Multi-Dimensional Classification via Decomposed Label Encoding

Bin-Bin Jia and Min-Ling Zhang *Senior Member, IEEE*

Abstract—In multi-dimensional classification (MDC), a number of class variables are assumed in the output space with each of them specifying the class membership w.r.t. one heterogeneous class space. One major challenge in learning from MDC examples lies in the heterogeneity of class spaces, where the modeling outputs from different class spaces are not directly comparable. To tackle this problem, we propose a new strategy named *decomposed label encoding* which enables modeling alignment for MDC in an encoded label space derived from one-vs-one (OvO) decomposition. Specifically, the original MDC output space is transformed into a ternary encoded label space by conducting OvO decomposition w.r.t. each class space. Then, the manifold structure in the feature space is exploited to enrich the labeling information in the encoded label space. Finally, the predictive model is induced by fitting the metric-aligned modeling outputs with enriched labeling information. Extensive experiments over twenty benchmark data sets clearly show the superiority of the proposed MDC strategy against state-of-the-art approaches.

Index Terms—machine learning, multi-dimensional classification, label encoding, one-vs-one decomposition

1 Introduction

In many real-world applications, the modeling problem can be formalized under the traditional multi-class classification framework, where each object is represented by one instance (feature vector) while associated with a single class variable. However, there are also other application scenarios where the objects' semantics cannot be simply characterized by a single class variable. For example, *e*-commerce websites usually need to simultaneously classify smartphones from the `brand` dimension (with the possible classes *Huawei*, *Samsung*, *Apple*, etc.), from the `operating system` dimension (with the possible classes *Android*, *iOS*, *Windows Phone*, etc.), from the `CPU brand` dimension (with the possible classes *Qualcomm*, *MediaTek*, *Hisilicon*, etc.), etc. Actually, similar application scenarios widely exist in real-word applications such as bioinformatics [7], [45], text classification [49], [50], resource allocation [1], etc. To characterize the rich semantics of such kind of objects, one natural solution is to associate multiple class variables with the objects, which results in the learning framework *multi-dimensional classification* (MDC) [25], [39], [43]. In contrast to multi-class classification, in MDC each example is also represented by one instance while associated with multiple class variables simultaneously. Here, each class variable corresponds to one specific class space which characterizes the objects' semantics from one dimension.

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ be the input (feature) space, and $\mathcal{Y} = C_1 \times C_2 \times \cdots \times C_q$ be the output space. Here, $\mathcal{Y}$ corresponds to the Cartesian product of q class spaces $C_j = \{c_1^j, c_2^j, \ldots, c_{K_j}^j\}$ $(1 \leq j \leq q)$ which consists of K_j possible class labels respectively. Given the MDC training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leq i \leq m\}$ with m training examples, for each example $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}$, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^\top \in \mathcal{X}$ is a d-dimensional feature vector and $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^\top \in \mathcal{Y}$ is the q-dimensional class vector associated with $\boldsymbol{x}_i$ with each component y_{ij} representing one possible class label in C_j, i.e., $y_{ij} \in C_j$. The task of multi-dimensional classification is to learn a mapping function $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$ which can return a proper class vector $f(\boldsymbol{x}_*) \in \mathcal{Y}$ for unseen instance $\boldsymbol{x}_*$.

Obviously, the MDC problem can be solved by training a number of independent multi-class classifiers, one per dimension. However, the simple decomposition strategy isn't consistent with the intention of MDC task which aims at inducing a unified model $f : \mathcal{X} \mapsto \mathcal{Y}$ for all dimensions. In other words, potential dependencies among class spaces should be considered when learning MDC models. An intuitive strategy in this way is to solve the MDC problem by considering all class variables as a single compound one, i.e., each distinct class combination in $\mathcal{D}$ is regarded as a new class. However, this powerset-like transformation strategy would suffer high computational cost due to its combinatorial nature and is incapable of predicting class combinations absent in the training set. Therefore, most existing MDC approaches focus on how to model class dependencies in appropriate ways, such as capturing pairwise class dependencies [2], [26], [27], learning a directed acyclic graph (DAG) structure for class spaces [5], [18], specifying chaining order over class spaces [42], [68], and grouping class spaces into super-classes [43], etc.

However, these approaches mainly deal with the MDC problem in the original output space $\mathcal{Y}$ which is quite challenging due to the heterogeneity of class spaces. Specifically, in MDC the output space consists of multiple heterogeneous class spaces, which is the essential difference between MDC and other related classification problems (e.g., multi-class/multi-label classification) [25]. The heterogeneity of class spaces makes the modeling outputs from different class spaces not directly comparable, which leads to the infeasibility of applying popular multi-class/multi-label classification techniques to learn from MDC examples. For example, ranking-based techniques are often utilized to distinguish

- *Bin-Bin Jia is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China and the College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China. E-mail: jiabb@seu.edu.cn*
- *Min-Ling Zhang (corresponding author) is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: zhangml@seu.edu.cn*

relevant and irrelevant labels when inducing multi-class/multi-label models [22], [71], while such techniques cannot be generalized to inducing MDC models.

To tackle the heterogeneity of class spaces, we propose to enable modeling alignment for MDC by employing the *label encoding* strategy. Although label encoding has been successfully applied to learning problems with non-unique labeling such as multi-label classification [21], [36], [37], [53], [73], its effectiveness in solving MDC problem is firstly investigated in this paper. Accordingly, a novel approach named DLEM, i.e., *Decomposed Label Encoding for Multi-dimensional classification*, is proposed by adapting the popular one-vs-one decomposition. Firstly, DLEM transforms the MDC output space into a ternary label space with negative, null or positive label assignment via OvO decomposition w.r.t. each class space. Then, the labeling information in the encoded label space is enriched by exploiting the manifold structure in the feature space. Finally, DLEM induces the predictive model by fitting the metric-aligned modeling outputs with enriched labeling information. Here, we would like to reiterate the differences and advantages of DLEM over existing MDC approaches, where the predictive model is induced in an encoded label space by DLEM while in the original heterogeneous label space by existing MDC approaches. By doing this, we expect DLEM can achieve better generalization performance, and extensive experiments over twenty benchmark data sets clearly show the superiority of DLEM against state-of-the-art MDC approaches.

The rest of this paper is organized as follows. Firstly, related works on MDC are briefly discussed. Secondly, technical details of the proposed approach are introduced. Thirdly, experimental results of comparative studies are reported. Finally, we conclude this paper.

2 RELATED WORK

Multi-dimensional classification has a close relationship with the widely studied multi-label classification (MLC) framework [17], [70], [72], while both of them can be regarded as specific instantiations of multi-output learning [65]. As per their mathematical definitions, each dimension in MDC corresponds to a multi-class variable while each label in MLC corresponds to a binary class variable. Furthermore, MDC usually assumes *heterogeneous* class spaces where different dimensions correspond to different semantic spaces, while MLC usually assumes *homogeneous* class space where different labels correspond to the relevancy of concepts in the same semantic space. Besides, one recent development in MDC named multi-dimensional partial label learning (MDPL) [64] considers a more complicated setting, where the ground-truth class label in each dimension is concealed in a candidate label set which makes the problem more challenging to be tackled.

Obviously, the MDC problem can be simply solved dimension by dimension, i.e., training a multi-class classifier for each class space independently. However, possible dependencies among class spaces are not considered by this intuitive strategy which would impact its generalization performance. Actually, one of the key challenges for MDC studies is how to learn a unified model for all dimensions instead of q independent models for each dimension. To induce a unified model for all dimensions, one strategy is to learn a directed acyclic graph (DAG) over class spaces [14], [46], [60], where different DAG structures correspond to different approaches which form a family of MDC models called multi-dimensional Bayesian network classifier (MBC) [5], [18]. Recent works on MBC mainly focus on designing efficient DAG structure learning algorithms, which is still challenging [3], [6], [77] due to large structure space. Another strategy is to train a chain of multi-class classifiers, one per class space, where predictions of preceding classifiers on the chain are used as extra features by the subsequent ones [42], [68]. Generally, the chaining order largely affects the performance of this strategy, but it is actually a NP-hard problem to determine an optimal one.

On the other hand, the MDC problem can be tackled by utilizing only one multi-class classifier, where each distinct class combination appearing in the training set can be treated as a new class. However, following this strategy, class combinations not appearing in the training set cannot be predicted for unseen instance and the computational complexity would be high due to the huge number of new classes. These deficiencies can be mitigated to some extent by grouping the class spaces into super-classes [43], but cannot be fully addressed due to the combinatorial nature. The MDC problem can also be tackled with a two-level strategy, where preliminary models are learned for each pair of class spaces via powerset transformation, and then meta models are learned for all class spaces based on the predictions of the preliminary models [2], [26], [27]. However, training classification models for pairwise class spaces leads to $\mathcal{O}(q^2)$ complexity which is computationally demanding.

In general, one-vs-rest and one-vs-one are two commonly used transformation strategies for multi-class classification problems. The M^{3}MDC approach decomposes each class space of MDC via one-vs-one strategy and then jointly solves the resulting binary classification problems by introducing a covariance regularization term [24]. However, the derived quadratic programming problem contains $m \cdot \sum_{j=1}^{q}(K_j - 1)$ variables which is usually too large making it difficult to be solved. The gMML approach conducts a multi-label like transformation for the MDC output space which can be regarded as one-vs-rest strategy and then learns a multi-output regressor for the resulting problem as well as a Mahalanobis distance metric [39]. However, the one-vs-rest encoded label space directly aligns class labels from different class spaces which is less reasonable due to the heterogeneity assumption in MDC.

It is worth noting that the label encoding strategy has been utilized in solving related learning problems such as multi-label classification. The pioneering work of multi-label prediction via compressed sensing [20] simply maps the sparse label space into a real-valued one with random sensing matrices which satisfy the restricted isometry property. The following works mainly focus on how to encode the label space into a more informative one via different ways, such as conducting principle label space transformation [56] or feature-aware label space transformation [12], [31], [33], [34], maximizing the margin between correct and incorrect encoded label vectors [36], [37], [73], learning neural networks to accomplish the encoding step [11], [30], [32], etc. There are also some works which claim better generalization performance by encoding the binary label space into another binary one instead of a real-valued one [51], [52], [75]. However, to the best of our knowledge, no existing works solve the MDC problem with label encoding strategy. In the next section, we will present the technical details of the proposed DLEM approach which deal with the MDC problem via decomposed label encoding.

3 THE DLEM APPROACH

The learning procedure of DLEM consists of three steps, including decomposed label encoding, labeling information enrichment, and predictive model induction. Technical details of these steps are scrutinized as follows.

3.1 Decomposed Label Encoding

Following the same notations defined in Section 1, for the MDC training set $\mathcal{D}$, let $\mathbf{X}$ be the instance matrix with size $m \times d$ where the ith row corresponds to the transpose of feature vector $\boldsymbol{x}_i \in \mathcal{X}$, and $\mathbf{Y}$ be the label matrix with size $m \times q$ where the ith row corresponds to the transpose of class vector $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^\top \in \mathcal{Y}$. According to OvO decomposition rule, $\mathbf{Y}$ can be transformed into a ternary encoded label matrix $\mathbf{L} = [\mathbf{L}^1, \mathbf{L}^2, \ldots, \mathbf{L}^q] \in \{-1, 0, +1\}^{m\times\ell}$. Here, $\mathbf{L}_j \in \{-1, 0, +1\}^{m\times\ell_j}$ corresponds to the encoded label matrix of the jth class space (i.e., the transformation of jth column of label matrix $\mathbf{Y}$) where $\ell_j = \binom{K_j}{2}$ $(1 \le j \le q)$, and $\ell = \sum_{j=1}^{q} \ell_j$. Without loss of generality, for $\mathbf{L}^j$, the ath column $(1 \le a \le \ell_j)$ corresponds to the pair of class labels (p_a^j, n_a^j) in C_j:

$$(p_a^j, n_a^j) = (c_u^j, c_{a+u-g_j(u-1)}^j), \qquad (1)$$
$$\text{when } 1 + g_j(u-1) \le a \le g_j(u) \;\; (1 \le u \le K_j - 1)$$

where $g_j(0) = 0$ and $g_j(t) = \sum_{v=1}^{t}(K_j - v)$ when $1 \le t \le K_j - 1$. It is easy to verify that $\ell_j = g_j(K_j - 1)$. Let l_{ia}^j be the element in ith row and ath column of $\mathbf{L}^j$, its value is determined as follows:

$$l_{ia}^j = \begin{cases} +1, & \text{if } y_{ij} = p_a^j \\ -1, & \text{if } y_{ij} = n_a^j \\ \;\;\,0, & \text{otherwise} \end{cases} \qquad (2)$$

Example 1. *Given the MDC data set $\mathcal{D}$ with $m = 4$ training examples, i.e., $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), (\boldsymbol{x}_3, \boldsymbol{y}_3), (\boldsymbol{x}_4, \boldsymbol{y}_4)\}$. Assume that the jth class space includes 4 class labels, i.e., $K_j = 4$ and $C_j = \{c_1^j, c_2^j, c_3^j, c_4^j\}$. For $\mathcal{D}$, assume that $y_{ij} = c_i^j$ $(1 \le i \le 4)$, i.e., the jth column of label matrix $\mathbf{Y}$ corresponds to $[c_1^j, c_2^j, c_3^j, c_4^j]^\top$. For the encoded label matrix $\mathbf{L}_j$ of the jth class space, according to Eq.(1), the first 3 columns $(1 \le a \le 3, \text{i.e.}, u = 1)$ correspond to (c_1^j, c_2^j), (c_1^j, c_3^j) and (c_1^j, c_4^j) respectively, the following 2 columns $(4 \le a \le 5, \text{i.e.}, u = 2)$ correspond to (c_2^j, c_3^j) and (c_2^j, c_4^j) respectively, and the last column $(a = 6, \text{i.e.}, u = 3)$ corresponds to (c_3^j, c_4^j). According to Eq.(2), it is easy to know that the value of $\mathbf{L}_j$ is as follows:*

$$\mathbf{L}_j = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

Here, each column of $\mathbf{L}$ corresponds to an OvO decomposition problem, where instances with label '$+1$' (or '-1') serve as positive (or negative) examples, and instances with label '0' are not considered in the current binary classification problem. Generally, we can simply train ℓ independent binary classifiers over examples with label '$+1$' and '-1' under the supervision of $\mathbf{L}$, one per column. However, all these binary classification problems originate from the MDC problem via OvO decomposition and should be solved in a joint manner due to potential relationships among them [24], [35]. Furthermore, the ternary labeling confidence with negative, null or positive might be inaccurate due to existence of possible label noise in real-world applications [76]. In this paper, DLEM aims at solving all OvO decomposition problems via a unified model with real-valued labeling confidence.

3.2 Labeling Information Enrichment

To obtain real-valued labeling confidence, DLEM enriches the labeling information residing in $\mathbf{L}$ which is initialized via Eq.(2) by leveraging the structural information in the feature space. Specifically, DLEM assumes that similar manifold structures exist in the input and output spaces.

Following the ideas of locally linear embedding [47], [62], each instance $\boldsymbol{x}$ can be reconstructed via linear combination of its k nearest neighbors, and this relationship also holds in the label space. For each training example $\boldsymbol{x}_i$ $(1 \le i \le m)$, the linear combination coefficients for its k nearest neighbors can be determined by solving the following optimization problem:

$$\min_{s_{ij_1}, \ldots, s_{ij_k}} \left\| \boldsymbol{x}_i - \sum_{j_r \in \mathcal{N}_k(\boldsymbol{x}_i)} s_{ij_r} \boldsymbol{x}_{j_r} \right\|_2^2, \;\; \text{s.t.} \; \sum_{a=1}^{k} s_{ij_a} = 1 \qquad (3)$$

where $\mathcal{N}_k(\boldsymbol{x}_i) = \{j_r \mid 1 \le r \le k\}$ represents the set of indices for $\boldsymbol{x}_i$'s k nearest neighbors. Furthermore, $\boldsymbol{s}_i = [s_{i1}, s_{i2}, \ldots, s_{im}]^\top$ where s_{ij} is determined by the above optimization problem if $j \in \mathcal{N}_k(\boldsymbol{x}_i)$ and $s_{ij} = 0$ otherwise. It is easy to know that Eq.(3) has the following closed-form solution:

$$[s_{ij_1}, \ldots, s_{ij_k}]^\top = \frac{\mathbf{C}_i^{-1}\mathbf{1}_k}{\mathbf{1}_k^\top \mathbf{C}_i^{-1} \mathbf{1}_k} \qquad (4)$$

where $\mathbf{C}_i = \mathbf{D}_i^\top \mathbf{D}_i \in \mathbb{R}^{k\times k}$, $\mathbf{D}_i = [\boldsymbol{x}_i - \boldsymbol{x}_{j_1}, \boldsymbol{x}_i - \boldsymbol{x}_{j_2}, \ldots, \boldsymbol{x}_i - \boldsymbol{x}_{j_k}] \in \mathbb{R}^{d\times k}$, and $\mathbf{1}_k$ is a column vector of all ones with length k.

Let $\mathbf{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_m]^\top \in \mathbb{R}^{m\times\ell}$ be the enriched label matrix of $\mathbf{L}$. After all $\boldsymbol{s}_i$ $(1 \le i \le m)$ have been obtained, $\mathbf{F}$ could be generated by solving the following optimization problem:

$$\min_{\mathbf{F}} \sum_{i=1}^{m} \left\| \boldsymbol{f}_i - \sum_{j_r \in \mathcal{N}_k(\boldsymbol{x}_i)} s_{ij_r} \boldsymbol{f}_j \right\|_2^2 + \lambda \left\| \mathbf{F} - \mathbf{L} \right\|_F^2 \qquad (5)$$

where λ is a trade-off parameter. The first term ensures the similar manifold structure to the feature space is kept in the enriched label space, and the second term ensures the obtained label matrix $\mathbf{F}$ should also be similar to the original label matrix $\mathbf{L}$.

The optimization problem can be equivalently reformulated as follows:

$$\min_{\mathbf{F}} \; \mathrm{tr}\left(\mathbf{F}^\top(\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top \mathbf{F}\right) + \lambda \left\| \mathbf{F} - \mathbf{L} \right\|_F^2 \qquad (6)$$

where $\mathrm{tr}(\cdot)$ computes the trace of a square matrix, $\mathbf{I}_m$ represents an $m \times m$ identity matrix, and $\mathbf{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m] \in \mathbb{R}^{m\times m}$.

Obviously, the second term in the objective function is convex w.r.t. $\mathbf{F}$. For the first term, because $(\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top$ is always positive semi-definite[1], we have $\mathbf{F}_{:j}^\top(\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top \mathbf{F}_{:j}$ is convex w.r.t. $\mathbf{F}_{:j}$ where $\mathbf{F}_{:j}$ is the jth column of $\mathbf{F}$. Furthermore, the first term can be expressed as $\sum_{j=1}^{\ell} \mathbf{F}_{:j}^\top(\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top \mathbf{F}_{:j}$ and the convexity can be preserved after summation operation which results in the convexity of the first term. Therefore, the objective function is jointly convex w.r.t. $\mathbf{F}$.

1. For any $\boldsymbol{x} \in \mathbb{R}^{m\times 1}$, we have $\boldsymbol{x}^\top(\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top \boldsymbol{x} = \|(\mathbf{I}_m - \mathbf{S})^\top \boldsymbol{x}\|_2^2 \ge 0$

Let $\Lambda(\mathbf{F})$ be the objective function of Eq.(6), the gradient of $\Lambda(\mathbf{F})$ is given as follows:

$$\frac{\partial \Lambda(\mathbf{F})}{\partial \mathbf{F}} = 2(\boldsymbol{I}_m - \mathbf{S})(\boldsymbol{I}_m - \mathbf{S})^\top \mathbf{F} + 2\lambda \mathbf{F} - 2\lambda \mathbf{L}$$

By setting $\frac{\partial \Lambda(\mathbf{F})}{\partial \mathbf{F}}$ to 0, we can obtain a closed-form solution of $\mathbf{F}$ as follows:

$$\mathbf{F} = \Big((\mathbf{I}_m - \mathbf{S})(\mathbf{I}_m - \mathbf{S})^\top + \lambda \mathbf{I}_m \Big)^{-1} (\lambda \mathbf{L}) \tag{7}$$

Thereafter, labeling information is aligned in the output space via the label encoding and enrichment procedure. Specifically, each element f_{ij} $(1 \le i \le m, 1 \le j \le \ell)$ in the real-valued matrix $\mathbf{F}$ can be regarded as the labeling confidence of the ith instance on the j encoded label.

3.3 Predictive Model Induction

As the enriched label matrix $\mathbf{F}$ is real-valued, it is natural to tackle the resulting problem with multi-output regression techniques [8]. Specifically, we can train a multi-output regressor over $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \boldsymbol{f}_i) \mid 1 \le i \le m\}$ by simply solving the following optimization problem:

$$\min_{\mathbf{W}} \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \right\|_2^2 + \frac{\gamma}{2} \|\mathbf{W}\|_F^2 \tag{8}$$

Here, γ is a trade-off parameter, $\phi(\cdot)$ is the (implicit) nonlinear mapping by kernel function $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and $\phi(\boldsymbol{x}_i) \in \mathbb{R}^{d' \times 1}$, $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_\ell] \in \mathbb{R}^{d' \times \ell}$ corresponds to the regression model to be determined. However, the above multi-output regressor actually deals with the ℓ output variables independently. Following the metric learning idea [36], [37], [39], [55], the ℓ output variables can be tackled in a joint manner by employing a Mahalanobis distance metric $\mathbf{M}$:

$$\min_{\mathbf{W}} \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \right\|_{\mathbf{M}}^2 + \frac{\gamma}{2} \|\mathbf{W}\|_F^2 \tag{9}$$

where $\|\boldsymbol{a} - \boldsymbol{b}\|_{\mathbf{M}}^2 = (\boldsymbol{a} - \boldsymbol{b})^\top \mathbf{M} (\boldsymbol{a} - \boldsymbol{b})$ returns the square of Mahalanobis distance between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The metric $\mathbf{M}$ aims at shortening the distance between $\mathbf{W}^\top \phi(\boldsymbol{x}_i)$ and $\boldsymbol{f}_i$ and enlarging the distance between $\mathbf{W}^\top \phi(\boldsymbol{x}_i)$ and non-$\boldsymbol{f}_i$s. Therefore, $\mathbf{M}$ can be determined by the following optimization problem [39], [67]:

$$\begin{aligned}
\min_{\mathbf{M} \succ 0} \;& \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \right\|_{\mathbf{M}}^2 \\
&+ \frac{1}{m \cdot k} \sum_{i=1}^{m} \sum_{i_r \in \mathcal{N}_k(\boldsymbol{f}_i)} \left\| \mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_{i_r} \right\|_{\mathbf{M}^{-1}}^2 \\
&+ \mu \cdot D(\mathbf{M}, \mathbf{I}_\ell)
\end{aligned} \tag{10}$$

where $\mathcal{N}_k(\boldsymbol{f}_i) = \{i_r \mid 1 \le r \le k\}$ is the set of indices for $\boldsymbol{f}_i$'s k nearest neighbors in $\mathbf{F}$, $D(\mathbf{M}, \mathbf{I}_\ell) = \operatorname{tr}(\mathbf{M}) + \operatorname{tr}(\mathbf{M}^{-1}) - 2\ell$ is the symmetrized LogDet divergence, and μ is a trade-off parameter. Here, the first term makes the distance between $\mathbf{W}^\top \phi(\boldsymbol{x}_i)$ and $\boldsymbol{f}_i$ closer, the second term makes the distance between $\mathbf{W}^\top \phi(\boldsymbol{x}_i)$ and non-$\boldsymbol{f}_i$s farther, and the third term penalizes the complexity of $\mathbf{M}$ to avoid overfitting.

Obviously, $\mathbf{M}$ should be known when solving the optimization problem w.r.t. $\mathbf{W}$ in Eq.(9), while $\mathbf{W}$ should be known when solving the optimization problem w.r.t. $\mathbf{M}$ in Eq.(10). The interaction between $\mathbf{W}$ and $\mathbf{M}$ prevents them from being calculated simultaneously. In this paper, we alternatingly calculate one of them while the remaining one is fixed until convergence.

Calculating $\mathbf{W}$ when $\mathbf{M}$ is fixed. Because there is the nonlinear mapping $\phi(\cdot)$ by kernel function κ, for the optimization problem in Eq.(9), we canot always obtain an explicit solution of $\mathbf{W}$. According to the Representer Theorem [48], under fairly general conditions, the predictive model can be expressed as a linear combination of the training instances. Let $\mathbf{\Phi} = [\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_m)]^\top \in \mathbb{R}^{m \times d'}$ be the nonlinear mapping instance matrix of $\mathbf{X}$, for the multi-output regression problem in Eq.(9), we have $\boldsymbol{w}_j = \sum_{i=1}^{m} \theta_{ji} \phi(\boldsymbol{x}_i) = \mathbf{\Phi}^\top \boldsymbol{\theta}_j$ and then $\mathbf{W} = \mathbf{\Phi}^\top \mathbf{\Theta}$, where $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\ell] \in \mathbb{R}^{m \times \ell}$ is the combination coefficients to be determined. Plugging $\mathbf{W} = \mathbf{\Phi}^\top \mathbf{\Theta}$ into the objective function in Eq.(9) which is denoted as $\Gamma(\mathbf{W})$:

$$\begin{aligned}
\Gamma(\mathbf{W}) =& \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{\Theta}^\top \mathbf{\Phi} \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \right\|_{\mathbf{M}}^2 + \frac{\gamma}{2} \left\| \mathbf{\Phi}^\top \mathbf{\Theta} \right\|_F^2 \\
=& \frac{1}{m} \left\| \mathbf{\Phi} \mathbf{\Phi}^\top \mathbf{\Theta} - \mathbf{F} \right\|_{\mathbf{M}}^2 + \frac{\gamma}{2} \left\| \mathbf{\Phi}^\top \mathbf{\Theta} \right\|_F^2 \\
=& \frac{1}{m} \operatorname{tr}\left((\mathbf{\Phi} \mathbf{\Phi}^\top \mathbf{\Theta} - \mathbf{F}) \mathbf{M} (\mathbf{\Phi} \mathbf{\Phi}^\top \mathbf{\Theta} - \mathbf{F})^\top \right) \\
&+ \frac{\gamma}{2} \operatorname{tr}\left(\mathbf{\Theta}^\top \mathbf{\Phi} \mathbf{\Phi}^\top \mathbf{\Theta} \right) \triangleq \Gamma(\mathbf{\Theta})
\end{aligned}$$

Let $\mathbf{K} = \mathbf{\Phi} \mathbf{\Phi}^\top \in \mathbb{R}^{m \times m}$ be the kernel matrix with (i,j)th element $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, then the gradient of $\Gamma(\mathbf{\Theta})$ w.r.t. $\mathbf{\Theta}$ is given as follows:

$$\begin{aligned}
\frac{\partial \Gamma(\mathbf{\Theta})}{\partial \mathbf{\Theta}} = \frac{1}{m} \big(& \mathbf{K}^\top \mathbf{K} \mathbf{\Theta} \mathbf{M} + \mathbf{K}^\top \mathbf{K} \mathbf{\Theta} \mathbf{M}^\top \\
& - \mathbf{K}^\top \mathbf{F} \mathbf{M} - \mathbf{K}^\top \mathbf{F} \mathbf{M}^\top \big) + \gamma \mathbf{K} \mathbf{\Theta}
\end{aligned}$$

By setting the above gradient to 0, we have:

$$\begin{aligned}
(m\gamma) \cdot (\mathbf{K}^\top \mathbf{K})^{-1} \mathbf{K} \mathbf{\Theta} + \mathbf{\Theta} (\mathbf{M} + \mathbf{M}^\top) \\
= (\mathbf{K}^\top \mathbf{K})^{-1} \mathbf{K}^\top \mathbf{F} (\mathbf{M} + \mathbf{M}^\top)
\end{aligned} \tag{11}$$

which is a Sylvester equation w.r.t. $\mathbf{\Theta}$ and can be solved by any off-the-shelf solvers.

Calculating $\mathbf{M}$ when $\mathbf{W}$ is fixed. The optimization problem in Eq.(10) can be equivalently reformulated as follows:

$$\min_{\mathbf{M} \succ 0} \operatorname{tr}(\mathbf{M}\mathbf{U}) + \operatorname{tr}(\mathbf{M}^{-1}\mathbf{V}) + \mu \cdot D(\mathbf{M}, \mathbf{I}_\ell) \tag{12}$$

Here,

$$\begin{aligned}
\mathbf{U} =& \frac{1}{m} \sum_{i=1}^{m} \Big(\mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \Big) \Big(\mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \Big)^\top \\
=& \frac{1}{m} \sum_{i=1}^{m} \Big(\mathbf{\Theta}^\top \mathbf{\Phi} \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \Big) \Big(\mathbf{\Theta}^\top \mathbf{\Phi} \phi(\boldsymbol{x}_i) - \boldsymbol{f}_i \Big)^\top \\
=& \frac{1}{m} (\mathbf{K} \mathbf{\Theta} - \mathbf{F})^\top (\mathbf{K} \mathbf{\Theta} - \mathbf{F})
\end{aligned} \tag{13}$$

$$\begin{aligned}
\mathbf{V} =& \frac{1}{m \cdot k} \sum_{i=1}^{m} \sum_{i_r \in \mathcal{N}_k(\boldsymbol{f}_i)} \Big(\mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_{i_r} \Big) \Big(\mathbf{W}^\top \phi(\boldsymbol{x}_i) - \boldsymbol{f}_{i_r} \Big)^\top \\
=& \frac{1}{m \cdot k} \sum_{i=1}^{m} \sum_{i_r \in \mathcal{N}_k(\boldsymbol{f}_i)} \Big(\mathbf{\Theta}^\top \mathbf{\Phi} \phi(\boldsymbol{x}_i) - \boldsymbol{f}_{i_r} \Big) \Big(\mathbf{\Theta}^\top \mathbf{\Phi} \phi(\boldsymbol{x}_i) - \boldsymbol{f}_{i_r} \Big)^\top \\
=& \frac{1}{m \cdot k} \sum_{r=1}^{k} (\mathbf{K} \mathbf{\Theta} - \mathbf{F}_r)^\top (\mathbf{K} \mathbf{\Theta} - \mathbf{F}_r)
\end{aligned} \tag{14}$$

Algorithm 1 The proposed DLEM approach.

Input: $\mathcal{D}$: MDC training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leq i \leq m\}$
k: number of nearest neighbors considered
λ, γ, μ: trade-off parameters
$\boldsymbol{x}_*$: unseen instance

Output: $\boldsymbol{y}_*$: predicted class vector for $\boldsymbol{x}_*$

1: Obtain the transformed label matrix $\mathbf{L}$ according to Eq.(1) and Eq.(2);
2: Obtain the enriched label matrix $\mathbf{F}$ according to Eq.(7);
3: Initialize $\mathbf{M} = \mathbf{I}_\ell$;
4: **repeat**
5: Obtain $\boldsymbol{\Theta}$ by solving the Sylvester equation in Eq.(11);
6: Obtain $\mathbf{M}$ according to Eq.(15);
7: **until** convergence
8: Obtain enriched label vector $\boldsymbol{f}_*$ for $\boldsymbol{x}_*$ according to Eq.(16);
9: Return $\boldsymbol{y}_*$ by applying OvO decoding rule over $\boldsymbol{l}_* = \mathrm{sign}(\boldsymbol{f}_*)$.

where $\mathbf{F}_r = [\boldsymbol{f}_{1_r}, \boldsymbol{f}_{2_r}, \ldots, \boldsymbol{f}_{m_r}]^\top$. Following [67], the optimization problem in Eq.(12) is strictly convex, and then its global minimum can be obtain when the gradient of the objective function vanishes. Specifically, by calculating the gradient w.r.t. $\mathbf{M}$ and setting it to 0, we can obtain:

$$(\mathbf{U} + \mu\mathbf{I}_\ell) - \mathbf{M}^{-1}(\mathbf{V} + \mu\mathbf{I}_\ell)\mathbf{M}^{-1} = 0$$

Here, the above equation is equivalent to $\mathbf{M}(\mathbf{U}+\mu\mathbf{I}_\ell)\mathbf{M} = (\mathbf{V}+\mu\mathbf{I}_\ell)$ which is actually a *Riccati equation* [4]. Its unique solution corresponds to the midpoint of the geodesic joining $(\mathbf{U}+\mu\mathbf{I}_\ell)^{-1}$ to $(\mathbf{V} + \mu\mathbf{I}_\ell)$, i.e.,

$$\mathbf{M} = (\mathbf{U} + \mu\mathbf{I}_\ell)^{-1} \#_{1/2} (\mathbf{V} + \mu\mathbf{I}_\ell) \tag{15}$$

where $\mathbf{A}\#_{1/2}\mathbf{B} = \mathbf{A}^{1/2}\left(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2}\right)^{1/2}\mathbf{A}^{1/2}$.

As the above two alternating optimizing steps converge, we can obtain the predictive model, i.e., the optimal values of $\mathbf{W}$ (or $\boldsymbol{\Theta}$ in Eq.(11)). Then, for unseen instance $\boldsymbol{x}_*$, its enriched label vector can be predicted as follows:

$$\boldsymbol{f}_* = \mathbf{W}^\top\phi(\boldsymbol{x}_*) = \boldsymbol{\Theta}^\top\mathbf{K}^* \tag{16}$$

where $\mathbf{K}^* = \boldsymbol{\Phi}\phi(\boldsymbol{x}_*) \in \mathbb{R}^{m\times 1}$ with elements $K_i^* = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_*)$ $(1 \leq i \leq m)$. Finally, we can determine $\boldsymbol{x}_*$'s binary label vector by $\boldsymbol{l}_* = \mathrm{sign}(\boldsymbol{f}_*)$, where $\mathrm{sign}(\cdot)$ represents the (element-wise) signed function. It is easy to know that the I_1^jth $\sim I_2^j$th elements in $\boldsymbol{l}_*$ belong to the jth class space, based on which we can predict the corresponding class label via OvO decoding rule of majority voting. Here, $I_1^j = \sum_{a=1}^{j-1}\binom{K_a}{2} + 1$ and $I_2^j = \sum_{a=1}^{j}\binom{K_a}{2}$.

In summary, Algorithm 1 presents the complete procedure of the proposed DLEM approach. Firstly, we obtain the transformed label matrix $\mathbf{L}$ (Step 1), and then obtain the enriched label matrix $\mathbf{F}$ (Step 2). After that, an alternating optimizing process is used to solve the multi-output regressor in Eq.(9) (Steps 3-7). Finally, the class vector for unseen instance is predicted by applying the OvO decoding rule over the binarized label vector $\boldsymbol{l}_*$ (Steps 8-9).

It is worth noting that, for the proposed DLEM approach, the first two steps provide the labeling information where the first step gives the basic labeling information which is further enriched by the second step, while the last step induces the predictive model supervised by the obtained labeling information where a multi-output regressor is learned to solve the resulting problem. In the case that the labeling information provided by the first two steps is inaccurate or the predictive model induced in the third step is under-performed, the generalization performance of DLEM would be impacted. Therefore, we can further explore more advanced techniques to obtain more accurate labeling information and more powerful predictive model in future. For example, for the labeling information enrichment step, following the idea in [28], [74], we can also attempt to obtain the coefficients and enriched labels in a unified formulation rather than separately optimizing Eq.(3) and Eq.(5). Moreover, following the idea of ALP-TMR [69], we can further consider the noise and outliers in instances, enriched labels and coefficients to improve model's robustness. Nonetheless, the technical choice of DLEM has made it achieve very competitive performance against state-of-the-art MDC baselines according to the experimental results which will be reported in the next section.

TABLE 1: Characteristics of the benchmark data sets.

Data Set	#Exam.	#Dim.	#Labels/Dim.	#Features†
Edm	154	2	3	$16n$
Oes97	334	16	3	$263n$
Jura	359	2	4,5	$9n$
Oes10	403	16	3	$298n$
Enb	768	2	2,4	$6n$
Song	785	3	3	$98n$
BeLaE	1930	5	5	$1n$,$44x$
Voice	3136	2	4,2	$19n$
Scm20d	8966	16	4	$61n$
Rf1	8987	8	4,4,3,4,4,3,4,3	$64n$
Thyroid	9172	7	5,5,3,2,4,4,3	$7n$, $22x$
Pain	9734	10	2,5,4,2,2,5,2,5,2,2	$136n$
Scm1d	9803	16	4	$280n$
CoIL2000	9822	5	6,10,10,4,2	$81x$
TIC2000	9822	3	6,4,2	$83x$
Flickr	12198	5	3,4,3,4,4	$1536n$
Disfa	13095	12	5,5,6,3,4,4,5,4,4,4,6,4	$136n$
Fera	14052	5	6	$136n$
Adult	18419	4	7,7,5,2	$5n$,$5x$
Default	28779	4	2,7,4,2	$14n$,$6x$

† n, x denote numeric and nominal features respectively.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Benchmark Data Sets

In this paper, we have collected a total of 20 real-world MDC data sets for comparative studies. Table 1 summarizes the detailed characteristics of all benchmark data sets, including *number of examples* (#Exam.), *number of dimensions* (#Dim.), *number of class labels per dimension* (#Labels/Dim.),[2] and *number of features* (#Features). More details can be found in Appendix A. To the best of our knowledge, this serves as the most comprehensive testbed for MDC studies with the largest number of real-world benchmark data sets [5], [24], [25], [39], [43], [63].

4.1.2 Compared Approaches

In this paper, the performance of DLEM is compared with seven state-of-the-art MDC approaches, including Binary Relevance (BR) [9], [43], [70], Class Powerset (CP) [26], [58], Ensembles of Classifier Chains (ECC) [42], [44], Ensembles of Super

2. If all dimensions contain the same number of class labels, then only this number is recorded; Otherwise, the number of class labels per dimension is recorded in turn.

TABLE 2: Experimental results (mean±std. deviation) of each MDC approach in terms of *Hamming Score*. In addition, •/∘ indicates whether DLEM is significantly superior/inferior to other compared MDC approaches on each data set (pairwise t-test at 0.05 significance level).

Data Set	DLEM	BR	CP	ECC	ESC	SEEM	MDKNN	gMML
Edm	.784 ± .059	.694 ± .047 •	.688 ± .060 •	.698 ± .053 •	.704 ± .066 •	.688 ± .103 •	.740 ± .122	.714 ± .083 •
Oes97	.738 ± .026	.607 ± .033 •	.188 ± .065 •	.590 ± .034 •	.573 ± .035 •	.711 ± .023 •	.730 ± .023	.724 ± .023 •
Jura	.717 ± .062	.586 ± .069 •	.570 ± .061 •	.559 ± .065 •	.558 ± .055 •	.578 ± .063 •	.652 ± .062 •	.606 ± .072 •
Oes10	.806 ± .016	.664 ± .019 •	.179 ± .041 •	.659 ± .024 •	.633 ± .020 •	.781 ± .013 •	.791 ± .022 •	.775 ± .017 •
Enb	.935 ± .024	.716 ± .029 •	.689 ± .020 •	.681 ± .035 •	.665 ± .022 •	.770 ± .028 •	.835 ± .028 •	.742 ± .027 •
Song	.785 ± .030	.771 ± .026 •	.769 ± .025 •	.770 ± .025 •	.766 ± .027 •	.777 ± .030 •	.777 ± .027	.788 ± .027
BeLaE	.412 ± .025	.423 ± .022	.354 ± .018 •	.408 ± .022	.374 ± .020 •	.398 ± .023	.395 ± .012 •	.417 ± .020
Voice	.958 ± .009	.940 ± .010 •	.916 ± .010 •	.930 ± .008 •	.931 ± .009 •	.936 ± .012 •	.943 ± .008 •	.842 ± .009 •
Scm20d	.882 ± .003	.632 ± .006 •	N/A	.608 ± .007 •	N/A	.770 ± .005 •	.866 ± .004 •	.600 ± .007 •
Rf1	.977 ± .002	.852 ± .005 •	.813 ± .010 •	.845 ± .004 •	.794 ± .007 •	.950 ± .002 •	.981 ± .001 ∘	.730 ± .007 •
Thyroid	.968 ± .002	.961 ± .002 •	.961 ± .002 •	.961 ± .002 •	.961 ± .002 •	.966 ± .003 •	.967 ± .003	.960 ± .002 •
Pain	.978 ± .002	.948 ± .004 •	.948 ± .004 •	.948 ± .004 •	.948 ± .004 •	.960 ± .003 •	.971 ± .003 •	.948 ± .004 •
Scm1d	.893 ± .003	.725 ± .007 •	N/A	.694 ± .007 •	N/A	.824 ± .004 •	.879 ± .002 •	.697 ± .007 •
CoIL2000	.904 ± .005	.874 ± .005 •	.738 ± .006 •	.858 ± .005 •	.851 ± .008 •	.921 ± .004 ∘	.877 ± .005 •	.894 ± .004 •
TIC2000	.885 ± .004	.892 ± .008 ∘	.872 ± .008 •	.884 ± .007	.884 ± .007	.916 ± .006 ∘	.864 ± .005 •	.895 ± .007 ∘
Flickr	.735 ± .005	.715 ± .006 •	.658 ± .008 •	.693 ± .005 •	.651 ± .007 •	.734 ± .006	.735 ± .006	.779 ± .004 ∘
Disfa	.949 ± .002	.885 ± .003 •	N/A	.884 ± .003 •	.878 ± .003 •	.913 ± .003 •	.937 ± .002 •	.884 ± .003 •
Fera	.812 ± .010	.599 ± .008 •	N/A	.588 ± .007 •	N/A	.675 ± .007 •	.763 ± .006 •	.589 ± .007 •
Adult	.679 ± .004	.701 ± .004 ∘	.682 ± .005	.702 ± .005 ∘	.675 ± .006 •	.706 ± .005 ∘	.699 ± .005 ∘	.705 ± .004 ∘
Default	.663 ± .002	.665 ± .004	.660 ± .004	.666 ± .004 ∘	.666 ± .004 ∘	.668 ± .004 ∘	.654 ± .003 •	.666 ± .004 ∘

TABLE 3: Experimental results (mean±std. deviation) of each MDC approach in terms of *Exact Match*. In addition, •/∘ indicates whether DLEM is significantly superior/inferior to other compared MDC approaches on each data set (pairwise t-test at 0.05 significance level).

Data Set	DLEM	BR	CP	ECC	ESC	SEEM	MDKNN	gMML
Edm	.625 ± .082	.389 ± .093 •	.467 ± .088 •	.395 ± .106 •	.454 ± .110 •	.455 ± .153 •	.585 ± .196	.487 ± .145 •
Oes97	.063 ± .050	.030 ± .028 •	.054 ± .046	.039 ± .040 •	.036 ± .042	.036 ± .031 •	.063 ± .048	.042 ± .038
Jura	.535 ± .077	.329 ± .110 •	.326 ± .099 •	.298 ± .103 •	.298 ± .098 •	.340 ± .095 •	.473 ± .085	.368 ± .119 •
Oes10	.094 ± .045	.064 ± .035 •	.077 ± .041 •	.074 ± .044 •	.067 ± .037 •	.077 ± .041	.089 ± .053	.079 ± .040
Enb	.870 ± .048	.431 ± .058 •	.379 ± .041 •	.362 ± .069 •	.330 ± .045 •	.539 ± .057 •	.669 ± .055 •	.483 ± .053 •
Song	.478 ± .062	.449 ± .060	.442 ± .055 •	.446 ± .055 •	.438 ± .059 •	.457 ± .062 •	.455 ± .065	.484 ± .059
BeLaE	.027 ± .014	.028 ± .010	.025 ± .009	.035 ± .012	.025 ± .008	.023 ± .012	.023 ± .009	.022 ± .009
Voice	.918 ± .017	.884 ± .017 •	.841 ± .016 •	.866 ± .015 •	.867 ± .016 •	.877 ± .021 •	.889 ± .015 •	.699 ± .017 •
Scm20d	.259 ± .013	.054 ± .006 •	N/A	.073 ± .009 •	N/A	.104 ± .008 •	.231 ± .011 •	.052 ± .007 •
Rf1	.833 ± .010	.322 ± .011 •	.319 ± .025 •	.322 ± .012 •	.275 ± .012 •	.690 ± .010 •	.858 ± .009 ∘	.138 ± .011 •
Thyroid	.803 ± .014	.743 ± .014 •	.743 ± .014 •	.743 ± .014 •	.742 ± .014 •	.784 ± .017 •	.791 ± .016 •	.741 ± .015 •
Pain	.866 ± .012	.751 ± .017 •	.751 ± .017 •	.751 ± .017 •	.751 ± .017 •	.778 ± .015 •	.834 ± .018 •	.750 ± .018 •
Scm1d	.291 ± .016	.115 ± .010 •	N/A	.123 ± .013 •	N/A	.179 ± .014 •	.257 ± .014 •	.102 ± .009 •
CoIL2000	.640 ± .014	.515 ± .012 •	.273 ± .012 •	.466 ± .013 •	.468 ± .019 •	.701 ± .014 ∘	.552 ± .014 •	.576 ± .015 •
TIC2000	.688 ± .009	.698 ± .019	.645 ± .019 •	.675 ± .016 •	.675 ± .016 •	.764 ± .016 ∘	.632 ± .018 •	.706 ± .018 ∘
Flickr	.226 ± .006	.187 ± .011 •	.125 ± .016 •	.168 ± .011 •	.114 ± .014 •	.211 ± .011 •	.228 ± .013	.287 ± .009 ∘
Disfa	.622 ± .013	.378 ± .011 •	N/A	.377 ± .011 •	.374 ± .011 •	.449 ± .016 •	.579 ± .010 •	.379 ± .011 •
Fera	.481 ± .020	.199 ± .013 •	N/A	.196 ± .013 •	N/A	.244 ± .005 •	.405 ± .012 •	.196 ± .013 •
Adult	.239 ± .008	.228 ± .006 •	.282 ± .012 ∘	.251 ± .009 ∘	.269 ± .011 ∘	.256 ± .010 ∘	.260 ± .010 ∘	.230 ± .009 •
Default	.181 ± .007	.177 ± .007	.182 ± .008	.179 ± .007	.179 ± .007	.185 ± .007 ∘	.177 ± .004	.177 ± .007 •

Class classifier (ESC) [43], Stacked dEpendency Exploitation for MDC (SEEM) [26], Multi-Dimensional *k* Nearest Neighbors (MDKNN) [27], and gMML [39]. Specifically, BR independently deals with each MDC dimension by training a multi-class classifier per dimension, while CP jointly deals with all MDC dimensions by training a single multi-class classifier where all distinct class combinations in training set are regarded as new classes. ECC jointly deals with all MDC dimensions by training a chain of multi-class classifiers, one per dimension, where the subsequent classifiers on the chain treat the predictions of preceding classifiers as extra features. ESC preprocesses all MDC dimensions by grouping them into super-classes, where each super-class is regarded as a new class variable. SEEM learns a multi-class classifier for each pair of class spaces via powerset transformation in the first level and then determine the class label of unseen instance w.r.t. each class space via adaptively stacking the predictive results from the first level. MDKNN makes maximum a posteriori (MAP) inference for each pair of class spaces based on their *k*NN counting statistics and then determine the class label of unseen instance w.r.t. each class space via consulting empirical *k*NN accuracy. gMML transforms the MDC output space into a new one via one-vs-rest strategy, and then alternatingly learns regression models for each transformed label as well as a Mahalanobis metric which can make the distance between the regression outputs and ground-truth label vector closer.

For BR, CP, ECC, ESC, SEEM and MDKNN, support vector machine (SVM) is used to instantiate their multi-class base learner. Specifically, the popular LIBSVM software [10] with default parameter setting is used in this paper. For ensemble approaches ECC and ESC, 67% examples randomly selected from training set are used to generate a total of 10 base models [43], and the predictive results are combined via majority voting. For SEEM

TABLE 4: Experimental results (mean±std. deviation) of each MDC approach in terms of *Sub-Exact Match*. In addition, •/∘ indicates whether DLEM is significantly superior/inferior to other compared MDC approaches on each data set (pairwise t-test at 0.05 significance level).

Data Set	DLEM	BR	CP	ECC	ESC	SEEM	MDKNN	gMML
Edm	.942 ± .048	1.00 ± .000 ∘	.909 ± .054	1.00 ± .000 ∘	.954 ± .055	.922 ± .082	.895 ± .100	.941 ± .065
Oes97	.108 ± .070	.072 ± .051	.072 ± .057 •	.078 ± .053	.060 ± .043	.087 ± .042	.120 ± .078	.099 ± .055
Jura	.900 ± .085	.844 ± .059	.813 ± .040 •	.819 ± .052 •	.819 ± .045 •	.816 ± .066 •	.830 ± .084	.844 ± .049
Oes10	.193 ± .072	.119 ± .059 •	.107 ± .044 •	.129 ± .047 •	.117 ± .048 •	.191 ± .053	.196 ± .059	.176 ± .038
Enb	1.00 ± .000	1.00 ± .000	1.00 ± .000	1.00 ± .000	1.00 ± .000	1.00 ± .000	1.00 ± .000	1.00 ± .000
Song	.878 ± .040	.868 ± .032	.868 ± .036	.869 ± .033	.862 ± .034 •	.878 ± .051	.878 ± .042	.883 ± .041
BeLaE	.135 ± .032	.132 ± .024	.093 ± .010 •	.134 ± .016	.110 ± .012	.116 ± .020	.111 ± .020 •	.130 ± .020
Voice	.999 ± .002	.996 ± .005	.991 ± .005 •	.995 ± .005 •	.995 ± .005 •	.995 ± .004 •	.997 ± .004	.985 ± .011 •
Scm20d	.511 ± .015	.105 ± .007 •	N/A	.128 ± .011 •	N/A	.225 ± .008 •	.472 ± .021 •	.100 ± .009 •
Rf1	.988 ± .004	.655 ± .017 •	.580 ± .022 •	.637 ± .012 •	.542 ± .014 •	.932 ± .006 •	.992 ± .003 ∘	.375 ± .014 •
Thyroid	.977 ± .004	.983 ± .004 ∘	.982 ± .004 ∘	.983 ± .004 ∘	.982 ± .004 ∘	.978 ± .004	.979 ± .004	.982 ± .005 ∘
Pain	.946 ± .008	.847 ± .010 •	.847 ± .010 •	.847 ± .010 •	.847 ± .010 •	.885 ± .006 •	.921 ± .008 •	.846 ± .010 •
Scm1d	.545 ± .014	.223 ± .016 •	N/A	.212 ± .014 •	N/A	.365 ± .014 •	.502 ± .013 •	.198 ± .015 •
CoIL2000	.908 ± .010	.873 ± .016 •	.576 ± .016 •	.851 ± .013 •	.820 ± .017 •	.923 ± .005 ∘	.872 ± .011 •	.903 ± .010
TIC2000	.966 ± .005	.979 ± .004 ∘	.972 ± .005 ∘	.977 ± .005 ∘	.976 ± .005 ∘	.985 ± .004 ∘	.962 ± .003 •	.978 ± .003 ∘
Flickr	.600 ± .015	.543 ± .015 •	.426 ± .018 •	.494 ± .014 •	.414 ± .017 •	.595 ± .019	.597 ± .016	.689 ± .016 ∘
Disfa	.845 ± .005	.596 ± .011 •	N/A	.592 ± .010 •	.575 ± .010 •	.703 ± .016 •	.800 ± .011 •	.590 ± .009 •
Fera	.734 ± .017	.387 ± .012 •	N/A	.375 ± .012 •	N/A	.496 ± .012 •	.648 ± .011 •	.378 ± .013 •
Adult	.610 ± .006	.657 ± .010 ∘	.599 ± .008 •	.651 ± .010 ∘	.586 ± .011 •	.660 ± .008 ∘	.638 ± .010 ∘	.669 ± .008 ∘
Default	.586 ± .005	.590 ± .008	.579 ± .007 •	.593 ± .008 ∘	.592 ± .009	.596 ± .008 ∘	.568 ± .008 •	.593 ± .008 ∘

TABLE 5: Win/tie/loss counts of pairwise t-test (at 0.05 significance level) between DLEM and each MDC approach.

Evaluation metric	DLEM against BR	CP	ECC	ESC	SEEM	MDKNN	gMML
HS	16/2/2	14/2/0	16/2/2	15/1/1	14/2/4	13/5/2	14/2/4
EM	16/4/0	12/3/1	17/2/1	13/3/1	14/2/4	10/8/2	14/4/2
SEM	9/7/4	11/3/2	11/4/5	10/5/2	8/8/4	9/9/2	7/8/5
In Total	**41/13/6**	**37/8/3**	**44/8/8**	**38/9/4**	**36/12/12**	**32/22/6**	**35/14/11**

and MDKNN, the number of nearest neighbors is set to 10 as recommended in their respective literatures [26], [27]. For gMML, the parameters are tuned according to [39]. For the proposed DLEM approach, we use the popular RBF kernel and set the three trade-off parameters and the number of nearest neighbors considered as $\lambda = 1$, $\gamma = 10$, $\mu = 1$ and $k = 6$.

4.1.3 Evaluation Metrics

In this paper, the generalization abilities of MDC approaches are measured via a total of three evaluation metrics, i.e., *Hamming Score* (HS), *Exact Match* (EM) and *Sub-Exact Match* (SEM). Specifically, let $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leq i \leq p\}$ be the test set, where $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^\top$ is the ground-truth class vector associated with $\boldsymbol{x}_i$. For the MDC model $f : \mathcal{X} \mapsto \mathcal{Y}$ to be evaluated, let $\hat{\boldsymbol{y}}_i = f(\boldsymbol{x}_i) = [\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iq}]^\top$ be the predicted class vector for $\boldsymbol{x}_i$, then the number of dimensions which are predicted correctly corresponds to $r^{(i)} = \sum_{j=1}^{q} \mathbf{1}_{y_{ij}=\hat{y}_{ij}}$. Here, $\mathbf{1}_\pi$ returns 1 if π is true and 0 otherwise. The detailed definitions of the three metrics are given as follows:

$$\mathrm{HS}_{\mathcal{S}}(f) = \frac{1}{p}\sum_{i=1}^{p} \frac{1}{q} \cdot r^{(i)}$$

$$\mathrm{EM}_{\mathcal{S}}(f) = \frac{1}{p}\sum_{i=1}^{p} \mathbf{1}_{r^{(i)}=q}$$

$$\mathrm{SEM}_{\mathcal{S}}(f) = \frac{1}{p}\sum_{i=1}^{p} \mathbf{1}_{r^{(i)}\geq q-1}$$

For all metrics, it is obvious that their values range in $[0, 1]$ and the *larger* the values the better the performance. For all benchmark data sets, we conduct ten-fold cross-validation and record both the mean metric value and standard deviation for comparative studies.

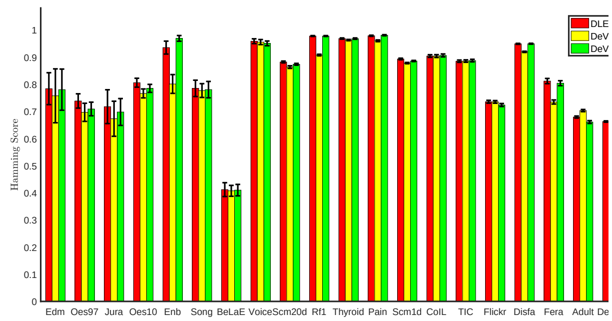
4.2 Experimental Results

Tables 2-4 report the detailed experimental results in terms of *Hamming Score*, *Exact Match* and *Sub-Exact Match* respectively. Moreover, pairwise t-test at 0.05 significance level is conducted to show whether DLEM achieves significantly superior/inferior performance against other compared MDC approaches on each data set. Accordingly, Table 5 summarizes the resulting win/tie/loss counts.

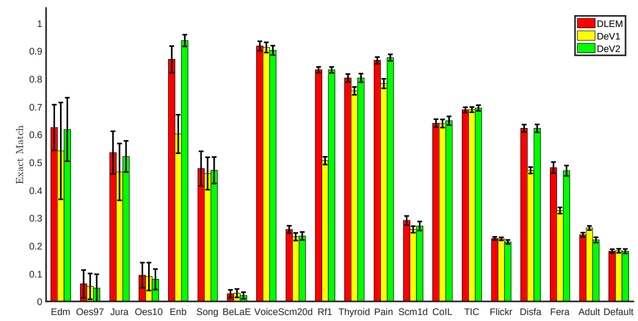
Based on the experimental results, the following observations can be made:

- Across all the 399 configurations[3] (20 data sets × 7 compared approaches × 3 metrics), DLEM achieves superior or at least comparable performance against the seven compared approaches in 349 cases.
- BR works by dealing with each dimension independently, which actually can be viewed as optimizing *Hamming Score*, while CP works by dealing with all dimension jointly via powerset transformation, which actually can be viewed as optimizing *Exact Match*. It is impressive to notice that DLEM still achieves superior performance against BR in 16 out of 20 cases in terms of *Hamming Score*, and against CP in 12 out of 16 cases in terms of *Exact Match*.

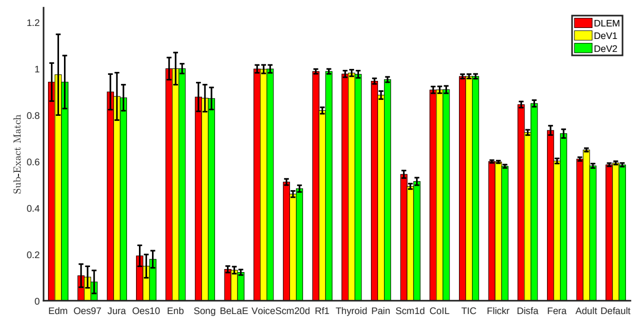
3. Due to the high computational complexity which leads to "out of memory" error for LIBSVM software, there are a total 21 configurations whose results are unavailable for some compared approaches.



(a) *hamming score*



(b) *exact match*



(c) *sub-exact match*

Fig. 1: Performance comparison between DLEM and its two variants.

- Both ECC and ESC work by explicitly considering class dependencies and utilizing ensemble strategy to account for the randomness in dependency modeling. As shown in Table 5, DLEM achieves superior performance against ECC in 44 out of 60 cases, and against ESC in 38 out of 51 cases. These results clearly validate the effectiveness of DLEM's label encoding strategy for dependency modeling.
- Both SEEM and MDKNN consider class dependencies in a two-level strategy, where pairwise (i.e., second-order) and high-order class dependencies are considered in the first and second level respectively. Compared with ECC and ESC, it is shown that more comparable and inferior cases occur though DLEM still achieves superior performance against SEEM in 36 out of 50 cases and against MDKNN in 32 out of 50 cases. Thus, it would be interesting to explore possible approaches which can integrate the two-level dependency modeling strategy into label encoding process to induce better learning models.
- gMML works by learning regression models in one-vs-rest decomposed label space. Note that DLEM achieves superior or at least comparable performance in 49 out of 60 cases. These results indicate the effectiveness of the one-vs-one decomposed label encoding strategy against the one-vs-rest decomposition strategy without modeling alignment.
- It is worth noting that against the compared approaches, most of the inferior cases (42 out of 50) for DLEM occur on data sets with nominal features including *BeLaE*, *Thyroid*, *CoIL2000*, *TIC2000*, *Adult* and *Default*. One potential reason lies in that the manifold structure identified in nominal feature space is less reliable, which impacts the quality of DLEM's enriched labeling information in the encoded label space derived via manifold structure preservation.
- It is also worth noting that DLEM achieves inferior performance against gMML over *Flickr* in terms of each metric. There are a total of 1536 features in *Flickr*, which might lead to less reliable manifold structure identified in such high dimensional feature space as dense sampling is no longer satisfied.
- To summarize, if one MDC data set is assumed to own good manifold structure in feature space, it is encouraged to try the proposed DLEM approach to induce the predictive model. Moreover, it is also worth further exploring some effective techniques (e.g., distance metric learning) to improve the quality of manifold structure in the future.

TABLE 6: Wilcoxon signed-ranks test between DLEM and its two degenerated versions in terms of each metric (at 0.05 significance level; p-values shown in the brackets).

DLEM	Evaluation Metrics		
against	HS	EM	SEM
DeV1	**win**[1.32e-03]	**win**[2.51e-02]	**win**[1.32e-03]
DeV2	**win**[3.04e-02]	**win**[2.98e-02]	**win**[1.13e-02]

4.3 Further Analysis

4.3.1 *Effectiveness of Algorithmic Design*

The performance of DLEM is also compared with its two degenerated versions to investigate the effectiveness of DLEM's algorithmic design. The two variants are denoted as DeV1 and DeV2 respectively:

- DeV1: This variant trains the multi-output regressor in Eq.(9) supervised by the ternary label matrix $\mathbf{L}$ instead of the enriched label matrix $\mathbf{F}$. In other words, DeV1 corresponds to the degenerated case without considering the enriched labeling information for model training.

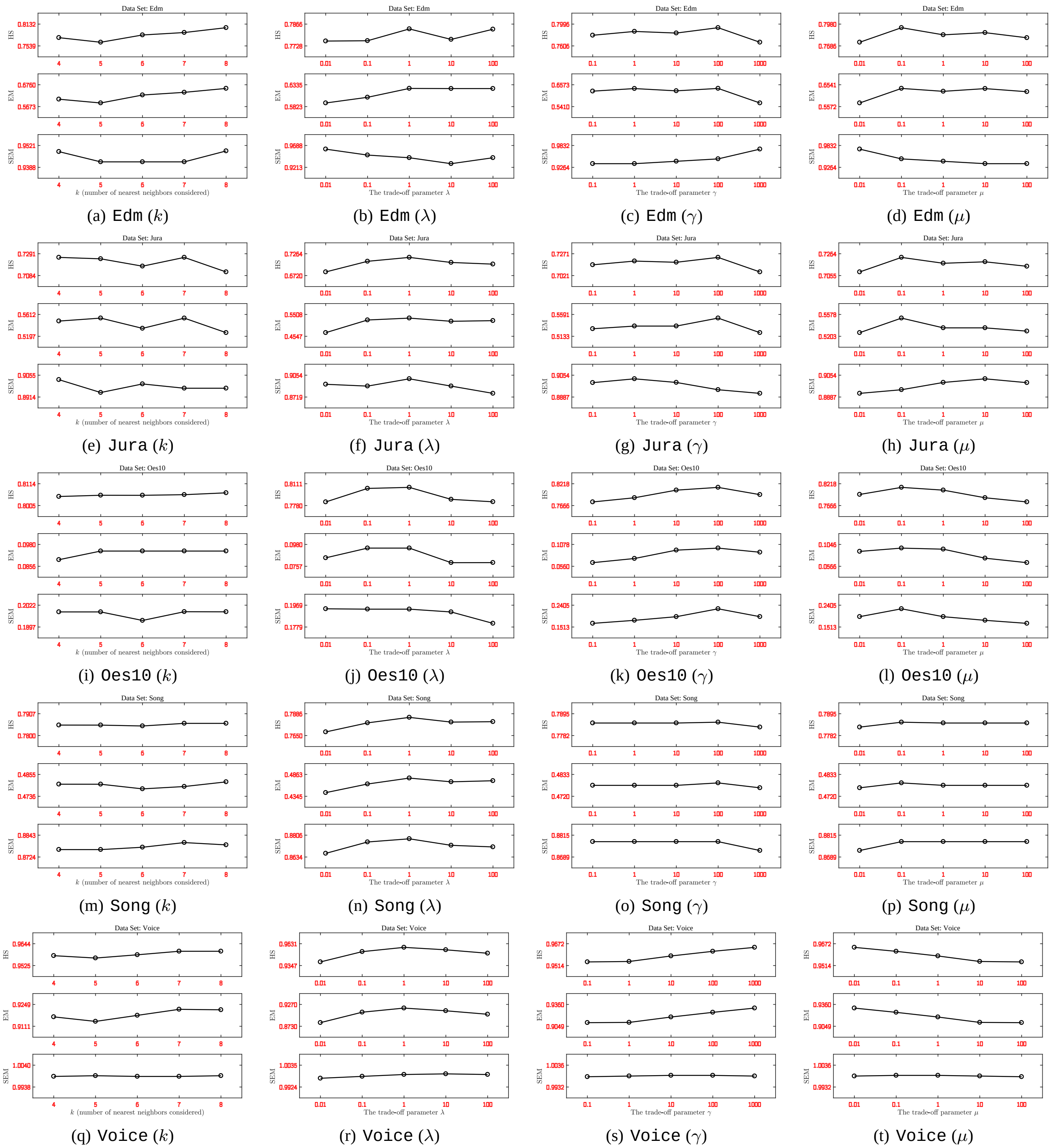


Fig. 2: Performance of DLEM varies as one of k, λ, γ and μ changes while others are fixed. For subfigures (a)(e)(i)(m)(q), k ranges in $\{4,5,6,7,8\}$ when fixing $\lambda=1$, $\gamma=10$ and $\mu=1$; For subfigures (b)(f)(j)(n)(r), λ ranges in $\{0.01,0.1,1,10,100\}$ when fixing $k=6$, $\gamma=10$ and $\mu=1$; For subfigures (c)(g)(k)(o)(s), γ ranges in $\{0.1,1,10,100,1000\}$ when fixing $k=6$, $\lambda=1$ and $\mu=1$; For subfigures (d)(h)(l)(p)(t), μ ranges in $\{0.01,0.1,1,10,100\}$ when fixing $k=6$, $\lambda=1$ and $\gamma=10$.

- DeV2: This variant employs the regressor in Eq.(8) instead of the regressor in Eq.(9) to accomplish the resulting multi-output regression task. In other words, DeV2 corresponds to the degenerated case without utilizing distance metric for model training.

Detailed experimental results are shown in Figure 1. Moreover, *Wilcoxon signed-ranks test* [15] serves as the statistical tool to test the relationship between DLEM and the two degenerated versions. Table 6 summarizes the statistical test results where the p-values for the corresponding tests are also shown in the brackets. It is shown that the performance of DLEM is better than the two variants over most data sets and achieves statistical superior performance against them in terms of each metric. These results clearly validate the effectiveness of DLEM's algorithmic design in utilizing enriched labeling information and distance metric for model training.

4.3.2 Parameter Sensitivity

As shown in Algorithm 1, there are a total of four parameters for DLEM to be tuned, i.e., the number of nearest neighbors considered k in Eq.(3) and Eq.(10), the trade-off parameters λ in Eq.(6), γ in Eq.(9), and μ in Eq.(10). Figure 2 illustrates how the performance of DLEM varies as one of these four parameters changes while others are fixed.

It is shown that the performance of DLEM is insensitive to k whose value is set to 6 in this paper. For λ, both small and large λ would lead to performance degradation of DLEM whose value is set to 1 in this paper. For γ and μ, the performance of DLEM is less sensitive to these two parameters, where the settings with moderate values $\gamma = 10$ and $\mu = 1$ serve as better choices for these trade-off parameters.

5 Conclusion

Most existing approaches solve the MDC problem in the original output space, while a novel approach named DLEM is proposed in this paper which solves the MDC problem in a transformed label space. Specifically, the original output variables are firstly encoded via one-vs-one decomposition. Then, the labeling information in the decomposed label space are enriched via manifold structure preservation identified in the feature space. Finally, a multi-output regression model with metric-aligned technique is learned for the resulting problem. The superiority of DLEM against state-of-the-art approaches is clearly validated via extensive comparative studies over the most up-to-date benchmark data sets.

It is shown that the effectiveness of subsequent steps is highly dependent on that of preceding steps for DLEM, whose coupling properties are worth further investigation in the future. Besides, it is also interesting to explore other alternatives for each step, e.g., other ways to instantiate the label encoding strategy and other strategies for labeling information enrichments.

Appendix A
Benchmark Data Sets

In this paper, a total of 20 benchmark data sets have been collected for comparative studies. Basic characteristics of all benchmark data sets have been summarized in Table 1 while further descriptions are given as follows.

- `Edm` aims at reconstructing the human operator's skill from historical examples to implement an automatic operator for an electrical discharge machining (EDM) machine [29]. The 2 class spaces correspond to two parameters (gap and flow) to be controlled during the process, and each class space includes three class labels w.r.t. possible actions: increasing the parameter, no action, and decreasing the parameter.
- `Oes97` and `Oes10` aim at estimating the relative number of full-time employees across different employment types for some specific metropolitan areas according to the occupational employment survey (OES) in years 1997 and 2010, respectively [54]. The 16 class spaces correspond to sixteen different employment types, and each class space includes three class labels w.r.t. relative representation of number of employees: small quantity, medium quantity, and large quantity.
- `Jura` aims at predicting land uses and rock types for some locations in Swiss Jura according to the measurements of concentrations of seven heavy metals [19]. The first class space corresponds to land uses with four possible types: forest, pasture, meadow, tillage, and the second class space corresponds to rock with five possible types: Argovian, Kimmeridgian, Sequanian, Portlandian, Quaternary.
- `Enb` aims at predicting some building parameters of energy buildings according to some other building parameters [57]. The first class space corresponds to overall height with two relative representations and the second class space corresponds to glazing area with four relative representations.
- `Song` aims at categorizing Chinese songs from three dimensions, including emotion, genre and scenario [25]. Each of the 3 class spaces includes three possible categories: happy, sad, cathartic for emotion, folk, Internet pop, pop for genre, and walk, wedding, nightclub for scenario.
- `BeLaE` aims at predicting students' answers to five questions in a questionnaire based on their age, sex and answers to other 43 questions [13], [40]. Each question is on the importance of certain properties of their future jobs, and the answer has a grade from '1' (completely unimportant) to '5' (very important).
- `Voice` aims at predicting the relative mean frequency and speaker's gender of a piece of human voice [25]. The first class space corresponds to mean frequency with four possible class labels: less than 120Hz, between 120Hz and 160Hz, between 160Hz and 200Hz, greater than 200Hz, and the second class space corresponds to gender with two possible class labels: male and female.
- `Scm20d` and `Scm1d` aim at predicting the relative mean price of products for 20 days in the future and for the next day, respectively [54]. Each class space corresponds to the relative mean price of one product with four possible grades from '1' (low) to '4' (high).
- `Rf1` aims at predicting the river flows for 48h in the future at eight specific locations in the Mississippi river network [54]. Each class space corresponds to the relative representation of river flows for one observation site with three or four grades.
- `Thyroid` aims at estimating the conditions of thyroid diagnoses according to physical test results of patients [16]. The 7 class spaces correspond to diagnosed conditions from seven aspects, including hyperthyroid, hypothyroid, binding protein, general health, replacement therapy, antithyroid treatment, and miscellaneous.
- `Pain` aims at estimating the facial action unit intensity of patients who are suffering from chronic shoulder pain while performing a range of arm motion exercises [38]. The 10 class spaces correspond to ten different facial action units with six intensity levels ranging from 0 (no pain) to 5 (strong pain). Some intensity levels for some action units are merged to alleviate the imbalanced class distribution in the original data set.
- `CoIL2000` aims at categorizing customers of an insurance company from different dimensions according to their product usage data and socio-demographic data

derived from zip area codes [61]. The 5 class spaces correspond to average age, customer main type, Roman Catholic, contribution private third party insurance, and number of mobile home policies, respectively. `TIC2000` is a variant of CoIL2000 where the two target variables customer main type and Roman Catholic are used as input attributes.

- `Flickr` aims at categorizing pictures in mirflickr25k [23] from different dimensions. We re-annotated all the pictures according to the MDC definition and just pick out part of them [25]. The 5 class spaces correspond to sky, people, night, plant, and indoor, respectively.
- `Disfa` aims at estimating the facial action unit intensity of young adults who are viewing 4-minute video clips [41]. The 12 class spaces correspond to twelve different facial action units with six intensity levels ranging from 0 (not present) to 5 (maximum intensity). Some intensity levels for some action units are merged to alleviate the imbalanced class distribution in the original data set.
- `Fera` aims at dealing with a similar task as Disfa. The data set is provided in FG 2015 Facial Expression Recognition and Analysis challenge (FERA 2015) [59] and only five different facial action units are used in output space.
- `Adult` aims at categorizing people from different dimensions based on their personal information [16]. This data set is also known as Census Income Data Set in UCI machine learning repository. The 4 class spaces correspond to work class, marital status, race, and sex, respectively.
- `Default` aims at categorizing credit card clients from different dimensions based on their personal information [66]. This data set is known as default of credit card clients Data Set in UCI machine learning repository [16]. The 4 class spaces correspond to gender, education, marital status, and default payment next month, respectively.

All these data sets are publicly available at http://palm.seu.edu.cn/zhangml/Resources.htm#MDC_data. More details about each data set's descriptions, original sources, references, data format in Matlab and preprocessing notes can also be found in the shared data repository.

References

[1] A. H. Al Muktadir, T. Miyazawa, P. Martinez-Julia, H. Harai, and V. P. Kafle, "Multi-target classification based automatic virtual resource allocation scheme," *IEICE Transactions on Information and Systems*, vol. 102, no. 5, pp. 898–909, 2019.

[2] J. Arias, J. A. Gamez, T. D. Nielsen, and J. M. Puerta, "A scalable pairwise class interaction framework for multidimensional classification," *International Journal of Approximate Reasoning*, vol. 68, pp. 194–210, 2016.

[3] M. Benjumeda, C. Bielza, and P. Larrañaga, "Tractability of most probable explanations in multidimensional Bayesian network classifiers," *International Journal of Approximate Reasoning*, vol. 93, pp. 74–87, 2018.

[4] R. Bhatia, *Positive Definite Matrices*. Princeton, NJ, USA: Princeton University Press, 2007.

[5] C. Bielza, G. Li, and P. Larrañaga, "Multi-dimensional classification with Bayesian networks," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.

[6] J. H. Bolt and L. C. van der Gaag, "Balanced sensitivity functions for tuning multi-dimensional Bayesian network classifiers," *International Journal of Approximate Reasoning*, vol. 80, pp. 361–376, 2017.

[7] H. Borchani, C. Bielza, C. Toro, and P. Larrañaga, "Predicting human immunodeficiency virus inhibitors using multi-dimensional Bayesian network classifiers," *Artificial Intelligence in Medicine*, vol. 57, no. 3, pp. 219–229, 2013.

[8] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Mining and Knowledge Discovery*, vol. 5, pp. 216–233, 2015.

[9] 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.

[10] 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, 2011, Article 27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] C. Chen, H. Wang, W. Liu, X. Zhao, T. Hu, and G. Chen, "Two-stage label embedding via neural factorization machine for multi-label classification," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 3304–3311.

[12] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Advances in Neural Information Processing Systems 25*, Lake Tahoe, NV, USA, 2012, pp. 1538–1546.

[13] W. Cheng, K. Dembczyński, and E. Hüllermeier, "Graded multilabel classification: The ordinal case," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 223–230.

[14] P. R. de Waal and L. C. van der Gaag, "Inference and learning in multi-dimensional Bayesian network classifiers," in *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Hammamet, Tunisia, 2007, pp. 501–511.

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

[16] D. Dua and C. Graff, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml.

[17] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, 2015, Article 52.

[18] S. Gil-Begue, C. Bielza, and P. Larrañaga, "Multi-dimensional Bayesian network classifiers: A survey," *Artificial Intelligence Review*, vol. 54, pp. 519–559, 2021.

[19] P. Goovaerts, *Geostatistics for natural resources evaluation*. New York, NY, USA: Oxford University Press, 1997.

[20] D. Hsu, S. M. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing," in *Advances in Neural Information Processing Systems 22*, Vancouver, Canada, 2009, pp. 772–780.

[21] 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.

[22] S.-J. Huang, G.-X. Li, W.-Y. Huang, and S.-Y. Li, "Incremental multi-label learning with active queries," *Journal Of Computer Science and Technology*, vol. 35, no. 2, pp. 234–246, 2020.

[23] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, Vancouver, Canada, 2008, pp. 39–43, http://press.liacs.nl/mirflickr/mirdownload.html.

[24] B.-B. Jia and M.-L. Zhang, "Maximum margin multi-dimensional classification," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, USA, 2020, pp. 4312–4319.

[25] B.-B. Jia and M.-L. Zhang, "Multi-dimensional classification via *k*NN feature augmentation," *Pattern Recognition*, vol. 106, 2020, Article 107423.

[26] B.-B. Jia and M.-L. Zhang, "Multi-dimensional classification via stacked dependency exploitation," *Science China Information Sciences*, vol. 63, no. 12, 2020, Article 222102.

[27] B.-B. Jia and M.-L. Zhang, "MD-KNN: An instance-based approach for multi-dimensional classification," in *Proceedings of the 25th International Conference on Pattern Recognition*, Milan, Italy, 2021, pp. 126–133.

[28] L. Jia, Z. Zhang, L. Wang, W. Jiang, and M. Zhao, "Adaptive neighborhood propagation by joint L2,1-norm regularized sparse coding for representation and classifications," in *Proceedings of the 16th International Conference on Data Mining*, Barcelona, Spain, 2016, pp. 201–210.

[29] A. Karalič and I. Bratko, "First order regression," *Machine Learning*, vol. 26, pp. 147–176, 1997.

[30] S.-Y. Li, S.-J. Huang, and S. Chen, "Crowdsourcing aggregation with deep Bayesian learning," *Science China Information Sciences*, vol. 64, no. 3, 2021, Article 130104.

[31] X. Li and Y. Guo, "Multi-label classification with feature-aware nonlinear label space transformation," in *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 3635–3642.

[32] Y. Li, M. Yang, Z. Xu, and Z. Zhang, "Learning with feature network and label network simultaneously," in *Proceedings of the 31st AAAI*

Conference on Artificial Intelligence, San Francisco, CA, USA, 2017, pp. 1410–1416.

[33] Z. Lin, G. Ding, J. Han, and L. Shao, "End-to-end feature-aware label space encoding for multilabel classification with many classes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2472–2487, 2018.

[34] Z. Lin, G. Ding, M. Hu, and J. Wang, "Multi-label classification via feature-aware implicit label space encoding," in *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 325–333.

[35] M. Liu, D. Zhang, S. Chen, and H. Xue, "Joint binary classifier learning for ECOC-based multi-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2335–2341, 2016.

[36] W. Liu and I. W. Tsang, "Large margin metric learning for multi-label prediction," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Austin, TX, USA, 2015, pp. 2800–2806.

[37] W. Liu, D. Xu, I. W. Tsang, and W. Zhang, "Metric learning for multi-output tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 408–422, 2019.

[38] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. A. Matthews, "Painful data: The UNBC-McMaster shoulder pain expression archive database," in *Proceedings of the 9th IEEE International Conference on Automatic Face and Gesture Recognition*, Santa Barbara, CA, USA, 2011, pp. 57–64.

[39] Z. Ma and S. Chen, "Multi-dimensional classification via a metric approach," *Neurocomputing*, vol. 275, pp. 1121–1131, 2018.

[40] Z. Ma and S. Chen, "A convex formulation for multiple ordinal output classification," *Pattern Recognition*, vol. 86, pp. 73–84, 2019.

[41] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, "DISFA: A spontaneous facial action intensity database," *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 151–160, 2013.

[42] J. Read, L. Martino, and D. Luengo, "Efficient monte carlo methods for multi-dimensional learning with classifier chains," *Pattern Recognition*, vol. 47, no. 3, pp. 1535–1546, 2014.

[43] J. Read, C. Bielza, and P. Larrañaga, "Multi-dimensional classification with super-classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1720–1733, 2014.

[44] 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.

[45] J. D. Rodríguez, A. Pérez, D. Arteta, D. Tejedor, and J. A. Lozano, "Using multidimensional Bayesian network classifiers to assist the treatment of multiple sclerosis," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1705–1715, 2012.

[46] J. D. Rodríguez and J. A. Lozano, "Multi-objective learning of multi-dimensional Bayesian classifiers," in *Proceedings of the 8th International Conference on Hybrid Intelligent Systems*, Barcelona, Spain, 2008, pp. 501–506.

[47] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[48] B. Schölkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press, 2002.

[49] F. Serafino, G. Pio, M. Ceci, and D. Malerba, "Hierarchical multi-dimensional classification of web documents with multiwebclass," in *Proceedings of the 18th International Conference on Discovery Science*, Banff, AB, Canada, 2015, pp. 236–250.

[50] H. Shatkay, F. Pan, A. Rzhetsky, and W. J. Wilbur, "Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users," *Bioinformatics*, vol. 24, no. 18, pp. 2086–2093, 2008.

[51] X. Shen, W. Liu, I. W. Tsang, Q.-S. Sun, and Y.-S. Ong, "Compact multi-label learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 4066–4073.

[52] X. Shen, W. Liu, I. W. Tsang, Q.-S. Sun, and Y.-S. Ong, "Multilabel prediction via cross-view search," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4324–4338, 2018.

[53] W. Siblini, P. Kuntz, and F. Meyer, "A review on dimensionality reduction for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 839–857, 2021.

[54] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. P. Vlahavas, "Multi-target regression via input space expansion: Treating targets as inputs," *Machine Learning*, vol. 104, pp. 55–98, 2016.

[55] Y.-P. Sun and M.-L. Zhang, "Compositional metric learning for multi-label classification," *Frontiers of Computer Science*, vol. 15, no. 5, 2021, Article 155320.

[56] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[57] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and Buildings*, vol. 49, pp. 560–567, 2012.

[58] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[59] M. F. Valstar, T. R. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn, "FERA 2015 - second facial expression recognition and analysis challenge," in *Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, Ljubljana, Slovenia, 2015, pp. 1–8.

[60] L. C. van der Gaag and P. R. de Waal, "Multi-dimensional Bayesian network classifiers," in *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models*, Prague, Czech Republic, 2006, pp. 107–114.

[61] P. van der Putten and M. van Someren, "CoIL challenge 2000: The insurance company case," Leiden Institute of Advanced Computer Science, Universiteit van Leiden, Technical Report 2000-09, 2000, http://kdd.ics.uci.edu/databases/tic/tic.data.html.

[62] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2007.

[63] H. Wang, C. Chen, W. Liu, K. Chen, T. Hu, and G. Chen, "Incorporating label embedding and feature augmentation for multi-dimensional classification," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, USA, 2020, pp. 6178–6185.

[64] H. Wang, W. Liu, Y. Zhao, T. Hu, K. Chen, and G. Chen, "Learning from multi-dimensional partial labels," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp. 2943–2949.

[65] D. Xu, Y. Shi, I. W. Tsang, Y. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 37, no. 7, pp. 2409 – 2429, 2020.

[66] I. Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[67] P. H. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016, pp. 2464–2471.

[68] J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza, and P. Larrañaga, "Bayesian chain classifiers for multidimensional classification," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, Spain, 2011, pp. 2192–2197.

[69] H. Zhang, Z. Zhang, M. Zhao, Q. Ye, M. Zhang, and M. Wang, "Robust triple-matrix-recovery-based auto-weighted label propagation for classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4538–4552, 2020.

[70] M.-L. Zhang, Y.-K. Li, X.-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.

[71] M.-L. Zhang, Q.-W. Zhang, J.-P. Fang, Y.-K. Li, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 2057–2070, 2021.

[72] 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.

[73] Y. Zhang and J. Schneider, "Maximum margin output coding," in *Proceedings of the 29th International Coference on Machine Learning*, Edinburgh, Scotland, UK, 2012, pp. 379–386.

[74] Z. Zhang, F. Li, L. Jia, J. Qin, L. Zhang, and S. Yan, "Robust adaptive embedded label propagation with weight learning for inductive classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3388–3403, 2018.

[75] W.-J. Zhou, Y. Yu, and M.-L. Zhang, "Binary linear compression for multi-label classification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3546–3552.

[76] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.

[77] M. Zhu, S. Liu, and J. Jiang, “A hybrid method for learning multi-dimensional Bayesian network classifiers based on an optimization model,” *Applied Intelligence*, vol. 44, no. 1, pp. 123–148, 2016.