Stable Label-Specific Features Generation for Multi-label Learning via Mixture-based Clustering Ensemble

Yi-Bo Wang, Jun-Yi Hang, and Min-Ling Zhang

***Abstract*—Multi-label learning deals with objects associated with multiple class labels, and aims to induce a predictive model which can assign a set of relevant class labels for an unseen instance. Since each class might possess its own characteristics, the strategy of extracting label-specific features has been widely employed to improve the discrimination process in multi-label learning, where the predictive model is induced based on the tailored features specific to each class label instead of the identical instance representations. As a representative approach, LIFT generates the label-specific features by conducting clustering analysis. However, its performance may be degraded due to the inherent instability of the single clustering algorithm. To improve this, a novel multi-label learning approach named SENCE (*stable label-Specific features gENeration for multi-label learning via mixture-based Clustering Ensemble*) is proposed, which stabilizes the generation process of the label-specific features via clustering ensemble techniques. Specifically, more stable clustering results are obtained by firstly augmenting the original instance representation with the cluster assignments from base clusters and then fitting a mixture model via the EM algorithm. Extensive experiments on seventeen benchmark data sets show that SENCE performs better than LIFT and other well-established multi-label learning algorithms.**

***Index Terms*—Multi-label learning, label-specific features, clustering ensemble, Expectation-Maximization algorithm.**

I. INTRODUCTION

MULTI-LABEL learning aims to build classification models for objects assigned with multiple semantics simultaneously, where each example is represented by a single instance and a set of relevant class labels [1]. As multi-label objects widely exist in the real world, multi-label learning has diverse applications, such as text categorization [2], image annotation [3], web mining [4], and bioinformatics analysis [5], etc.

In recent years, significant amount of algorithms have been proposed for multi-label learning. One common strategy adopted by the most existing approaches is to build a predictive model based on the identical instance representations for each class label [1]. However, this strategy might be suboptimal as each class label is supposed to have distinct characteristics of its own. For instance, in text categorization, features corresponding to word terms *voting*, *reform* and *government* would be informative in discriminating political and non-political documents, while features related to world term *piano*, *Mozart* and *sonata* would be informative in discriminating musical and non-musical documents. Therefore, the strategy of *label-specific features* [6] has been proposed to benefit the discrimination of different class labels.

As a representative approach for label-specific features, LIFT [6] utilizes clustering techniques to investigate the underlying properties of the feature space for each class label. Nevertheless, the clustering in LIFT tends to be unstable due to the inherent instability of the single clustering method [7]. To address this, clustering ensemble techniques [8]–[10] can be utilized to obtain clustering results with stronger stability and robustness. With the assumption that the clustering results of related labels should be similar, LIFTACE [8] employs clustering ensemble techniques to integrate the preliminary clustering results of all class labels based on the consensus similarity matrix. However, it fails to utilize the information embodied in the original data representation during the combination process of clustering ensemble.

To address above issues, a novel approach named SENCE, i.e. *stable label-Specific features gENeration for multi-label learning via mixture-based Clustering Ensemble*, is proposed, which stabilizes the clustering process via a two-stage method. Firstly, several base clusters are exploited to conduct clustering analysis on positive and negative instances of each class label. Then, base cluster assignments are combined via a tailored EM procedure, where a mixture model is fitted on clustering-augmented instances. After that, a predictive model is induced based on the label-specific features derived from the improved generation strategy.

In this paper, we advance label-specific features generation via a novel clustering combination strategy, which is an essential step in clustering ensemble. The novel strategy can fully leverage the information hidden in the original data representation and encoded in each cluster assignment to avoid the suboptimal results of existing techniques. Comprehensive experiments over 17 benchmark data sets indicate the effectiveness of SENCE.

The rest of this paper is organized as follows. Section II briefly reviews related works on multi-label learning. Section III presents the proposed approach SENCE. Section IV reports the experimental results on 17 benchmark datasets. Finally, Section V concludes.

II. RELATED WORKS

The task of multi-label learning has been extensively studied in recent years. Generally, the major challenge for multi-

Yi-Bo Wang, Jun-Yi Hang and Min-Ling Zhang are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: {wang_yb,hangjy,zhangml}@seu.edu.cn).

Manuscript received X, X; revised September X, X.

label learning is its huge output space which is exponential to the number of class labels. Therefore, exploiting *label correlations* is regarded as a common strategy to facilitate the learning process. Roughly speaking, existing approaches can be grouped into three categories based on the order of correlations [1], [11], i.e. *first-order* approaches, *second-order* approaches and *high-order* approaches. First-order approaches tackle multi-label learning problem in a label-by-label manner [3], [12]. Second-order approaches exploit pairwise relationships between class labels [13], [14]. High-order approaches exploit relationships among a subset of class labels or all class labels [15]–[17].

In addition to exploiting label correlations in the output space, another strategy for facilitating multi-label learning is to manipulate the input space. The most straightforward feature manipulation strategy is to conduct dimensionality reduction [18]–[20] or feature selection [21]–[24], which is also a common strategy used in multi-class learning, over the original feature space. Besides, there are also some other ways, such as generating meta-level features [25], [26] with strong discriminative information from the original representation, constructing multi-view representations for multi-label data [27]–[29], etc. Note that all these feature manipulation strategies employ identical feature representation for all labels in the discrimination process.

Instead, label-specific features generation serves as an alternative feature manipulation strategy, which extracts the most discriminative features for each individual label. Some works generate label-specific features by selecting a different subset of the original features for each class label [30]–[33]. Based on the sparse assumption, the most pertinent and discriminative features for each label can be identified using spectral clustering and LASSO algorithms [34].

In addition to conducting label-specific feature selection in the original feature space, it is also feasible to derive label-specific features from a transformed feature space. For example, LIFT [6] performs clustering analysis on the positive and negative instances of each class label, and generates label-specific features by querying the distances between the instance and the clustering centers. To improve this, attribute reduction [35] can be employed in the process of label-specific features construction to remove redundant information in generated label-specific features. Some other works aim to enrich the label-specific features by exploiting the nearest neighbor rule [36], exploring spatial topology structure [37], jointly considering label-specific features generation and classification model induction [38], generating BiLabel-specific features based on heuristic prototype selection and embedding [39], or imposing structured sparsity regularization over the label-specific features [40].

Recently, clustering ensemble techniques have been considered to enhance the process of label-specific features generation. However, the off-the-shelf clustering ensemble techniques employed in previous methods fail to utilize the information embodied in the original data representation [8], [41]. In this paper, we propose a novel clustering ensemble strategy for label-specific feature generation, where the information hidden in the original data representation and encoded in each cluster assignment is taken into consideration simultaneously to facilitate the generation of more stable clustering. We will detail our approach in the next section.

III. The Proposed Approach

A. Preliminaries

Formally, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space including q class labels. Given the multi-label training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$ where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T \in \mathcal{X}$ is the d-dimensional feature vector and $Y_i \subseteq \mathcal{Y}$ is the set of relevant labels associated with $\boldsymbol{x}_i$, the task of multi-label learning is to induce a predictive model $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from $\mathcal{D}$ which can assign a set of relevant labels $h(\boldsymbol{u}) \subseteq \mathcal{Y}$ for an unseen instance $\boldsymbol{u} \in \mathcal{X}$. Specifically, LIFT learns from $\mathcal{D}$ by taking two steps i.e. *label-specific features construction* and *predictive model induction*.

In the first step, for each class label $l_k \in \mathcal{Y}$, instances are divided into positive set and negative set as follows:

$$\begin{aligned} \mathcal{P}_k &= \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D}, l_k \in Y_i\} \\ \mathcal{N}_k &= \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D}, l_k \notin Y_i\} \end{aligned} \tag{1}$$

Then LIFT performs k-means to partition both sets into m_k disjoint clusters where clustering centers are denoted as $\{\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, \ldots, \boldsymbol{p}_{m_k}^k\}$ and $\{\boldsymbol{n}_1^k, \boldsymbol{n}_2^k, \ldots, \boldsymbol{n}_{m_k}^k\}$ respectively. Thereafter, the mapping $\phi_k : \mathcal{X} \rightarrow \mathcal{Z}_k$ from the original d-dimensional input space $\mathcal{X}$ to the $2m_k$-dimensional label-specific feature space w.r.t. l_k can be created as follows:

$$\phi_k(\boldsymbol{x}) = [d(\boldsymbol{x}, \boldsymbol{p}_1^k), \ldots, d(\boldsymbol{x}, \boldsymbol{p}_{m_k}^k), d(\boldsymbol{x}, \boldsymbol{n}_1^k), \ldots, d(\boldsymbol{x}, \boldsymbol{n}_{m_k}^k)] \tag{2}$$

Here, $d(\cdot, \cdot)$ returns the Euclidean distance between two feature vectors.

In the second step, a new binary training set B_k is constructed from the original training set $\mathcal{D}$ according to the label-specific features generated by the mapping ϕ_k:

$$\mathcal{B}_k = \{(\phi_k(\boldsymbol{x}_i), Y_i(k)) \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D}\} \tag{3}$$

where $Y_i(k) = +1$ if $l_k \in Y_i$ and $Y_i(k) = -1$ otherwise. Based on $\mathcal{B}_k$, a classification model $g_k : \mathcal{Z}_k \rightarrow \mathbb{R}$ for l_k is induced by invoking any binary learner $\mathfrak{L}$. Given an unseen instance $\boldsymbol{u} \in \mathcal{X}$, its relevant label set is predicted as:

$$Y = \{l_k \mid g_k(\phi_k(\boldsymbol{u})) > 0, 1 \leq k \leq q\} \tag{4}$$

B. SENCE

SENCE learns from $\mathcal{D}$ by taking four elementary stages, which aims to induce a multi-label classification model with the generated label-specific features. The first two stages are designed to stabilize the clustering process via clustering ensemble techniques. Specifically, the first stage augments the original instance representations based on the cluster assignments from base clusters. The second stage fits a mixture model on augmented instances via the EM algorithm to obtain more stable clustering results. The third stage constructs label-specific features, and the fourth stage induces the predictive models, which are consistent with the corresponding stages in LIFT. To facilitate understanding, the notations set in SENCE are summarized in Table I.

TABLE I: The set of notations for SENCE.

Notations	Description
m	number of training examples
d	number of features in input space
q	number of class labels in label space
$\mathcal{X}$	the d dimensional feature space, i.e. $\mathcal{X} = \mathbb{R}^d$
$\mathcal{Y}$	the label space where $\mathcal{Y} = \{l_1, l_2, \dots, l_q\}$
r	the number of base clusters
m_k	the number of mixture components w.r.t. class label l_k
α_j	the mixing coefficient of jth mixture component
$\boldsymbol{\mu}_j$	the d dimensional mean vector of jth mixture component
$\boldsymbol{\Sigma}_j$	the covariance matrix of jth mixture component
$v_{pj}(l)$	The probability of the instance belonging to the lth cluster in pth base cluster of jth mixture component
$\mathcal{D}$	the multi-label training set where $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$
$\boldsymbol{x}_i$	the ith feature vector where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T \in \mathcal{X}$
$\boldsymbol{t}_i$	the ith cluster assignment vector where $\boldsymbol{t}_i = (t_i^1, t_i^2, \dots, t_i^r)$
Y_i	the ith set of relevant labels where $Y_i \subseteq \mathcal{Y}$
$\boldsymbol{u}$	the unseen instance where $\boldsymbol{u} \in \mathcal{X}$

1) Clustering-based Feature Augmentation: For each class label l_k, SENCE divides instances into positive set and negative set donated as $\mathcal{P}_k$ and $\mathcal{N}_k$ respectively according to Eq.(1). To mitigate the inherent instability of the single clustering method, different from LIFT, SENCE employs multiple base clusters on $\mathcal{P}_k$ and $\mathcal{N}_k$ to derive cluster assignments and re-represents $\mathcal{P}_k$ and $\mathcal{N}_k$ as follows:

$$\begin{aligned} \overline{\mathcal{P}_k} &= \{[\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{x}_i \in \mathcal{P}_k\} \\ \overline{\mathcal{N}_k} &= \{[\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{x}_i \in \mathcal{N}_k\} \end{aligned} \quad (5)$$

Here, $\boldsymbol{t}_i = (t_i^1, t_i^2, \dots, t_i^r)$ is a *cluster assignment vector*, where r is the number of base clusters and the pth element indicates the cluster assignment given by the pth base cluster. The cluster assignment vector $\boldsymbol{t}_i$ is regarded as extra features to augment the original instance $\boldsymbol{x}_i$. Thus, such feature representation of instances in $\overline{\mathcal{P}_k}$ and $\overline{\mathcal{N}_k}$ can fully encode the information embodied in the original data representation and the cluster assignments, which makes the following label-specific features extraction more stable and robust.

2) Clustering Combination via A Mixture Model: Existing clustering ensemble methods work in two steps, i.e. clustering generation and clustering combination. In the clustering generation step, similar to existing clustering ensemble methods, SENCE exploits several base clusters to conduct clustering analysis on positive and negative instances of each class label. As the original features and the augmented features are generated in different ways, existing clustering combination methods might be suboptimal. Thus, in the clustering combination step, instead of directly combining base cluster assignments as existing clustering ensemble methods do, SENCE innovatively performs another clustering analysis on augmented instances which treat the original features and the augmented features in different ways. This novel clustering combination strategy can leverage the information hidden in the original data representation and encoded in each cluster assignment to facilitate the generation of more stable clustering.

Assume that instances in $\overline{\mathcal{P}_k}$ are drawn from a finite mixture distribution parameterized by $\boldsymbol{\Theta} = \{\alpha_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \boldsymbol{\vartheta}_j \mid 1 \leq j \leq m_k\}$, i.e.

$$\begin{aligned} P([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{\Theta}) &= \sum_{j=1}^{m_k} \alpha_j P_j([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{\theta}_j) \\ &= \sum_{j=1}^{m_k} \alpha_j P_j(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) P_j(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_j) \end{aligned} \quad (6)$$

where m_k is the number of mixture components which also corresponds to the number of clusters in the final ensemble clustering. Each mixture component is parameterized by $\boldsymbol{\theta}_j$ while $\alpha_j > 0$ is regarded as the mixing coefficient corresponding to the prior probability of each clusters. In addition, $\sum_{j=1}^{m_k} \alpha_j = 1$. Note that random variables $\boldsymbol{x}_i$ and $\boldsymbol{t}_i$ are assumed to be conditionally independent to make the problem tractable. This assumption is reasonable since $\boldsymbol{t}_i$ describes the inherent structure of the whole training set, which is relatively immune to a certain data point $\boldsymbol{x}_i$.

In this paper, the instance $\boldsymbol{x}_i$ is modeled as a random variable drawn from a marginal distribution described as a mixture of *Gaussian distributions* according to Eq.(6), i.e.

$$\begin{aligned} P(\boldsymbol{x}_i) &= \sum_{\boldsymbol{t}_i} P([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{\Theta}) = \sum_{j=1}^{m_k} \alpha_j P_j(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \\ &= \sum_{j=1}^{m_k} \alpha_j \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)} \end{aligned} \quad (7)$$

Here, each mixture component is parameterized by $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$, where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the d-dimensional *mean vector* and the *covariance matrix* for each mixture component respectively.

Similarly, the cluster assignment vector $\boldsymbol{t}_i$ is modeled as a random variable drawn from a marginal distribution described as a mixture of *multinomial distributions* according to Eq.(6), i.e.

$$P(\boldsymbol{t}_i) = \sum_{\boldsymbol{x}_i} P([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{\Theta}) = \sum_{j=1}^{m_k} \alpha_j P_j(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_j) \quad (8)$$

Here, each mixture component is parameterized by $\boldsymbol{\vartheta}_j$. Assume that the elements of the cluster assignment vector $\boldsymbol{t}_i$ are conditionally independent, then:

$$P_j(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_j) = \prod_{p=1}^{r} P_j^{(p)}(t_i^p \mid \boldsymbol{\vartheta}_j^{(p)}) = \prod_{p=1}^{r} \prod_{l=1}^{k^{(p)}} v_{pj}(l)^{\delta(t_i^p, l)} \quad (9)$$

where $k^{(p)}$ is the number of clusters in the pth base cluster. In addition, $\delta(t_i^p, l)$ is the *Kronecker* δ function which returns 1 if t_i^p is equal to l and 0 otherwise. The probability of the instance belonging to the lth cluster is defined as $v_{pj}(l)$ with $\sum_{l=1}^{k^{(p)}} v_{pj}(l) = 1$.

Based on the above assumptions, the problem of clustering combination is now transformed into a maximum likelihood

TABLE II: The pseudo-code of SENCE.

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$ $(\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\}, \boldsymbol{x}_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y})$

r, ϖ: the number of base clusters and the ratio parameter $\varpi \in [0,1]$ in Eq.(17)

$\mathfrak{L}$: the binary training algorithm

$\boldsymbol{u}$: an unseen instance

Outputs:

Y: the predicted label set for $\boldsymbol{u}$

Process:

for $k = 1$ to q **do**
  Form $\mathcal{P}_k$ and $\mathcal{N}_k$ according to Eq.(1);
  Obtain cluster assignment vector $\boldsymbol{t}_i$ for each instance by performing clustering on $\mathcal{P}_k$ and $\mathcal{N}_k$ several times;
  Re-represent $\mathcal{P}_k$ and $\mathcal{N}_k$ as $\overline{\mathcal{P}_k}$ and $\overline{\mathcal{N}_k}$ according to Eq.(5);
  Initialize parameters $\boldsymbol{\Theta}^{\overline{\mathcal{P}_k}} = \{\alpha_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\mu}_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\Sigma}_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\vartheta}_j^{\overline{\mathcal{P}_k}} \mid 1 \leq j \leq m_k\}$ and $\boldsymbol{\Theta}^{\overline{\mathcal{N}_k}} = \{\alpha_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\mu}_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\Sigma}_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\vartheta}_j^{\overline{\mathcal{N}_k}} \mid 1 \leq j \leq m_k\}$;
  repeat
    Estimate the posterior distribution of the hidden variable z_i for each instance in $\overline{\mathcal{P}_k}$ according to Eq.(11);
    Update parameters $\boldsymbol{\Theta}^{\overline{\mathcal{P}_k}} = \{\alpha_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\mu}_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\Sigma}_j^{\overline{\mathcal{P}_k}}, \boldsymbol{\vartheta}_j^{\overline{\mathcal{P}_k}} \mid 1 \leq j \leq m_k\}$ according to Eq.(12)-(15);
  until convergence;
  repeat
    Estimate the posterior distribution of the hidden variable z_i for each instance in $\overline{\mathcal{N}_k}$ according to Eq.(11);
    Update parameters $\boldsymbol{\Theta}^{\overline{\mathcal{N}_k}} = \{\alpha_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\mu}_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\Sigma}_j^{\overline{\mathcal{N}_k}}, \boldsymbol{\vartheta}_j^{\overline{\mathcal{N}_k}} \mid 1 \leq j \leq m_k\}$ according to Eq.(12)-(15);
  until convergence;
  Divide $\mathcal{P}_k$ into m_k clusters $\boldsymbol{\mathcal{C}}^{\mathcal{P}_k} = \{\mathcal{C}_1^{\mathcal{P}_k}, \mathcal{C}_2^{\mathcal{P}_k}, \ldots, \mathcal{C}_{m_k}^{\mathcal{P}_k}\}$ according to Eq.(16);
  Divide $\mathcal{N}_k$ into m_k clusters $\boldsymbol{\mathcal{C}}^{\mathcal{N}_k} = \{\mathcal{C}_1^{\mathcal{N}_k}, \mathcal{C}_2^{\mathcal{N}_k}, \ldots, \mathcal{C}_{m_k}^{\mathcal{N}_k}\}$ according to Eq.(16);
  Create the mapping ϕ_k for l_k defined in Eq.(2) based on $\boldsymbol{\mathcal{C}}^{\mathcal{P}_k}$ and $\boldsymbol{\mathcal{C}}^{\mathcal{N}_k}$;
end for
for $k = 1$ to q **do**
  Form $\mathcal{B}_k$ according to Eq.(3);
  Induce g_k by invoking $\mathfrak{L}$ on $\mathcal{B}_k$, i.e. $g_k \leftarrow \mathfrak{L}(\mathcal{B}_k)$;
end for
Return the predicted label set $Y = \{l_k \mid g_k(\phi_k(\boldsymbol{u})) > 0, 1 \leq k \leq q\}$.

estimation problem. The optimal parameter $\boldsymbol{\Theta}^*$ w.r.t. $\overline{\mathcal{P}_k}$ is found by maximizing the log-likelihood function as follows:

$$\begin{aligned}\boldsymbol{\Theta}^* &= \arg\max_{\boldsymbol{\Theta}} L(\overline{\mathcal{P}_k}|\boldsymbol{\Theta}) = \arg\max_{\boldsymbol{\Theta}} \ln(\prod_{i=1}^{|\overline{\mathcal{P}_k}|} P([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid \boldsymbol{\Theta})) \\ &= \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{|\overline{\mathcal{P}_k}|} \ln(\sum_{j=1}^{m_k} \alpha_j P_j(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) P_j(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_j)) \end{aligned} \tag{10}$$

The optimal parameter $\boldsymbol{\Theta}^*$ w.r.t. $\overline{\mathcal{N}_k}$ is found in the same way.

However, as all the parameters $\boldsymbol{\Theta} = \{\alpha_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \boldsymbol{\vartheta}_j \mid 1 \leq j \leq m_k\}$ are unknown, the problem in Eq.(10) cannot generally be solved in a closed form. Thus, the *EM algorithm* is used to optimize Eq.(10). In order to perform the EM algorithm, the hidden variable $z_i \in \{1, 2, \ldots, m_k\}$ is introduced to represent the corresponding mixture component generating $[\boldsymbol{x}_i, \boldsymbol{t}_i]$, i.e. $z_i = j$ if $[\boldsymbol{x}_i, \boldsymbol{t}_i]$ belongs to the jth mixture component. According to the *Bayes' theorem*, the E-step of the EM algorithm can be obtained by estimating the posterior distribution of the hidden variable z_i as follows:

$$\begin{aligned}\gamma_{ij} &= P(z_i = j \mid [\boldsymbol{x}_i, \boldsymbol{t}_i]) \\ &= \frac{P(z_i = j) P([\boldsymbol{x}_i, \boldsymbol{t}_i] \mid z_i = j)}{P([\boldsymbol{x}_i, \boldsymbol{t}_i])} \\ &= \frac{\alpha_j P_j(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) P_j(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_j)}{\sum_{l=1}^{m_k} \alpha_l P_l(\boldsymbol{x}_i \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) P_l(\boldsymbol{t}_i \mid \boldsymbol{\vartheta}_l)} \end{aligned} \tag{11}$$

In other word, γ_{ij} gives the posterior probability that $[\boldsymbol{x}_i, \boldsymbol{t}_i]$ is drawn from the jth mixture component. Given the value of γ_{ij} from the E-step, the M-step aims to maximize the log-likelihood function $L(\overline{\mathcal{P}_k}|\boldsymbol{\Theta})$. The mean vector $\boldsymbol{\mu}_j$ and the covariance matrix $\boldsymbol{\Sigma}_j$ are derived as follows:

$$\frac{\partial L(\overline{\mathcal{P}_k} \mid \boldsymbol{\Theta})}{\partial \boldsymbol{\mu}_j} = 0 \Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \gamma_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \gamma_{ij}} \tag{12}$$

$$\frac{\partial L(\overline{\mathcal{P}_k} \mid \boldsymbol{\Theta})}{\partial \boldsymbol{\Sigma}_j} = 0 \Rightarrow \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \gamma_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \gamma_{ij}} \tag{13}$$

With the constraint $\sum_{j=1}^{m_k} \alpha_j = 1$, *Lagrange multipliers* are invoked to update the mixing coefficients:

$$\frac{\partial (L(\overline{\mathcal{P}_k} \mid \boldsymbol{\Theta}) + \lambda(\sum_{j=1}^{m_k} \alpha_j - 1))}{\partial \alpha_j} = 0 \Rightarrow \alpha_j = \frac{1}{m} \sum_{i=1}^{|\overline{\mathcal{P}_k}|} \gamma_{ij} \tag{14}$$

Similarly, the optimal value of $v_{pj}(l)$ is obtained as follows:

$$\frac{\partial (L(\overline{\mathcal{P}_k} \mid \boldsymbol{\Theta}) + \lambda(\sum_{l=1}^{k^{(p)}} v_{pj}(l) - 1))}{\partial v_{pj}(l)} = 0$$

TABLE III: Characteristics of the experimental data sets.

Data set	$\lvert\mathcal{S}\rvert$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$LCard(\mathcal{S})$	$LDen(\mathcal{S})$	$DL(\mathcal{S})$	$PDL(\mathcal{S})$	Domain
flags	194	19	7	3.392	0.485	54	0.278	images
CAL500	502	68	174	26.044	0.150	502	1.000	audio
emotions	593	72	6	1.868	0.311	27	0.046	audio
medical	978	1449	45	1.245	0.028	94	0.096	text
llog	1,208	484	74	1.180	0.016	286	0.196	text
enron	1,702	1001	53	3.378	0.064	753	0.442	text
image	2,000	294	5	1.236	0.247	20	0.010	image
scene	2,407	294	5	1.074	0.179	15	0.006	image
yeast	2,417	103	14	4.237	0.303	198	0.082	biology
slashdot	3,659	805	22	1.181	0.054	119	0.033	text
corel5k	5,000	410	374	3.522	0.009	3175	0.635	image
arts	5,000	462	26	1.636	0.063	462	0.092	text
reference	5,570	29	33	1.187	0.036	240	0.043	text
health	8,116	115	32	1.649	0.052	314	0.039	text
entertainment	8,166	99	21	1.437	0.068	278	0.034	text
business	8,718	132	30	1.623	0.054	211	0.024	text
NUS-WIDE-c	10,000	128	81	2.403	0.030	2,448	0.245	image
socity	10,973	55	27	1.674	0.062	885	0.081	text

$$\Rightarrow v_{pj}(l) = \frac{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \delta(t_i^p, l)\gamma_{ij}}{\sum_{i=1}^{|\overline{\mathcal{P}_k}|} \sum_{l=1}^{k(p)} \delta(t_i^p, l)\gamma_{ij}} \tag{15}$$

In summary, for each iteration, the E-step estimates the posterior distribution of the hidden variable z_i according to the current parameters while the M-step updates the optimal values of all parameters according to Eq.(12)-(15).

3) Label-Specific Features Construction: According to the induced mixing distribution on $\overline{\mathcal{P}_k}$, $\mathcal{P}_k$ is divided into m_k disjoint clusters donated as $\{\mathcal{C}_1^{\mathcal{P}_k}, \mathcal{C}_2^{\mathcal{P}_k}, \ldots, \mathcal{C}_{m_k}^{\mathcal{P}_k}\}$. The final cluster assignment of each instance in $\mathcal{P}_k$ can be defined as follows:

$$\lambda_i = \underset{j\in\{1,2,\ldots,m_k\}}{\arg\max}\ \gamma_{ij} \tag{16}$$

Similarly, $\mathcal{N}_k$ is divided into m_k disjoint clusters denoted as $\{\mathcal{C}_1^{\mathcal{N}_k}, \mathcal{C}_2^{\mathcal{N}_k}, \ldots, \mathcal{C}_{m_k}^{\mathcal{N}_k}\}$ in the same way. Notice that the number of clusters retained for $\mathcal{P}_k$ is equal to $\mathcal{N}_k$ in order to mitigate the risk of *class-imbalance*, i.e. $|\mathcal{P}_k| \ll |\mathcal{N}_k|$. Specifically, the value of m_k is set as:

$$m_k = \lceil \varpi \cdot \min\left(|\mathcal{P}_k|, |\mathcal{N}_k|\right) \rceil \tag{17}$$

Here, $\varpi \in$ [0,1] is a ratio parameter controlling the number of clusters $\mathcal{P}_k$ and $\mathcal{N}_k$ retained, and $|\cdot|$ returns the set cardinality.

Conceptually, cluster centers characterize the inherent structure of the positive and negative instances. Thus, clustering centers can be used as prototypes to construct label-specific features which are derived from more stable clustering. Similar to LIFT, the mapping $\phi_k : \mathcal{X} \rightarrow \mathcal{Z}_k$ can be created according to Eq.(2).

4) Predictive Model Induction: Similar to LIFT, SENCE transforms the training set $\mathcal{D}$ into a new binary training set $\mathcal{B}_k$ for each class label according to Eq.(3). Any binary learner $\mathfrak{L}$ can be applied to induce a classification model $g_k : \mathcal{Z}_k \rightarrow \mathbb{R}$ for l_k based on $\mathcal{B}_k$. After that, an associated label set is predicted for an unseen example $\boldsymbol{u} \in \mathcal{X}$ according to Eq.(4)

Table II summarizes the procedure of SENCE. SENCE firstly performs clustering several times to re-represent instances for each label (step 2 to 4); After that, the EM algorithm is used to yield more stable clustering (step 5 to 15) and label-specific features are constructed for each class label (step 16); Then, a family of q binary classification models are induced based on the constructed label-specific features (step 18 to 21); Finally, an unseen instance is fed to the learned models for predicting the relevant labels (step 22).

IV. EXPERIMENTS

A. Experimental Setup

Given the multi-label data set $\mathcal{S} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$, $|\mathcal{S}|$, $dim(\mathcal{S})$ and $L(\mathcal{S})$ denote the number of examples, number of features and number of possible class labels respectively. In addition, several other multi-label properties [1], [15] are denoted as:

- $LCard(\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m} |Y_i|$: *label cardinality* measures the average number of labels per example;
- $LDen(\mathcal{S}) = \frac{LCard(\mathcal{S})}{L(\mathcal{S})}$: *label density* normalizes $LCard(\mathcal{S})$ by the number of possible labels;
- $DL(\mathcal{S}) = |\{Y \mid (\boldsymbol{x}, Y) \in \mathcal{S}\}|$: *distinct label sets* counts the number of distinct label sets existing in $\mathcal{S}$;
- $PDL(S) = \frac{DL(\mathcal{S})}{|\mathcal{S}|}$: *proportion of distinct label sets* normalizes $DL(\mathcal{S})$ by the number of examples.

Table III summarizes the detailed characteristics of the benchmark multi-label data sets employed in the experiments. Data sets shown in Table III are roughly ordered by $|\mathcal{S}|$. The 17 benchmark data sets exhibit diversified multi-label properties which provide a solid basis for thorough performance evaluation.

To validate the effectiveness of the proposed approach, six state-of-the-art multi-label learning approaches are used for comparative studies.

- LPLC [42]: A second-order multi-label learning approach which exploits the local positive and negative pairwise label correlations by maximizing kNN-based posterior probability. $[k = 10, \alpha = 0.1]$
- LIFT [6]: A first-order multi-label learning approach, which induces classifiers with the label-specific features generated via conducting clustering analysis for each class label. [Base learner: linear kernel SVM, $r = 0.1$]

TABLE IV: Experimental results of the comparing approaches on the first nine data sets (↓: the smaller the better; ↑: the larger the better).

Comparing algorithm	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
					Hamming loss↓				
SENCE	0.271±0.042	0.138±0.006	**0.177±0.019**	0.011±0.002	**0.017±0.001**	0.050±0.009	**0.153±0.013**	**0.074±0.005**	**0.188±0.008**
LPLC	0.292±0.035	0.150±0.006	0.216±0.024	0.018±0.003	0.020±0.001	0.067±0.013	0.230±0.012	0.128±0.009	0.227±0.009
LIFT	0.267±0.058	0.138±0.006	0.183±0.019	0.012±0.003	0.018±0.001	0.049±0.008	0.154±0.014	0.078±0.006	0.191±0.007
LLSF	0.278±0.042	**0.137±0.007**	0.197±0.020	0.011±0.003	0.018±0.001	0.048±0.008	0.193±0.011	0.111±0.006	0.199±0.008
MLSF	0.292±0.060	0.138±0.007	0.207±0.022	**0.010±0.002**	0.018±0.001	0.055±0.010	0.185±0.020	0.110±0.014	0.211±0.013
LIFTACE	**0.265±0.052**	0.138±0.006	0.179±0.018	0.012±0.002	**0.017±0.001**	**0.047±0.008**	0.155±0.013	0.078±0.005	0.190±0.007
WRAP	0.285±0.030	**0.137±0.007**	0.237±0.024	0.125±0.037	0.018±0.001	0.072±0.029	0.198±0.012	0.120±0.006	0.210±0.007
Comparing algorithm					*Ranking loss*↓				
	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
SENCE	**0.202±0.049**	**0.182±0.007**	**0.138±0.029**	**0.024±0.012**	**0.134±0.019**	**0.085±0.018**	**0.133±0.020**	**0.056±0.007**	**0.160±0.011**
LPLC	0.226±0.046	0.228±0.016	0.178±0.028	0.072±0.011	0.330±0.018	0.208±0.046	0.199±0.026	0.107±0.010	0.188±0.011
LIFT	0.220±0.049	0.183±0.007	0.146±0.023	0.025±0.012	0.148±0.020	**0.085±0.017**	0.144±0.022	0.061±0.007	0.164±0.013
LLSF	0.232±0.048	0.188±0.014	0.172±0.022	0.032±0.016	0.223±0.021	0.104±0.014	0.178±0.021	0.091±0.010	0.169±0.013
MLSF	0.256±0.059	0.210±0.009	0.170±0.032	0.031±0.019	**0.134±0.028**	0.096±0.019	0.182±0.018	0.105±0.020	0.208±0.022
LIFTACE	0.222±0.055	0.183±0.007	0.147±0.027	0.028±0.012	0.154±0.021	**0.085±0.019**	0.145±0.023	0.060±0.005	0.164±0.012
WRAP	0.237±0.048	0.180±0.007	0.202±0.024	0.165±0.042	0.224±0.022	0.152±0.042	0.184±0.025	0.092±0.012	0.181±0.014
Comparing algorithm					*One-error*↓				
	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
SENCE	**0.186±0.092**	0.116±0.028	**0.231±0.059**	0.147±0.041	0.652±0.050	0.253±0.042	**0.253±0.032**	**0.179±0.022**	**0.209±0.019**
LPLC	0.240±0.083	0.210±0.052	0.297±0.045	0.312±0.056	0.789±0.029	0.540±0.124	0.347±0.040	0.249±0.028	0.236±0.024
LIFT	0.251±0.105	0.124±0.031	0.242±0.051	0.162±0.042	0.643±0.044	0.255±0.051	0.273±0.038	0.197±0.022	0.214±0.018
LLSF	0.249±0.103	0.120±0.033	0.280±0.068	0.143±0.047	0.686±0.036	0.255±0.043	0.334±0.040	0.258±0.024	0.221±0.021
MLSF	0.282±0.093	0.132±0.038	0.286±0.059	**0.140±0.043**	0.701±0.029	0.328±0.055	0.340±0.043	0.292±0.046	0.252±0.033
LIFTACE	0.255±0.128	0.124±0.031	0.249±0.057	0.163±0.039	**0.635±0.042**	**0.249±0.044**	0.271±0.039	0.191±0.020	0.215±0.027
WRAP	0.212±0.080	**0.115±0.029**	0.308±0.050	0.518±0.118	0.838±0.040	0.325±0.084	0.350±0.038	0.263±0.026	0.242±0.030
Comparing algorithm					*Coverage*↓				
	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
SENCE	**0.524±0.047**	0.754±0.014	**0.277±0.033**	**0.038±0.016**	0.176±0.026	**0.239±0.051**	**0.161±0.016**	**0.060±0.006**	**0.447±0.017**
LPLC	0.550±0.045	0.861±0.022	0.309±0.031	0.090±0.011	0.370±0.021	0.456±0.103	0.208±0.021	0.094±0.009	0.471±0.016
LIFT	0.542±0.043	0.756±0.015	0.285±0.035	0.039±0.016	0.193±0.028	0.241±0.048	0.169±0.018	0.064±0.006	0.453±0.019
LLSF	0.549±0.045	**0.748±0.016**	0.307±0.030	0.042±0.016	0.273±0.027	0.278±0.051	0.196±0.018	0.090±0.009	0.454±0.017
MLSF	0.558±0.054	0.820±0.026	0.299±0.047	0.047±0.024	**0.172±0.035**	0.255±0.055	0.197±0.017	0.101±0.015	0.524±0.038
LIFTACE	0.540±0.049	0.760±0.013	0.284±0.037	0.042±0.016	0.200±0.030	0.243±0.053	0.170±0.018	0.064±0.004	0.454±0.018
WRAP	0.550±0.048	0.753±0.014	0.337±0.047	0.188±0.043	0.274±0.029	0.356±0.080	0.198±0.021	0.092±0.011	0.466±0.019
Comparing algorithm					*Average precision*↑				
	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
SENCE	**0.824±0.045**	0.502±0.015	**0.826±0.036**	0.887±0.032	0.440±0.045	0.672±0.046	**0.834±0.019**	**0.896±0.012**	**0.776±0.012**
LPLC	0.800±0.033	0.461±0.022	0.784±0.030	0.748±0.042	0.250±0.021	0.472±0.096	0.772±0.026	0.843±0.014	0.753±0.015
LIFT	0.806±0.047	0.498±0.014	0.818±0.025	0.876±0.030	0.445±0.038	0.675±0.028	0.823±0.024	0.887±0.011	0.772±0.012
LLSF	0.795±0.041	**0.505±0.023**	0.794±0.027	**0.893±0.031**	0.400±0.032	0.673±0.032	0.784±0.023	0.845±0.014	0.762±0.013
MLSF	0.783±0.047	0.473±0.014	0.795±0.037	0.887±0.032	0.393±0.030	0.623±0.049	0.783±0.022	0.824±0.029	0.721±0.022
LIFTACE	0.804±0.052	0.498±0.016	0.817±0.031	0.875±0.026	**0.446±0.040**	**0.687±0.051**	0.824±0.024	0.889±0.010	0.772±0.013
WRAP	0.799±0.044	0.503±0.013	0.766±0.027	0.568±0.093	0.272±0.024	0.600±0.034	0.778±0.025	0.841±0.017	0.743±0.017
Comparing algorithm					*Macro-averaging AUC*↑				
	flags	CAL500	emotions	medical	language log	enron	image	scene	yeast
SENCE	**0.699±0.050**	0.527±0.027	**0.858±0.024**	0.922±0.035	0.733±0.033	0.695±0.023	**0.871±0.025**	**0.953±0.005**	**0.707±0.015**
LPLC	0.674±0.087	0.529±0.027	0.821±0.034	0.831±0.033	0.562±0.029	0.583±0.028	0.815±0.023	0.922±0.008	0.685±0.022
LIFT	**0.699±0.057**	0.529±0.020	0.844±0.025	0.923±0.035	**0.747±0.034**	**0.704±0.033**	0.860±0.026	0.949±0.005	0.694±0.017
LLSF	**0.699±0.027**	**0.553±0.047**	0.828±0.026	0.929±0.017	0.729±0.032	0.667±0.034	0.824±0.024	0.922±0.008	0.693±0.017
MLSF	0.683±0.056	0.524±0.019	0.835±0.029	**0.935±0.033**	0.706±0.046	0.646±0.026	0.823±0.025	0.915±0.016	0.633±0.016
LIFTACE	0.689±0.047	0.524±0.024	0.844±0.026	0.918±0.032	0.738±0.032	**0.704±0.025**	0.860±0.026	0.949±0.006	0.693±0.013
WRAP	0.696±0.076	0.466±0.034	0.797±0.027	0.407±0.061	0.333±0.027	0.485±0.056	0.816±0.026	0.907±0.012	0.629±0.026

- LLSF [30]: A second-order multi-label learning approach based on label-specific features generated by retaining a different subset of original features for each class label. $[\alpha = 0.5, \beta = 0.5, \gamma = 0.5]$
- MLSF [34]: A high-order multi-label learning approach based on label-specific features, which performs sparse regression to generate tailored features by retaining a different subset of original features for a group of class labels. $[\epsilon = 0.01, \alpha = 0.8, \gamma = 0.01]$
- LIFTACE [8]: A high-order multi-label learning approach based on label-specific features generated by considering label correlations via clustering ensemble techniques. [Base learner: linear kernel SVM, $r = 0.1$, $\gamma = 10$]
- WRAP [38]: A high-order multi-label learning approach which performs label-specific feature generation and classification model induction in a joint manner. [$\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\lambda_3 = 0.1$, $\alpha = 0.9$]

For each comparing approach, parameter configurations suggested in respective literature are stated above. For SENCE shown in Table II, the parameter configuration corresponds to $\varpi = 0.4$ and $r = 5$. Moreover, LIBSVM [43] is employed as the binary learning algorithm $\mathfrak{L}$ and k-means algorithm is employed as the base clustering algorithm.

In addition, given the test set $\mathcal{T} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq t\}$ and a family of q learned functions $\{f_1, f_2, \ldots, f_q\}$, six evaluation metrics [1] widely-used in multi-label learning are utilized in this paper to evaluate the performance of each comparing approach:

- *Hamming loss*:

$$hloss = \frac{1}{t}\sum_{i=1}^{t} |h(\boldsymbol{x}_i) \triangle Y_i|$$

Hamming loss evaluates the fraction of instance-label pairs which are misclassified. Here, $h(\boldsymbol{x_i}) = \{l_k \mid f_k(\boldsymbol{x}_i) > 0, 1 \leq k \leq q\}$ corresponds to the predicted set of relevant labels for $\boldsymbol{x}_i$, and $\triangle$ stands for the symmetric difference between two sets.

TABLE V: Experimental results of the comparing approaches on the other nine data sets (↓: the smaller the better; ↑: the larger the better).

Comparing algorithm	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
					Hamming loss↓				
SENCE	**0.055±0.002**	0.010±0.000	**0.052±0.001**	0.036±0.001	0.049±0.005	0.067±0.001	**0.030±0.001**	**0.026±0.000**	0.059±0.001
LPLC	0.105±0.008	0.010±0.009	0.091±0.007	0.038±0.001	0.058±0.001	0.077±0.002	0.049±0.002	0.029±0.000	0.065±0.001
LIFT	0.058±0.003	0.010±0.000	**0.052±0.001**	0.042±0.013	0.051±0.006	0.071±0.011	0.031±0.003	**0.026±0.000**	0.059±0.001
LLSF	0.063±0.002	**0.009±0.000**	0.054±0.001	**0.035±0.001**	**0.047±0.001**	**0.066±0.002**	0.043±0.001	0.027±0.000	0.059±0.001
MLSF	0.061±0.002	**0.009±0.000**	0.054±0.004	0.037±0.002	0.048±0.002	**0.066±0.002**	**0.030±0.001**	0.027±0.001	0.059±0.001
LIFTACE	0.058±0.003	0.010±0.000	0.053±0.001	0.037±0.009	0.056±0.017	0.085±0.029	0.041±0.015	**0.026±0.000**	0.060±0.009
WRAP	0.076±0.002	**0.009±0.000**	0.062±0.002	0.036±0.001	**0.047±0.001**	0.068±0.002	**0.030±0.001**	0.030±0.000	**0.058±0.001**
Comparing algorithm					*Ranking loss*↓				
	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
SENCE	**0.107±0.013**	**0.197±0.046**	**0.109±0.007**	0.112±0.005	0.081±0.007	**0.141±0.007**	0.050±0.005	**0.102±0.004**	0.147±0.004
LPLC	0.469±0.025	0.714±0.018	0.424±0.022	0.311±0.018	0.192±0.011	0.286±0.015	0.137±0.011	0.277±0.013	0.313±0.008
LIFT	0.119±0.010	0.201±0.043	0.110±0.006	0.117±0.014	0.081±0.006	0.146±0.008	0.051±0.005	0.108±0.003	0.148±0.004
LLSF	0.122±0.009	0.410±0.081	0.137±0.013	0.138±0.009	0.136±0.009	0.185±0.013	0.182±0.011	0.106±0.003	0.185±0.010
MLSF	0.130±0.007	0.212±0.044	0.119±0.016	0.111±0.005	0.082±0.006	0.174±0.030	0.062±0.007	0.137±0.046	0.149±0.004
LIFTACE	0.117±0.014	0.205±0.044	0.110±0.006	0.113±0.009	0.081±0.015	0.157±0.033	0.059±0.014	0.109±0.003	0.150±0.010
WRAP	0.179±0.015	0.223±0.043	0.146±0.008	**0.108±0.004**	**0.077±0.003**	0.143±0.007	**0.049±0.003**	0.128±0.005	**0.144±0.005**
Comparing algorithm					*One-error*↓				
	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
SENCE	**0.342±0.026**	0.765±0.055	**0.445±0.015**	0.564±0.029	0.509±0.069	0.645±0.019	**0.139±0.014**	**0.469±0.018**	**0.475±0.013**
LPLC	0.705±0.024	0.874±0.072	0.826±0.014	0.558±0.028	0.485±0.018	0.617±0.024	0.143±0.013	0.545±0.017	0.518±0.015
LIFT	0.373±0.026	0.765±0.054	0.449±0.018	0.657±0.177	0.547±0.114	0.677±0.084	**0.139±0.014**	0.472±0.017	0.478±0.014
LLSF	**0.342±0.021**	0.816±0.027	0.460±0.018	**0.546±0.026**	**0.431±0.016**	**0.582±0.022**	0.278±0.014	0.474±0.017	0.504±0.011
MLSF	0.401±0.018	0.779±0.044	0.474±0.039	0.564±0.027	0.458±0.017	0.630±0.014	0.140±0.014	0.512±0.029	0.479±0.012
LIFTACE	0.368±0.021	0.757±0.052	0.452±0.015	0.604±0.119	0.573±0.207	0.733±0.171	0.393±0.374	0.472±0.014	0.514±0.126
WRAP	0.493±0.022	**0.745±0.069**	0.605±0.029	0.566±0.029	0.477±0.016	0.647±0.017	**0.139±0.014**	0.644±0.012	0.481±0.013
Comparing algorithm					*Coverage*↓				
	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
SENCE	**0.124±0.013**	**0.437±0.077**	**0.167±0.008**	0.127±0.005	0.124±0.007	**0.177±0.006**	0.091±0.007	**0.199±0.007**	0.215±0.005
LPLC	0.325±0.017	0.826±0.053	0.333±0.014	0.261±0.014	0.225±0.012	0.282±0.013	0.168±0.011	0.309±0.012	0.320±0.008
LIFT	0.136±0.010	0.445±0.070	0.169±0.007	0.133±0.014	0.124±0.007	0.180±0.007	0.092±0.007	0.208±0.005	0.216±0.004
LLSF	0.140±0.009	0.736±0.071	0.211±0.015	0.159±0.008	0.200±0.011	0.228±0.014	0.246±0.012	0.200±0.005	0.262±0.013
MLSF	0.148±0.007	0.469±0.075	0.181±0.024	0.127±0.005	0.132±0.011	0.233±0.045	0.118±0.012	0.270±0.097	0.217±0.005
LIFTACE	0.133±0.015	0.449±0.071	0.168±0.007	0.129±0.010	0.125±0.014	0.192±0.030	0.100±0.013	0.211±0.005	0.217±0.010
WRAP	0.196±0.015	0.495±0.063	0.209±0.009	**0.124±0.005**	**0.121±0.004**	0.178±0.006	**0.090±0.005**	0.240±0.007	**0.212±0.006**
Comparing algorithm					*Average precision*↑				
	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
SENCE	**0.745±0.020**	**0.210±0.038**	**0.637±0.014**	0.542±0.018	0.611±0.040	0.528±0.016	0.855±0.011	0.535±0.011	0.570±0.009
LPLC	0.445±0.021	0.091±0.031	0.358±0.013	0.521±0.019	0.603±0.012	0.498±0.019	0.814±0.012	0.469±0.012	0.524±0.009
LIFT	0.722±0.020	0.207±0.042	0.633±0.011	0.495±0.092	0.595±0.061	0.509±0.045	0.854±0.011	0.525±0.012	0.568±0.010
LLSF	0.742±0.015	0.142±0.017	0.622±0.015	**0.561±0.019**	**0.644±0.008**	**0.550±0.017**	0.723±0.012	**0.542±0.012**	0.551±0.009
MLSF	0.701±0.012	0.198±0.028	0.613±0.028	0.540±0.018	0.636±0.009	0.526±0.009	0.850±0.012	0.486±0.031	0.566±0.009
LIFTACE	0.727±0.018	**0.210±0.039**	0.632±0.011	0.523±0.061	0.586±0.108	0.472±0.111	0.738±0.168	0.525±0.011	0.553±0.057
WRAP	0.628±0.016	0.209±0.043	0.529±0.019	0.546±0.018	0.636±0.009	0.528±0.013	**0.857±0.010**	0.410±0.008	**0.573±0.010**
Comparing algorithm					*Macro-averaging AUC*↑				
	slashdot	corel5k	arts	reference	health	entertainment	business	NUS-WIDE-c	society
SENCE	0.871±0.012	0.601±0.048	0.747±0.016	0.542±0.031	0.619±0.022	0.586±0.024	0.525±0.023	0.736±0.016	0.534±0.023
LPLC	0.654±0.012	0.517±0.025	0.575±0.015	0.567±0.014	0.611±0.016	0.587±0.011	0.578±0.018	0.622±0.017	0.551±0.011
LIFT	0.867±0.011	**0.603±0.044**	0.748±0.019	0.566±0.030	0.628±0.045	0.592±0.032	0.575±0.032	0.683±0.010	0.545±0.018
LLSF	**0.875±0.012**	0.595±0.043	**0.749±0.016**	**0.620±0.033**	**0.692±0.042**	**0.628±0.017**	**0.671±0.024**	**0.763±0.011**	**0.603±0.013**
MLSF	0.846±0.011	0.561±0.029	0.739±0.014	0.555±0.032	0.636±0.026	0.591±0.020	0.578±0.026	0.706±0.015	0.551±0.011
LIFTACE	0.871±0.014	0.599±0.040	0.742±0.014	0.555±0.034	0.637±0.039	0.593±0.035	0.618±0.032	0.678±0.021	0.566±0.021
WRAP	0.774±0.020	0.234±0.075	0.606±0.021	0.432±0.060	0.484±0.038	0.555±0.031	0.538±0.030	0.533±0.016	0.567±0.019

- *Ranking loss*:

$$rloss = \frac{1}{t}\sum_{i=1}^{t}\frac{|\{(l_k,l_j) \mid f_k(\boldsymbol{x}_i) \leq f_j(\boldsymbol{x}_i), (l_k,l_j) \in Y_i \times \overline{Y_i}\}|}{|Y_i||\overline{Y_i}|}$$

  Ranking loss evaluates the fraction of relevant-irrelevant label pairs which are reversely ordered. Here, $\overline{Y_i}$ is the complementary set of $Y_i \subseteq \mathcal{Y}$.

- *One-error*:

$$one\text{–}error = \frac{1}{t}\sum_{i=1}^{t}[\![\arg\max_{l_k \in \mathcal{Y}} f_k(\boldsymbol{x}_i) \notin Y_i]\!]$$

  One-error evaluates the fraction of examples whose top-ranked predicted label is not in the ground-truth relevant label set. Here, $[\![\pi]\!]$ returns 1 if predicate π holds and 0 otherwise.

- *Coverage*:

$$coverage = \frac{1}{q}\left(\frac{1}{t}\sum_{i=1}^{t}\max_{l_k \in Y_i} rank(\boldsymbol{x}_i, l_k) - 1\right)$$

  Coverage evaluates the average number of steps needed to move down the ranked label list in order to cover all relevant labels. Here, $rank(\boldsymbol{x}_i, l_k) = \sum_{j=1}^{q}[\![f_j(\boldsymbol{x_i}) \geq f_k(\boldsymbol{x_i})]\!]$ returns the rank of l_k when all class labels in $\mathcal{Y}$ are sorted in descending order according to $\{f_1(\boldsymbol{x}_i), f_2(\boldsymbol{x}_i), \ldots, f_q(\boldsymbol{x}_i)\}$.

- *Average precision*:

$$avgprec = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i|}\sum_{l_k \in Y_i}\frac{|\mathcal{R}(\boldsymbol{x}_i, l_k)|}{rank(\boldsymbol{x}_i, l_k)}$$

  Average precision evaluates the average fraction of relevant labels which rank higher than a particular relevant label. Here, $\mathcal{R}(\boldsymbol{x}_i, l_k) = \{l_j \mid rank(\boldsymbol{x}_i, l_j) \leq rank(\boldsymbol{x}_i, l_k), l_j \in Y_i\}$

- *Macro-averaging AUC*:

$$AUC_{marco} = \frac{1}{q}\sum_{k=1}^{q}\frac{|\{(\boldsymbol{x}',\boldsymbol{x}'') \mid f_k(\boldsymbol{x}') \geq f_k(\boldsymbol{x}''), (\boldsymbol{x}',\boldsymbol{x}'') \in \mathcal{P}_k \times \mathcal{N}_k\}|}{|\mathcal{P}_k||\mathcal{N}_k|}$$

  Macro-averaging AUC evaluates the average AUC value across all class labels.

B. Experimental Results

Ten-fold cross-validation is performed on each benchmark data set, where the mean metric value as well as standard deviation are recorded. Tables IV and V report the detailed experimental results in terms of each evaluation metric where the best performance on each data set is shown in boldface.

TABLE VI: Friedman statistics F_F in terms of each evaluation metric as well as the critical value at 0.05 significance level (# comparing approaches $n = 7$, # data sets $N = 18$).

Evaluation metric	F_F	critical value
Hamming loss	10.1962	
Ranking los	27.0046	
One-error	5.9978	2.1888
Coverage	24.9081	
Average precision	9.7773	
Macro-averaging AUC	14.5575	

In addition, the widely-accepted *Friedman test* [44] is employed here for statistical comparisons of multiple algorithms over a number of data sets. Table VI summarizes the Friedman statistics F_F and the corresponding critical values on each evaluation metric at $\alpha = 0.05$ significance level. As shown in Table VI, the null hypothesis of "equal" performance among comparing approaches should be clearly rejected in terms of each evaluation metric.

Therefore, the *Bonferroni-Dunn test* [45] is employed as the *post-hoc test* [44] to analyze the relative performance among comparing approaches where SENCE is treated as the control approach. Here, the difference between the average ranks of SENCE and one comparing approach is calibrated with the *critical difference* (CD). Here, their performance difference is deemed to be significant if the average ranks of SENCE and one comparing algorithm differ by at least one CD. In this paper, we have CD=1.8996 at significance level $\alpha = 0.05$ as $k = 7$ and $N = 18$.

Based on the reported experimental results, the following observations can be made:

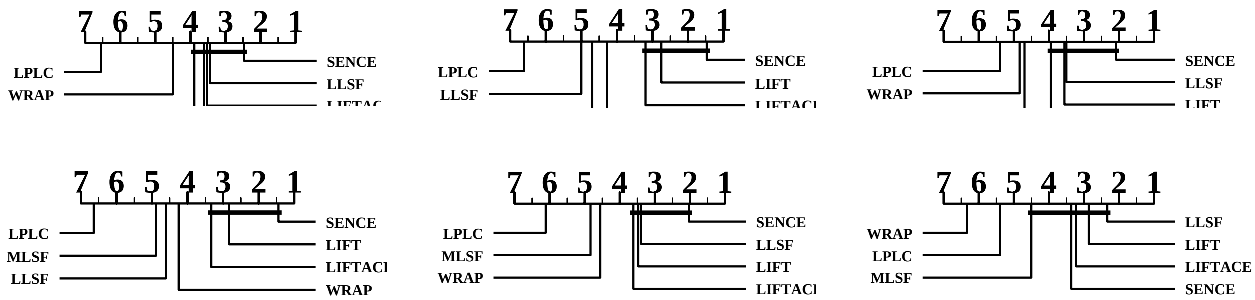
- As shown in Fig. 1, it is impressive that SENCE achieves the lowest rank in terms of all evaluation metrics except *macro-averaging AUC*. Furthermore, all comparing approaches except LPLC and WRAP achieve statistically comparable performance in terms of *macro-averaging AUC*.
- Comparing with approaches without label-specific features, SENCE significantly outperforms LPLC in terms of all evaluation metrics. These results clearly indicate the effectiveness of constructed label-specific features for multi-label label learning.
- Among approaches with label-specific features, SENCE significantly outperforms LLSF, MLSF and WRAP in terms of *ranking loss* and *coverage*. SENCE is comparable to LIFT in terms of all evaluation metrics. Furthermore, pairwise t-tests at 0.05 significance level show that SENCE achieves superior or at least comparable performance than LIFT in 97.2% cases out of 108 cases (18 data sets $\times$ 6 evaluation metrics). These results clearly indicate our proposed clustering ensemble-based strategy for label-specific features serves a more effective way in achieving stable clustering and strong generalization performance.
- SENCE is comparable to LIFTACE in terms of all evaluation metrics. Further pairwise t-tests at 0.05 significance level show that SENCE achieves superior or at least comparable performance than LIFTACE in 96.3% cases out of 109 cases (18 data sets $\times$ 6 evaluation metrics). These results clearly validate the effectiveness of the proposed clustering ensemble strategy employed in SENCE, as both SENCE and LIFTACE utilize clustering ensemble to facilitate the label-specific features construction.

All metric values are normalized in [0,1], where for the first four metrics the smaller the metric value the better the performance and for the other two metrics the larger the metric value the better the performance.

C. Further Analysis

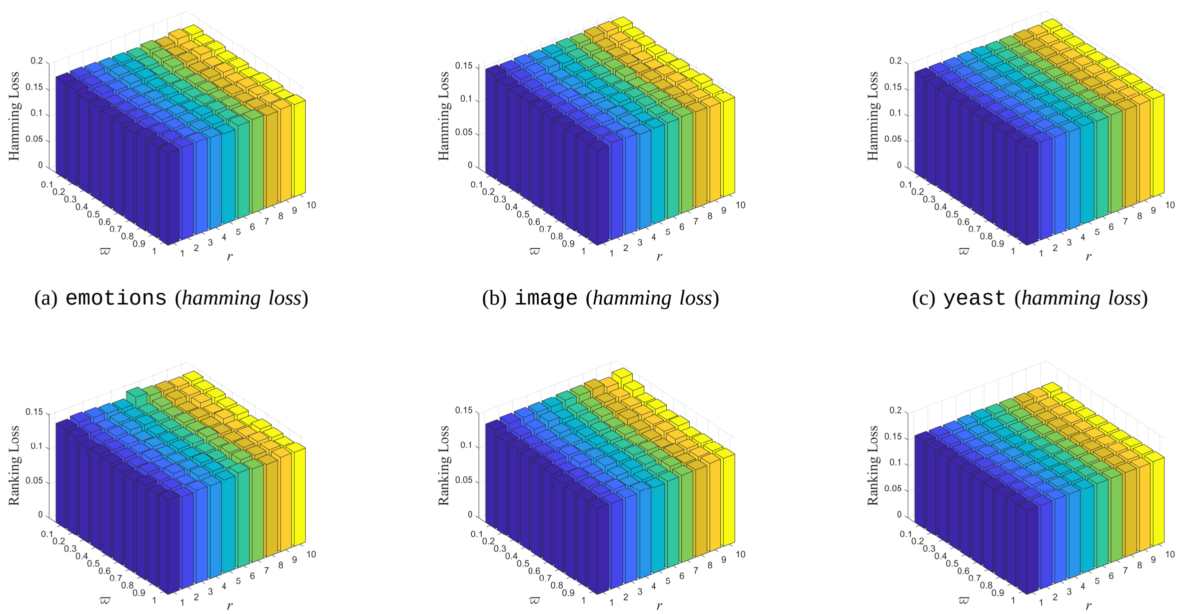
1) Parameter Sensitivity: As shown in Table II, there are two parameters for SENCE to be tuned, i.e. the number of base clusters r and the ratio parameter ϖ. Fig.2 illustrates how the performance of SENCE changes with varying parameter configurations $\varpi \in \{0.1, 0.2, \cdots, 1\}$ and $r \in \{1, 2, \cdots, 10\}$ on three benchmark data sets (evaluation metrics: *hamming loss* and *ranking loss*). As shown in Fig.2, the performance of SENCE is relatively stable as the value of r increases under fixed value of ϖ. On the other hand, the performance of SENCE becomes stable as the value of ϖ increases beyond 0.4 under fixed value of r. Therefore, the value of ϖ and r is fixed to be 0.4 and 5 respectively for comparative studies in this paper.

2) Base Learner: Among the six comparing algorithms employed in Subsection IV-A, three of them are tailored towards concrete learning techniques. Specifically, LPLC is adapted from *k-nearest neighbor* while LLSF and WRAP adapted from *linear regression*. On the other hand, LIFT, LIFTACE and MLSF work in similar way as SENCE by transforming the multi-label learning problem so that any base learner can be applied thereafter. Considering that SENCE, LIFT, LIFTACE and MLSF rely on the choice of base leaner $\mathfrak{L}$ to instantiate the learning approaches, Table VII reports the performance of them on 8 data sets instantiated with different choices of base learner $\mathfrak{L}$ ($\mathfrak{L} \in \{$SVM, k-Nearest Neighbor (kNN), Classification And Regression Tree (CART)$\}$). As shown in Table VII, the following observations can be made: (a) The choice of base learner has significant influence on the performance of each algorithm; (b) SENCE achieves superior or comparable performance than other algorithms in most



(d) *Coverage* (e) *Average precision* (f) *Macro-averaging AUC*

Fig. 1: Comparison of SENCE (control approach) against six comparing approaches with the *Bonferroni-Dunn test*. Approaches not connected with SENCE in the CD diagram are considered to have significantly different performance from the control approach (CD=1.8996 at 0.05 significance level).



(a) `emotions` (*hamming loss*) (b) `image` (*hamming loss*) (c) `yeast` (*hamming loss*)

(d) `emotions` (*ranking loss*) (e) `image` (*ranking loss*) (f) `yeast` (*ranking loss*)

Fig. 2: Performance of SENCE changes with varying parameter configurations $\varpi \in \{0.1, 0.2, \cdots, 1\}$ and $r \in \{1, 2, \cdots, 10\}$ (Data sets: `emotions`, `image`, `yeast`; First row: *hamming loss*, the smaller the better; Second row: *ranking loss*, the smaller the better).

cases with different base learners; (c) SENCE tends to perform better when SVM is used as the base learner other than kNN and CART.

3) Ablation Study: In training phase, SENCE employs multiple base clusters and a mixture model to yield the final clustering. To analyze the rationality of these components, ablation study on two variants of SENCE is further conducted in this subsection. Specifically, SENCE$^{\mathcal{K}}$ employs k-means to obtain clustering results on augmented instances instead of a mixture model; SENCE$^{\mathcal{M}}$ employs one mixture gaussian model to yield clustering results on original instance representations without feature augmentation.

Table VIII reports the detailed experimental results of SENCE and its two variants SENCE$^{\mathcal{K}}$, SENCE$^{\mathcal{M}}$ on 8 benchmark data sets. Compared with SENCE$^{\mathcal{M}}$, SENCE achieves statistically superior or comparable performance in all cases. These results clearly validate the usefulness of multiple base clusters which augment the original instance representations with cluster assignments. Compared with SENCE$^{\mathcal{K}}$, SENCE achieves statistically superior or comparable performance in all cases. These results clearly indicate that the mixture model might be more effective for integrating the preliminary clustering results.

4) Algorithmic Complexity: Let $\mathcal{F}_{\mathfrak{L}}(m, b)$ be the training complexity of the binary learner $\mathfrak{L}$ w.r.t. m training examples and b-dimensional features, the training complexity of SENCE corresponds to $\mathcal{O}\Big(q\big(I(md^2 + r\lceil\varpi\cdot m\rceil^2 + \lceil\varpi\cdot m\rceil d^3) + \mathcal{F}_{\mathfrak{L}}(m, \lceil\varpi\cdot m\rceil)\big)\Big)$, where d^3 is derived from the covariance matrix inversion and I is the number of iterations. The testing complexity of SENCE over unseen instance $\boldsymbol{u}$ corresponds to $\mathcal{O}\Big(q\big(d\lceil\varpi\cdot m\rceil + \mathcal{F}'_{\mathfrak{L}}(\lceil\varpi\cdot m\rceil)\big)\Big)$, where $\mathcal{F}_{\mathfrak{L}}(b)'$ is the testing

TABLE VII: Experimental results of the comparing approaches instantiated with different base learners $\mathfrak{L}$ ($\mathfrak{L} \in$ {SVM, k-Nearest Neighbor (kNN), Classification And Regression Tree (CART)}). In addition, •/○ indicates whether the performance of SENCE is statistically superior/inferior to the comparing approaches on each data set (pairwise t-test at 0.05 significate level).

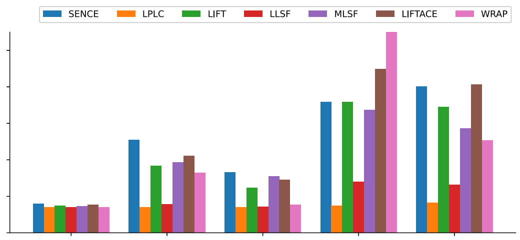
Base learner	Comparing algorithm	*Hamming loss*↓								win/tie/loss counts
		CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
$\mathfrak{L}$ =SVM	LIFT	0.138±0.006	0.183±0.019•	0.078±0.006•	0.191±0.007•	0.052±0.001	0.042±0.013	0.051±0.006	0.026±0.000•	**4/4/0**
	MLSF	0.138±0.007	0.207±0.022•	0.110±0.014•	0.211±0.013•	0.054±0.004	0.037±0.002	0.048±0.002	0.027±0.001•	**4/4/0**
	LIFTACE	0.138±0.006	0.179±0.018	0.078±0.005•	0.190±0.007•	0.053±0.001	0.037±0.009	0.056±0.017	0.026±0.000•	**3/5/0**
	SENCE	0.138±0.006	0.177±0.019	0.074±0.005	0.188±0.008	0.052±0.001	0.036±0.001	0.049±0.005	0.026±0.000	**In Total: 11/13/0**
$\mathfrak{L}$ = kNN	LIFT	0.153±0.007	0.214±0.021	0.096±0.005	0.211±0.004•	0.059±0.001	0.036±0.001	0.050±0.001	0.028±0.001•	**2/6/0**
	MLSF	0.148±0.006○	0.214±0.026	0.096±0.008	0.210±0.010	0.083±0.002•	0.038±0.002•	0.051±0.002	0.029±0.000•	**3/4/1**
	LIFTACE	0.154±0.007	0.211±0.021	0.096±0.006	0.212±0.005•	0.059±0.001•	0.036±0.001	0.051±0.001•	0.028±0.001•	**4/4/0**
	SENCE	0.152±0.008	0.212±0.017	0.098±0.005	0.207±0.005	0.059±0.001	0.036±0.001	0.050±0.001	0.027±0.001	**In Total: 9/14/1**
$\mathfrak{L}$ =CART	LIFT	0.190±0.005•	0.258±0.026	0.128±0.010	0.258±0.008	0.082±0.003•	0.048±0.001	0.064±0.002•	0.039±0.001•	**4/4/0**
	MLSF	0.201±0.010•	0.268±0.033	0.145±0.013•	0.285±0.008•	0.082±0.003•	0.049±0.002•	0.069±0.002•	0.045±0.001•	**7/1/0**
	LIFTACE	0.190±0.004•	0.268±0.023	0.127±0.006	0.258±0.009	0.081±0.002•	0.048±0.001•	0.064±0.001•	0.039±0.001•	**5/3/0**
	SENCE	0.185±0.005	0.260±0.023	0.129±0.007	0.257±0.008	0.074±0.002	0.047±0.002	0.062±0.002	0.036±0.001	**In Total: 16/8/0**
Base learner	Comparing algorithm	*One-error*↓								win/tie/loss counts
		CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
$\mathfrak{L}$ =SVM	LIFT	0.124±0.031	0.242±0.051	0.197±0.022•	0.214±0.018	0.449±0.018	0.657±0.177	0.547±0.114	0.472±0.017	**1/7/0**
	MLSF	0.132±0.038	0.286±0.059•	0.292±0.046•	0.252±0.033•	0.474±0.039•	0.564±0.027	0.458±0.017○	0.512±0.029•	**5/2/1**
	LIFTACE	0.124±0.031	0.249±0.057•	0.191±0.020•	0.215±0.027	0.452±0.015	0.604±0.119	0.573±0.207	0.472±0.014	**2/6/0**
	SENCE	0.116±0.028	0.231±0.059	0.179±0.022	0.209±0.019	0.445±0.015	0.564±0.029	0.509±0.069	0.469±0.018	**In Total: 8/15/1**
$\mathfrak{L}$ = kNN	LIFT	0.092±0.027	0.292±0.059	0.221±0.015	0.221±0.023	0.537±0.032•	0.542±0.040	0.442±0.015	0.481±0.023	**1/7/0**
	MLSF	0.104±0.023	0.276±0.038	0.248±0.027	0.169±0.029○	0.766±0.017•	0.556±0.029	0.467±0.028	0.520±0.051•	**2/5/1**
	LIFTACE	0.116±0.044	0.283±0.053	0.231±0.035	0.221±0.016	0.526±0.033•	0.541±0.034	0.453±0.014	0.478±0.017	**1/7/0**
	SENCE	0.104±0.034	0.297±0.073	0.232±0.022	0.224±0.027	0.504±0.030	0.541±0.033	0.449±0.022	0.483±0.016	**In Total: 4/19/1**
$\mathfrak{L}$ =CART	LIFT	0.012±0.014	0.334±0.081	0.313±0.035	0.207±0.024•	0.613±0.014	0.664±0.025	0.553±0.026	0.580±0.011	**1/7/0**
	MLSF	0.022±0.026	0.371±0.081	0.385±0.045•	0.213±0.032•	0.561±0.024○	0.661±0.031	0.559±0.025	0.641±0.051•	**3/4/1**
	LIFTACE	0.010±0.014	0.337±0.065	0.305±0.028	0.211±0.026•	0.616±0.023•	0.684±0.019	0.565±0.024	0.569±0.014	**2/6/0**
	SENCE	0.006±0.013	0.344±0.068	0.309±0.033	0.170±0.036	0.592±0.026	0.680±0.018	0.550±0.012	0.564±0.020	**In Total: 6/17/1**
Base learner	Comparing algorithm	*Average precision*↑								win/tie/loss counts
		CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
$\mathfrak{L}$ =SVM	LIFT	0.498±0.014•	0.818±0.025	0.887±0.011•	0.772±0.012•	0.633±0.011	0.495±0.092	0.595±0.061	0.525±0.012•	**4/4/0**
	MLSF	0.473±0.014•	0.795±0.037•	0.824±0.029•	0.721±0.022•	0.613±0.028•	0.540±0.018	0.636±0.009	0.486±0.031•	**6/2/0**
	LIFTACE	0.498±0.016	0.817±0.031•	0.889±0.010•	0.772±0.013•	0.632±0.011	0.523±0.061	0.586±0.108	0.525±0.011•	**4/4/0**
	SENCE	0.502±0.015	0.826±0.036	0.896±0.012	0.776±0.012	0.637±0.014	0.542±0.018	0.611±0.040	0.535±0.011	**In Total: 14/10/0**
$\mathfrak{L}$ = kNN	LIFT	0.407±0.021	0.764±0.046	0.836±0.011	0.727±0.010	0.469±0.020	0.465±0.028	0.536±0.010○	0.425±0.008○	**0/6/2**
	MLSF	0.412±0.017	0.778±0.032	0.835±0.016	0.720±0.011•	0.376±0.015•	0.479±0.027○	0.551±0.019○	0.385±0.039•	**3/3/2**
	LIFTACE	0.405±0.020•	0.768±0.032	0.834±0.017	0.727±0.011	0.477±0.015○	0.458±0.024	0.520±0.010•	0.426±0.007○	**2/4/2**
	SENCE	0.410±0.019	0.763±0.051	0.829±0.010	0.728±0.010	0.460±0.016	0.459±0.022	0.526±0.009	0.418±0.006	**In Total: 5/13/6**
$\mathfrak{L}$ =CART	LIFT	0.306±0.008	0.701±0.038	0.764±0.021	0.630±0.013	0.431±0.011	0.423±0.018	0.466±0.015	0.308±0.011○	**0/7/1**
	MLSF	0.286±0.018•	0.690±0.045	0.711±0.032•	0.609±0.016•	0.453±0.019○	0.428±0.026	0.462±0.018	0.264±0.025•	**4/3/1**
	LIFTACE	0.307±0.011	0.696±0.033	0.770±0.019	0.628±0.009	0.429±0.020	0.409±0.012	0.460±0.013	0.314±0.009○	**0/7/1**
	SENCE	0.310±0.016	0.693±0.039	0.755±0.019	0.624±0.014	0.436±0.021	0.416±0.011	0.468±0.009	0.290±0.011	**In Total: 4/17/3**

TABLE VIII: Experimental results of SENCE and its two ablated variants on eight data sets. In addition, •/○ indicates whether the performance of SENCE is statistically superior/inferior to the variants on each data set (pairwise t-test at 0.05 significate level).

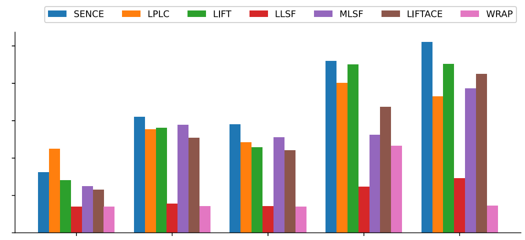
Comparing algorithm	*Hamming loss*↓								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.138±0.007	0.186±0.018	0.080±0.005•	0.195±0.008•	0.055±0.001•	0.036±0.001	0.051±0.010	0.026±0.000•	**4/4/0**
SENCE$^{\mathcal{K}}$	0.137±0.007	0.199±0.016•	0.087±0.004•	0.201±0.010•	0.056±0.002•	0.037±0.007	0.048±0.008	0.026±0.000•	**5/3/0**
SENCE	0.138±0.006	0.177±0.019	0.074±0.005	0.188±0.008	0.052±0.001	0.036±0.001	0.049±0.005	0.026±0.000	**In Total: 9/7/0**
Comparing algorithm	*Ranking loss*↓								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.181±0.006	0.151±0.022•	0.066±0.006•	0.167±0.010•	0.136±0.007•	0.114±0.007	0.082±0.007	0.114±0.005•	**5/3/0**
SENCE$^{\mathcal{K}}$	0.181±0.005	0.160±0.025•	0.071±0.004•	0.173±0.011•	0.137±0.005•	0.118±0.009	0.079±0.010	0.111±0.004•	**5/3/0**
SENCE	0.182±0.007	0.138±0.029	0.056±0.007	0.160±0.011	0.109±0.007	0.112±0.005	0.081±0.007	0.102±0.004	**In Total: 10/6/0**
Comparing algorithm	*One-error*↓								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.118±0.027	0.242±0.051	0.204±0.025•	0.225±0.024•	0.511±0.018•	0.569±0.031	0.527±0.148	0.477±0.018•	**4/4/0**
SENCE$^{\mathcal{K}}$	0.120±0.030	0.278±0.052•	0.217±0.019•	0.227±0.028•	0.514±0.017•	0.598±0.092	0.488±0.138	0.480±0.017•	**5/3/0**
SENCE	0.116±0.028	0.231±0.059	0.179±0.022	0.209±0.019	0.445±0.015	0.564±0.029	0.509±0.069	0.469±0.018	**In Total: 9/7/0**
Comparing algorithm	*Coverage*↓								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.750±0.018	0.290±0.035•	0.069±0.004•	0.454±0.015•	0.199±0.009•	0.129±0.007	0.126±0.007	0.218±0.007•	**5/3/0**
SENCE$^{\mathcal{K}}$	0.752±0.012	0.294±0.033•	0.073±0.003•	0.462±0.014•	0.200±0.007•	0.134±0.011	0.125±0.010	0.214±0.006•	**5/3/0**
SENCE	0.754±0.014	0.277±0.033	0.060±0.006	0.447±0.017	0.167±0.008	0.127±0.005	0.124±0.007	0.199±0.007	**In Total: 10/6/0**
Comparing algorithm	*Average precision*↑								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.499±0.013	0.814±0.027	0.880±0.013•	0.766±0.014•	0.581±0.014•	0.536±0.024	0.601±0.070	0.519±0.011•	**4/4/0**
SENCE$^{\mathcal{K}}$	0.498±0.012	0.800±0.032•	0.872±0.009•	0.757±0.014•	0.578±0.012•	0.524±0.040	0.626±0.067	0.520±0.012•	**5/3/0**
SENCE	0.502±0.015	0.826±0.036	0.896±0.012	0.776±0.012	0.637±0.014	0.542±0.018	0.611±0.040	0.535±0.011	**In Total: 9/7/0**
Comparing algorithm	*Macro-averaging AUC*↑								win/tie/loss counts
	CAL500	emotions	scene	yeast	arts	reference	health	NUS-WIDE-c	
SENCE$^{\mathcal{G}}$	0.516±0.013	0.834±0.028•	0.945±0.006•	0.654±0.021•	0.637±0.024•	0.565±0.038	0.597±0.034•	0.611±0.013•	**6/2/0**
SENCE$^{\mathcal{K}}$	0.520±0.026	0.828±0.025•	0.937±0.005•	0.641±0.018•	0.639±0.020•	0.554±0.029	0.596±0.032	0.621±0.017•	**5/3/0**
SENCE	0.527±0.027	0.858±0.024	0.953±0.005	0.707±0.015	0.747±0.016	0.542±0.031	0.619±0.022	0.736±0.016	**In Total: 11/5/0**

complexity of $\mathfrak{L}$ in predicting one unseen instance with b-dimensional features.

Fig.3 illustrates the execution time (training phase as well as testing phase) of all the comparing algorithms investigated in Subsection IV-A on five benchmark data sets `emotions`, `enron`, `image`, `corel5k`, and `NUS-WIDE-c`. Across the 5 data sets, their number of examples, features and class labels range from 593 to 10,000, 72 to 1001, and 5 to 374 respectively. The training time of SENCE is relatively comparable to the comparing approaches except LPLC and LLSF. Furthermore, the test time of SENCE is higher than LLSF and WRAP while relatively comparable to the other comparing approaches. Note that due to the cubic computational complexity of SENCE w.r.t. d (i.e. the number of features in input space), the proposed approach may have problem when applied to data sets with high-dimensionality features. We will



(a) Training time of comparing approaches



(b) Test time of comparing approaches

Fig. 3: Running time (training/test) of each comparing approach on five benchmark data sets. For histogram illustration, the y-axis corresponds to the logarithm of running time.

leave it for future work.

V. Conclusion

In this paper, the problem of generating label-specific features for multi-label learning is investigated. A novel approach for label-specific features generation is proposed, which stabilizes the generation process of the label-specific features via clustering ensemble techniques. Specifically, the final clustering used to construct label-specific features is obtained by fitting a mixture model on instances augmented with the base cluster assignments via the EM algorithm. Comprehensive experimental studies validate the effectiveness of the proposed approach against state-of-the-art multi-label learning algorithms. In the future, it is interesting to consider generating label-specific features by exploiting label correlations based on the proposed SENCE and investigate a more general joint distribution by taking dependency of the original instance and corresponding cluster assignment vector into account.

References

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.
[2] A. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Working Notes of the AAAI'99 Workshop on Text Learning*, Orlando, FL, 1999.
[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
[4] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2004, pp. 649–656.
[5] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.
[6] M.-L. Zhang and L. Wu, "LIFT: Multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2014.
[7] A. Topchy, J. A. K, and W. Punch, "A mixture model for clustering ensembles," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, Florida, USA, 2004, pp. 379–390.
[8] W. Zhan and M.-L. Zhang, "Multi-label learning with label-specific features via clustering ensemble," in *Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics*, Tokyo, Japan, 2017, pp. 129–136.
[9] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.
[10] H. Ayad and M. Kamel, "Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors," in *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, Surrey, UK, 2003, pp. 166–175.
[11] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1–38, 2015.
[12] M.-L. Zhang, Y.-K. Li, Y.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: An overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.
[13] C. Brinker, E. Loza Mencía, and J. Fürnkranz, "Graded multilabel classification by pairwise comparisons," in *Proceedings of the 14th IEEE International Conference on Data Mining*, Shenzhen, China, 2014, pp. 731–736.
[14] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.
[15] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.
[16] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.
[17] M. Huang, F. Zhuang, X. Zhang, X. Ao, Z. Niu, M.-L. Zhang, and Q. He, "Supervised representation learning for multi-label classification," *Machine Learning*, vol. 108, no. 5, pp. 747–763, 2019.
[18] L. Sun, S. Ji, and J. Ye, *Multi-label dimensionality reduction*. Chapman and Hall/CRC, Boca Ration, FL, 2013.
[19] C. Yan, X. Chang, M. Luo, Q. Zheng, X. Zhang, Z. Li, and F. Nie, "Self-weighted robust lda for multiclass classification with edge classes," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 1, pp. 1–19, 2020.
[20] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, "Compound rank-k projections for bilinear analysis," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 7, pp. 1502–1513, 2015.
[21] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 57–78, 2018.
[22] Z. Cai and W. Zhu, "Feature selection for multi-label classification using neighborhood preservation," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 320–330, 2017.
[23] C. Yan, Q. Zheng, X. Chang, M. Luo, C.-H. Yeh, and A. G. Hauptman, "Semantics-preserving graph propagation for zero-shot object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 8163–8176, 2020.
[24] Y. Chen, X. Yang, J. Li, P. Wang, and Y. Qian, "Fusing attribute reduction accelerators," *Information Sciences*, vol. 587, pp. 354–370, 2022.
[25] S. Canuto, M. A. Gonçalves, and F. Benevenuto, "Exploiting new sentiment-based meta-level features for effective sentiment analysis," in *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, San Francisco, CA, 2016, pp. 53–62.
[26] Y. Yang and S. Gopal, "Multilabel classification with meta-level features in a learning-to-rank framework," *Machine Learning*, vol. 88, no. 1-2, pp. 47–68, 2012.
[27] X. Wu, Q.-G. Chen, Y. Hu, D. Wang, X. Chang, X. Wang, and M.-L. Zhang, "Multi-view multi-label learning with view-specific information extraction," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macau, China, 2019, pp. 3884–3890.
[28] C. Zhang, Z. Yu, Q. Hu, P. Zhu, X. Liu, and X. Wang, "Latent semantic aware multi-view multi-label classification," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018, pp. 4414–4421.

[29] W. Zhan and M.-L. Zhang, "Inductive semi-supervised multi-label learning with co-training," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1305–1314.

[30] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3309–3323, 2016.

[31] W. Weng, Y. Chen, C. Chen, S. Wu, and J. Liu, "Non-sparse label specific features selection for multi-label classification," *Neurocomputing*, vol. 377, pp. 85–94, 2020.

[32] X.-Y. Jia, S.-S. Zhu, and W.-W. Li, "Joint label-specific features and correlation information for multi-label learning," *Journal of Computer Science and Technology*, vol. 35, no. 2, pp. 247–258, 2020.

[33] J. Zhang, C. Li, D. Cao, Y. Lin, S. Su, L. Dai, and S. Li, "Multi-label learning with label-specific features by resolving label correlations," *Knowledge-Based Systems*, vol. 159, pp. 148–157, 2018.

[34] L. Sun, M. Kudo, and K. Kimura, "Multi-label classification with meta-label-specific features," in *Proceedings of the 23rd International Conference on Pattern Recognition*, Cancun, Mexico, 2016, pp. 1612–1617.

[35] S. Xu, X. Yang, H. Yu, D.-J. Yu, J. Yang, and E. C. C. Tsang, "Multi-label learning with label-specific feature reduction," *Knowledge-Based Systems*, vol. 104, pp. 52–61, 2016.

[36] W. Weng, Y. Lin, S. Wu, Y. Li, and Y. Kang, "Multi-label learning based on label-specific features and local pairwise label correlation," *Neurocomputing*, vol. 273, pp. 385–394, 2018.

[37] J. Ma, H. Zhang, and T. W. S. Chow, "Multilabel classification with label-specific features and classifiers: A coarse- and fine-tuned framework," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 1028–1042, 2019.

[38] Z.-B. Yu and M.-L. Zhang, "Multi-label classification with label-specific feature generation: A wrapped approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, in press.

[39] M.-L. Zhang, J.-P. Fang, and Y.-B. Wang, "Bilabel-specific features for multi-label classification," *ACM Transactions on Knowledge Discovery from Data*, vol. 16, no. 1, pp. 1–23, 2021.

[40] Z.-S. Chen and M.-L. Zhang, "Multi-label learning with regularization enriched label-specific features," in *Proceedings of the 11th Asian Conference on Machine Learning*, Nagoya, Japan, 2019, pp. 411–424.

[41] C.-Y. Zhang and Z.-S. Li, "Multi-label learning with label-specific features via weighting and label entropy guided clustering ensemble," *Neurocomputing*, vol. 419, pp. 59–69, 2021.

[42] J. Huang, G. Li, S. Wang, Z. Xue, and Q. Huang, "Multi-label classification by exploiting local positive and negative pairwise label correlation," *Neurocomputing*, vol. 257, pp. 164–174, 2017.

[43] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. Article 27, 2011, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[44] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[45] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.