approximation space $NAS = \langle U, N \rangle$, which is the largest union of neighborhood information particles that can be completely contained in X.

Definition 5 For a neighborhood decision system $NDT = \langle U, A, D, \delta \rangle$, D partitions U into N equivalence classes: $X_1, X_2, \ldots, X_N \cdot \forall B \subseteq A$, we define the upper and lower approximations of the decision attribute D with respect to B as

$$\underline{N_B}D = \bigcup_{i=1}^{N} \underline{N_B}X_i \tag{2}$$

$$\overline{N_B}D = \bigcup_{i=1}^{N} \overline{N_B}X_i \tag{3}$$

respectively [50], where $\delta_B(x_i)$ is the informative neighborhood particle generated by attribute B and metric Δ.

The lower approximation of decision attribute D, also called the decision-positive region, is denoted by POS(D).

The size of the positive region reflects the degree to which the classification problem is separable in a given attribute space, with larger positive regions indicating areas of overlap (i.e., fewer boundaries) for each category. We can describe such classification problems in more detail using this set of attributes.

$$\mathrm{POS}(D) = \{x_i \mid \delta_B(x_i) \subseteq D, x_i \in U\} \tag{4}$$

Definition 6 Suppose that A, B are two sets; we define the degree to which A is contained in B, $I(A, B)$, as follows [51].

$$I(A, B) = \frac{\mathrm{Card}(A \cap B)}{\mathrm{Card}(A)} \tag{5}$$

When $A = \emptyset$ or $B = \emptyset$, we define $I(A, B) = 0 \cdot I(A, B)$ reflects the importance of B to A.

The dependency of decision attribute D on condition attribute B is defined as follows [52]:

$$\gamma_B(D) = \mathrm{Card}\left(\underline{N_B}D\right)/\mathrm{Card}(U) \tag{6}$$

where $\gamma_B(D)$ denotes the proportion of samples in the sample set that can be included by a decision according to the description of condition attribute B.

The positive region of the decision is larger if the decision attribute D is more dependent on the condition attribute B.

3 Proposed method

3.1 Improvements to neighborhood particles based on average nearest neighbors

Given a decision system $NDT = \langle U, C, D \rangle$, $U = \{x_1, x_2, \ldots, x_n\}$ represents a non-empty set of instances, C represents the feature set corresponding to the instance set, and D represents the decision attribute set. The traditional single-label method for neighborhood information particle division is unsuitable for multi-label data. For general data, a group of instances with the same attribute value or label value is called an equivalence class. Similarly, for mixed data, a group of instances with similar attribute values or label values is called a neighborhood class. In this paper, the margin of particles in the sample is used for granulating the neighborhood size.

Definition 7 Given a sample x, the margin of x relative to a set of samples U is defined as follows:

$$m(x) = \Delta(x, NS(x)) - \Delta(x, NT(x)) \tag{7}$$

where $NS(x)$ denotes the instance from U that has the shortest distance from x and whose label class is different from that of x and $NT(x)$ denotes the instance from U that has the shortest distance from x and has the same label class as x; we call these instances the nearest miss and the nearest hit, respectively. $\Delta(x, NS(x))$ denotes the distance between x and $NS(x)$, and $\Delta(x, NT(x))$ denotes the distance between x and $NT(x)$. We call $\delta(x) = \{y \mid \Delta(x, y) \le m(x)\}$ the neighborhood particle about x. To facilitate the setting of neighborhood information particles, we set $m(x) = 0$ when $m(x) < 0$.

A sample may have a positive or negative effect on different labels. Thus, for a given sample, the degree of granularity may depend on the label used.

Definition 8 For a sample x and label $l_k \in L$, the margin of x with respect to l_k is

$$m_{l_k}(x) = \Delta_{l_k}\left(x, NS_{l_k}(x)\right) - \Delta_{l_k}\left(x, NT_{l_k}(x)\right), l_k \in L \tag{8}$$

As noted above, each sample has a different label and correspondingly a different granularity. Depending on the different decision views, we need to combine all the single-label granularities of a given sample to form a multi-label granularity [53]. Therefore, in this paper, we choose the average granularity (i.e., the average nearest neighborhood, also known as the neutral view) to represent the multi-label granularity of a sample [54].

$$m^{neu}(x) = \frac{1}{L}\sum_{i=1}^{L} m_{l_k}(x) \tag{9}$$

To solve the problem of the granularity selection of δ, combining Eqs. 1 and 9, the new neighborhood of the sample is defined as

$$\delta_B(x_i) = \left\{x_j \mid x_j \in U, \Delta_B(x_i, x_j) \le m^{neu}(x_i)\right\} \tag{10}$$

We have defined a new neighborhood information particle to solve the problem of selecting the neighborhood granularity, which is caused by multi-label data. In addition, the average nearest neighbor reflects the relationship between features in an instance. This new neighborhood model considers the relationships between features and is based on improved neighborhood information.

3.2 Label correlation

Definition 9 In the neighborhood decision system $NDS = \langle U, C \cup D, \delta \rangle$, for any instance $x_i, y_i = \left\{y_i^1, y_i^2, \ldots, y_i^m\right\}$ is its corresponding label vector, and l_j is a label in the label space $L = D$. When x_i belongs to category l_j, the corresponding value of y_i^j is 1. We define $D^j = \left\{x_i \mid \forall x_i \in U, y_i^j = 1\right\}$, that is, the set of all instances in U that belong to category l_j. Through the definition of multi-label decision space, we can expand the decision-positive region of single-label decision making, using Eq. 4. For a certain feature subset $B \subseteq C$, the lower approximation of the decision l_j about B is

$$\mathrm{POS}\left(D^j\right)' = \underline{N_B} D^{j'} = \left\{x_i \mid \delta_B(x_i) \subseteq D^j, x_i \in U\right\} \tag{11}$$

Multi-label data differs from single-label data in that it is necessary to consider the importance of the labels and the correlation between them because the labels of each instance are always somehow related.

Definition 10 For a sample x_i and the corresponding feature vector Y_i, that is, $D = \left\{\left(x_i, Y_i\right) \mid 1 \le i \le N, x_i \in U, Y_i \in L\right\}, N$ is the number of instances in the training set and $l_i, l_j \in L (1 \le i, j \le k)$ are any two labels in the label space L. The correlation between l_i and l_j is calculated by mutual information:

$$MI\left(l_i, l_j\right) = \sum_{k=1}^{M}\sum_{q=1}^{M} P\left(l_{ik}, l_{jq}\right) \log \frac{P\left(l_{ik} \mid l_{jq}\right)}{P\left(l_{jq}\right)} \tag{12}$$

A labeled undirected graph (WUG) = (V, E, W) can be constructed by applying Eq. 12. $V = L = \left\{l_1, l_2, \ldots, l_m\right\}$ represents the set of nodes of the undirected graph, $E = \left\{\left(l_i, l_j\right) \mid l_i, l_j \in L\right\}$ represents its set of edges, and $w\left(l_i, l_j\right) = MI\left(l_i, l_j\right)$ represents the weight of each edge [55]. The importance of each node in this undirected graph is defined as follows:

$$LW\left(l_i\right) = (1 - d) + d \sum_{l_j \in SN(l_i)} \frac{LW\left(l_j\right) w\left(l_i, l_j\right)}{SW\left(l_j\right)} \tag{13}$$

$$SW\left(l_j\right) = \sum_{l_j} w\left(l_i, l_j\right) \tag{14}$$

$LW\left(l_i\right)$ and $LW\left(l_j\right)$ represent the weight divisions of nodes l_i and l_j, respectively. $SN\left(l_i\right)$ is the set of nodes with edges

to label l_i, and $w(l_i, l_j) = MI(l_i, l_j)$ represents the correlation between nodes. Equation 10 is used to calculate $SN(l_i)$, which denotes the sum for the correlation for all edges starting from $l_j \cdot d$ is the damping coefficient, for which it is recommended to use $d = 0.85$ citerk58. For ease of calculation, an initial weight value can be set for all nodes; this is usually $1/L$, where L is the total number of nodes, that is, the total number of labels [56]. Using this algorithm, we can calculate the correlation between node l_i (i.e., label l_i) and other nodes l_j related to it, as well as the structure of the graph (WUG). Through label correlation, we obtain the weight of each label in the label space and we complete the exploration of label correlation.

3.3 Feature selection based on neighborhood rough sets

The multi-label domain decision system $NDS = \langle U, C \cup D, \delta \rangle$ is handled in a similar manner to the single-label decision system. By extending the rough set importance theory for multi-label data (Eq. 6) and combining the multi-label neighborhood particles (Eq. 10) and label correlation (Eq. 13), we obtain the importance of the feature subset $B(B \subseteq C)$ for the decision attribute set $D = L = \{l_1, l_2, \ldots, l_m\}$:

$$\gamma_B(D)' = \sum_{l_j \in L}^{m} \frac{\mathrm{Card}\left(\mathrm{POS}\left(D^j\right)'\right) LW\left(l_j\right)}{\mathrm{Card}(U)} \tag{15}$$

The above equation reflects the importance of the decision-positive region and the corresponding decision attributes of the feature subset B. It solves the problems of granularity selection and feature association for multi-label NRS.

According to Eq. 15, in the neighborhood decision system $NDS = \langle U, C \cup D, \delta \rangle, B \subseteq C$ is a feature subset, $a \in C - B$, and the degree of importance of a to B is defined as follows:

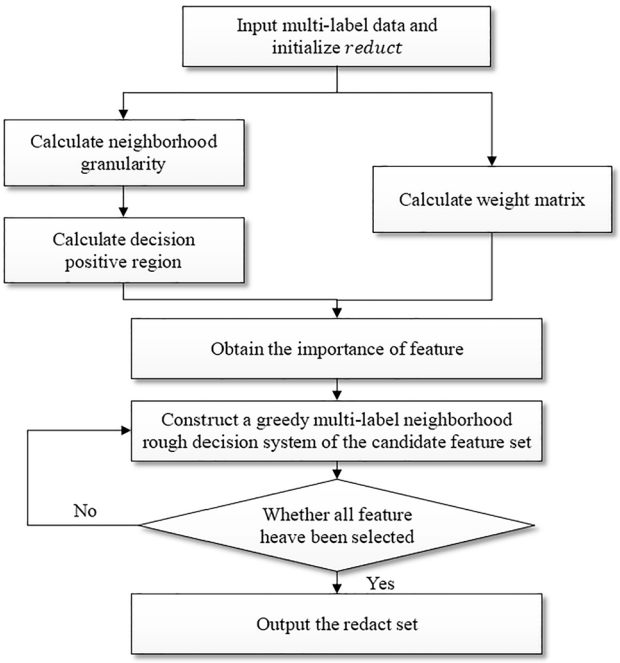

Fig. 1 Framework of static multi-label feature selection

$$\mathrm{SIG}(a, B, D)' = \gamma_{B \cup a}(D)' - \gamma_B(D)' \tag{16}$$

In the new importance model, we have added label importance and label relevance to the NRS model. The new NRS model reflects the fusion of feature information and label correlation.

For the above NRS model, we construct a greedy forward search-based multi-label feature selection algorithm. To illustrate the proposed algorithm more clearly, the framework of the algorithm is presented in Fig. 1.

According to the framework shown in Fig. 1, our proposed forward greedy [57] multi-label feature selection algorithm behaves as follows. The final reduced reduct is the best subset after feature selection on the feature space.

Algorithm 1 Static Multilabel Feature Selection Algorithm based on Label Importance and Label Correlation (SMFS-LILC).

Input: Neighbourhood decision system $\langle U, C \cup D \rangle$.
Output: $reduct$: the final reduced selected feature set.
1 Calculate the weight matrix $LW(L)$ using Formula (13);
2 Initialize $reduct \leftarrow \varnothing$;
3 **if** $reduct = \varnothing$ **then**
4 | let $POS(D)^{'} = 0$;
5 **end**
6 **if** $reduct \neq \varnothing$ **then**
7 | $\forall x_i \in U$, calculate the average approximate neighbors $d_{red}(x_i)$ under a using (9);
8 | $\forall x_i \in U$, calculate the neighborhood $\delta_{red}(x_i)$ using (10);
9 | $\forall$ label l_j of x_i, calculate $POS\left(D^j\right)^{'}$ using (11);
10 | Calculate $\gamma_{red}(D)^{'}$ using (15);
11 **end**
12 $\forall\ a\ \epsilon\ C - reduct$, repeat steps 7-10 and calculate $\gamma_{red \cup a}(D)^{'}$;
13 Calculate $SIG(a, red, D)^{'} = \gamma_{red \cup a}(D)^{'} - \gamma_a(D)^{'}$;
14 $Set\ a_k, SIG(a_k, red, D)' = \max\left(SIG(a, red, D)'\right)$
15 **if** $SIG(a_k, red, D)^{'} > 0$ **then**
16 **end**
17 $reduct \cup a_k \rightarrow reduct$;
18 go to step 12;
19 **else**
20 | Return reduct;
21 **end**
22 output reduced $reduct$.

In Algorithm 1, steps 1–5 perform the preparation work when multiple items of labeled data arrive. Our reduct set reduct starts from the empty set and calculates the label weights *LW*(*L*) of the entire label space. This step requires the traversal of the entire label space and the construction of an undirected graph. Assuming that the number of labels in the label space is *L*, the time complexity of the calculation of the correlation between each pair of labels is $\mathrm{O}\left(|\ L\ |^2\right)$, and that of the calculation of each label weight is $O(1)$. Therefore, the time complexity of steps 1–5 is $O\left(|\ L\ |^2 + 1\right) = O\left(|\ L\ |^2\right)$. Steps 6–21 are divided into two parts: calculating the neighborhood of the instance and analyzing whether the instance and the neighborhood are important. First, by selecting the average approximation neighborhood as the domain granularity standard (step 6–12), this step requires searching for the nearest hit or miss for each instance; assuming that the instance space is *U*, the time complexity of this step is $\mathrm{O}\left(|\ U\ |^2\right)$. Next, the neighborhood corresponding to the instance is determined and the decision-positive region and attribute importance are calculated (step 13–21). The time complexity for determining the neighborhood of each instance is $\mathrm{O}(n \log n)$, and the time complexities of the calculations of the decision-positive region and importance are both $O(1)$, so the overall time complexity of the calculation of the instance domain is $\mathrm{O}\left(|\ U\ |^2 + |\ U\ | \log |\ U\ | + 1 + 1\right) = \mathrm{O}\left(|\ U\ |^2\right)$. The time complexity for determining whether the samples in the instance neighborhood are consistent is $O(n)$ and, if the number of features in the feature space is *C*, the time complexity of steps 6–21 is $\mathrm{O}\left(|\ C\ ||\ U\ |^2\right)$. Therefore, the time complexity of Algorithm 1 is $\mathrm{O}\left(|\ L\ |^2 + |\ C\ ||\ U\ |^2\right)$.

3.4 Dynamic multi-label feature selection algorithm based on label importance and label correlation

Algorithm 1, similarly to most feature selection algorithms, assumes that all candidate features are available to the algorithm before feature selection. In contrast, with flow features, all features cannot be collected before learning starts because they arrive, dynamically and incrementally, over time. Therefore, we propose an online multi-label flow feature selection algorithm based on Algorithm 1 combined

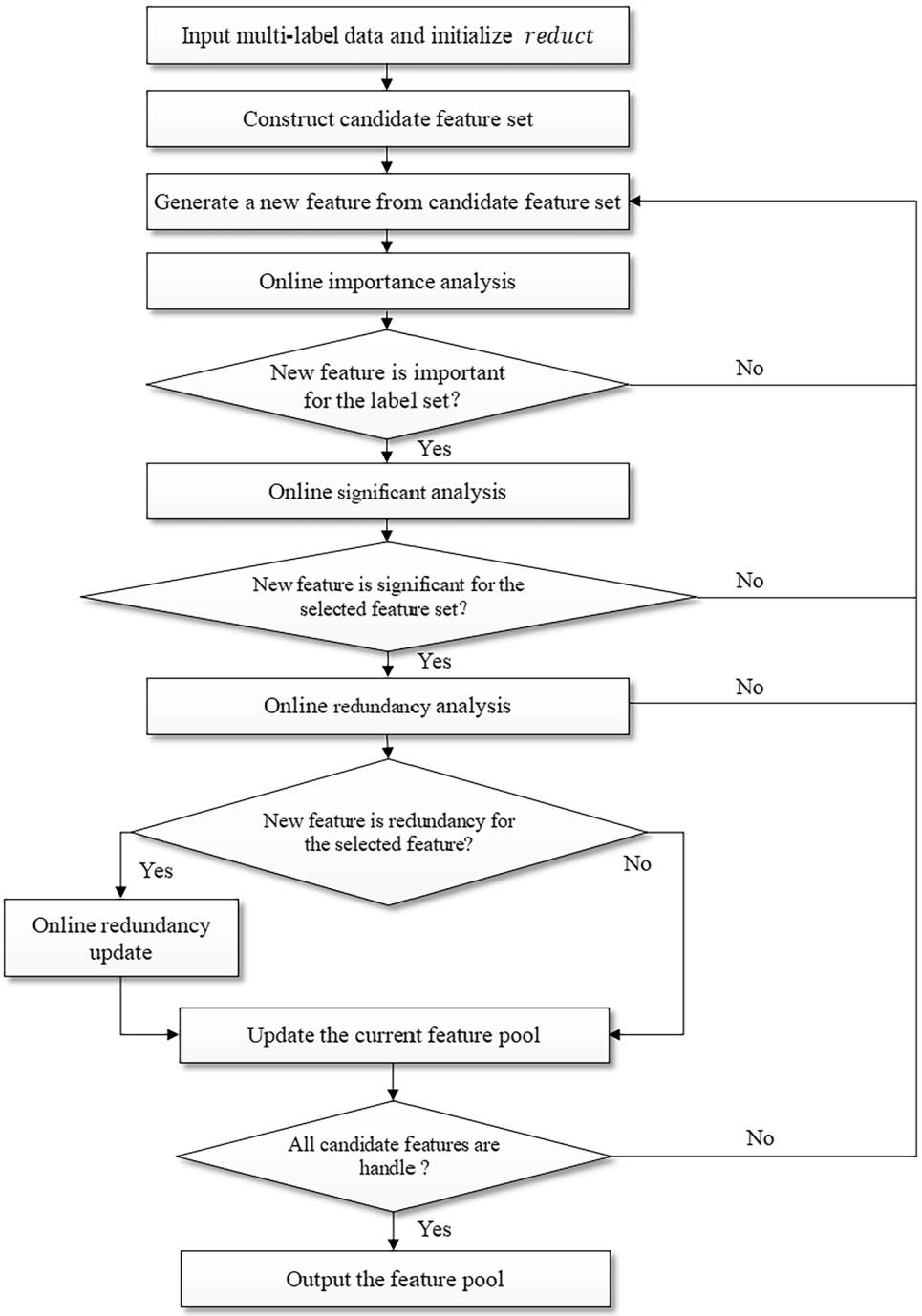

Fig. 2 Flow feature selection framework

with the online flow feature selection framework [51], to solve the multi-label flow feature selection problem.

In the multi-label flow feature decision system $NFDS = \langle U, C \cup L, t \rangle$, $U = \{x_1, x_2, \ldots, x_n\}$ represents a series of non-empty sample sets, C represents the feature set corresponding to the sample, L represents the label set, and t represents the arrival time of the flow feature. F_t denotes the newly arrived feature at time t, and S_{t-1} denotes the reduced feature subset reduct at time t.

3.4.1 Importance analysis

For a newly arrived feature F_t, the first step is to perform importance analysis on F_t. The purpose of importance analysis is to evaluate whether F_t is beneficial to the label set L, that is, to evaluate the importance of F_t to the whole label set L. We define a parameter δ to assess the importance of F_t and use Eq. 15 to calculate the importance $\gamma_{F_t}(D)'$ of F_t to the entire label set. If $\gamma_{F_t}(D)' < \delta$, we consider F_t to be unimportant to the label set L, so F_t is discarded.

3.4.2 Significance analysis

After the above importance analysis, we believe that F_t is important to the label set L. However, we also need to consider the relationship between F_t and the current reduced feature set S_{t-1}. The purpose of significance analysis is to evaluate the relevance of the newly arrived feature F_t to the subset of features at time t, that is, to check whether F_t is significant in the current feature subset. F_t is compared with the average value Avg_γ of the importance of each feature in the current feature subset S_{t-1}

Here we use the iterative method to calculate the average Avg_γ:

$$\mathrm{Avg}_\gamma = \mathrm{Avg}_{\gamma-1} + \frac{\gamma_{F_i}(D)' - \mathrm{Avg}_{\gamma-1}}{\mid F_i \mid} \tag{17}$$

where $\mathrm{Avg}_1 = \gamma_{F_1}(D)'$.

If $\gamma_{F_t}(D)' \geq \mathrm{Avg}_\gamma$, the importance of the new feature F_t to the label set L is greater than or equal to the average importance of the already achieved features in S_{t-1}. Therefore, we consider F_t to be a significant feature, which should be preserved.

3.4.3 Redundancy analysis

After the above significance analysis, we already know that F_t is beneficial to the current S_{t-1}. However, we also need to analyze the relationship between F_t and the features in S_{t-1}. The purpose of redundancy analysis is to compare the contributions of features F_k and F_t to S_{t-1} in the current reduction set S_{t-1}. When the contributions of two features are the same, they are repeated, and one of them must be discarded.

For two features F_t and F_k, if $\mathrm{SIG}(F_t, S_{t-1}, D)' = \mathrm{SIG}(F_k, S_{t-1}, D)'$, F_t and F_k have the same degree of contribution to S_{t-1}. Therefore, we compare $\gamma_{F_k}(D)'$ and $\gamma_{F_t}(D)$. If $\gamma_{F_k}(D)' \geq \gamma_{F_t}(D)'$, we preserve F_k and discard F_t; if $\gamma_{F_k}(D)' < \gamma_{F_t}(D)'$, we preserve F_t and discard F_k.

The flow feature selection framework, illustrated in Fig. 2, is based on online importance analysis, significance analysis, and redundancy analysis. In this framework, a training set with known feature sizes is used to simulate flow features, and each flow feature is generated from the candidate feature set. In the framework shown in Fig. 2, we propose a dynamic multi-label feature selection algorithm that considers label importance and label correlation (Algorithm 2), which incorporates the above three types of analysis.

Table 1 Dataset introduction

Dataset	Instances	Features	Labels	Training	Test	Card	Density
Arts	5000	462	26	2000	3000	1.6360	0.0629
Birds	645	260	20	322	323	1.470	0.074
Business	5000	438	30	2000	3000	1.588	0.053
CAL500	502	68	174	251	251	26.044	0.150
Computer	5000	681	33	2000	3000	1.509	0.046
Emotion	593	72	6	391	202	1.869	0.311
Health	5000	612	32	2000	3000	1.662	0.052
Scene	2317	294	6	1211	1196	1.074	0.179
Yeast	2417	103	14	1499	918	4.238	0.303

Algorithm 2 Dynamic Multi-label Feature Selection Algorithm Based on Label Importance and Label Correlation (DMFS-LILC).

Input: Multi-label flow feature decision system $NFDS = \langle U, C \cup L, t \rangle$, F_t: newly arrived feature at time t; S_{t-1}: reduced feature subset at time t; δ importance weight.
Output: $reduct$: final reduced selected feature set.

Calculate the weight matrix $LW(L)$ using (13);
Initialize $reduct \leftarrow \varnothing, Mean_\gamma = 0$;
$F_t \leftarrow$ flow in a new feature, calculate $\gamma_{F_t}(D)^{'}$ using (15);
/* Importance Analysis*/
if $red = \varnothing$ **then**
  if $\gamma_{F_t}(D)^{'} < \delta$ then discard F_t and go to step 30;
  else $reduct = reduct \cup F_t$ and go to step 30;
end
/*Significance Analysis*
else
  calculate $Mean_\gamma$ using (17);
end
if $\gamma_{F_t}(D)^{'} > Mean_\gamma$ **then**
  $reduct = reduct \cup F_t$ and go to step 30;
end
else
  if $\gamma_{F_t}(D)^{'} < Mean_\gamma$, discard F_t and go to step 30; **then**
    else
      /* Redundancy Analysis */
      while $\exists F_k \in reduct$, **do**
        if $\gamma_{F_k}(D)^{'} = \gamma_{F_t}(D)^{'}$ *and* $SIG(F_t, S_{t-1} - F_k, D)^{'} < SIG(F_k, S_{t-1}, D)^{'}$,
        then discard F_k, and go to step 30;
        else if $\gamma_{F_k}(D)^{'} = \gamma_{F_t}(D)^{'}$ *and* $SIG(F_t, S_{t-1} - F_k, D)^{'} \geq SIG(F_k, S_{t-1}, D)^{'}$,
        then
        $reduct = reduct - F_k$;
        $reduct = reduct \cup F_t$ and go to step 30;
      end
    end
  end
end
Repeat step 4 until no new feature F_t is available;
Output: $reduct$.

Table 2 Comparison of average precision (↑) of eight feature selection algorithms

Datasets	MDDMspc	MDDMproj	PMU	RF_ML	NRPS	MFSF	SMFS-LILC	DMFS-LILC
ARTS	0.5003	0.4849	0.4955	0.4862	0.5062	0.5109	0.5082	**0.5147**
Birds	0.5818	0.5821	0.6894	0.6559	0.6785	0.6834	0.6886	**0.6987**
Business	0.8702	0.8698	0.8721	0.8729	0.8758	0.8736	0.8719	**0.8855**
Cal500	0.4791	0.4791	0.4779	0.4792	0.4826	0.4920	0.4989	**0.4989**
Computer	0.6347	0.6225	0.6312	0.6285	0.6485	0.6495	0.6446	**0.6585**
Emotion	0.7730	0.7300	0.7346	0.7553	0.7786	0.7786	0.7814	**0.8002**
Health	0.6607	0.6653	0.6797	0.6699	0.6981	0.6880	0.6938	**0.6957**
Scene	0.7336	0.7255	0.7899	0.7674	0.8037	0.8062	0.8356	**0.8378**
Yeast	0.7278	0.7084	0.7478	0.7432	0.7519	0.7551	**0.7607**	0.7581
AVERAGE	0.6624	0.6520	0.6798	0.6732	0.6915	0.6930	0.6982	**0.7053**

Table 3 Comparison of ranking loss (↓) of eight feature selection algorithms

Datasets	MDDMspc	MDDMproj	PMU	RF_ML	NRPS	MFSF	SMFS-LILC	DMFS-LILC
Arts	0.1552	0.1588	0.1546	0.1538	0.1525	0.1530	0.1519	**0.1474**
Birds	0.1613	0.1666	0.1426	0.1503	0.1465	0.1321	0.1302	**0.1267**
Business	0.0416	0.0419	0.0405	0.0420	0.0401	0.0408	0.0405	**0.0398**
CAL500	0.1918	0.1918	0.1910	0.1902	0.1896	0.1903	0.1854	**0.1839**
Computer	0.0934	0.0962	0.0946	0.0921	0.0890	0.0890	0.0917	**0.0898**
Emotion	0.2130	0.2315	0.2164	0.1866	0.1834	0.1794	0.1726	**0.1686**
Health	0.0685	0.0671	0.0659	0.0633	0.0631	**0.0621**	0.0635	0.0627
Scene	0.1084	0.1190	0.1104	0.1042	0.1036	0.1021	**0.0976**	**0.0976**
Yeast	0.1830	0.1938	0.1811	0.1828	0.1794	0.1756	**0.1705**	0.1733
AVERAGE	0.1351	0.1407	0.1330	0.1295	0.1275	0.1249	0.1227	**0.1211**

Table 4 Comparison of coverage (↓) of eight feature selection algorithms

Datasets	MDDMspc	MDDMproj	PMU	RF_ML	NRPS	MFSF	SMFS-LILC	DMFS-LILC
Arts	0.1552	0.1588	0.1546	0.1538	0.1525	0.1530	0.1519	**0.1474**
Birds	0.1613	0.1666	0.1426	0.1503	0.1465	0.1321	0.1302	**0.1267**
Business	0.0416	0.0419	0.0405	0.0420	0.0401	0.0408	0.0405	**0.0398**
CAL500	0.1918	0.1918	0.1910	0.1902	0.1896	0.1903	0.1854	**0.1839**
Computer	0.0934	0.0962	0.0946	0.0921	0.0890	0.0890	0.0917	**0.0898**
Emotion	0.2130	0.2315	0.2164	0.1866	0.1834	0.1794	0.1726	**0.1686**
Health	0.0685	0.0671	0.0659	0.0633	0.0631	**0.0621**	0.0635	0.0627
Scene	0.1084	0.1190	0.1104	0.1042	0.1036	0.1021	**0.0976**	**0.0976**
Yeast	0.1830	0.1938	0.1811	0.1828	0.1794	0.1756	**0.1705**	0.1733
AVERAGE	0.1351	0.1407	0.1330	0.1295	0.1275	0.1249	0.1227	**0.1211**

The main computation performed by Algorithm 2 is the computation of dependencies between features. At time t, S_{t-1} is the number of features in the currently selected feature set. Algorithm 2 assesses whether the new feature F_t, arriving at time t, needs to be retained and decides how to retain it. The entire process is an online selection problem that comprises three main parts: importance analysis, significance analysis, and redundancy analysis, which are marked in Algorithm 2. The feature calculation performed by the algorithm is taken from Algorithm 1, and the time complexity of the selection of a single feature is $O(|U| \log |U|)$. In the best case, online selection can obtain the best subset immediately, so the time complexity is $O(|L|^2 + |L||U| \log |U|)$. However, in most cases, Algorithm 2 is neither simple nor optimistic, and it needs to be updated online for S_t Because the time complexity of the S_t update depends on the calculation of feature dependencies, in the worst case it is necessary to go through all selected features to process F_t, and therefore the worst-case time complexity is $O(|L|^2 + |S_{t-1}||L||U| \log |U|)$.

Table 5 Comparison of one-error (↓) of eight feature selection algorithms

Datasets	MDDMspc	MDDMproj	PMU	RF_ML	NRPS	MFSF	SMFS-LILC	DMFS-LILC
Arts	0.6635	0.6761	0.6484	0.6757	0.6588	0.6573	0.6355	**0.6187**
Birds	0.5511	0.6009	0.4619	0.4728	0.4338	0.4419	0.4365	**0.4025**
Business	0.1302	0.1307	0.1256	0.1280	0.1247	0.1263	0.1225	**0.1196**
Cal500	0.1474	0.1474	0.1195	0.1195	0.1172	0.1160	0.1076	**0.1076**
Computer	0.4574	0.4624	0.4460	0.4454	0.4453	0.4431	0.4320	**0.4202**
Emotion	0.3619	0.3700	0.3732	0.3453	0.3218	0.3218	0.3218	**0.3020**
Health	0.4378	0.4254	0.4193	0.4336	0.4013	**0.3947**	0.4013	0.4080
Scene	0.3067	0.2988	0.3928	0.3015	0.2736	**0.2625**	0.2642	0.2642
Yeast	0.2558	0.2534	0.2436	0.2481	**0.2366**	0.2460	0.2386	**0.2366**
AVERAGE	0.3680	0.3739	0.3589	0.3522	0.3348	0.3344	0.3289	**0.3199**

Table 6 Comparison of Hamming loss (↓) of eight feature selection algorithms

Datasets	MDDMspc	MDDMproj	PMU	RF_ML	NRPS	MFSF	SMFS-LILC	DMFS-LILC
Arts	0.0616	0.0622	0.0615	0.0627	0.0612	0.0623	0.0602	**0.0584**
Birds	0.0632	0.0637	0.0587	0.0607	0.0605	0.0657	0.0611	**0.0576**
Business	0.0409	0.0410	0.0272	0.0342	**0.0267**	0.0336	0.0277	0.0273
Cal500	0.1376	0.1376	0.1334	0.1366	0.1226	0.1229	0.1004	**0.1004**
Computer	0.0415	0.0413	0.0405	0.0415	0.0402	0.0401	0.0407	**0.0388**
Emotion	0.2480	0.2517	0.2421	0.2426	0.2476	0.2476	0.2324	**0.2294**
Health	0.0455	0.0441	0.0440	0.0463	0.0446	0.0443	0.0422	**0.0415**
Scene	0.1347	0.1384	0.1285	0.1400	0.1219	0.1184	0.1231	**0.1074**
Yeast	0.2170	0.2179	0.2075	0.2064	0.2104	0.2128	**0.1983**	0.2004
AVERAGE	0.1100	0.1109	0.1048	0.1079	0.1040	0.1053	0.0985	**0.0957**

4 Experiment

4.1 Datasets and experimental design

To validate the performance of our proposed algorithms, we used nine benchmark datasets from various application domains as our experimental data [27, 58]. The Arts, Business, Computer, Health, and Scene datasets are all from Yahoo and are widely used for web text classification. The Birds dataset identifies classes of birds by recordings of their calls. It contains 645 sound samples, 260 features extracted from the sound recordings, and 20 labels. (One of the samples, with nonexistent labels, represents background noise.) The Cal500 dataset is a dataset of 500 English songs. The Emotions dataset is also a music dataset, which consists of 593 music samples, each belonging to one of six classes. The Yeast dataset is used to predict functional classes of yeast genes and consists of 2417 samples, each representing a gene and 14 actionable tags. Table 1 shows standard statistics for the nine multi-label datasets: the number of samples, number of features, number of labels, number of samples in the training set, number of samples in the test set, label cardinality, and label density.

In our experiments, we compared our proposed algorithms with several multi-label feature selection algorithms, including MDDM, PMU, RF-ML, NRPS [59], and MFSF [60], all of which reflect the effectiveness of feature selection from different perspectives.

The experiments used five evaluation criteria, namely average precision (AP), ranking loss (RL), coverage (CV), one-error (OE), and Hamming loss (HL), to evaluate the performance of all multi-label feature selection algorithms [61]. These five criteria were designed to evaluate performance from different perspectives, and there are usually several algorithms that achieve the best performance with respect to all these criteria at the same time. Finally, the performance of all algorithms was evaluated using the MLKNN ($K = 10$) classifier [62].

Because each sample of the multi-label data corresponds to a set of labels, the evaluation method for multi-label data is more complicated than that for traditional single-label data. The set $T = \{(x_i, y_i) \mid 1 \le i \le N\}$ represents a given test set, where $y_i \subseteq L$ is the correct label subset and $Y_i' \subseteq L$ represents the binary label vector predicted by the multi-label classification algorithm.

Average precision (AP): AP is the average fraction of labels ranked higher than a specific label $\gamma \in y_i$. A larger

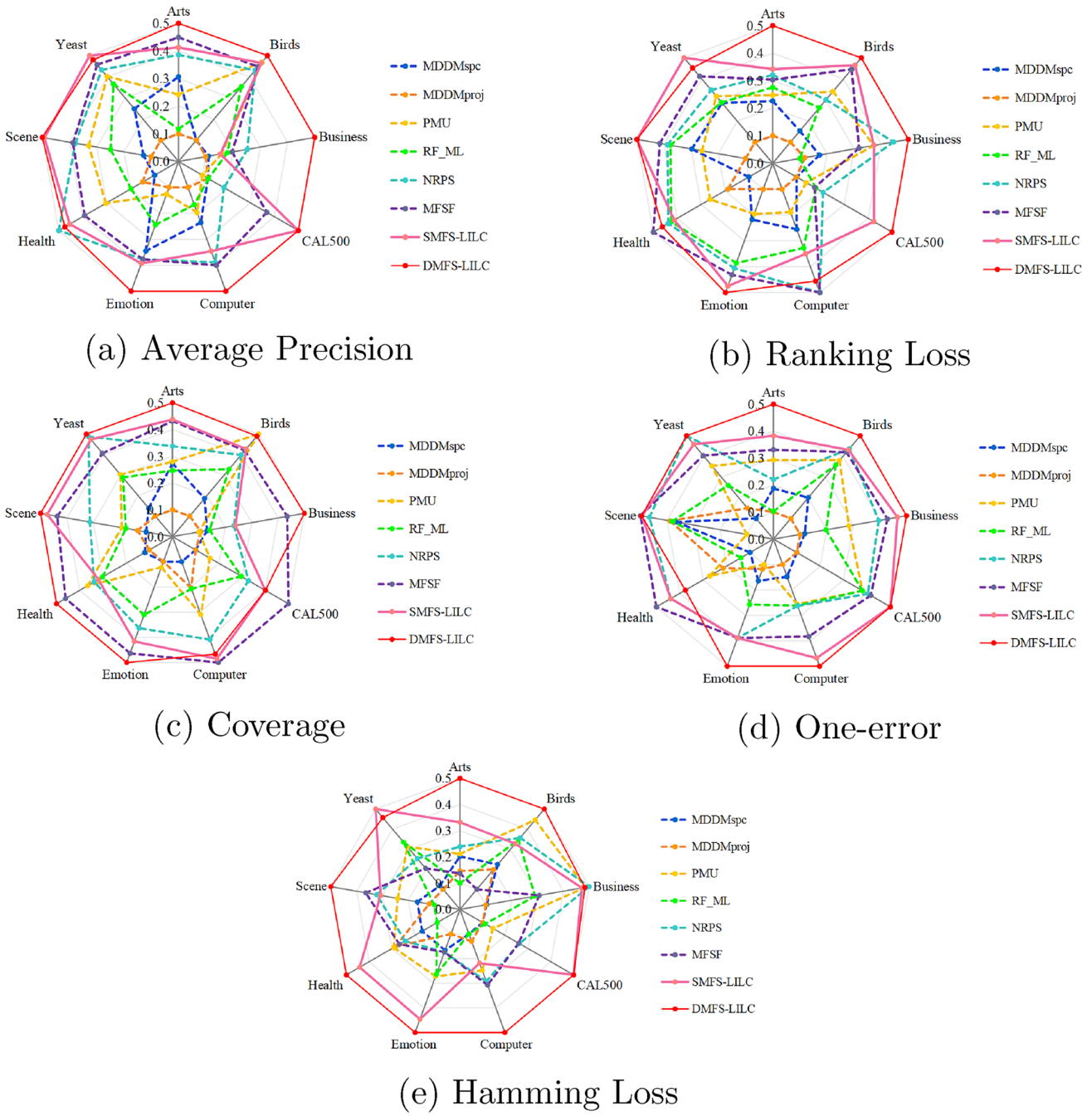

Fig. 3 Spider web diagrams for stability analysis

value of AP corresponds to a better prediction performance of the whole classifier.

$$AP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\,y_i\,|}\sum_{\gamma\in y_i}\frac{|\,\{\gamma' \in y_i : r_i(\gamma') \le r_i(\gamma)\}\,|}{r_i(\gamma)} \tag{18}$$

where $r_i(\gamma)$ denotes the rank of the corresponding label $l \in L$ after the given sample x_i is predicted by the learning algorithm.

Hamming loss (HL): HL indicates the number of times a sample-label instance is misclassified.

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{|\,Y_i' \oplus y_i\,|}{M} \tag{19}$$

where $\oplus$ denotes the XOR operation; a smaller value of HL corresponds to a better result.

Ranking loss (RL): RL indicates how many irrelevant tags are ranked higher than relevant tags. RL is the average probability of an item that is not in the set of relevant labels being ranked (in the resultant ranking) among items that are in the set of relevant labels.

$$RL = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\ y_i\ ||\ \bar{y}_l\ |}\ |\ \left\{\left(\lambda_1,\lambda_2\right)\ |\ \lambda_1 \le \lambda_2, \left(\lambda_1,\lambda_2\right) \in y_i \times \overline{y_l}\right\} \tag{20}$$

where λ_i denotes the real-valued likelihood between the label value of x_i and each $l_i \in L$ after classification by the multi-label classifier, and $\overline{y_l}$ denotes the complementary set of y_i. A smaller value of RL corresponds to a better result.

Coverage (CV): CV evaluates how many steps are needed, on average, to traverse the list of labels in such a manner that all the ground-truth labels of the instance are covered.

$$CV = \frac{1}{N}\sum_{i=1}^{N}\max_{\lambda\in y_i}\text{rank}(\lambda) - 1 \tag{21}$$

where rank(λ) denotes the rank of λ. If $\lambda_1 > \lambda_2$, then $\text{rank}\left(\lambda_1\right) < \text{rank}\left(\lambda_2\right)$. A smaller value of CV corresponds to a better result.

One-error (OE): OE is the probability that the label ranked first in the output result does not belong to the actual label set.

$$QE = \frac{1}{N}\sum_{i=1}^{N}\left[\left[\text{argmax}_{y_i\subseteq L} f\left(x_i,y_i\right)\right] \notin Y_i'\right] \tag{22}$$

where $[|\ \pi\ |] = \begin{cases} 1, \pi \text{ is true} \\ 0, \pi \text{ is false} \end{cases}$. A smaller value of OE corresponds to a better result.

Of these evaluation criteria, AP, CV, OE, and RL focus on the label ranking performance of each instance, whereas HL focuses on the label set prediction performance of each instance.

Table 7 Friedman test ($k = 8, N = 9$) summary of F_F value and critical value of each evaluation criterion on $\alpha = 0.10$

Evaluation metric	F_F	Critical value ($\alpha = 0.10$)
Average precision	21.567	1.82
Coverage	23.4273	
Hamming loss	11.7293	
One-error	14.716	
Ranking loss	25.0716	

4.2 Experimental results

4.2.1 Evaluation of predictive performance of algorithms

We compared the two proposed algorithms-the static multi-label feature selection algorithm (SMFS-LILC) and dynamic multi-label feature selection algorithm (DMFS-LILC)-with MDDMproj, MDDMspc, PMU, RF-ML, NRPS, and MFSF with respect to predictive classification performance. The first four of these are widely used multi-label classification algorithms, and the last two are multi-label feature selection algorithms proposed in the past two years that combine NRS with flow features. To ensure comparable results, the features obtained by all algorithms were ranked, and the final feature subset of all algorithms contained the same number of features as the final feature subset of DMFS-LILC. Because all algorithms in the comparison use the results of feature selection as the result of feature ranking, we present in Tables 2, 3, 4, 5 and 6 the detailed experimental results for all algorithms on each classification dataset. Each evaluation criterion is labeled by "↓" to mean "smaller is better" or "↑" to mean "larger is better". In addition, the best predictive classification performance, with respect to each evaluation criterion, is shown in **bold**, the second-best performance is underlined, and the average performance of each algorithm is shown in *italics*.

The experimental results, shown in Tables 2, 3, 4, 5 and 6, are as follows:

(1) With respect to AP, DMFS-LILC outperformed the existing algorithms on all seven datasets, whereas SMFS-LILC achieved suboptimal performance on five datasets. The two proposed algorithms achieved good performance on all multi-label datasets in the experiment.
(2) With respect to RL, OE, and HL, DMFS-LILC achieved the best performance on six multi-label datasets. In addition, it achieved second-best or close to second-best performance on the remaining datasets. With respect to RL and HL, the predictive classification performance of DMFS-LILC was also very close to the optimal performance of another existing algorithm on the multi-label datasets. The performance achieved by DMFS-LILC was close to the optimal performance. In contrast, SMFS-LILC achieved suboptimal performance on five datasets and optimal performance on two datasets, with respect to RL. In particular, with respect to OE, SMFS-LILC achieved suboptimal performance on all datasets.
(3) With respect to CV, DMFS-LILC significantly outperformed all existing algorithms on at least five multi-label datasets. Although SMFS-LILC performed worse than MFSF and (on some datasets) DMFS-LILC per-

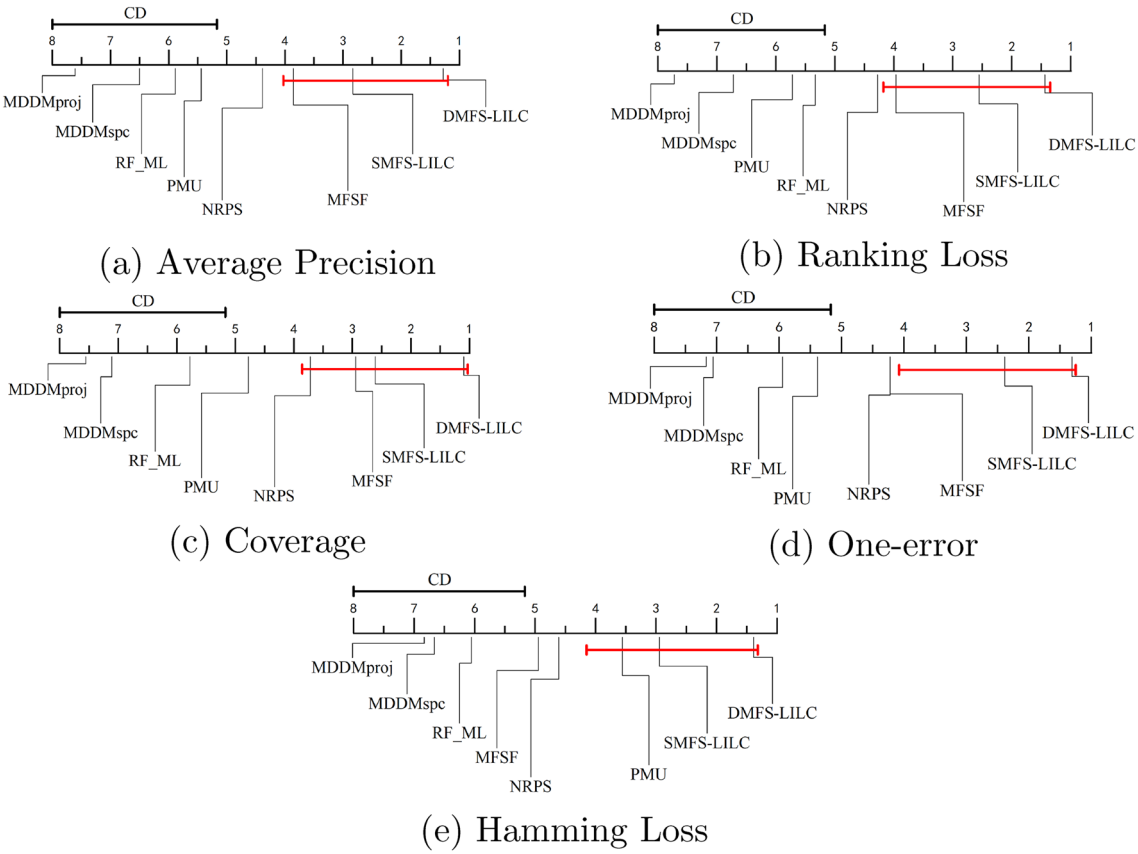

Fig. 4 Bonferroni-Dunn test of SMFS-LILC and DMFS-LILC in comparison with existing algorithms

formed worse than the existing algorithms, the CV achieved by DMFS-LILC and SMFS-LILC was not very different from that of the two existing algorithms that performed better. In addition, on the datasets on which performance was less good, the results of DMFS-LILC were still good. In addition, SMFS-LILC achieved suboptimal performance on five datasets. In summary, DMFS-LILC and SMFS-LILC did not perform significantly better than existing algorithms with respect to the CV evaluation criterion.

(4) In general, with respect to all the criteria, the average classification performance of DMFS-LILC was significantly better than that of all existing algorithms, and SMFS-LILC was the second best with respect to average performance. These experimental results show that DMFS-LILC and SMFS-LILC achieved better performance than the existing algorithms.

Because of differences in the data types and other aspects of the evaluation criteria, prediction performance is expected to vary. To clearly assess the differences between the algorithms, the prediction performance was normalized to [0.1, 0.5], following [63]. Figure 3 shows the stability indicators of the normalized AP, HL, RL, CV, and OE. Each corner of the spider graph in Fig. 3 represents a different dataset and each colored line represents a different algorithm.

If the area of the graph composed of lines of a specific color is large and its shape is similar to a regular nonagon, the performance and stability of the corresponding algorithm are good. A stability value of approximately 0.5 is considered to be a good value. From Fig. 3, the following observations can be made:

(1) With respect to AP, DMFS-LILC achieved the best stability because its shape closely approximates a regular nonagon and has the largest enclosed area.

(2) With respect to RL, OE, and HL, DMFS-LILC maintained stability on at least six datasets.

(3) With respect to CV, the nonagons of DMFS-LILC and SMFS-LILC are similar to those of NRPS and MFSF. Therefore, their performance advantages over the existing algorithms are not as obvious as for other evaluation criteria.

(4) For all the evaluation criteria, the shapes of DMFS-LILC and SMFS-LILC have areas that are larger than, or similar to, those of the existing algorithms, and they are closer to regular nonagons. In fact, a comprehensive

analysis of the results indicates that the performance and stability of the SMFS-LILC algorithm are second best, whereas the stability of DMFS-LILC is optimal.

4.2.2 Statistical test

Because some experimental results are quite similar, statistical tests can be used to verify whether these results differ significantly. We used the Friedman test to systematically analyze the differences between the results of the algorithms in the comparison. This is a widely accepted method of statistically comparing the results of multiple algorithms for significant differences across many datasets [64]. The method is as follows. Given k algorithms and N multi-label datasets, $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j v$ represents the average rank of the jth algorithm on all datasets, where r_i^j is the rank of algorithm j on the ith dataset. Under the null hypothesis (where it is assumed that the classification performance of all algorithms under each evaluation criterion are equal, that is, the ranks of all algorithms are equal), the Friedman test is defined as

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \text{ where } \chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4}\right) \tag{23}$$

where F_F follows an F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. Table 7 summarizes the F_F values and the corresponding critical values of each evaluation criterion after the Friedman test statistics [65].

As shown in Table 7, the null hypothesis is clearly rejected for all evaluation criteria with a significance level of $\alpha = 0.10$. Next, we used a post-hoc test to further determine the differences in the statistical performance of the various algorithms. Because our purpose was to compare the performance of the two proposed methods with that of the other algorithms, the Bonferroni–Dunn test was used [66]. The performance of two compared algorithms is considered to be significantly different if the distance between the average ranks of the two algorithms exceeds the following critical difference (CD).

$$CD_\alpha = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \tag{24}$$

For the Bonferroni–Dunn test, at a significance level of $\alpha = 0.01$, we have $q_\alpha = 2.450$, so we obtain $CD_\alpha = 2.8290$.

To visualize the relative performance of SMFS-LILC and DMFS-LILC compared with that of the other six algorithms, we plotted the CD for each evaluation criterion, with the average ranking of each compared algorithm on the axis. We consider the rightmost algorithm to be the best, so the lowest ranking on the axis is on the right. The CD plots for all evaluation criteria are shown in Fig. 4.

From Fig. 4 we can observe the following:

(1) SMFS-LILC and DMFS-LILC are significantly better than MDDMspc, MDDMproj, RF-ML, and MFSF with respect to all evaluation criteria. In particular, DMFS-LILC has obvious advantages compared with them.
(2) SMFS-LILC is statistically superior to, or at least comparable to, MFSF and NRPS with respect to all evaluation criteria, and DMFS-LILC also shows significant advantages over those algorithms, with respect to some criteria.
(3) Although the classification performance of SMFS-LILC and MFSF is comparable, the average classification performance of DMFS-LILC in Tables 2, 3, 4, 5 and 6 is significantly better than that of the other algorithms in the comparison. In summary, DMFS-LILC has significantly stronger performance than the other algorithms.

5 Conclusion

In this paper, we propose an NRS model based on label importance and label correlation. We first define a new neighborhood particle by the mean nearest neighborhood method, to better correlate the information between features of multiply labeled data, and solve the problem of neighborhood granularity caused by such data. The feature correlation weights are then obtained by calculating the mutual information between features, and the new neighborhood lower bound approximation is combined with the feature weights to obtain a new feature subset importance model. On the basis of this model, we propose a new static forward greedy algorithm (SMFS-LILC) for multi-label feature selection. In addition, we propose a dynamic feature selection algorithm (DMFS-LILC), based on SMFS-LILC, to evaluate features that arrive incrementally over time by importance analysis, significance analysis, and redundancy analysis to solve the multi-label stream feature problem. Experimental results showed that our algorithms are competitive with existing commonly used algorithms. However, the time complexity of the proposed algorithms is relatively high, compared with that of state-of-the-art multi-label feature selection methods. Therefore, in future work, we hope to reduce the computation time of the algorithm. Furthermore, by solving multi-label problems using label importance and label correlation, or by handling features and labels by mutual information methods, these methods can also be extended to the feature selection problem of label distribution.

Data Availability The datasets supporting Table 1 are publicly available in Mulan Library at https://mulan.sourceforge.net/datasets.html. The data in Tables 2, 3, 4, 5 and 6 was generated through the code in this article using the datasets in Table 1. The code is available from the corresponding author by request.

References

1. Mitchell TM, Mitchell TM (1997) Machine learning 1(9):13–16
2. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Warehous Min (IJDWM) 3(3):1–13
3. Fakhari A, Moghadam AME (2013) Combination of classification and regression in decision tree for multi-labeling image annotation and retrieval. Appl Soft Comput 13(2):1292–1302
4. Lewis DD, Yang Y, Russell-Rose T, Li F (2004) RCV1: a new benchmark collection for text categorization research. J Mach Learn Res 5:361–397
5. Liu W, Wang H, Shen X, Tsang IW (2021) The emerging trends of multi-label learning. IEEE Trans Pattern Anal Mach Intell 44(11):7955–7974
6. Zhang M-L, Zhou Z-H, Tsoumakas G (2009) Learning from multi-label data. In: ECML/PKDD, vol 9
7. Schapire RE, Singer Y (2000) Boostexter: a boosting-based system for text categorization. Mach Learn 39(2):135–168
8. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recogn 37(9):1757–1771
9. Elisseeff A, Weston J (2001) A kernel method for multi-labelled classification. Adv Neural Inf Process Syst 14
10. Barutcuoglu Z, Schapire RE, Troyanskaya OG (2006) Hierarchical multi-label prediction of gene function. Bioinformatics 22(7):830–836
11. Wu M, Su W, Chen L, Pedrycz W, Hirota K (2020) Two-stage fuzzy fusion based-convolution neural network for dynamic emotion recognition. IEEE Trans Affective Comput
12. Trohidis K, Tsoumakas G, Kalliris G, Vlahavas IP et al (2008) Multi-label classification of music into emotions. ISMIR 8:325–330
13. Yang F, Zhong Z, Luo Z, Cai Y, Lin Y, Li S, Sebe N (2021) Joint noise-tolerant learning and meta camera shift adaptation for unsupervised person re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4855–4864
14. Gopal S, Yang Y (2010) Multilabel classification with meta-level features. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval, pp 315–322
15. Lee J, Kim D-W (2015) Fast multi-label feature selection based on information-theoretic feature ranking. Pattern Recogn 48(9):2761–2771
16. Kumar V, Minz S (2016) Multi-view ensemble learning: an optimal feature set partitioning for high-dimensional data classification. Knowl Inf Syst 49(1):1–59
17. Lin Y, Hu Q, Liu J, Chen J, Duan J (2016) Multi-label feature selection based on neighborhood mutual information. Appl Soft Comput 38:244–256
18. Yu Y, Pedrycz W, Miao D (2014) Multi-label classification by exploiting label correlations. Expert Syst Appl 41(6):2989–3004
19. Wu X, Yu K, Wang H, Ding W (2010) Online streaming feature selection. In: ICML
20. Chen H, Li T, Luo C, Horng S-J, Wang G (2015) A decision-theoretic rough set approach for dynamic data mining. IEEE Trans Fuzzy Syst 23(6):1958–1970
21. Chen D, Yang Y (2013) Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models. IEEE Trans Fuzzy Syst 22(5):1325–1334
22. Hu Q, Pan W, Zhang L, Zhang D, Song Y, Guo M, Yu D (2011) Feature selection for monotonic classification. IEEE Trans Fuzzy Syst 20(1):69–81
23. Wu X, Zhu X, Wu G-Q, Ding W (2013) Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97–107
24. Lin Y, Hu Q, Liu J, Duan J (2015) Multi-label feature selection based on max-dependency and min-redundancy. Neurocomputing 168:92–103
25. Javidi MM, Eskandari S (2018) Streamwise feature selection: a rough set method. Int J Mach Learn Cybernet 9(4):667–676
26. Hotelling H (1992) Relations between two sets of variates. In: Breakthroughs in statistics. Springer, Berlin, pp 162–190
27. Zhang Y, Zhou Z-H (2010) Multilabel dimensionality reduction via dependence maximization. ACM Trans Knowl Discov Data (TKDD) 4(3):1–21
28. Yu K, Yu S, Tresp V (2005) Multi-label informed latent semantic indexing. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, pp 258–265
29. Lee J, Kim D-W (2013) Feature selection for multi-label classification using multivariate mutual information. Pattern Recogn Lett 34(3):349–357
30. Spolaôr N, Cherman EA, Monard MC, Lee HD (2013) Relieff for multi-label feature selection. In: 2013 Brazilian conference on intelligent systems. IEEE, pp 6–11
31. Arslan S, Ozturk C (2019) Multi hive artificial bee colony programming for high dimensional symbolic regression with feature selection. Appl Soft Comput 78:515–527
32. Chen S-B, Zhang Y-M, Ding CH, Zhang J, Luo B (2019) Extended adaptive lasso for multi-class and multi-label feature selection. Knowl-Based Syst 173:28–36
33. Jiang Z, Liu K, Yang X, Yu H, Fujita H, Qian Y (2020) Accelerator for supervised neighborhood based attribute reduction. Int J Approx Reason 119:122–150
34. Dong H, Sun J, Li T, Ding R, Sun X (2020) A multi-objective algorithm for multi-label filter feature selection problem. Appl Intell 50(11):3748–3774
35. Sun L, Yin T, Ding W, Qian Y, Xu J (2021) Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy. IEEE Trans Fuzzy Syst 30(5):1197–1211
36. Ding W, Lin C-T, Cao Z (2018) Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping PSO with nearest-neighbor memeplexes. IEEE Trans Cybern 49(7):2744–2757
37. Li A-D, Xue B, Zhang M (2021) Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. Appl Soft Comput 106:107302
38. Zhang J, Luo Z, Li C, Zhou C, Li S (2019) Manifold regularized discriminative feature selection for multi-label learning. Pattern Recogn 95:136–150
39. Doquire G, Verleysen M (2011) Feature selection for multi-label classification problems. In: Advances in computational intelligence: 11th international work-conference on artificial

neural networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings, Part I 11. Springer, pp 9–16
40. Zhu Y, Kwok JT, Zhou Z-H (2017) Multi-label learning with global and local label correlation. IEEE Trans Knowl Data Eng 30(6):1081–1094
41. Yang P, Sun X, Li W, Ma S, Wu W, Wang H (2018) SGM: sequence generation model for multi-label classification. arXiv preprint arXiv:1806.04822
42. Jian L, Li J, Shu K, Liu H (2016) Multi-label informed feature selection. IJCAI 16:1627–33
43. Yu K, Wu X, Ding W, Pei J (2016) Scalable and accurate online feature selection for big data. ACM Trans Knowl Discov Data (TKDD) 11(2):1–39
44. Paul D, Jain A, Saha S, Mathew J (2021) Multi-objective PSO based online feature selection for multi-label classification. Knowl-Based Syst 222:106966
45. Lin Y, Hu Q, Liu J, Li J, Wu X (2017) Streaming feature selection for multilabel learning based on fuzzy mutual information. IEEE Trans Fuzzy Syst 25(6):1491–1507
46. Wang J, Wang M, Li P, Liu L, Zhao Z, Hu X, Wu X (2015) Online feature selection with group structure analysis. IEEE Trans Knowl Data Eng 27(11):3029–3041
47. Yu K, Wu X, Ding W, Pei J (2014) Towards scalable and accurate online feature selection for big data. In: 2014 IEEE international conference on data mining. IEEE, pp 660–669
48. Fan Y, Liu J, Wu S (2022) Exploring instance correlations with local discriminant model for multi-label feature selection. Appl Intell 52(7):8302–8320
49. Fan Y, Chen B, Huang W, Liu J, Weng W, Lan W (2022) Multi-label feature selection based on label correlations and feature redundancy. Knowl Based Syst 241:108256
50. Chen P, Lin M, Liu J (2020) Multi-label attribute reduction based on variable precision fuzzy neighborhood rough set. IEEE Access 8:133565–133576
51. Liu J, Lin Y, Du J, Zhang H, Chen Z (2023) Zhang J (2022) ASFS: a novel streaming feature selection for multi-label data based on neighborhood rough set. Appl Intell 53(2):1707–1724
52. Wu Y, Liu J, Yu X, Lin Y, Li S (2022) Neighborhood rough set based multi-label feature selection with label correlation. Concurr Comput Pract Exp 34(22):7162
53. Qian Y, Liang J, Pedrycz W, Dang C (2010) Positive approximation: an accelerator for attribute reduction in rough set theory. Artif Intell 174(9–10):597–618
54. Liu J, Lin Y, Lin M, Wu S, Zhang J (2017) Feature selection based on quality of information. Neurocomputing 225:11–22
55. Hashemi A, Dowlatshahi MB, Nezamabadi-Pour H (2020) MGFS: a multi-label graph-based feature selection algorithm via pagerank centrality. Expert Syst Appl 142:113024
56. Sen T, Chaudhary, DK (2017) Contrastive study of simple pagerank, hits and weighted pagerank algorithms. In: 2017 7th International conference on cloud computing, data science & engineering-confluence. IEEE, pp 721–727
57. Hu Q, Zhao H, Yu D (2008) Efficient symbolic and numerical attribute reduction with neighborhood rough sets. Pattern Recogn Artif Intell 21(6):732–738
58. Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I (2011) Mulan: a java library for multi-label learning. J Mach Learn Res 12:2411–2414
59. Cai Z, Zhu W (2017) Feature selection for multi-label classification using neighborhood preservation. IEEE/CAA J Autom Sin 5(1):320–330
60. Xu J, Shen K, Sun L (2022) Multi-label feature selection based on fuzzy neighborhood rough sets. Complex Intell Syst 8(3):2105–2129
61. Hu Q, Yu D, Liu J, Wu C (2008) Neighborhood rough set based heterogeneous feature subset selection. Inf Sci 178(18):3577–3594
62. Zhang M-L, Zhou Z-H (2007) ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
63. Li Y, Lin Y, Liu J, Weng W, Shi Z, Wu S (2018) Feature selection for multi-label learning based on kernelized fuzzy rough sets. Neurocomputing 318:271–286
64. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat 11(1):86–92
65. Dong J, Fu J, Zhou P, Li H, Wang X (2022) Improving spoken language understanding with cross-modal contrastive learning. Proc Interspeech 2022:2693–2697
66. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64