An Improved LSML with Global and Local Label Correlation

Yichun Liu, Qing Ai*, Yiying Wang, Yang Yu, Bo Cui and Yuting Xu

***Abstract*—LSML is a novel method for learning label-specific features in multi-label classification with missing labels. Due to its outstanding performance, LSML has garnered widespread attention. However, it has certain limitations: A) LSML assumes that correlated labels share similar label-specific features, which may not hold true in uncertain cases. B) LSML overlooks local label correlations that can impact classification performance. To address these issues, we propose an improved version of LSML, called LSMLLC, which incorporates both global and local label correlations. First, we introduce a label-level regularizer to capture global label correlations directly from the label output, rather than relying on the coefficient matrix. Second, we account for local label correlations using a k-nearest neighbor mechanism. Finally, we employ the accelerated proximal gradient algorithm to efficiently solve the classification model. Extensive experimental results demonstrate that LSMLLC outperforms existing methods in multi-label learning tasks.**

***Index Terms*—multi-label classification, missing labels, label-specific features, label correlation learning**

I. Introduction

TO address the issue that an instance may have multiple labels in the real world, multi-label learning has emerged. Multi-label learning algorithms can be categorized into two primary groups: problem transformation (PT) methods and algorithm adaptation (AA) methods. PT methods transform the multi-label classification problem into multiple single-label classification tasks. Its representative algorithms are Binary Relevance (BR) [1], calibrated label ranking [2], random-k-labelsets [3], and classifier chains [4]. AA methods directly apply the existing classification algorithm to multi-label datasets. Its representative algorithms are Multi-label Decision Tree [5], Rank-SVM [6], ML-KNN [7], and CML [8].

The traditional algorithms typically assume that all labels share the same features [9]. However, in multi-label classification tasks, each label has its corresponding features. The label-specific feature learning algorithm extracts distinct features for each label, enhancing classification performance. Its representative algorithms are LIFT [10], LLSF [11], LLSF-DL [11], LSML [12], and CLML [13].

The labels are also correlated with each other [14]. According to the label correlation, multi-label learning algorithms can be divided into three main categories. First-order strategy ignores the correlations between the labels. Its representative algorithms are ML-KNN [7], LIFT [10], ML-localkNN [15], and MLSAkNN [16]. Second-order strategy considers pairwise correlations between labels. Its representative algorithms are Rank-SVM [6], LLSF [11], LSML [12], CLML [13] and RFSFS [17]. High-order strategy considers the correlations between a label and all other labels. Its representative algorithms are LLSF-DL [11], GLOCAL [18], KLLI [19], LFFS [20], and GLFS [21].

LSML is an efficient multi-label classification algorithm with missing labels, which utilizes the label correlation coefficient matrix to recover the missing labels, and learns the label-specific features in the new supplementary label matrix. Due to the excellent performance of LSML, it has been gained widespread attention. However, LSML has certain limitations. A) LSML [12] bases on a assumption: if labels are associated, they have similar label-specific features. However, this assumption is not always right in uncertain cases [13]. B) LSML ignores local label correlations that affect the classification performance. To address above issues, we propose an improved LSML based on the global and local label correlations (LSMLLC). First, we introduce a novel assumption: if labels exhibit strong correlations, the label outputs should be similar, rather than their coefficient matrices. Further, we utilize a label-level regularizer to capture the global label correlation on the label outputs instead of the coefficient matrix. Second, we also consider the local label correlation through the k-nearest neighbor mechanism. Finally, the accelerated proximal gradient algorithm is employed to solve the classification model efficiently. Extensive experimental results show that our proposed LSMLCL achieves better performance in multi-label learning tasks.

This paper is organized as follows. Section 2 presents an overview of relevant works on multi-label learning. Section 3 introduces our LSMLLC method, including the classification model, decision function, optimization, proofs and analysis of complexity. Section 4 analyzes the experimental result and investigates parameter sensitivity. Finally, we conclude the paper in Section 5.

II. Related works

A. Multi-Label Learning with Label-Specific Features learning

Zhang et al. proposed LIFT [10] algorithm, which learned the label-specific features through conducting clustering anal-

Manuscript received Oct 18, 2024; revised Jan 17, 2025.

This work was funded in part by the Basic Research Project of Education Department of Liaoning Province in China (JYTMS20230929).

Yichun Liu is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(ustlliuyichun@163.com).

Qing Ai is a professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(lyaiqing@126.com).

Yiying Wang is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(yiyingwang2024@126.com).

Yang Yu is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(yu_yang0927@163.com).

Bo Cui is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(ustlcuibo@163.com).

Yuting Xu is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning China(ustlxuyuting@163.com).

ysis, and using these new features to build the model. Subsequently, Huang et al. proposed LLSF [11] algorithm, which introduced a novel strategy for learning label-specific features. LLSF made the assumption that each label was associated with only a subset of features. Based on this assumption, LLSF effectively identified the discriminative features for each label using the l_1-norm of the coefficient matrix. LLSF-DL [11] extended LLSF by using a sparse stacking approach. LLSF-DL treated the label matrix predicted by LLSF as additional features to be added to the training set for training. However, LLSF ignored instance correlation learning and common feature learning. To address above issues, Li et al. proposed CLML [13] algorithm, which effectively learned both label-specific and common features by leveraging the relationships between labels and instances. This approach took into account the associated information to capture the underlying correlations, allowing for a more comprehensive representation of the data.

B. Multi-Label Learning with Missing labels

He et al. proposed MLMF [22] algorithm, which explored label correlations. The algorithm constructed a conditional dependence network, where input features and other label variables served as parent nodes for each label. Simultaneously, MLMF introduced the l_1-norm regularizer to select the sparse feature structure. However, MLMF neglected the importance of matrix completion. Reshma Rastogi and Sayed Mortaza proposed MLLCRS-ML [23] algorithm that utilized the structural property of the datasets along with pairwise label correlations (both positive and negative) to recover missing labels. Further, Sanjay Kumar and Reshma Rastogi proposed a new algorithm to recover the missing labels, called LRMML [24], which introduced the auxiliary label matrix to recover missing label information. LRMML leveraged the auxiliary label matrix and the low-rank constraint to capture label correlations. Additionally, LRMML enforced maximum separation between different label subspaces in order to distinguish labels, enhancing classification performance.

C. Multi-Label Learning with Label Correlations Learning

Cheng et al. proposed LSLC-ML [25] algorithm, which detected both positive and negative relationships among labels in datasets with incomplete label information. The algorithm retrieved the missing labels and used these obtained label-specific features. However, LSLC-ML neglected the instance correlations that affect the classification performance. To address this issue, Feng et al. proposed RMFL [26] algorithm, which represented a multi-label learning approach aimed at resolving the problem of missing labels through the use of regularized matrix factorization. The algorithm encapsulated the latent information pertaining to instances and labels, optimizing the model by integrating both instance space data and label correlations. Sun et al. proposed the GLC [27] algorithm that introduced a new matrix aimed at leveraging the global structural information of labels derived from different subspaces and captured local correlations among labels.

D. LSML

Let $\mathbf{X} = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^{n\times d}$ represent the training data with n samples and d dimensions. Let $\mathbf{Y} = [y_1, y_2, \ldots, y_n]^\top \in \{-1, 0, 1\}^{d\times l}$ be the corresponding class labels of the training data, where $y_{ij} = 1$ denotes the i-th example has the j-th label and $y_{ij} = -1$ denotes the i-th example does not have the j-th label and $y_{ij} = 0$ denotes the label of the i-th sample is unobserved.

LSML utilized the label correlation coefficient matrix to recover missing labels. It then extracted label-specific features for each label based on the newly completed label matrix. The final objective function can be reformulated into an equation.

$$\min_{W,C} \left(\frac{1}{2} \|XW - YC\|_F^2 + \frac{\lambda_1}{2} \|YC - Y\|_F^2 \right) + \left(\lambda_2 \|C\|_1 + \lambda_3 \|W\|_1 + \lambda_4 \operatorname{tr}\left(WLW^T\right) \right), \quad (1)$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are penalty parameters.

III. LSMLLC

A. Multi-label learning with missing labels and label-specific features

In this paper, we utilize the l_1-norm regularizer to obtain the label-specific features. The original problem is defined

$$\min_W \frac{1}{2} \|XW - Y\|_F^2 + \lambda_1 \|W\|_1, \quad (2)$$

where $W = \left[w^1, w^2, \ldots, w^d\right] \in \mathbb{R}^{d\times 1}$ represents the coefficient matrix of the regression model, λ_1 is the trade-off parameter that adjust the sparse regularization term.

The traditional algorithms typically assume that all labels for the training samples are available. However, in real-world applications, only a subset of the labels can be observed. [28]–[30]. To address the issue, we assume that missing labels can be reconstructed by using their correlations with other labels. Thus, the incomplete label matrix is augmented into a new label matrix through using the label correlation coefficient matrix. We defined

$$\min_{W,C} \frac{1}{2} \|XW - YC\|_F^2 + \frac{\alpha}{2} \|YC - Y\|_F^2 + \lambda_1 \|W\|_1 + \lambda_2 \|C\|_1, \quad (3)$$

where $C \in \mathbb{R}^{l\times l}$ represents the label correlations coefficient matrix. We obtain the new label matrix through the label correlation coefficient matrix. The model enables the redefinition of each label by considering high-order correlations among labels. Besides, in practical applications, a class label is only related to a subset of labels. Thus, the label correlations coefficient matrix C is constrained by l_1-norm that can select the most important label correlation information.

B. Multi-label learning with global and local label correlations

In real applications, it is assumed that labels with correlation exhibit similar label-specific features. This assumption may not always be accurate. To address this issue, we propose another new assumption that two labels exhibit a strong correlation, their corresponding outputs may be similar. We utilize the new output matrix XW instead of the coefficient matrix W. Finally, we introduce a graph

Laplacian regularization term to enhance model stability. The objective function is then defined

$$\sum_{i,j=1}^{l} G_{ij} \left\|(XW)_i - (XW)_j\right\| = \operatorname{tr}\left((XW)(G^{\Delta} - G)(XW)^T\right) = \operatorname{tr}\left((XW)L_1(XW)^T\right), \tag{4}$$

where $G_{ij} \in G$ is the label correlation between label y_i and y_j, which is determined by the cosine similarity. $L_1 = G^{\Delta} - G$ is the $l \times l$ label Laplacian matrix of G. G^{Δ} is the diagonal matrix and $G^{\Delta}_{ii} = \sum_{j=1}^{l} G_{ij}$. The objective function is then defined

$$\min_{W,C}\left(\frac{1}{2}\|XW - YC\|_F^2 + \frac{\alpha}{2}\|YC - Y\|_F^2 + \frac{\beta}{2}\operatorname{tr}((XW)L_1(XW)^T) + \lambda_1\|W\|_1 + \lambda_2\|C\|_1\right) \tag{5}$$

In our approach, we thoroughly analyze both global and local correlations among labels. However, the evaluating ability of the cosine similarity is limited to noise and redundant features. Thus, we utilize the k-nearest neighbors (KNN) method to evaluate an instance similarity matrix H. For instance, the similarity of the i-th and j-th instances is defined

$$H_{ij} = \begin{cases} 1, & \text{if } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

If two instances exhibit a strong correlation, their predicted labels are likely to be similar. To achieve this, we employ a graph Laplacian regularization term, defined as follows:

$$\sum_{i,j=1}^{n} H_{ij} \left\|(x_iW) - (x_jW)\right\| = \operatorname{tr}\left((XW)^T L_2(XW)\right), \tag{7}$$

where L_2 is the $n \times n$ laplacian matrix of H.

The final objective function can be reformulated into an equation.

$$\min_{W,C}\left(\frac{1}{2}\|XW - YC\|_F^2 + \frac{\alpha}{2}\|YC - Y\|_F^2 + \frac{\beta}{2}\operatorname{tr}((XW)L_1(XW)^T) + \frac{\gamma}{2}\operatorname{tr}((XW)^T L_2(XW)) + \lambda_1\|W\|_1 + \lambda_2\|C\|_1\right), \tag{8}$$

where $\alpha, \beta, \gamma, \lambda_1$ and λ_2 are penalty parameters.

C. Optimization

Due to the non-smooth nature of the l_1-norm regularization term, the optimization problem (7) is convex in nature, but it is non-smooth overall. Therefore, this paper employs the accelerated proximal gradient descent algorithm to address problem (7). The model has two solution variables; we denote ϕ as W and C. Simplifying the objective framework to:

$$\min_{\phi}\{\Gamma(\phi) = \hbar(\phi) + \psi(\phi)\}, \tag{9}$$

where

$$\hbar(\phi) = \frac{1}{2}\|XW - YC\|_F^2 + \frac{\alpha}{2}\|YC - Y\|_F^2 + \frac{\beta}{2}\operatorname{tr}\left((XW)L_1(XW)^T\right) + \frac{\gamma}{2}\operatorname{tr}\left((XW)^T L_2(XW)\right) \tag{10}$$

$$\psi(\phi) = \lambda_1\|W\|_1 + \lambda_2\|C\|_1 \tag{11}$$

Both $\hbar(\phi)$ and $\psi(\phi)$ are convex, but $\psi(\phi)$ is non-smooth. For any $L > 0$, we define the second-order approximation of the former:

$$Q_L\left(\phi, \phi^{(k)}\right) = \hbar\left(\phi^{(k)}\right) + \left\langle \nabla\hbar\left(\phi^{(k)}\right), \phi - \phi^{(k)}\right\rangle + \frac{L}{2}\left\|\phi - \phi^{(k)}\right\|_F^2 + \psi(\phi) \tag{12}$$

For any $L \geq L_f$, it can be stated that $Q_L(\phi, \phi^k) \geq \Gamma(\phi)$, where L_f is the Lipschitz constant. We do not directly minimize $\Gamma(\phi)$, the proximal gradient algorithm approximates the objective function L_f by minimizing many separable quadratic approximations. By defining the update $F^{(k)} = \phi^{(k)} - \frac{1}{L}\nabla\hbar\left(\phi^{(k)}\right)$, the solution for ϕ can be obtained through the function $Q_L\left(\phi, \phi^{(k)}\right)$.

$$\phi^* = \arg\min_{\phi} Q_L\left(\phi, \phi^{(k)}\right) = \arg\min_{\phi}\left(g(\phi) + \frac{L}{2}\left\|\phi - F^{(k)}\right\|_F^2\right), \tag{13}$$

where $F^{(k)} = \phi^{(k)} - \frac{1}{L}\nabla\hbar\left(\phi^{(k)}\right)$, by setting t_k in a sequence, the convergence of the model can be accelerated, and the update of ϕ is given by $\phi^{(k)} = \phi_k + \frac{t_{k-1}-1}{t_k}(\phi_k - \phi_{k-1})$, where $t_{k+1}^2 - t_{k+1} \leq t_k^2$ and ϕ_k is the k-th iteration.

1) Updating W: First, C is fixed to update W, and the partial derivative $\frac{\partial}{\partial W}$ is computed.

$$\nabla_W\hbar(\phi) = X^TXW - X^TYC + \beta X^TXWL_1 + \gamma X^TL_2XW \tag{14}$$

The update process for W can be obtained through equation (13).

$$W^{(k)} = W_k + \frac{t_{k-1}-1}{t_k}(W_k - W_{k-1})$$
$$W^{(k+1)} = \operatorname{prox}_{\varepsilon}\left(W^k - \frac{1}{L}\nabla f\left(W^{(k)}, C\right)\right), \tag{15}$$

where τ represents the step size, regarding $\psi(\phi)$, the l_1-norm of W can be derived from the element-wise soft-thresholding operator:

$$\operatorname{prox}_{\varepsilon}(W_{ij}) = (|W_{ij}| - \tau)_+ \operatorname{sign}(W_{ij}), \tag{16}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

2) Updating C: First, W is fixed to update C, and the partial derivative C is computed.

$$\nabla_S f(\phi) = (1+\alpha)Y^TYC - Y^TXW - \alpha Y^TY \tag{17}$$

Similarly, we can obtain the update process for C:

$$C^{(k)} = C_k + \frac{t_{k-1}-1}{t_k}(C_k - C_{k-1})$$
$$C^{(k+1)} = \operatorname{prox}_{\varepsilon}\left(C^{(k)} - \frac{1}{L}\nabla f\left(W, C^{(k)}\right)\right), \tag{18}$$

where ε represents the step size, regarding $\psi(\phi)$, the representation of the l_1-norm of C can be derived from the definition of the element-wise soft-thresholding operator.

$$\operatorname{prox}_{\varepsilon}(S_{ij}) = (|S_{ij}| - \varepsilon)_+ \operatorname{sign}(S_{ij}), \tag{19}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

D. Proof of Lipschitz Continuity

Lipschitz is essential in accelerated proximal gradient algorithms. We give $\phi_1 = (W_1, C_1)$ and $\phi_2 = (W_2, C_2)$. Based on equations (14) and (17), we can obtain the following expressions.

$$\begin{aligned}
&\|\nabla f(\phi_1) - \nabla f(\phi_2)\|_F^2 \\
&= \left\|X^T X \Delta W + \beta X^T X \Delta W L_1 + \gamma X^T L_2 X \Delta W \right. \\
&\quad \left. + (1+\alpha) Y^T Y \Delta C \right\|_F^2 \\
&= \left\|X^T X \Delta W + \beta X^T X \Delta C L_1 + \gamma X^T L_2 X \Delta W \right\|_F^2 \\
&\quad + \left\|(1+\alpha) Y^T Y \Delta C \right\|_F^2 \\
&\leq 3\left(\left\|X^T X\right\|_2^2 + \left\|\beta X^T X\right\|_2^2 \cdot \|L_1\|_2^2 + \left\|\gamma X^T L_2 X\right\|_2^2\right) \\
&\quad \|\Delta W\|_F^2 + 2\left(\left\|Y^T Y\right\|_2^2 + \left\|\alpha Y^T Y\right\|_2^2\right) \|\Delta C\|_F^2 \\
&\leq \left(3\left(\left\|X^T X\right\|_2^2 + \left\|\beta X^T X\right\|_2^2 \cdot \|L_1\|_2^2 + \left\|\gamma X^T L_2 X\right\|_2^2\right)\right. \\
&\quad \left. + 2\left(\left\|Y^T Y\right\|_2^2 + \left\|\alpha Y^T Y\right\|_2^2\right)\right) \|\Delta W\|_F^2,
\end{aligned} \tag{20}$$

where $\Delta W = W_1 - W_2$, $\Delta C = C_1 - C_2$.

Therefore, the objective function for the lipschitz constant is expressed as

$$L_f = \sqrt{\begin{array}{l} 3\left(\left\|X^T X\right\|_2^2 + \left\|\beta X^T X\right\|_2^2 \cdot \|L_1\|_2^2 + \left\|\gamma X^T L_2 X\right\|_2^2\right) \\ \quad + 2\left(\left\|Y^T Y\right\|_2^2 + \left\|\alpha Y^T Y\right\|_2^2\right) \end{array}} \tag{21}$$

Algorithm LSMLLC.

Input: Train data matrix $X \in \mathbb{R}^{n \times d}$, train label matrix $Y \in \mathbb{R}^{n \times l}$, and weighting parameters $\alpha, \beta, \gamma, \lambda_1$, and λ_2.

Output: Coefficient matrix $W \in \mathbb{R}^{d \times l}$.

1) **Initialization:** $W_0, W_1 \leftarrow \operatorname{rand}(d, l)$; $C_0, C_1 \leftarrow \operatorname{rand}(l, l)$; $\phi^{(1)} = \{W_1, C_1\}$; $t_0, t_1 \leftarrow 1$; $k \leftarrow 1$;
2) Calculate label correlation matrix S by calculating cosine similarity on Y; Calculate H by using k-nearest neighbors; Calculate the Lipschitz constant L_f;
3) **repeat:**
   1. $W^{(k)} \leftarrow W_k + \frac{t_{k-1}-1}{t_k}(W_k - W_{k-1})$;
   2. $F_W^{(k)} \leftarrow W^{(k)} - \frac{1}{L}\nabla_W f(W^{(k)}, C_k)$;
   3. $W_{k+1} \leftarrow \operatorname{prox}_{\mathcal{E}} \frac{\lambda_1}{L}(F_W^{(k)})$ (by according (14)).
   4. $W^{(k+1)} \leftarrow W^{(k)}$;
   5. $C^{(k)} \leftarrow C_k + \frac{t_{k-1}-1}{t_k}(C_k - C_{k-1})$;
   6. $F_C^{(k)} \leftarrow C^{(k)} - \frac{1}{L}\nabla_C f(W_k, C^{(k)})$;
   7. $C_{k+1} \leftarrow \operatorname{prox}_{\varepsilon} \frac{\lambda_2}{L}(F_C^{(k)})$ (by according (17)).
   8. $C^{(k+1)} \leftarrow C^{(k)}$;
   9. $t_{k+1} \leftarrow \frac{1+\sqrt{4t_k^2+1}}{2}$;
   10. $k \leftarrow k+1$;
4) **until converge**;
5) **Return** W;

TABLE I
AN OVERVIEW OF THE FRIEDMAN STATISTIC $F_F(K=7, N=5)$ ALONG WITH ITS CRITICAL VALUE BASED ON SIX ASSESSMENT METRICS.

Evaluation Metric	F_F	Critical Value(0.05)
One Error	12.642857	
Ranking Loss	9.596939	
Average Precision	18.750000	2.363751
Hamming Loss	20.923469	
AUC	11.693878	

E. Analysis of complexity

The time complexity of the LSMLLC algorithm can be divided into two main parts: preprocessing and iterative optimization. The primary operations include computing distances between labels and instances, leading to a time complexity of $O(n^2 \cdot d)$. Iterative optimization involves updating matrices W_x and C_y in each iteration, with the number of iterations bounded by 'maxIter'. Each iteration involves matrix multiplications and updates, resulting in a time complexity of

$$O\left(\max Iter \times \left(nd^2 + ndl + n^2 d + nl^2 + nld\right)\right).$$

Overall, the total time complexity is

$$O\left(n^2 d + \max Iter \times \left(nd^2 + ndl + n^2 d + nl^2 + nld\right)\right).$$

IV. Experiments and evaluation

We conduct a comparative evaluation of our algorithm against six existing algorithms for label-specific feature learning and label correlation learning across seven distinct datasets. The performance and parameter sensitivity are analyzed using five evaluation metrics. The experiments are conducted on an Intel(R) Xeon(R) CPU E5-2678 v3 processor running at 2. 50GHz, with 32 GB of memory. These algorithms have been developed and run in MATLAB R2021a [31].

A. Datasets

We conduct experiments on seven multi-label benchmark datasets to evaluate the performance of LSMLLC. The summary of experimental dataset characteristics is shown in Table II.

TABLE II
THE CHARACTERISTICS OF THE EXPERIMENTAL DATASETS.

Dataset	Domain	#Complete-Sample	#Missing-Sample	#Sample	#Feature	#Label
genbase	biology	451	194	645	1186	27
emotions	music	415	178	593	72	6
yeast	biology	1691	725	2416	103	14
arts	text (web)	3500	1500	5000	462	26
science	text (web)	3500	1500	5000	743	40
bibtex	text	5176	2219	7395	1836	159
delicious	text (web)	11273	4832	16105	500	983

TABLE III
AVERAGE_PRECISION OF SIX DIFFERENT ALGORITHMS ON SEVEN DATASETS.

Algorithm	Average_precision↑						
	genbase	emotions	yeast	art	science	bibtex	delicious
LIFT	0.9696 ± 0.0015	0.6414 ± 0.0108	0.7408 ± 0.0016	0.5748 ± 0.0045	0.5111 ± 0.0042	0.4236 ± 0.0014	0.2896 ± 0.0006
LLSF	0.9373 ± 0.0144	0.6245 ± 0.0027	0.6874 ± 0.0057	0.6057 ± 0.0036	0.5258 ± 0.0041	0.4508 ± 0.0063	0.2783 ± 0.0004
LLSF-DL	0.9654 ± 0.0031	0.6133 ± 0.0102	0.7191 ± 0.0080	0.4549 ± 0.0044	0.5245 ± 0.0073	0.4412 ± 0.0033	0.3464 ± 0.0008
LSML	0.9828 ± 0.0025	0.6122 ± 0.0108	0.7399 ± 0.0010	0.5856 ± 0.0032	0.5525 ± 0.0063	0.5019 ± 0.0014	0.3513 ± 0.0006
CLML	0.9654 ± 0.0022	0.6086 ± 0.0063	0.6676 ± 0.0043	0.5917 ± 0.0071	0.5230 ± 0.0222	0.4322 ± 0.0059	0.2560 ± 0.0004
LRLSF	0.9888 ± 0.0052	0.5806 ± 0.0297	0.7119 ± 0.0016	0.5888 ± 0.0058	0.5874 ± 0.0031	0.4246 ± 0.0019	0.3000 ± 0.0006
LSMLLC	0.9898 ± 0.0015	0.7002 ± 0.0069	0.7430 ± 0.0016	0.6108 ± 0.0045	0.5702 ± 0.0042	0.5031 ± 0.0014	0.3520 ± 0.0006

TABLE IV
ONE_ERROR OF SIX DIFFERENT ALGORITHMS ON SEVEN DATASETS.

Algorithm	One_error↓						
	genbase	emotions	yeast	art	science	bibtex	delicious
LIFT	0.0000 ± 0.0000	0.4568 ± 0.0170	0.2458 ± 0.0056	0.5265 ± 0.0092	0.6133 ± 0.0036	0.5053 ± 0.0051	0.3885 ± 0.0025
LLSF	0.0000 ± 0.0000	0.5136 ± 0.0060	0.3058 ± 0.0135	0.4810 ± 0.0080	0.5849 ± 0.0127	0.5032 ± 0.0084	0.4737 ± 0.0062
LLSF-DL	0.0000 ± 0.0000	0.5284 ± 0.0164	0.2339 ± 0.0038	0.7093 ± 0.0126	0.5983 ± 0.0131	0.5254 ± 0.0043	0.3670 ± 0.0047
LSML	0.0050 ± 0.0055	0.5056 ± 0.0170	0.2634 ± 0.0029	0.4732 ± 0.0030	0.5133 ± 0.0073	0.4344 ± 0.0010	0.3421 ± 0.0021
CLML	0.0000 ± 0.0000	0.5062 ± 0.0000	0.3619 ± 0.0133	0.5053 ± 0.0106	0.6036 ± 0.0256	0.5414 ± 0.0118	0.5057 ± 0.0049
LRLSF	0.0000 ± 0.0000	0.5062 ± 0.0269	0.2203 ± 0.0000	0.4922 ± 0.0048	0.4951 ± 0.0084	0.5393 ± 0.0064	0.4083 ± 0.0042
LSMLLC	0.0000 ± 0.0000	0.4136 ± 0.0157	0.2403 ± 0.0056	0.4677 ± 0.0092	0.5247 ± 0.0036	0.4132 ± 0.0051	0.3387 ± 0.0025

TABLE V
RANKING_LOSS OF SIX DIFFERENT ALGORITHMS ON SEVEN DATASETS.

Algorithm	Ranking_loss↓						
	genbase	emotions	yeast	art	science	bibtex	delicious
LIFT	0.0333 ± 0.0006	0.3850 ± 0.0146	0.1835 ± 0.0015	0.1432 ± 0.0027	0.1427 ± 0.0027	0.1367 ± 0.0015	0.1471 ± 0.0005
LLSF	0.0132 ± 0.0041	0.4003 ± 0.0045	0.2649 ± 0.0063	0.1333 ± 0.0029	0.1254 ± 0.0053	0.1404 ± 0.0041	0.1777 ± 0.0014
LLSF-DL	0.0355 ± 0.0035	0.4229 ± 0.0111	0.1805 ± 0.0015	0.1735 ± 0.0006	0.1133 ± 0.0040	0.1023 ± 0.0022	0.1229 ± 0.0005
LSML	0.0067 ± 0.0019	0.3927 ± 0.0146	0.1875 ± 0.0011	0.1893 ± 0.0028	0.1682 ± 0.0018	0.1138 ± 0.0016	0.1530 ± 0.0015
CLML	0.0357 ± 0.0032	0.4218 ± 0.0196	0.2708 ± 0.0024	0.1308 ± 0.0027	0.1286 ± 0.0041	0.1539 ± 0.0035	0.1311 ± 0.0002
LRLSF	0.0351 ± 0.0035	0.4560 ± 0.0132	0.2440 ± 0.0027	0.1702 ± 0.0042	0.1443 ± 0.0017	0.1219 ± 0.0067	0.3824 ± 0.0016
LSMLLC	0.0020 ± 0.0006	0.2798 ± 0.0103	0.1805 ± 0.0015	0.1531 ± 0.0027	0.1320 ± 0.0027	0.1125 ± 0.0015	0.1520 ± 0.0005

TABLE VI
HAMMING_LOSS OF SIX DIFFERENT ALGORITHMS ON SEVEN DATASETS.

Algorithm	Hamming_loss↓						
	genbase	emotions	yeast	art	science	bibtex	delicious
LIFT	0.0086 ± 0.0030	0.3146 ± 0.0076	0.2701 ± 0.0015	0.0604 ± 0.0002	0.0384 ± 0.0005	0.0140 ± 0.0000	0.0185 ± 0.0000
LLSF	0.0361 ± 0.0030	0.3572 ± 0.0223	0.2423 ± 0.0028	0.0564 ± 0.0004	0.0345 ± 0.0002	0.0149 ± 0.0002	0.0255 ± 0.0001
LLSF-DL	0.0046 ± 0.0003	0.3844 ± 0.0137	0.3041 ± 0.0002	0.0618 ± 0.0000	0.0340 ± 0.0006	0.0148 ± 0.0000	0.0186 ± 0.0000
LSML	0.0057 ± 0.0035	0.3148 ± 0.0076	0.2666 ± 0.0011	0.0543 ± 0.0001	0.0309 ± 0.0005	0.0135 ± 0.0000	0.0185 ± 0.0001
CLML	0.0468 ± 0.0015	0.3317 ± 0.0008	0.3001 ± 0.0011	0.0619 ± 0.0000	0.0349 ± 0.0008	0.0213 ± 0.0002	0.0541 ± 0.0002
LRLSF	0.0295 ± 0.0058	0.3074 ± 0.0112	0.2740 ± 0.0042	0.0566 ± 0.0001	0.0340 ± 0.0039	0.0369 ± 0.0008	0.0368 ± 0.0002
LSMLLC	0.0040 ± 0.0030	0.3239 ± 0.0109	0.2701 ± 0.0015	0.0550 ± 0.0005	0.0308 ± 0.0005	0.0135 ± 0.0000	0.0184 ± 0.0000

TABLE VII
AUC OF SIX DIFFERENT ALGORITHMS ON SEVEN DATASETS.

Algorithm	AUC↑ genbase	emotions	yeast	art	science	bibtex	delicious
LIFT	0.9459 ± 0.0011	0.6029 ± 0.0157	0.8031 ± 0.0014	0.8205 ± 0.0029	0.8345 ± 0.0017	0.8637 ± 0.0013	0.8583 ± 0.0007
LLSF	0.9430 ± 0.0063	0.5802 ± 0.0040	0.7241 ± 0.0059	0.8208 ± 0.0025	0.8502 ± 0.0051	0.8515 ± 0.0064	0.8199 ± 0.0017
LLSF-DL	0.9433 ± 0.0040	0.5626 ± 0.0078	0.7871 ± 0.0065	0.7959 ± 0.0009	0.8567 ± 0.0034	0.8934 ± 0.0008	0.8772 ± 0.0007
LSML	0.9826 ± 0.0029	0.5694 ± 0.0157	0.7997 ± 0.0010	0.7648 ± 0.0020	0.8030 ± 0.0023	0.8798 ± 0.0013	0.8453 ± 0.0009
CLML	0.9432 ± 0.0037	0.5579 ± 0.0160	0.7157 ± 0.0023	0.8381 ± 0.0026	0.8373 ± 0.0124	0.8314 ± 0.0043	0.8717 ± 0.0002
LRLSF	0.9453 ± 0.0025	0.5339 ± 0.0097	0.7497 ± 0.0024	0.7912 ± 0.0036	0.8268 ± 0.0017	0.8801 ± 0.0022	0.7674 ± 0.0008
LSMLLC	0.9899 ± 0.0011	0.6902 ± 0.0088	0.8053 ± 0.0014	0.8115 ± 0.0029	0.8403 ± 0.0017	0.8813 ± 0.0013	0.8461 ± 0.0007

TABLE VIII
AVERAGE_PRECISION RANK OF SIX COMPARED ALGORITHMS ACROSS SEVEN DATASETS.

Algorithm	Average_precision genbase	emotions	yeast	art	science	bibtex	delicious	Average
LIFT	4	2	3	6	7	7	5	4.85714
LLSF	7	3	6	2	4	3	6	4.42857
LLSF-DL	6	4	4	7	5	4	3	4.71428
LSML	3	5	2	5	3	2	2	3.14285
CLML	5	6	7	3	6	5	7	5.57142
LRLSF	2	7	5	4	1	6	4	4.14285
LSMLLC	1	1	1	1	2	1	1	1.14285

TABLE IX
ONE_ERROR RANK OF SIX COMPARED ALGORITHMS ACROSS SEVEN DATASETS.

Algorithm	One_error genbase	emotions	yeast	art	science	bibtex	delicious	Average
LIFT	1	2	4	6	7	4	4	4.00000
LLSF	1	6	6	3	4	3	6	4.14285
LLSF-DL	1	7	2	7	5	5	3	4.28571
LSML	7	3	5	2	2	1	2	3.14285
CLML	1	4	7	5	6	7	7	5.28571
LRLSF	1	5	1	4	1	6	5	3.28571
LSMLLC	1	1	3	1	3	2	1	1.71428

TABLE X
RANKING_LOSS RANK OF SIX COMPARED ALGORITHMS ACROSS SEVEN DATASETS.

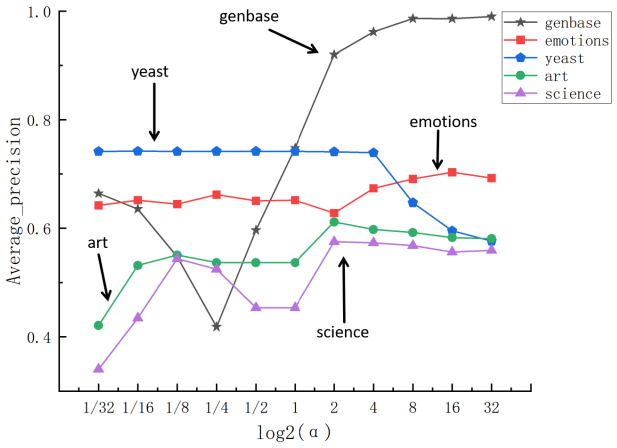
Algorithm	Ranking_loss genbase	emotions	yeast	art	science	bibtex	delicious	Average
LIFT	4	2	3	3	5	5	3	3.57142
LLSF	3	4	6	2	2	6	6	4.14285
LLSF-DL	6	6	1	6	1	1	1	3.14285
LSML	2	3	4	7	7	3	5	4.42857
CLML	7	5	7	1	3	7	2	4.57142
LRLSF	5	7	5	5	6	4	7	5.57142
LSMLLC	1	1	2	4	4	2	4	2.57142

TABLE XI
HAMMING_LOSS RANK OF SIX COMPARED ALGORITHMS ACROSS SEVEN DATASETS.

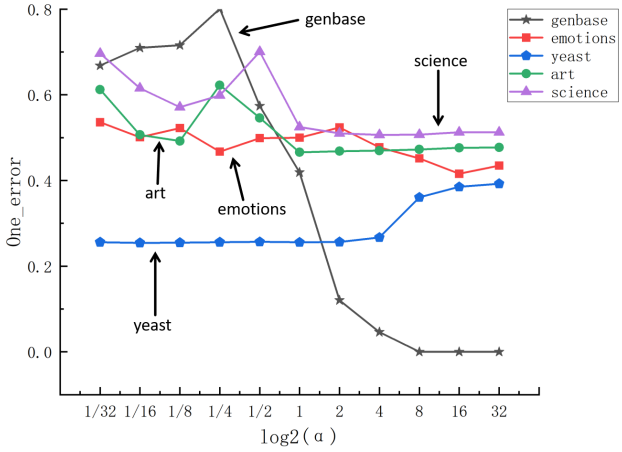
Algorithm	Hamming_loss genbase	emotions	yeast	art	science	bibtex	delicious	Average
LIFT	4	2	3	5	7	5	2	4.00000
LLSF	6	6	1	3	5	6	5	4.57142
LLSF-DL	2	7	7	6	3	1	4	4.28571
LSML	3	3	2	2	2	3	2	2.42857
CLML	7	5	5	7	3	7	6	5.71428
LRLSF	5	1	6	4	6	4	7	4.71428
LSMLLC	1	4	3	1	1	1	1	1.71428

TABLE XII
AUC RANK OF SIX COMPARED ALGORITHMS ACROSS SEVEN DATASETS.

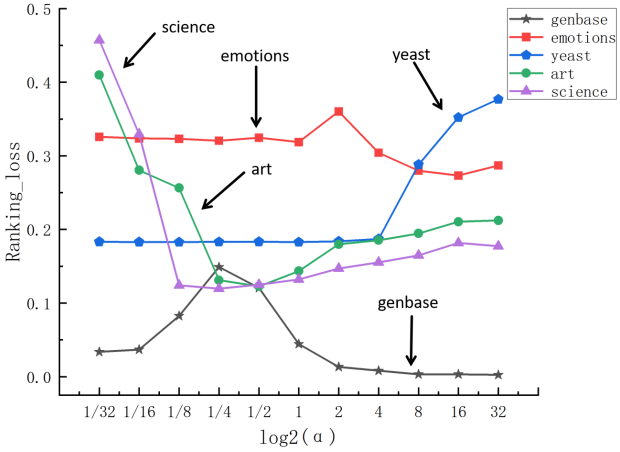
Algorithm	AUC genbase	emotions	yeast	art	science	bibtex	delicious	Average
LIFT	3	2	2	3	5	5	3	3.28571
LLSF	7	3	6	2	2	6	6	4.57142
LLSF-DL	5	5	4	5	1	1	1	3.14285
LSML	2	4	3	7	7	4	5	4.57142
CLML	6	6	7	1	4	7	2	4.71428
LRLSF	4	7	5	6	6	3	7	5.42857
LSMLLC	1	1	1	4	3	2	4	2.28571



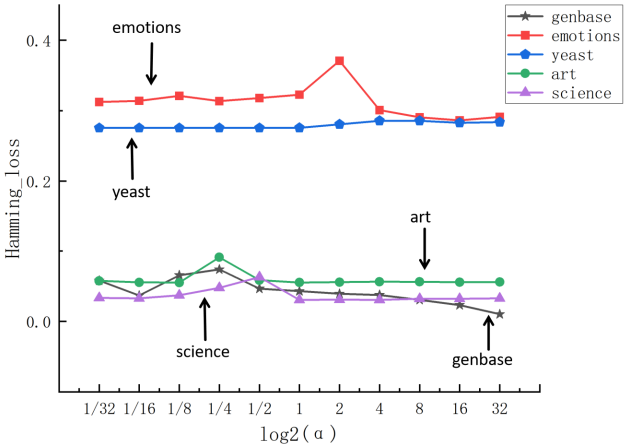
(a) Average precision



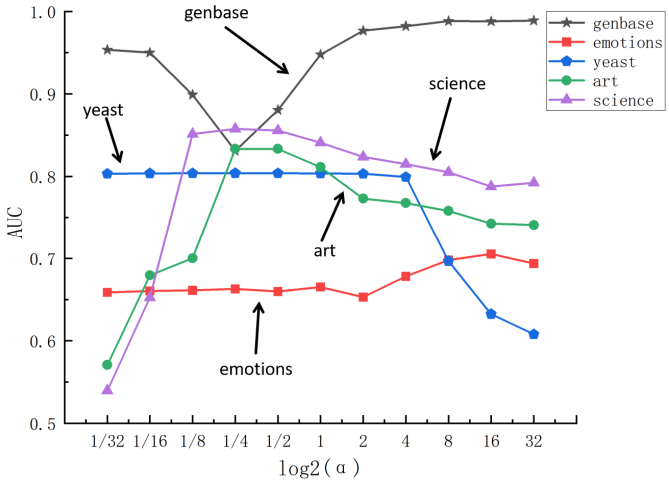
(b) One error



(c) Ranking loss

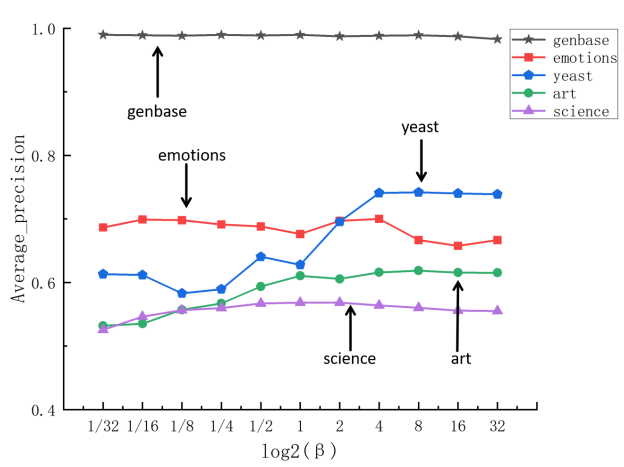


(d) Hamming loss

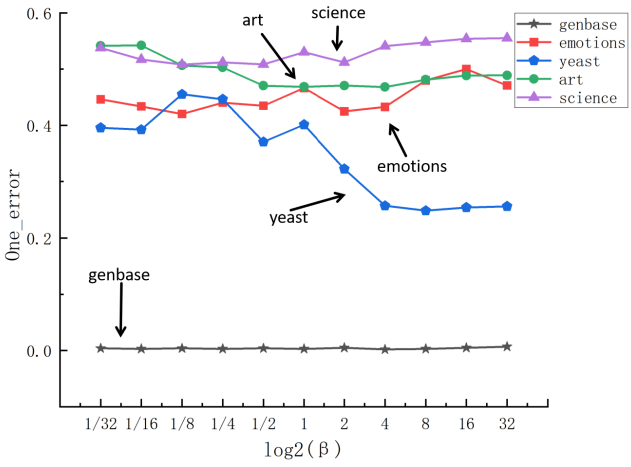


(e) AUC

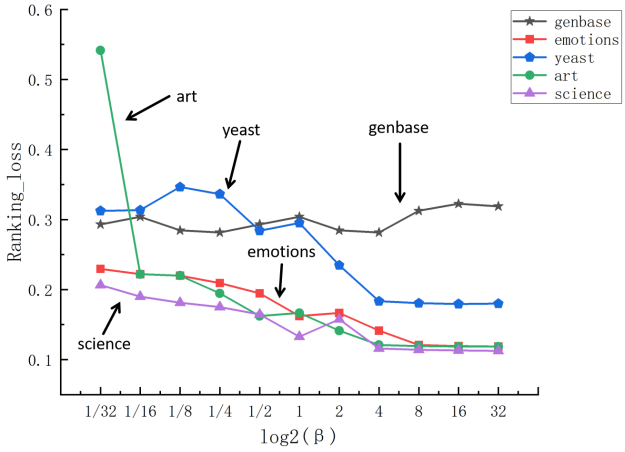
Fig.1. The experimental results of LSMLCL as α increasing across five datasets.



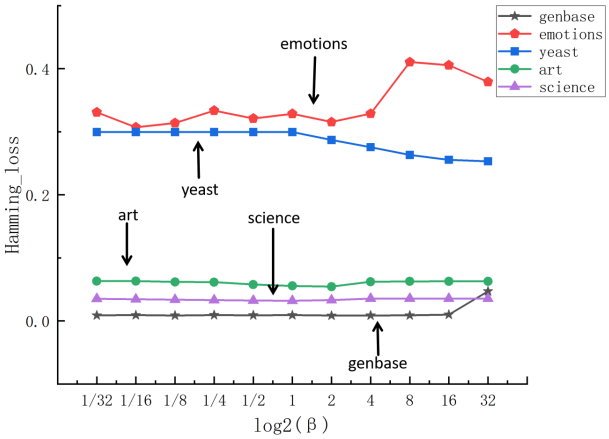
(a) Average precision



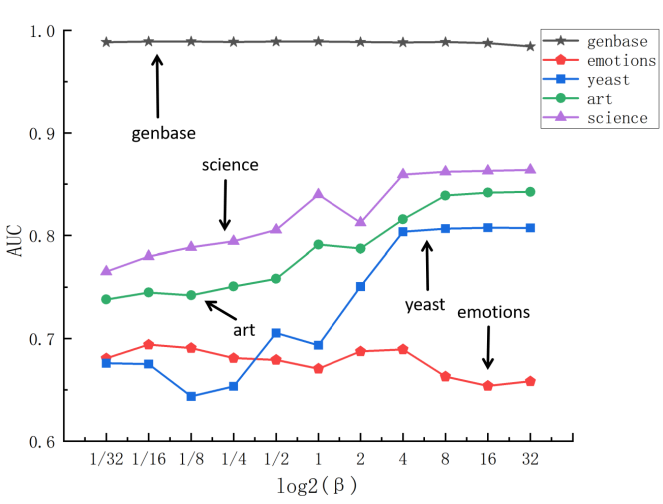
(b) One error



(c) Ranking loss

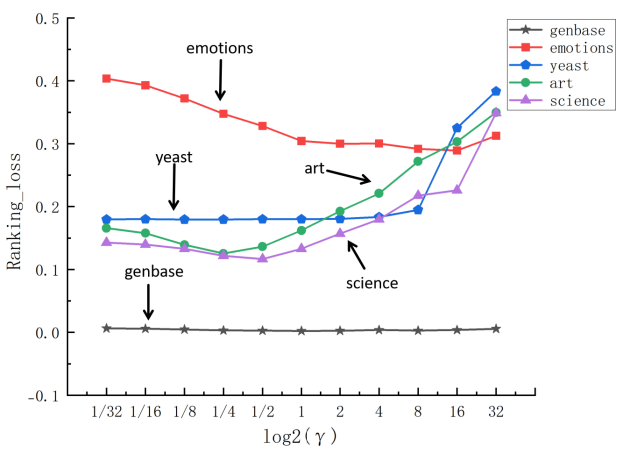


(d) Hamming loss

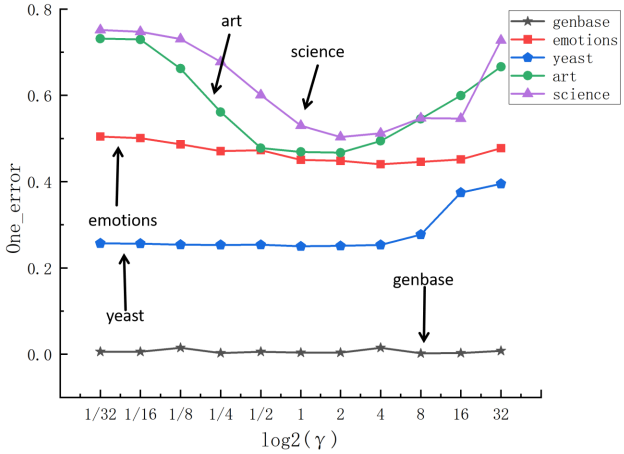


(e) AUC

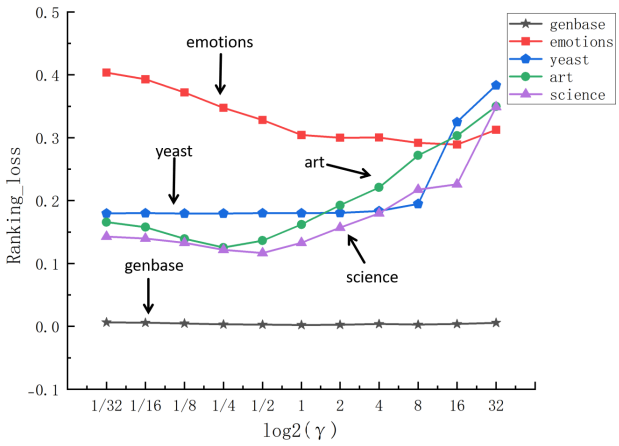
Fig.2. The experimental results of LSMLCL as β increasing across five datasets.



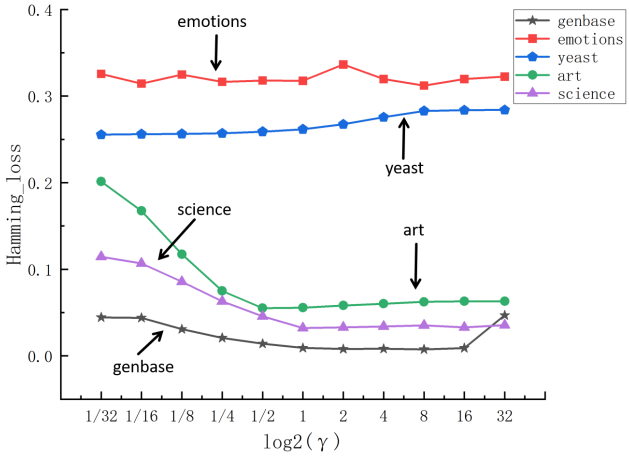
(a) Average precision



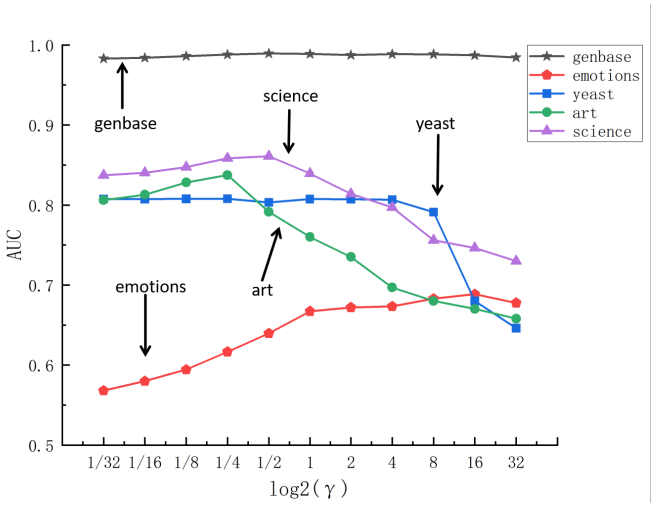
(b) One error



(c) Ranking loss

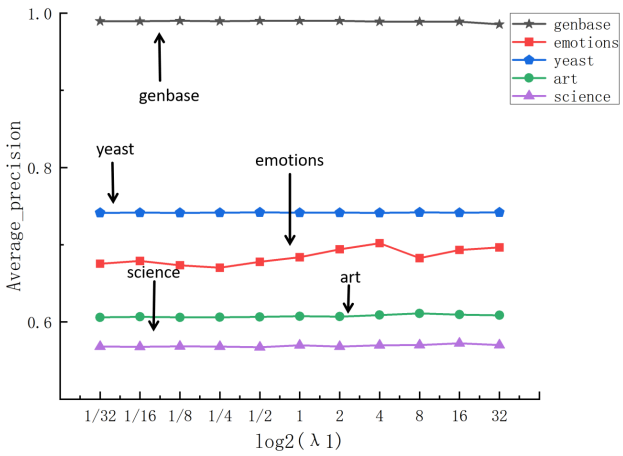


(d) Hamming loss

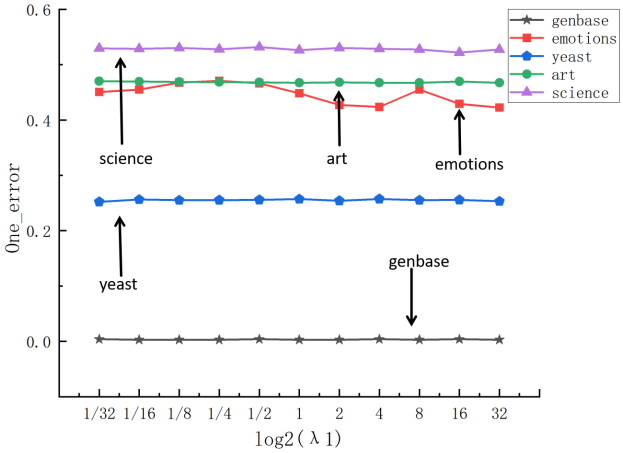


(e) AUC

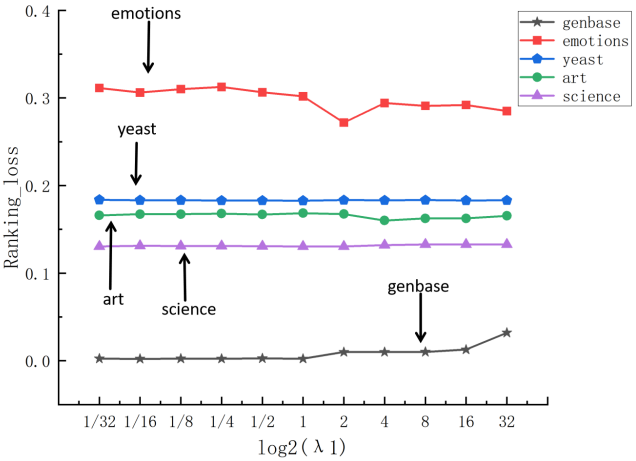
Fig.3. The experimental results of LSMLCL as γ increasing across five datasets.



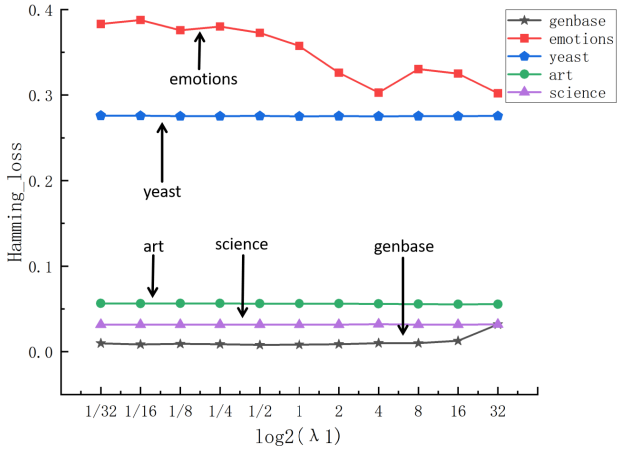
(a) Average precision



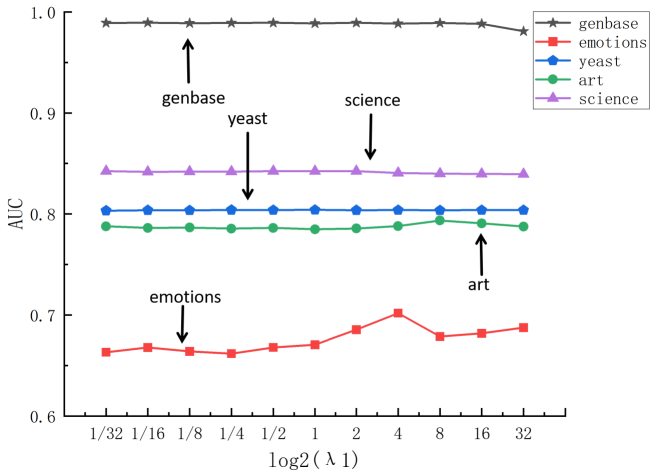
(b) One error



(c) Ranking loss

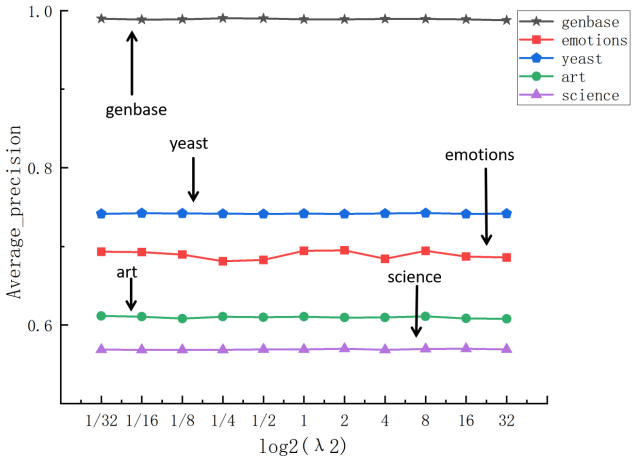


(d) Hamming loss

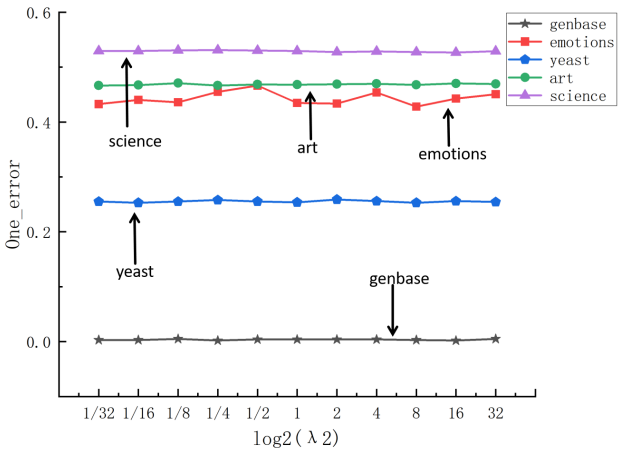


(e) AUC

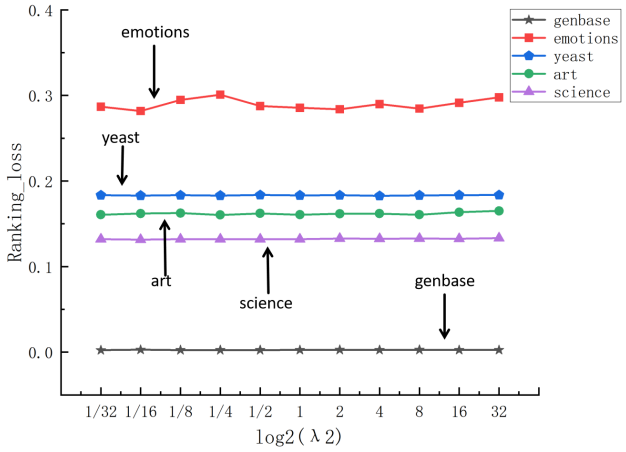
Fig.4. The experimental results of LSMLCL as λ_1 incresing across five datasets.



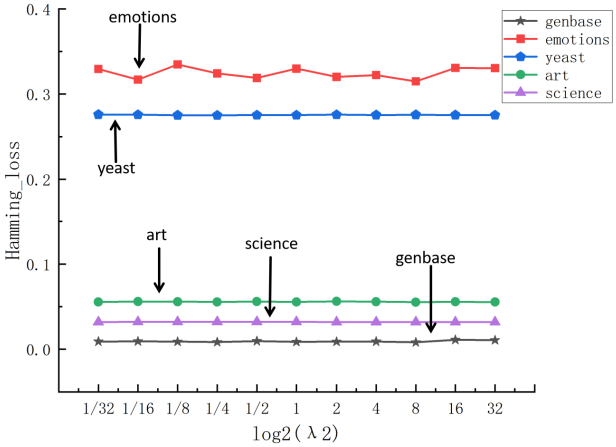
(a) Average precision



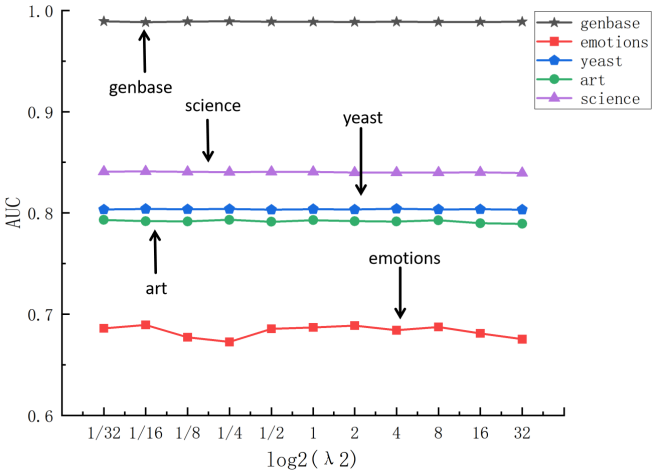
(b) One error



(c) Ranking loss

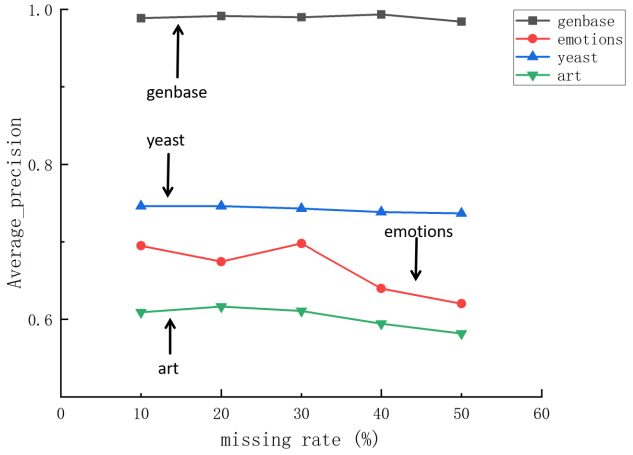


(d) Hamming loss

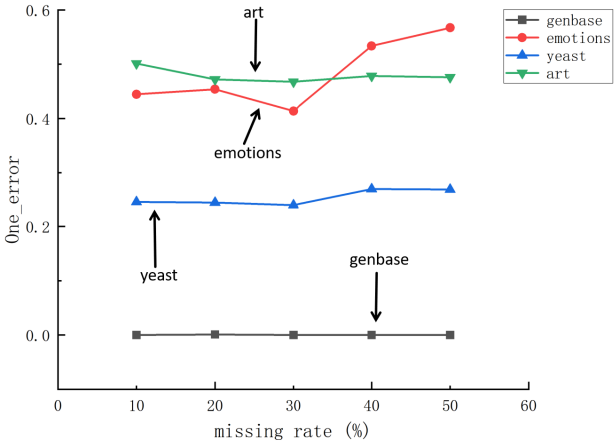


(e) AUC

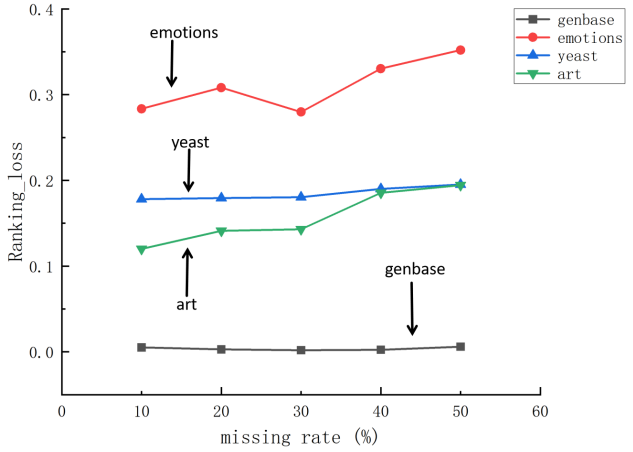
Fig.5. The experimental results of LSMLCL as λ_2 incresing across five datasets.



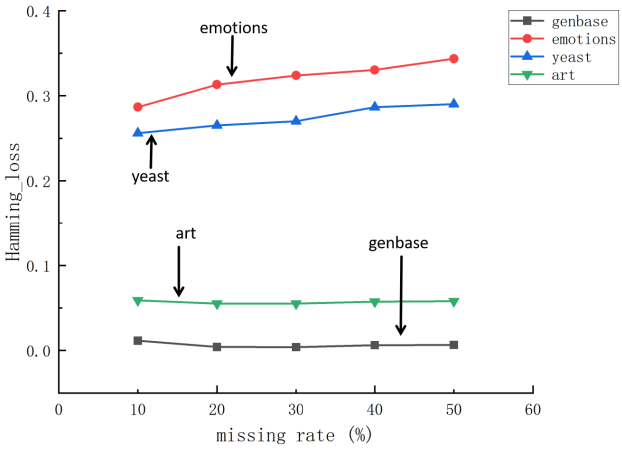
(a) Average precision



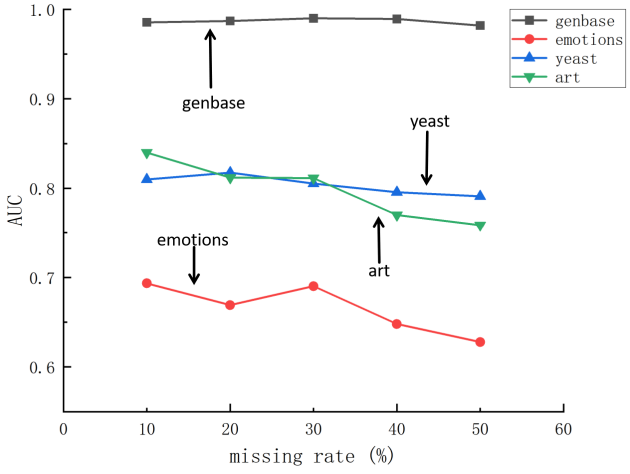
(b) One error



(c) Ranking loss

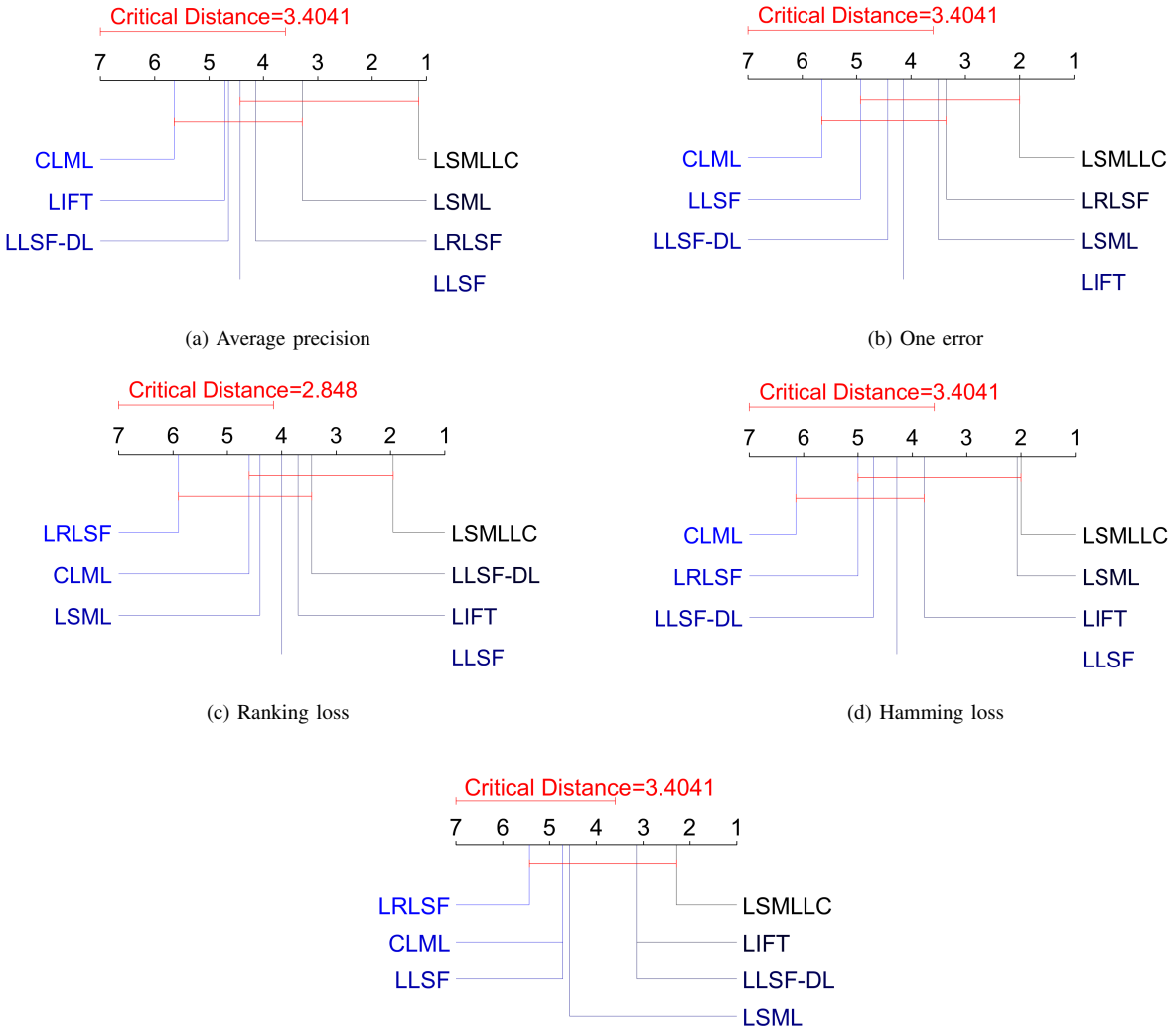


(d) Hamming loss



(e) AUC

Fig.6. The experimental results of the missing ratio of LSMLCL across four datasets.



(a) Average precision

(b) One error

(c) Ranking loss

(d) Hamming loss

(e) AUC

Fig.7. The experimental results of LSMLCL as α increasing across five datasets.

B. Comparative algorithms

We conduct a comparative analysis of LSMLLC against six state-of-the-art algorithms. Meanwhile, we select five evaluation metrics: Average_precision, One_error, Ranking_loss, Hamming_loss and AUC [32], [33]. The descriptions and parameter configurations for these algorithms are detailed as follows:

LIFT [10] extracts label-specific features through class clusters and subsequently employs LIBSVM [34] for classification. It includes one parameter λ_1, which is configured to 0.1 in the experimental arrangement.

LLSF [11] utilizes label-specific features and label correlations learning, which is the pioneering method for leveraging label-specific features. The parameter values λ_1 and λ_2 range from 2^{-5} to 2^5 with a step of 1, where threshold $\tau = 0.5$.

LLSF-DL [11] learns class-dependent labels in a sparse superposition and combines higher-order label correlations. The parameter values λ_1, λ_2, and λ_3 range from 2^{-5} to 2^5 with a step of 1, where the threshold $\tau = 0.5$.

LSML [12] is a multi-label classification algorithm designed to tackle the problem of missing labels. It learned label-specific features for each label in the new complete label matrix. The parameter values $\lambda_1, \lambda_2, \lambda_3$, and λ_4 range from 2^{-5} to 2^5 with a step of 1, where the threshold $\tau = 0$.

CLML [13] both uses label-specific features and common features for multi-label classification and investigated label correlations directly through the labels themselves rather than relying on coefficient matrices. The parameter values $\lambda_1, \lambda_2, \lambda_3$, and λ_4 range from 2^{-5} to 2^5 with a step of 1, where the threshold $\tau = 0.5$.

LRLSF [35] proposes a robust global label correlation strategy by self-expression matrix and incorporates a manifold regularization term to capture local label correlations. The parameter values $\lambda_1, \lambda_2, \lambda_3$, and λ_4 range from 2^{-5} to 2^5 with a step of 1, where the threshold $\tau = 0.5$.

C. Experimental results

We assess the classification performance of each algorithm across five benchmark datasets through five-fold cross-validation. The detailed results are shown in Tables III through VII, and the rankings based on these results are shown in Tables VIII through XII. For each evaluation metric, '↓' indicates that a smaller value signifies better performance, whereas '↑' indicates that a larger value signifies better performance. As shown in Tables VIII through XII, our LSMLLC method consistently surpasses LIFT [10], LLSF [11], LLSF-DL [11], LSML [12], CLML [13], and LRLSF [35] across seven datasets in terms of average metrics. To assess the notable performance differences between our approach and the competing algorithms, we conducted a Friedman test [36]. Table I reports each evaluation metric along with its corresponding F_F and critical value. At an important level of $\alpha = 0.05$, we dismissed the null hypothesis suggesting that all competing algorithms perform equally, as the F_F exceeded the critical value, indicating significant differences among the competing algorithms. We utilized the Nemenyi test [36] to assess if LSMLLC demonstrates superior performance in comparison to other algorithms, designating LSMLLC as the reference algorithm. Compare the average rank differences between any pair of algorithms with the critical difference (CD) using the Nemenyi test, where $CD = q_\alpha \sqrt{\frac{6N}{N-1}}$, $CD = 3.4041(K = 7, N = 5)$ at a significance level of $\alpha = 0.05$. The CD diagrams for each evaluation metric are shown in Fig. 7. Each subplot displays a comparison of algorithms, where two algorithms are connected by a line if their average ranks fall within the same critical difference (CD) range. In other words, algorithms that are not connected by a line are considered to have a significant difference in performance for that evaluation metric. It is clear from the diagrams that LSMLLC outperforms the other six algorithms.

D. Parameter sensitivity analysis

We use five parameters, namely α, β, γ, λ_1, and λ_2. In this context, parameter α regulates the discrepancy between the recovered label matrix and the original incomplete label matrix. Parameters β and γ regulate the contributions of the global and local label correlations, respectively. Parameters λ_1 and λ_2 regulate the sparsity of label-specific features and the label correlations coefficient matrix, respectively. We set the values of all parameters in the range of $[2^{-5}, 2^5]$ and conduct experiments by varying one parameter at a time while maintaining the others at their optimal settings. Fig. 1 through 5 demonstrate the effect of these parameters across five datasets: Genbase, Emotions, Yeast, Arts, and Science.

Specifically, Fig. 1 through 5 show how five parameters affect five datasets. It is clear that small changes in the α value lead to noticeable variations in performance and results. In other words, if the weight between the original missing matrix and the new recovery label matrix is very low, it is difficult to complete the missing label. As the value of α increases, its performance tends to stabilize and become excellent. Increasing the value of β can significantly enhance performance, as it indicates a higher importance of global label correlations. When the value of γ is between $\frac{1}{4}$ and 4, the overall performance is optimized, suggesting that local label correlations within this range have the most significant effect on performance. The other two parameters λ_1 and λ_2 have little impact on the overall performance.

We not only analyze the impact of each parameter on various metrics but also conduct a detailed discussion on the missing rate metric of our algorithm across four datasets. Fig. 6 shows how different missing rates affect the four datasets: Genbase, Emotions, Yeast, and Arts.

The performance of most evaluation metrics tends to deteriorate as the missing rate increases, although certain metrics may occasionally improve due to the random nature of the missing data.

V. Conclusion

In this paper, we propose a novel method called LSMLLC, designed to effectively learn label correlations and label-specific features for multi-label classification with missing labels. Extensive experimental results demonstrate that missing labels can severely impact the performance of multi-label classifiers. Our LSMLLC algorithm achieves superior performance compared to other state-of-the-art methods across seven benchmark datasets. Additionally, we confirm the critical role of label correlations and label-specific features in addressing the challenges posed by missing labels. Future

research will focus on extending this algorithm into a multi-view learning framework.

REFERENCES

[1] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[2] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73:133–153, 2008.

[3] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE transactions on knowledge and data engineering*, 23(7):1079–1089, 2010.

[4] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85:333–359, 2011.

[5] Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In *European conference on principles of data mining and knowledge discovery*, pages 42–53. Springer, 2001.

[6] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. *Advances in neural information processing systems*, 14, 2001.

[7] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[8] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200, 2005.

[9] Wang, Jiayin, Zhou, Ronghe, Wang, Yukun, and Li, Zhongfeng. Efficient Improved Henry Gas Solubility Optimization and Its Application in Feature Selection Problems. *Engineering Letters*, vol. 32, no 10, pp2023-2040, 2024.

[10] Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):107–120, 2014.

[11] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE transactions on knowledge and data engineering*, 28(12):3309–3323, 2016.

[12] Jun Huang, Feng Qin, Xiao Zheng, Zekai Cheng, Zhixiang Yuan, Weigang Zhang, and Qingming Huang. Improving multi-label classification with missing labels by learning label-specific features. *Information Sciences*, 492:124–146, 2019.

[13] Junlong Li, Peipei Li, Xuegang Hu, and Kui Yu. Learning common and label-specific features for multi-label classification with correlation information. *Pattern recognition*, 121:108259, 2022.

[14] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

[15] Juan A Romero-del Castillo, Manuel Mendoza-Hurtado, Domingo Ortiz-Boyer, and Nicolás García-Pedrajas. Local-based k values for multi-label k-nearest neighbors rule. *Engineering Applications of Artificial Intelligence*, 116:105487, 2022.

[16] Martha Roseberry, Bartosz Krawczyk, Youcef Djenouri, and Alberto Cano. Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams. *Neurocomputing*, 442:10–25, 2021.

[17] Yonghao Li, Liang Hu, and Wanfu Gao. Multi-label feature selection via robust flexible sparse regularization. *Pattern Recognition*, 134:109074, 2023.

[18] Yue Zhu, James T Kwok, and Zhi-Hua Zhou. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1081–1094, 2017.

[19] Xiaozhen Fu, Deyu Li, and Yanhui Zhai. Multi-label learning with kernel local label information. *Expert Systems with Applications*, 207:118027, 2022.

[20] Yuling Fan, Baihua Chen, Weiqin Huang, Jinghua Liu, Wei Weng, and Weiyao Lan. Multi-label feature selection based on label correlations and feature redundancy. *Knowledge-Based Systems*, 241:108256, 2022.

[21] Jia Zhang, Hanrui Wu, Min Jiang, Jinghua Liu, Shaozi Li, Yong Tang, and Jinyi Long. Group-preserving label-specific feature selection for multi-label learning. *Expert Systems with Applications*, 213:118861, 2023.

[22] Zhi-Fen He, Ming Yang, Yang Gao, Hui-Dong Liu, and Yilong Yin. Joint multi-label classification and label correlations with missing labels and feature selection. *Knowledge-Based Systems*, 163:145–158, 2019.

[23] Reshma Rastogi and Sayed Mortaza. Multi-label classification with missing labels using label correlation and robust structural learning. *Knowledge-Based Systems*, 229:107336, 2021.

[24] Sanjay Kumar and Reshma Rastogi. Low rank label subspace transformation for multi-label learning with missing labels. *Information Sciences*, 596:53–72, 2022.

[25] Ziwei Cheng and Ziwei Zeng. Joint label-specific features and label correlation for multi-label learning with missing label. *Applied Intelligence*, 50(11):4029–4049, 2020.

[26] Lei Feng, Jun Huang, Senlin Shu, and Bo An. Regularized matrix factorization for multilabel learning with missing labels. *IEEE transactions on cybernetics*, 52(5):3710–3721, 2020.

[27] Lijuan Sun, Songhe Feng, Jun Liu, Gengyu Lyu, and Congyan Lang. Global-local label correlation for partial multi-label learning. *IEEE Transactions on Multimedia*, 24:581–593, 2021.

[28] Qing Ai, Fei Li, Xiangna Li, Ji Zhao, Wenhui Wang, Qingyun Gao, and Fei Zhao. An improved mltsvm using label-specific features with missing labels. *Applied Intelligence*, 53(7):8039–8060, 2023.

[29] Ai, Qing, Li, Fei, Gao, Qingyun, and Zhao, Fei. An Improved LST-KSVC Based on Energy Model. *Engineering Letters*, vol. 30, no. 4, pp 1436-1443, 2022.

[30] Li, Fei, Ai, Qing, Li, Xiangna, Wang, Wenhui, Gao, Qingyun, and Zhao, Fei. Intuitionistic fuzzy least squares MLTSVM for noisy label data using label-specific features and local label correlation. *Expert Systems with Applications*, 260:125351, 2025.

[31] Keerthika, V., and Prahalatha, R. Series Solution of Non-linear Initial Value Problems on Certain Physical Systems through MATLAB. *IAENG International Journal of Applied Mathematics*, vol. 54, no. 10, pp 2118-2127, 2024.

[32] Yu, Lie, Hu, Gaotong, Ding, Lei, Luo, Na, and Zhang, Yong. Gait Pattern Recognition based on Multi-sensors Information Fusion through PSO-SVM Model. *Engineering Letters*, vol. 32, no. 5, pp 974-980 2024.

[33] Zhang, Lan, Hu, Lei, and Yu, Lina. The Feasibility of Solving the Satisfiability Problem Using Various Machine Learning Approaches. *Engineering Letters*, vol.32, no. 5, pp 1004-1011 2024.

[34] CHANG Chih-C. Libsvm: a library for support vector machines. *http://www. csie. ntu. edu. tw/˜ cjlin/libsvm*, 2001.

[35] Yong Yang, Hongmei Chen, Yong Mi, Chuan Luo, Shi-Jinn Horng, and Tianrui Li. Multi-label feature selection based on stable label relevance and label-specific features. *Information Sciences*, 648:119525, 2023.

[36] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.