

Maximum Margin Multi-Dimensional Classification

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

Abstract—Multi-dimensional classification (MDC) assumes *heterogeneous* class spaces for each example, where class variables from different class spaces characterize semantics of the example along different dimensions. The heterogeneity of class spaces leads to incomparability of the modeling outputs from different class spaces which is the major difficulty in designing MDC approaches. In this paper, we make a first attempt towards adapting maximum margin techniques for MDC problem and a novel approach named M^3MDC is proposed. Specifically, M^3MDC maximizes the margins between each pair of class labels w.r.t. individual class variable while models relationship across class variables (as well as class labels within individual class variable) via covariance regularization. The resulting formulation admits convex objective function with nonlinear constraints, which can be solved via alternating optimization with quadratic programming (QP) or closed-form solution in either alternating step. Comparative studies on the most comprehensive real-world MDC data sets to date are conducted and it is shown that M^3MDC achieves highly competitive performance against state-of-the-art MDC approaches.

Index Terms—machine learning, multi-dimensional classification, maximum margin, class dependencies.

I. INTRODUCTION

IN traditional supervised learning, one popular learning task is to train classification models supervised by one class variable, e.g., multi-class classification. However, in many real-world applications, the simplifying assumption that each example is associated with only one class variable does not fit well. For example, news websites usually need to simultaneously classify a news document from the `topic` dimension (with possible classes *Sci&Tech*, *politics*, *social*, *sports*, etc.), from the `mood` dimension (with possible classes *good news*, *neutral news*, *bad news*), and from the `zone` dimension (with possible classes *domestic*, *intra-/inter-continental*, etc.). Actually, the need to characterize the semantics of objects from different dimensions arises in diverse application scenarios such as text classification [41], [40], computer vision [13], [14], [48], bioinformatics [38], [7], [6], [32], [16], software testing [39], resource allocation [34] etc., where the resulting learning problem can be naturally formalized under the multi-dimensional classification (MDC) framework [35], [30], [23]. Specifically, each MDC example is represented by a single instance while associated with multiple class variables simultaneously. Here, each class variable corresponds to one heterogeneous class space which characterizes the object's semantics from one specific dimension.

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ denote the input (feature) space and $\mathcal{Y} = C_1 \times C_2 \times \dots \times C_q$ denote the output space which corresponds to the Cartesian product of q heterogeneous class spaces. Here, each heterogeneous class space C_j ($1 \leq j \leq q$) consists of K_j possible class labels, i.e., $C_j = \{c_1^j, c_2^j, \dots, c_{K_j}^j\}$. Given the MDC training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid 1 \leq i \leq N\}$, for each MDC example $(\mathbf{x}_i, \mathbf{y}_i)$, $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^\top \in \mathcal{X}$ is a d -dimensional feature vector and $\mathbf{y}_i = [y_{i1}, y_{i2}, \dots, y_{iq}]^\top \in \mathcal{Y}$ is the class vector associated with \mathbf{x}_i where each component $y_{ij} \in C_j$ corresponds to the relevant class label for \mathbf{x}_i in C_j . The learning task of multi-dimensional classification is to train a predictive model $f : \mathcal{X} \mapsto \mathcal{Y}$ over \mathcal{D} which can predict a proper class vector $f(\mathbf{x}_*) \in \mathcal{Y}$ for unseen instance \mathbf{x}_* .

To learn predictive models from MDC training examples, one intuitive strategy is to train a multi-class classification model for each class space independently. However, dependencies among class spaces are completely ignored by the independent modeling which would impact the generalization performance of resulting MDC predictive model. Another intuitive strategy is to treat every distinct combination of class variables appearing in the training set as a new class, and then induce a single multi-class classification model in the transformed class space. However, the resulting multi-class classification model is difficult to be learned due to the huge number of possible classes in the transformed class space and is incapable of predicting combinations of class variables absent in the training set. Generally, modeling class dependencies in appropriate ways is one of the key challenges in designing MDC approaches. Therefore, existing MDC approaches aim to model class dependencies in different ways such as specifying chaining order over class spaces [51], [36], assuming directed acyclic graph (DAG) structure over class spaces [2], [4], [1], and grouping class spaces into super-classes [35].

Maximum margin is one of the most popular and powerful machine learning techniques which has been successfully adapted to tackle various learning tasks [45], [20], [49], [19], [8]. As per the intrinsic characteristics of MDC problem, modeling outputs from the heterogeneous class spaces are not directly comparable. On the other hand, dependencies among class spaces are expected to be exploited to help improve the generalization performance of classification model. In light of the above modeling challenges, a first attempt towards adapting maximum margin techniques for solving MDC problem is investigated in this paper. Accordingly, a novel MDC approach named M^3MDC , i.e., *Maximum Margin for Multi-Dimensional Classification*, is proposed. To account for the incompatibility of modeling outputs from heterogeneous class spaces, M^3MDC chooses to maximize the classification margin on individual class spaces via one-vs-one (OvO) decomposition. Furthermore, dependencies among class spaces are modeled by

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

M³MDC via covariance regularization. The derived M³MDC formulation admits convex objective function with nonlinear constraints, which can be solved via alternating optimization with QP or closed-form solution in either alternating step. Comparative studies are conducted over a total of fifteen real-world MDC data sets, which serve as the most comprehensive basis to date for MDC performance evaluation [2], [35], [30], [23], [44]. Experimental results show that M³MDC achieves highly competitive performance against state-of-the-art MDC approaches.

We organize the remainder of this paper as follows. Section II presents the *technical details* of M³MDC. Section III briefly discusses *related works* on MDC. Section IV reports the *experimental results* of comparative studies. Finally, we *conclude* this paper in Section V. This paper is an extension of our preliminary work [22]. The main differences include: (1) The introduction and related work parts have been updated to reflect state-of-the-art research progress on MDC; (2) The derivation of the proposed approach has been revised to improve readability; (3) A kernelized version of the original approach has been proposed; (4) The comparative studies have been extended by adding five newly-collected benchmark datasets, one recently proposed compared approach [30], analyses on the effects of initialization, and enriching the correlation analyses.

II. THE MAXIMUM MARGIN MDC APPROACH

To maximize the margin between a pair of modeling outputs, the essential prerequisite is that these modeling outputs are comparable to each other. However, due to intrinsic characteristics of MDC that class spaces are *heterogeneous*, the modeling outputs of class labels from different class spaces are not directly comparable. In this section we present technical details of M³MDC, which considers the margins between each pair of class labels on individual class spaces while models class dependencies via covariance regularization.

To obtain margins between each pair of class labels, M³MDC tackles the multi-class classification problem w.r.t. each class space via OvO decomposition. Across all class spaces, there will be a total of $m = \sum_{j=1}^q \binom{K_j}{2}$ pairs of class labels under the OvO decomposition. Without loss of generality, for the i th pair of class labels l_+^i and l_-^i , let $\mathcal{D}^i = \{(\mathbf{x}_j^i, y_j^i) \mid 1 \leq j \leq n_i\}$ denote the corresponding OvO decomposition training set. Here, we have $\mathbf{x}_j^i \in \mathcal{X}$, $y_j^i = +1$ (or -1) when l_+^i (or l_-^i) is relevant, and n_i is the number of examples in MDC training set \mathcal{D} for which either l_+^i or l_-^i is relevant. Assuming that the examples in \mathcal{D}^i can be separated perfectly by hyperplane (\mathbf{w}_i, b_i) , we can define the margin of (\mathbf{w}_i, b_i) as $2/\|\mathbf{w}_i\|$ by appropriately normalizing (\mathbf{w}_i, b_i) [10], where $\|\cdot\|$ returns the norm of vectors. Then, the maximum margin hyperplane for \mathcal{D}^i can be obtained by maximizing $2/\|\mathbf{w}_i\|$ or equivalently minimizing $\|\mathbf{w}_i\|^2/2$. Considering the more general case where no hyperplane is capable of correctly classifying all training examples in \mathcal{D}^i , we can model the empirical risk by introducing slack variables $\xi^i = [\xi_1^i, \dots, \xi_{n_i}^i]$. Considering all pairs of class labels, let $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$, $\mathbf{b} = (b_1, \dots, b_m)^\top$ and

$\xi = [\xi^1, \xi^2, \dots, \xi^m]^\top \in \mathbb{R}^{\sum_{i=1}^m n_i \times 1}$, the maximum margin formulation for MDC can be given as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \xi} \quad & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ \text{s.t.} \quad & y_j^i (\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) \geq 1 - \xi_j^i, \\ & \xi_j^i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_i \end{aligned} \quad (1)$$

where λ_1 is the trade-off parameter for model complexity term, $\langle \cdot, \cdot \rangle$ computes two vectors' inner product, and $\text{tr}(\cdot)$ returns the trace of a square matrix.

Obviously, the above formulation only deals with each pair of class labels independently while ignores potential dependencies among class spaces. Similar to the covariance regularization strategy for dependency modeling [56], [27], [25], [31], we introduce the column covariance matrix of \mathbf{W} (i.e. $\mathbf{C} \in \mathbb{R}^{m \times m}$) to model the relationships among all \mathbf{w}_i s in \mathbf{W} . Thereafter, the optimization problem in Eq.(1) is transformed to:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \xi, \mathbf{C}} \quad & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) \\ \text{s.t.} \quad & \mathbf{C} \succeq 0, \text{tr}(\mathbf{C}) \leq 1, \\ & y_j^i (\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) > 1 - \xi_j^i, \\ & \xi_j^i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_i \end{aligned} \quad (2)$$

where λ_2 is the trade-off parameter for covariance regularization term. In addition, $\mathbf{C} \succeq 0$ ensures that the covariance matrix is positive semi-definite and $\text{tr}(\mathbf{C}) \leq 1$ further penalizes its complexity. Here, minimizing the covariance regularization with above constraints can be regarded as maximum a posteriori (logarithm) estimation¹ of \mathbf{W} with assuming the matrix-variate normal distribution¹ over it, i.e., $\mathbf{W} \sim \mathcal{MN}_{d \times m}(\mathbf{W} \mid \mathbf{0}_{d \times m}, \mathbf{I}_d \otimes \mathbf{C})$, where $\mathbf{0}_{d \times m}$ is a $d \times m$ zero matrix and \mathbf{I}_d is a $d \times d$ identity matrix which means the features are assumed uncorrelated with each other.

For the objective function in Eq.(2), it is obvious that the first two terms are convex w.r.t. \mathbf{W} and \mathbf{b} . For the third term, it can be reformulated as the summation of d items, i.e., $\text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) = \sum_{i=1}^d \mathbf{W}_i \mathbf{C}^{-1} \mathbf{W}_i^\top$, where \mathbf{W}_i denotes the i th row of \mathbf{W} . Note that $\mathbf{W}_i \mathbf{C}^{-1} \mathbf{W}_i^\top$ is convex w.r.t. \mathbf{W}_i and \mathbf{C} when \mathbf{C} is positive semi-definite, and the convexity can be preserved after summation operation which results in the convexity of the third term w.r.t. \mathbf{W} and \mathbf{C} [56]. Therefore, the objective function in Eq.(2) is convex w.r.t. \mathbf{W} , \mathbf{b} and \mathbf{C} , respectively. In this paper, an alternating procedure is derived to accomplish the optimization task [54]. Specifically, the two sets of parameters $\{\mathbf{W}, \mathbf{b}\}$ and $\{\mathbf{C}\}$ are optimized alternately until convergence.

¹Let $\mathcal{MN}_{d \times m}(\mathbf{X} \mid \mathbf{M}, \mathbf{U} \otimes \mathbf{V})$ be a matrix-variate normal distribution with mean $\mathbf{M} \in \mathbb{R}^{d \times m}$, row covariance matrix $\mathbf{U} \in \mathbb{R}^{d \times d}$ and column covariance matrix $\mathbf{V} \in \mathbb{R}^{m \times m}$. The corresponding probability density function is defined as $p(\mathbf{X} \mid \mathbf{M}, \mathbf{U} \otimes \mathbf{V}) = \frac{\exp(-\frac{1}{2} \text{tr}(\mathbf{U}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{V}^{-1}(\mathbf{X}-\mathbf{M})^\top))}{(2\pi)^{md/2} |\mathbf{U}|^{m/2} |\mathbf{V}|^{d/2}}$.

Optimizing w.r.t. \mathbf{W} and \mathbf{b} when \mathbf{C} is fixed. When we fix \mathbf{C} , the optimization problem in Eq.(2) can be reformulated as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \xi} \quad & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) \\ \text{s.t.} \quad & y_j^i (\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) > 1 - \xi_j^i, \\ & \xi_j^i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_i \end{aligned} \quad (3)$$

For the above optimization problem, it is easy to obtain its Lagrange function as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{W}, \mathbf{b}, \xi, \alpha, \beta) = & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i [y_j^i (\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) - 1 + \xi_j^i] \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \beta_j^i \xi_j^i \end{aligned} \quad (4)$$

where $\alpha = (\alpha_1^1, \dots, \alpha_{n_1}^1, \dots, \alpha_1^m, \dots, \alpha_{n_m}^m)^\top \in \mathbb{R}^{\sum_{j=1}^m n_j \times 1}$, $\beta = (\beta_1^1, \dots, \beta_{n_1}^1, \dots, \beta_1^m, \dots, \beta_{n_m}^m)^\top \in \mathbb{R}^{\sum_{j=1}^m n_j \times 1}$, and $\alpha_j^i, \beta_j^i \geq 0$. Then, by setting the gradients of \mathcal{L} w.r.t. \mathbf{W} , \mathbf{b} , and ξ_j^i to 0, the following conditions can be obtained respectively [55]:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = 0 \Rightarrow \mathbf{W} = \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i y_j^i \mathbf{x}_j^i \mathbf{e}_i^\top \mathbf{C} (\lambda_1 \mathbf{C} + \lambda_2 \mathbf{I}_m)^{-1} \quad (5)$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = 0 \Rightarrow \sum_{j=1}^{n_i} \alpha_j^i y_j^i = 0, \quad (1 \leq i \leq m) \quad (6)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_j^i} = 0 \Rightarrow \alpha_j^i + \beta_j^i = 1 \quad (7)$$

where \mathbf{e}_i is the i th column of identity matrix \mathbf{I}_m . Plugging Eq.(6) and Eq.(7) into Eq.(4), the terms related to b_i and ξ_j^i will be eliminated respectively, then Eq.(4) can be simplified as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{W}, \alpha) = & \frac{1}{2} \text{tr}(\mathbf{W}(\lambda_1 \mathbf{I}_m + \lambda_2 \mathbf{C}^{-1})\mathbf{W}^\top) \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i y_j^i \langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i \end{aligned} \quad (8)$$

Let $\mathbf{M} = (\lambda_1 \mathbf{C} + \lambda_2 \mathbf{I}_m)^{-\top} \mathbf{C}^\top$ and $M_{i_1 i_2} = \mathbf{e}_{i_1}^\top \mathbf{M} \mathbf{e}_{i_2}$. Plugging Eq.(5) into the first term (denoted by \mathcal{L}_1) of Eq.(8), we have:

$$\mathcal{L}_1 = \frac{1}{2} \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2} \rangle$$

Plugging Eq.(5) into the second term (denoted by \mathcal{L}_2) of Eq.(8) and note that $\mathbf{w}_i = \mathbf{W} \mathbf{e}_i$, we have:

$$\mathcal{L}_2 = - \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2} \rangle$$

Plugging \mathcal{L}_1 and \mathcal{L}_2 into Eq.(8), we can obtain the dual function, i.e., $\Gamma(\alpha) = \min_{\mathbf{W}, \mathbf{b}} \mathcal{L}(\mathbf{W}, \mathbf{b})$, as follows:

$$\begin{aligned} \Gamma(\alpha) = & - \frac{1}{2} \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2} \rangle \\ & + \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i \end{aligned} \quad (9)$$

Then, the dual problem $\max_{\alpha} \Gamma(\alpha)$ can be equivalently formulated as $\min_{\alpha} -\Gamma(\alpha)$:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2} \rangle \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i \\ \text{s.t.} \quad & \sum_{j=1}^{n_i} \alpha_j^i y_j^i = 0 \quad (1 \leq i \leq m), \quad 0 \leq \alpha_j^i \leq 1 \end{aligned} \quad (10)$$

Obviously, the above problem is a QP problem with m equality constraints which can be solved by any off-the-shelf QP solver. However, the number of variables α_j^i is usually too large making this QP problem difficult to be solved efficiently. Specifically, the number of variables equals the total number of training examples in m OvO binary training sets, i.e., $\sum_{j=1}^m n_j = N \cdot \sum_{j=1}^q (K_j - 1)$. Here, we further decompose the dual QP problem into m sub-QP problems each with one equality constraint as follows:

$$\begin{aligned} \min_{\alpha^i} \quad & \frac{1}{2} \sum_{j_1=1}^{n_i} \sum_{j_2=1}^{n_i} \alpha_{j_1}^i \alpha_{j_2}^i y_{j_1}^i y_{j_2}^i M_{ii} \langle \mathbf{x}_{j_1}^i, \mathbf{x}_{j_2}^i \rangle \\ & - \sum_{j=1}^{n_i} (1 - S_j^i) \alpha_j^i \\ \text{s.t.} \quad & \sum_{j=1}^{n_i} \alpha_j^i y_j^i = 0, \quad 0 \leq \alpha_j^i \leq 1 \end{aligned} \quad (11)$$

where $1 \leq i \leq m$, $\alpha^i = (\alpha_1^i, \dots, \alpha_{n_i}^i)^\top \in \mathbb{R}^{n_i \times 1}$, and

$$S_j^i = y_j^i \sum_{i_1 \neq i} \frac{1}{2} (M_{ii_1} + M_{i_1 i}) \sum_{j_1=1}^{n_{i_1}} \alpha_{j_1}^{i_1} y_{j_1}^{i_1} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_1}^{i_1} \rangle \quad (12)$$

To solve the optimization problem in Eq.(10), we can initialize $\alpha = \mathbf{0}$ and then repeatedly solve the m sub-QP problems in Eq. (11) until all α_j^i s meet Karush-Kuhn-Tucker (KKT) conditions.

To validate KKT conditions, the decision value of each \mathbf{x}_j^i , i.e., $\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i$, needs to be obtained. For $\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle$, its value can be obtained by plugging α into Eq.(5), i.e.,

$$\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle = \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \alpha_{j_1}^{i_1} y_{j_1}^{i_1} M_{i i_1} \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_j^i \rangle \quad (13)$$

For b_i , however, the situation is somewhat complicated. When there are α_j^i s in $(0, 1)$, we have $y_j^i(\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) = 1$ such that $b_i = y_j^i - \langle \mathbf{w}_i, \mathbf{x}_j^i \rangle$. When there aren't α_j^i s in $(0, 1)$, i.e., either $\alpha_j^i = 0$ or $\alpha_j^i = 1$, the value of b_i can be obtained by resorting to a number of inequalities. Specifically, in the case that $\alpha_j^i = 0$, $y_j^i(\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) \geq 1$ should hold, while in the case that $\alpha_j^i = 1$, $y_j^i(\langle \mathbf{w}_i, \mathbf{x}_j^i \rangle + b_i) \leq 1$ should hold. Each of the inequalities can give rise to one upper or lower bound of b_i . Let \mathbf{b}_i^{up} and $\mathbf{b}_i^{\text{low}}$ denote the set of all these upper and lower bounds respectively, the value of b_i can be determined by $b_i = \frac{1}{2}(\max(\mathbf{b}_i^{\text{low}}) + \min(\mathbf{b}_i^{\text{up}}))$, where $\max(\mathbf{b}_i^{\text{low}})$ and $\min(\mathbf{b}_i^{\text{up}})$ return the maximum of $\mathbf{b}_i^{\text{low}}$ and minimum of \mathbf{b}_i^{up} respectively.

Optimizing w.r.t. \mathbf{C} when \mathbf{W} and \mathbf{b} are fixed. When we fix \mathbf{W} and \mathbf{b} , the optimization problem in Eq.(2) can be reformulated as follows:

$$\min_{\mathbf{C}} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top), \text{ s.t. } \mathbf{C} \succeq 0, \text{tr}(\mathbf{C}) \leq 1 \quad (14)$$

As per the property $\text{tr}(\mathbf{X}\mathbf{Y}\mathbf{Z}) = \text{tr}(\mathbf{Y}\mathbf{Z}\mathbf{X})$ and the constraint $\text{tr}(\mathbf{C}) \leq 1$, we can lower-bound the objective in Eq.(14) as follows:

$$\begin{aligned} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) &= \text{tr}(\mathbf{C}^{-1}\mathbf{W}^\top\mathbf{W}) \\ &\geq \text{tr}(\mathbf{C}^{-1}\mathbf{W}^\top\mathbf{W})\text{tr}(\mathbf{C}) \\ &= \text{tr}(\mathbf{C}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{C}^{-\frac{1}{2}})\text{tr}(\mathbf{C}^{\frac{1}{2}}\mathbf{C}^{\frac{1}{2}}) \\ &\geq (\text{tr}(\mathbf{C}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{C}^{\frac{1}{2}}))^2 = (\text{tr}(\mathbf{A}^{\frac{1}{2}}))^2 \end{aligned} \quad (15)$$

where $\mathbf{A} = \mathbf{W}^\top\mathbf{W}$. The last inequality in Eq.(15) holds based on the fact that both \mathbf{A} and \mathbf{C} are symmetric matrices as well as the following Lemma:

Lemma 1. *Given two matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{\ell_1 \times \ell_2}$, then the following inequality holds:*

$$\text{tr}(\mathbf{U}^\top\mathbf{U})\text{tr}(\mathbf{V}^\top\mathbf{V}) \geq (\text{tr}(\mathbf{U}^\top\mathbf{V}))^2$$

The left-hand side of the inequality can reach its minimum in the right-hand side when $\mathbf{U} = \mu \cdot \mathbf{V}$ where μ is one constant.

Proof. According to the property of matrix,

$$\begin{aligned} \text{tr}(\mathbf{U}^\top\mathbf{U}) &= \sum_{i=1}^{\ell_2} \sum_{j=1}^{\ell_1} U_{ij}^2 = \langle \text{vec}\mathbf{U}, \text{vec}\mathbf{U} \rangle = \|\text{vec}\mathbf{U}\|^2 \\ \text{tr}(\mathbf{V}^\top\mathbf{V}) &= \sum_{i=1}^{\ell_2} \sum_{j=1}^{\ell_1} V_{ij}^2 = \langle \text{vec}\mathbf{V}, \text{vec}\mathbf{V} \rangle = \|\text{vec}\mathbf{V}\|^2 \\ \text{tr}(\mathbf{U}^\top\mathbf{V}) &= \sum_{i=1}^{\ell_2} \sum_{j=1}^{\ell_1} U_{ij}V_{ij} = \langle \text{vec}\mathbf{U}, \text{vec}\mathbf{V} \rangle \end{aligned}$$

Here, $\text{vec}\mathbf{U}, \text{vec}\mathbf{V}$ denote the vectorized form of \mathbf{U}, \mathbf{V} . As per Cauchy-Schwarz inequality $\|\text{vec}\mathbf{U}\| \cdot \|\text{vec}\mathbf{V}\| \geq |\langle \text{vec}\mathbf{U}, \text{vec}\mathbf{V} \rangle|$, and let's square both sides of this inequality, then we have $\|\text{vec}\mathbf{U}\|^2 \cdot \|\text{vec}\mathbf{V}\|^2 \geq (\langle \text{vec}\mathbf{U}, \text{vec}\mathbf{V} \rangle)^2$ which is actually the result to be proved. The equality relationship holds only when $\text{vec}\mathbf{U} = \mu \cdot \text{vec}\mathbf{V}$, i.e., $\mathbf{U} = \mu \cdot \mathbf{V}$. \square

According to Eq.(15), $\text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top)$ can reach its minimum value $(\text{tr}(\mathbf{A}^{\frac{1}{2}}))^2$ when both $\text{tr}(\mathbf{C}) = 1$ and $\mathbf{A}^{\frac{1}{2}}\mathbf{C}^{-\frac{1}{2}} =$

TABLE I
THE PSEUDO-CODE OF M^3MDC .

Inputs:	
\mathcal{D} :	MDC training set $\{(\mathbf{x}_i, \mathbf{y}_i) \mid 1 \leq i \leq N\}$
λ_1, λ_2 :	trade-off parameters
\mathbf{x}_* :	unseen instance
Outputs:	
\mathbf{y}_* :	predicted class vector for \mathbf{x}_*
Process:	
1:	Transform the MDC training set \mathcal{D} into a total of $m = \sum_{j=1}^q \binom{K_j}{2}$ binary training sets via OvO decomposition w.r.t. each class space;
2:	Initialize $\mathbf{C} = \frac{1}{m}\mathbf{I}_m$ and $\boldsymbol{\alpha} = \mathbf{0}$;
3:	repeat
4:	while not all $\boldsymbol{\alpha}$ meet KKT conditions do
5:	for $i = 1$ to m do
6:	Solve the sub-QP problem in Eq.(11);
7:	end for
8:	end while
9:	Calculate \mathbf{C} according to Eq.(16);
10:	until convergence
11:	Obtain binary votings \mathbf{y}_*^b for \mathbf{x}_* according to Eq.(18);
12:	Return \mathbf{y}_* according to Eq.(19).

$\mu\mathbf{C}^{\frac{1}{2}}$ hold. Therefore, it is not difficult to have the following closed-form solution for \mathbf{C} :

$$\mathbf{C} = \frac{\mathbf{A}^{\frac{1}{2}}}{\text{tr}(\mathbf{A}^{\frac{1}{2}})} \quad (16)$$

Here, based on Eq.(5), the matrix \mathbf{A} can be expressed in terms of the inner product of instances:

$$\begin{aligned} \mathbf{A} &= \mathbf{W}^\top\mathbf{W} \\ &= \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} \mathbf{M}_{:i_1} \mathbf{M}_{:i_2}^\top \langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2} \rangle \end{aligned} \quad (17)$$

where $\mathbf{M}_{:i_1}$ ($\mathbf{M}_{:i_2}$) denotes the i_1 th (i_2)th column of \mathbf{M} .

As the above two alternating optimizing steps converge, we can obtain the predictive model, i.e., the optimal values of \mathbf{W} (or Lagrange multiplier vector $\boldsymbol{\alpha}$) and \mathbf{b} . Then, the class vector for unseen instance \mathbf{x}_* can be predicted. Specifically, a binary voting vector \mathbf{y}_*^b with m elements is returned as follows:

$$\begin{aligned} \mathbf{y}_*^b &= \text{sign}(\mathbf{W}^\top\mathbf{x}_* + \mathbf{b}) \\ &= \text{sign}\left(\sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_{ij}^i y_j^i \mathbf{M}e_i \langle \mathbf{x}_j^i, \mathbf{x}_* \rangle + \mathbf{b}\right) \end{aligned} \quad (18)$$

where $\text{sign}(\cdot)$ represents the (element-wise) signed function. Conceptually, for the i th pair of class labels l_+^i and l_-^i , one vote is recorded for l_+^i (l_-^i) if the i th element \mathbf{y}_*^b takes the value of $+1$ (-1). Here, the $\sum_{a=1}^{j-1} \binom{K_a}{2} + 1 \sim \sum_{a=1}^j \binom{K_a}{2}$ binary votes in \mathbf{y}_*^b correspond to the j th class space. Correspondingly, among the $\binom{K_j}{2}$ binary votes w.r.t. the j th class space $C_j = \{c_1^j, c_2^j, \dots, c_{K_j}^j\}$, let s_r^j denote the number of

recorded votes for c_r^j ($1 \leq r \leq K_j$). Then, the multi-dimensional prediction $\mathbf{y}_* = [y_{*1}, \dots, y_{*q}]^\top \in \mathcal{Y}$ for \mathbf{x}_* is determined by the OvO decoding rule (ties are broken at random):

$$y_{*j} = c_{\hat{r}}^j, \quad \text{where } \hat{r} = \arg \max_{1 \leq r \leq K_j} s_r^j \quad (1 \leq j \leq q) \quad (19)$$

In summary, the complete procedure of the proposed M³MDC approach is presented in Table I. Specifically, we firstly decompose the original MDC problem via OvO rule w.r.t. each class space (Step 1), based on which the covariance regularization is introduced for dependency modeling among class spaces. Then, an alternating procedure is invoked to solve the resulting optimization problem (Steps 2-10). Finally, the multi-dimensional prediction on unseen instance is obtained by querying the modeling outputs with OvO decoding rule (Steps 11-12).

Computational complexity. The QP problem in Eq.(11) with r variables can be solved with $\mathcal{O}(r^3)$ time complexity [33], and the square root for one matrix in Eq.(16) with $s \times s$ elements can be obtained with $\mathcal{O}(s^3)$ time complexity [3]. Then, the proposed M³MDC approach has computational complexity of $\mathcal{O}(T_1 \cdot T_2 \cdot m \cdot N^3 + T_1 \cdot m^3)$, where T_1 corresponds to the number of alternating rounds (i.e., Steps 3-10) and T_2 corresponds to the number of iterations in solving m sub-QP problems (i.e., Steps 4-8). Note that $\mathcal{O}(N^3)$ is the worse-case complexity of solving each Eq.(11) because the number of examples belonging to each OvO decomposition problem is less than the number of training examples in \mathcal{D} , i.e., N .

Kernel extension. The above derivations for M³MDC aim to learn linear model in the d -dimensional original feature space. When the data distribution is complicated, it might be better to learn nonlinear model in a d' -dimensional transformed feature space with the help of mapping function $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$. Then, we can reformulate the linear maximum margin MDC model in Eq.(2) as the following nonlinear version:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \boldsymbol{\xi}, \mathbf{C}} \quad & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) \\ \text{s.t.} \quad & \mathbf{C} \succeq 0, \text{tr}(\mathbf{C}) \leq 1, \\ & y_j^i (\langle \mathbf{w}_i, \phi(\mathbf{x}_j^i) \rangle + b_i) > 1 - \xi_j^i, \\ & \xi_j^i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_i \end{aligned} \quad (20)$$

Here, note that $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_m] \in \mathbb{R}^{d' \times m}$. This problem can also be solved by alternately optimizing the two sets of parameters $\{\mathbf{W}, \mathbf{b}\}$ and $\{\mathbf{C}\}$. When we fix \mathbf{C} , the optimization problem in Eq.(20) can be reformulated as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \boldsymbol{\xi}} \quad & \sum_{i=1}^m \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(\mathbf{W}\mathbf{W}^\top) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^\top) \\ \text{s.t.} \quad & y_j^i (\langle \mathbf{w}_i, \phi(\mathbf{x}_j^i) \rangle + b_i) > 1 - \xi_j^i, \\ & \xi_j^i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_i \end{aligned} \quad (21)$$

The dual problem of Eq.(21) is as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \langle \phi(\mathbf{x}_{j_1}^{i_1}), \phi(\mathbf{x}_{j_2}^{i_2}) \rangle \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i \\ \text{s.t.} \quad & \sum_{j=1}^{n_i} \alpha_j^i y_j^i = 0 \quad (1 \leq i \leq m), \quad 0 \leq \alpha_j^i \leq 1 \end{aligned}$$

The computation of inner product in d' -dimensional space would be intractable if d' is large (or even approaching infinity). To avoid such operation, kernel trick can be utilized where a kernel function can be defined as follows:

$$\kappa(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) = \langle \phi(\mathbf{x}_{j_1}^{i_1}), \phi(\mathbf{x}_{j_2}^{i_2}) \rangle$$

Then, the dual problem of Eq.(21) can be rewritten as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} M_{i_1 i_2} \kappa(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) \\ & - \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i \\ \text{s.t.} \quad & \sum_{j=1}^{n_i} \alpha_j^i y_j^i = 0 \quad (1 \leq i \leq m), \quad 0 \leq \alpha_j^i \leq 1 \end{aligned} \quad (22)$$

which can be solved similarly to the problem in Eq.(10). When we fix \mathbf{W} and \mathbf{b} , the optimization problem in Eq.(20) can also be reformulated as Eq.(14). For its closed-form solution, the matrix \mathbf{A} in Eq.(16) can be obtained with the help of kernel function $\kappa(\cdot, \cdot)$ as follows:

$$\mathbf{A} = \sum_{i_1=1}^m \sum_{j_1=1}^{n_{i_1}} \sum_{i_2=1}^m \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_1}^{i_1} y_{j_2}^{i_2} \mathbf{M}_{:i_1} \mathbf{M}_{:i_2}^\top \kappa(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$$

When the optimal values of \mathbf{W} (actually the Lagrange multiplier vector $\boldsymbol{\alpha}$) and \mathbf{b} are obtained, the binary voting vector \mathbf{y}_* with m elements can be obtained similarly to Eq.(18) as follows:

$$\mathbf{y}_*^b = \text{sign} \left(\sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_j^i y_j^i \mathbf{M} \mathbf{e}_i \kappa(\mathbf{x}_j^i, \mathbf{x}_*) + \mathbf{b} \right) \quad (23)$$

Based on \mathbf{y}_*^b , we can obtain the final prediction for unseen instance via OvO decoding according to Eq.(19).

III. RELATED WORK

In this section, we briefly discuss learning settings related to MDC as well as existing approaches in learning from MDC examples.

On one hand, MDC can be regarded as a specific instantiation of multi-output learning [46], where each output in MDC corresponds to a discrete class variable taking values among a number of possible class labels. On the other hand, the traditional multi-class classification (MCC) can be regarded as a degenerated version of MDC by assuming only one class

space.² If ordinal relationship exists among the class labels of each class space, the MDC problem can be generalized to the problem of multiple ordinal output classification [31]. The MDC problem is also closely-related to multi-label classification (MLC) [53], [17], [52], which can be regarded as a degenerated version of MDC by only assuming binary-valued class variable for each class space. More importantly, the essential difference between MDC and MLC lies in whether the class space is *heterogeneous* or *homogeneous*. Generally, MDC assumes *heterogeneous* class spaces which characterize object’s semantics along different dimensions, while MLC assumes *homogeneous* class space which characterizes the relevancy of specific concepts along one dimension. Therefore, one should avoid directly aligning the modeling outputs of class labels residing in different class spaces when designing MDC models.

By treating each class space independently, the MDC problem can be tackled by solving a number of multi-class classification problems. However, this intuitive strategy ignores possible dependencies among class spaces and would lead to sub-optimal solutions. A straightforward strategy to consider class dependencies is to train a single multi-class classifier where every distinct class combination appearing in the training set is regarded as a new class. However, this strategy is incapable of predicting class combinations absent in the training set and is difficult to be learned due to the huge number of possible classes in the transformed class space. Therefore, it might be helpful to group all class spaces into several super-classes before subsequent MDC model induction [35]. Moreover, the q MDC class spaces can be jointly solved by training a chain of q multi-class classifiers (one per class space) where the predictive outputs of preceding classifiers in the chain are treated as extra features by subsequent classifiers [51], [36]. Besides, a number of existing MDC approaches assume a DAG structure over class spaces to explicitly model potential class space dependencies, where different DAG structures lead to a family of MDC models called multi-dimensional Bayesian network classifier [43], [11], [37], [18]. Recent works further explore efficient structure learning strategies [50], [5], [57], [1] to tackle the demanding training complexity of DAG-based MDC approaches.

Maximum margin techniques have been widely adapted to solve learning problems related to MDC such as MCC and MLC. For MCC problem, one can work with margin-based classification models by transforming the original MCC problem into a number of binary classification problems via different decompositions (e.g., one-vs-one, one-vs-rest, and many-vs-many), or by maximizing multi-class margins directly [21], [49]. For MLC, one can also work with margin-based classification models via binary decomposition, such as maximizing margins between a pair of class labels [15], [47], or by maximizing output coding margins [28], [29], [42], etc. It is worth noting that, to model dependencies among class spaces, the regularization covariance term $\text{tr}(\mathbf{W}\mathbf{C}^{-1}\mathbf{W}^T)$ in Eq.(2) has been utilized by M³MDC to help learn a set

TABLE II
CHARACTERISTICS OF THE BENCHMARK DATA SETS.

Data Set	#Exam.	#Dim.	#Labels/Dim.	#Features [†]
Edm	154	2	3	16 n
Flare1	323	3	3,4,2	10 x
Cal500	502	10	2	68 n
Music	591	6	2	71 n
Enb	768	2	2,4	6 n
Song	785	3	3	98 n
WQplants	1060	7	4	16 n
WQanimals	1060	7	4	16 n
WaterQuality	1060	14	4	16 n
BeLaE	1930	5	5	44 n ,1 x
Yeast	2417	14	2	103 n
Voice	3136	2	4,2	19 n
Thyroid	9172	7	5,5,3,2,4,4,3	7 n ,22 x
TIC2000	9822	3	6,4,2	83 x
Adult	18418	4	7,7,5,2	5 n ,5 x

[†] n and x denote numeric and nominal features respectively.

of classifiers in a joint manner. Nonetheless, different from existing works on covariance regularization [56], [27], [31], M³MDC aims at solving MDC problem by coupling covariance regularization with empirical loss of OvO decomposition w.r.t. each class space.

IV. EXPERIMENTS

To validate the effectiveness of M³MDC in learning from multi-dimensional examples, extensive comparative studies are conducted in this section. Firstly, Subsection IV-A introduces experimental setup including the employed data sets, compared approaches and evaluation metrics. Then, Subsection IV-B reports the detailed experimental results with statistical comparisons. Lastly, Subsection IV-C further investigates properties of M³MDC based on correlation, sensitivity, convergence and parameter initialization analyses.

A. Experimental Setup

1) *Benchmark Data Sets*: In this paper, we have collected a total of fifteen real-world MDC data sets for thorough comparative studies. To the best of our knowledge, this serves as the most comprehensive basis for MDC performance evaluation in terms of the number of benchmark data sets [2], [35], [30], [23], [44]. The characteristics of all benchmark data sets are summarized in Table II, including *number of examples* (#Exam.), *number of class spaces* (#Dim.), *number of class labels w.r.t each class space* (#Labels/Dim.),³ and *number of features* (#Features).

2) *Compared Approaches*: In this paper, five well-established MDC approaches have been employed as compared approaches, including BR [35], [24], ECC [36], ECP [35], [23], ESC [35] and gMML [30]:

²Furthermore, the recently proposed dual set multi-label learning problem [26] can also be regarded as a degenerated version of MDC by assuming two class spaces.

³If the number of class labels w.r.t. each class space is identical, then only this number is recorded; Otherwise, the number of class labels w.r.t. each class space is recorded in turn.

- BR works by training a number of independent multi-class classifiers, one per class space. Therefore, BR does not consider dependencies among class spaces in model induction.
- ECC works by training a chain of multi-class classifiers, one per class space, where the predictions of preceding classifiers in the chain are used as extra features in training the subsequent ones. Therefore, ECC exploits dependency modeling via the specified chaining order over class spaces.
- ECP works by training a single multi-class classifier via powerset transformation in output space, where all distinct class combinations in output space are treated as new classes. Therefore, ECP exploits dependency modeling via powerset transformation.
- ESC works by grouping the original class variables into super-classes, where each super-class is treated as a new class variable and all distinct class combinations in this super-class are treated as its new classes.
- gMML works by alternately learning regression models for each class label and a Mahalanobis metric characterizing the closeness between regression outputs and ground-truth labeling information.

For ensemble approaches ECC, ECP and ESC, the base MDC model is trained over a random cut of 67% examples from the original MDC training set and a total of ten base classifiers are used [35], [23]. Furthermore, we aggregate predictions of base MDC models via majority voting. For all the compared approaches (except gMML which doesn't necessitate base multi-class classifier), support vector machine (SVM) is used as the base multi-class classifier. Specifically, LIBSVM [9] with either linear kernel or RBF kernel is used to implement the base multi-class classifier. Here, it is worth noting that we employ LIBSVM to implement the base classifier which solves the multi-class classification problems via OvO decomposition for fair comparison between M³MDC and the compared approaches. For ESC, the classifier chains model is used to solve the resulting problem obtained by super-class partition, and the fine-tune mechanism is not used because it does not bring significant performance improvements [35]. For gMML, parameters λ , t , γ and k are tuned from the range $\{1, 10, 100\}$, $\{0.3, 0.5, 0.7\}$, $\{0, 0.1, 0.2\}$ and $\{5, 10, 15, 20, 25\}$ respectively, and η is fixed as 3 as recommended in the corresponding literature [30]. As shown in Table I, the two trade-off parameters λ_1, λ_2 for M³MDC are set as 0.001 and 0.1 respectively.

Table III summarizes the computational complexity of all compared approaches. For M³MDC and gMML, its time complexity has been analyzed in Section II and in [30] (Subsection 4.4). For BR, ECC, ECP, ESC, the multi-class classification problem is solved by the binary classifier SVM with the help of OvO decomposition. The main complexity of SVM [10] corresponds to solving the dual QP problem [33]. In Table III, E denotes the number of base learners (i.e., ensemble size) in ECC, ECP, ESC. For ESC [35], θ and K_j^θ denote the number of super-classes and the number of class labels in the j th super-class. For gMML [30], k denotes the

TABLE III
THE TIME COMPLEXITY OF M³MDC AND ALL COMPARED APPROACHES.

Algo.	Time complexity
M ³ MDC	$\mathcal{O}(T_1 \cdot T_2 \cdot m \cdot N^3 + T_1 \cdot m^3)$
BR	$\mathcal{O}(m \cdot N^3)$
ECC	$\mathcal{O}(E \cdot m \cdot N^3)$
ECP	$\mathcal{O}(E \cdot (\prod_{j=1}^q K_j)^2 \cdot N^3)$
ESC	$\mathcal{O}(E \cdot m \cdot N^3 + E \cdot (\prod_{j=1}^\theta K_j^\theta)^2 \cdot N^3)$
gMML	$\mathcal{O}(d^3 + (\sum_{j=1}^q K_j)^3 + Nd^3 + Nd(\sum_{j=1}^q K_j) + Nk)$

number of nearest neighbors considered.

3) *Evaluation Metrics*: In this paper, the widely-used three metrics, i.e., *Hamming Score*, *Exact Match* and *Sub-Exact Match* [2], [35], [30], [23], [24], are employed to measure the generalization performance of MDC approaches.⁴ Specifically, let $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid 1 \leq i \leq p\}$ be the test set where the ground-truth class vector associated with \mathbf{x}_i is $\mathbf{y}_i = [y_{i1}, y_{i2}, \dots, y_{iq}]^\top$. For the MDC predictive function f to be evaluated, the class vector of \mathbf{x}_i predicted by f is denoted as $\hat{\mathbf{y}}_i = f(\mathbf{x}_i) = [\hat{y}_{i1}, \hat{y}_{i2}, \dots, \hat{y}_{iq}]^\top$. Then, the number of class spaces which f predicts correctly can be calculated as $r^{(i)} = \sum_{j=1}^q \llbracket y_{ij} = \hat{y}_{ij} \rrbracket$. Here, the predicate $\llbracket \pi \rrbracket$ returns 1 if π holds and 0 otherwise. Accordingly, formal definitions of the employed evaluation metrics correspond to:

- *Hamming Score*:

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

- *Exact Match*:

$$\text{EMatch}_{\mathcal{S}}(f) = \frac{1}{p} \sum_{i=1}^p \llbracket r^{(i)} = q \rrbracket$$

- *Sub-Exact Match*:

$$\text{SEMATCH}_{\mathcal{S}}(f) = \frac{1}{p} \sum_{i=1}^p \llbracket r^{(i)} \geq q - 1 \rrbracket$$

In a nutshell, *Hamming Score* returns the average fraction of correctly predicted class spaces. *Exact Match* returns the proportion of test examples whose class spaces are predicted correctly in full. *Sub-Exact Match* returns the proportion of test examples for which at least $q - 1$ class spaces are predicted correctly. Obviously, *Sub-Exact Match* corresponds to a relaxed version of *Exact Match*, where the value of *Exact Match* might be rather low when the MDC task has a large number of class spaces. For all three metrics, the *larger* the values the better the performance. In this paper, ten-fold cross validation⁵ is conducted on all benchmark data sets, and both

⁴In some literatures, *Hamming Score* and *Exact Match* are also termed as *Class Accuracy* and *Example Accuracy* [35], or *Mean Accuracy* and *Global Accuracy* [2].

⁵In this paper, each data set is randomly split into ten equal-sized folds. Generally, the random splitting would result in stratified sampling approximately and ensure that the training examples in each fold cover all classes. Exceptions might occur when the number of examples belonging to one class is limited. In ten-fold cross validation, M³MDC (as well as the compared methods) is trained on nine folds where the training examples do cover all classes in our experimental studies.

TABLE IV

PREDICTIVE PERFORMANCE (MEAN±STD. DEVIATION) OF EACH COMPARED APPROACH ON THE BENCHMARK DATA SETS (LINEAR KERNEL FOR M^3 MDC, BR, ECC, ECP AND ESC). THE PERFORMANCE RANK ON EACH DATA SET IS SHOWN IN THE PARENTHESES WHERE WE ALSO HIGHLIGHT THE BEST PERFORMANCE IN BOLDFACE AND UNDERLINE THE SECOND PERFORMANCE.

(a) Hamming Score

Data Set	M^3 MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.727±0.089(1)	0.689±0.070(6)	0.695±0.065(5)	0.721±0.082(2)	0.698±0.089(4)	0.714±0.083(3)
Flare1	0.923±0.033(2)	0.922±0.034(4)	0.922±0.034(4)	0.921±0.036(6)	0.923±0.033(2)	0.925±0.034(1)
Cal500	0.630±0.010(2)	0.628±0.011(3)	0.625±0.015(4)	0.616±0.015(5)	0.616±0.019(5)	0.631±0.014(1)
Music	0.811±0.022(3)	0.808±0.023(4)	0.814±0.025(1)	0.799±0.032(6)	0.813±0.028(2)	0.800±0.018(5)
Enb	0.793±0.026(1)	0.734±0.029(5)	0.754±0.033(3)	0.728±0.043(6)	0.759±0.043(2)	0.742±0.027(4)
Song	0.796±0.028(1)	0.793±0.023(2)	0.790±0.024(3)	0.786±0.029(6)	0.790±0.029(3)	0.788±0.027(5)
WQplants	0.660±0.013(1)	0.657±0.016(2)	0.654±0.016(4)	0.647±0.015(6)	0.651±0.017(5)	0.655±0.015(3)
WQanimals	0.632±0.014(1)	0.630±0.014(3)	0.630±0.014(3)	0.629±0.013(6)	0.631±0.014(2)	0.630±0.015(3)
WaterQuality	0.646±0.012(1)	0.644±0.013(2)	0.643±0.013(3)	0.628±0.015(6)	0.641±0.013(5)	0.643±0.013(3)
BeLaE	0.454±0.021(1)	0.447±0.015(4)	0.451±0.018(2)	0.413±0.017(6)	0.450±0.015(3)	0.417±0.020(5)
Yeast	0.802±0.006(1)	0.801±0.006(3)	0.797±0.007(5)	0.795±0.007(6)	0.802±0.006(1)	0.800±0.005(4)
Voice	0.970±0.008(1)	0.964±0.007(2)	0.961±0.008(3)	0.955±0.013(5)	0.961±0.009(3)	0.842±0.009(6)
Thyroid	0.965±0.002(1)	0.965±0.002(1)	0.965±0.002(1)	0.965±0.002(1)	0.965±0.002(1)	0.960±0.002(6)
TIC2000	0.935±0.004(1)	0.934±0.004(4)	0.935±0.004(1)	0.926±0.005(5)	0.935±0.004(1)	0.895±0.007(6)
Adult	0.711±0.004(1)	0.710±0.004(2)	0.710±0.004(2)	0.708±0.004(4)	0.708±0.004(4)	0.705±0.004(6)

(b) Exact Match

Data Set	M^3 MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.500±0.151(3)	0.442±0.125(6)	0.454±0.123(5)	0.559±0.136(1)	0.512±0.142(2)	0.487±0.145(4)
Flare1	0.821±0.073(1)	0.821±0.073(1)	0.817±0.078(5)	0.817±0.078(5)	0.821±0.073(1)	0.821±0.075(1)
Cal500	0.016±0.016(3)	0.016±0.016(3)	0.020±0.016(2)	0.026±0.028(1)	0.014±0.013(5)	0.014±0.013(5)
Music	0.281±0.074(4)	0.272±0.075(5)	0.346±0.079(2)	0.343±0.076(3)	0.350±0.078(1)	0.252±0.056(6)
Enb	0.586±0.051(1)	0.469±0.059(5)	0.508±0.066(3)	0.456±0.086(6)	0.518±0.085(2)	0.483±0.053(4)
Song	0.486±0.069(1)	0.479±0.059(6)	0.481±0.057(4)	0.484±0.054(2)	0.481±0.062(4)	0.484±0.059(2)
WQplants	0.100±0.034(1)	0.097±0.033(2)	0.093±0.037(3)	0.093±0.028(3)	0.093±0.037(3)	0.092±0.035(6)
WQanimals	0.059±0.022(5)	0.058±0.022(6)	0.061±0.023(4)	0.065±0.018(1)	0.064±0.024(2)	0.062±0.023(3)
WaterQuality	0.008±0.008(1)	0.007±0.008(2)	0.006±0.008(3)	0.001±0.003(6)	0.006±0.008(3)	0.006±0.008(3)
BeLaE	0.033±0.016(2)	0.031±0.013(4)	0.031±0.016(4)	0.035±0.016(1)	0.032±0.013(3)	0.022±0.009(6)
Yeast	0.157±0.018(4)	0.151±0.017(5)	0.207±0.014(3)	0.252±0.012(1)	0.237±0.017(2)	0.134±0.018(6)
Voice	0.941±0.017(1)	0.929±0.014(2)	0.923±0.016(4)	0.912±0.025(5)	0.924±0.016(3)	0.699±0.017(6)
Thyroid	0.777±0.014(1)	0.773±0.015(2)	0.772±0.014(4)	0.773±0.014(2)	0.771±0.014(5)	0.741±0.015(6)
TIC2000	0.815±0.011(1)	0.812±0.011(4)	0.814±0.012(2)	0.791±0.014(5)	0.814±0.012(2)	0.706±0.018(6)
Adult	0.252±0.011(4)	0.247±0.009(5)	0.260±0.008(3)	0.310±0.009(1)	0.310±0.009(1)	0.230±0.009(6)

(c) Sub-Exact Match

Data Set	M^3 MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.955±0.053(1)	0.935±0.061(3)	0.935±0.069(3)	0.883±0.074(5)	0.883±0.074(5)	0.941±0.065(2)
Flare1	0.951±0.036(2)	0.947±0.039(5)	0.951±0.036(2)	0.947±0.039(5)	0.951±0.036(2)	0.957±0.039(1)
Cal500	0.082±0.046(2)	0.074±0.037(5)	0.080±0.031(3)	0.078±0.036(4)	0.086±0.038(1)	0.072±0.041(6)
Music	0.687±0.067(1)	0.674±0.067(3)	0.676±0.064(2)	0.640±0.064(6)	0.662±0.075(4)	0.652±0.040(5)
Enb	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)
Song	0.905±0.039(1)	0.903±0.033(2)	0.891±0.036(4)	0.878±0.040(6)	0.892±0.038(3)	0.883±0.041(5)
WQplants	0.289±0.052(1)	0.287±0.055(2)	0.283±0.049(4)	0.281±0.049(6)	0.282±0.049(5)	0.286±0.053(3)
WQanimals	0.236±0.028(1)	0.229±0.034(4)	0.229±0.032(4)	0.230±0.032(3)	0.232±0.032(2)	0.227±0.033(6)
WaterQuality	0.051±0.025(1)	0.051±0.024(1)	0.050±0.023(3)	0.035±0.018(6)	0.046±0.022(5)	0.049±0.024(4)
BeLaE	0.162±0.029(2)	0.158±0.023(4)	0.164±0.025(1)	0.135±0.024(5)	0.159±0.024(3)	0.130±0.020(6)
Yeast	0.273±0.028(4)	0.269±0.029(5)	0.288±0.023(3)	0.304±0.020(2)	0.310±0.030(1)	0.266±0.026(6)
Voice	0.999±0.001(1)	0.999±0.002(1)	0.998±0.002(3)	0.998±0.003(3)	0.998±0.002(3)	0.985±0.011(6)
Thyroid	0.982±0.004(1)	0.982±0.004(1)	0.981±0.004(5)	0.981±0.005(5)	0.982±0.004(1)	0.982±0.005(1)
TIC2000	0.991±0.004(1)	0.989±0.003(4)	0.990±0.003(2)	0.987±0.003(5)	0.990±0.003(2)	0.978±0.003(6)
Adult	0.665±0.009(3)	0.669±0.009(1)	0.662±0.009(4)	0.638±0.007(5)	0.638±0.008(5)	0.669±0.008(1)

the mean metric value and standard deviation are recorded for comparative studies.

B. Experimental Results

Detailed experimental results are reported in Tables IV-V, where the performance rank on each data set is also

TABLE V
 PREDICTIVE PERFORMANCE (MEAN±STD. DEVIATION) OF EACH COMPARED APPROACH ON THE BENCHMARK DATA SETS (RBF KERNEL FOR M³MDC, BR, ECC, ECP AND ESC). THE PERFORMANCE RANK ON EACH DATA SET IS SHOWN IN THE PARENTHESES WHERE WE ALSO HIGHLIGHT THE BEST PERFORMANCE IN BOLDFACE AND UNDERLINE THE SECOND PERFORMANCE.

(a) Hamming Score						
Data Set	M ³ MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.721±0.056(1)	0.694±0.047(6)	0.698±0.053(5)	0.714±0.077(2)	0.704±0.066(4)	0.714±0.083(2)
Flare1	0.923±0.033(2)	0.923±0.033(2)	0.923±0.033(2)	0.923±0.033(2)	0.923±0.033(2)	0.925±0.034(1)
Cal500	0.618±0.019(2)	0.613±0.019(3)	0.613±0.021(3)	0.613±0.020(3)	0.593±0.029(6)	0.631±0.014(1)
Music	0.801±0.023(1)	0.739±0.014(3)	0.738±0.028(4)	0.687±0.044(6)	0.695±0.033(5)	0.800±0.018(2)
Enb	0.745±0.029(1)	0.716±0.029(3)	0.681±0.035(4)	0.667±0.019(5)	0.665±0.022(6)	0.742±0.027(2)
Song	0.773±0.027(2)	0.771±0.026(3)	0.770±0.025(4)	0.769±0.027(5)	0.766±0.027(6)	0.788±0.027(1)
WQplants	0.652±0.016(2)	0.649±0.016(3)	0.648±0.016(4)	0.647±0.015(5)	0.647±0.015(5)	0.655±0.015(1)
WQanimals	0.630±0.013(1)	0.628±0.013(3)	0.628±0.013(3)	0.628±0.013(3)	0.628±0.013(3)	0.630±0.015(1)
WaterQuality	0.640±0.013(2)	0.639±0.013(3)	0.638±0.012(4)	0.627±0.017(6)	0.638±0.012(4)	0.643±0.013(1)
BeLaE	0.437±0.019(1)	0.423±0.022(2)	0.408±0.022(4)	0.354±0.018(6)	0.374±0.020(5)	0.417±0.020(3)
Yeast	0.791±0.007(2)	0.775±0.006(4)	0.776±0.008(3)	0.739±0.009(6)	0.741±0.012(5)	0.800±0.005(1)
Voice	0.962±0.008(1)	0.940±0.010(2)	0.930±0.008(4)	0.905±0.009(5)	0.931±0.009(3)	0.842±0.009(6)
Thyroid	0.961±0.003(1)	0.961±0.002(1)	0.961±0.002(1)	0.961±0.002(1)	0.961±0.002(1)	0.960±0.002(6)
TIC2000	0.904±0.007(1)	0.892±0.008(3)	0.884±0.007(4)	0.850±0.006(6)	0.884±0.007(4)	0.895±0.007(2)
Adult	0.699±0.004(4)	0.701±0.004(3)	0.702±0.005(2)	0.675±0.006(5)	0.675±0.006(5)	0.705±0.004(1)

(b) Exact Match						
Data Set	M ³ MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.460±0.118(3)	0.389±0.093(6)	0.395±0.106(5)	0.486±0.129(2)	0.454±0.110(4)	0.487±0.145(1)
Flare1	0.821±0.073(1)	0.821±0.073(1)	0.821±0.073(1)	0.821±0.073(1)	0.821±0.073(1)	0.821±0.075(1)
Cal500	0.018±0.020(3)	0.006±0.010(6)	0.024±0.025(1)	0.022±0.025(2)	0.010±0.017(5)	0.014±0.013(4)
Music	0.257±0.065(1)	0.078±0.041(6)	0.135±0.071(5)	0.194±0.063(4)	0.200±0.068(3)	0.252±0.056(2)
Enb	0.490±0.057(1)	0.431±0.058(3)	0.362±0.069(4)	0.335±0.037(5)	0.330±0.045(6)	0.483±0.053(2)
Song	0.453±0.056(2)	0.449±0.060(3)	0.446±0.055(4)	0.442±0.059(5)	0.438±0.059(6)	0.484±0.059(1)
WQplants	0.092±0.031(4)	0.092±0.030(4)	0.094±0.029(1)	0.094±0.029(1)	0.094±0.029(1)	0.092±0.035(4)
WQanimals	0.058±0.022(2)	0.056±0.024(3)	0.056±0.024(3)	0.056±0.025(3)	0.056±0.024(3)	0.062±0.023(1)
WaterQuality	0.007±0.008(1)	0.006±0.008(2)	0.006±0.008(2)	0.001±0.003(6)	0.006±0.008(2)	0.006±0.008(2)
BeLaE	0.031±0.010(2)	0.028±0.010(3)	0.035±0.012(1)	0.025±0.009(4)	0.025±0.008(4)	0.022±0.009(6)
Yeast	0.058±0.018(5)	0.014±0.007(6)	0.067±0.016(4)	0.139±0.007(1)	0.138±0.021(2)	0.134±0.018(3)
Voice	0.926±0.016(1)	0.884±0.017(2)	0.866±0.015(4)	0.825±0.016(5)	0.867±0.016(3)	0.699±0.017(6)
Thyroid	0.748±0.015(1)	0.743±0.014(2)	0.743±0.014(2)	0.742±0.014(4)	0.742±0.014(4)	0.741±0.015(6)
TIC2000	0.732±0.018(1)	0.698±0.019(3)	0.675±0.016(4)	0.587±0.016(6)	0.675±0.016(4)	0.706±0.018(2)
Adult	0.216±0.010(6)	0.228±0.006(5)	0.251±0.009(3)	0.269±0.011(1)	0.269±0.011(1)	0.230±0.009(4)

(c) Sub-Exact Match						
Data Set	M ³ MDC	BR [35], [24]	ECC [36]	ECP [35], [23]	ESC [35]	gMML [30]
Edm	0.981±0.031(3)	1.000±0.000(1)	1.000±0.000(1)	0.941±0.049(5)	0.954±0.055(4)	0.941±0.065(5)
Flare1	0.951±0.036(2)	0.951±0.036(2)	0.951±0.036(2)	0.951±0.036(2)	0.951±0.036(2)	0.957±0.039(1)
Cal500	0.068±0.037(6)	0.070±0.036(5)	0.072±0.029(2)	0.090±0.037(1)	0.072±0.028(2)	0.072±0.041(2)
Music	0.635±0.060(2)	0.454±0.063(4)	0.476±0.084(3)	0.436±0.077(6)	0.446±0.073(5)	0.652±0.040(1)
Enb	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)	1.000±0.000(1)
Song	0.869±0.040(2)	0.868±0.032(4)	0.869±0.033(2)	0.868±0.038(4)	0.862±0.034(6)	0.883±0.041(1)
WQplants	0.287±0.048(1)	0.284±0.051(3)	0.282±0.050(4)	0.282±0.048(4)	0.282±0.048(4)	0.286±0.053(2)
WQanimals	0.231±0.030(1)	0.226±0.031(3)	0.226±0.031(3)	0.225±0.031(6)	0.226±0.031(3)	0.227±0.033(2)
WaterQuality	0.045±0.023(2)	0.044±0.024(4)	0.045±0.023(2)	0.033±0.018(6)	0.043±0.020(5)	0.049±0.024(1)
BeLaE	0.151±0.023(1)	0.132±0.024(3)	0.134±0.016(2)	0.093±0.010(6)	0.110±0.012(5)	0.130±0.020(4)
Yeast	0.236±0.023(2)	0.110±0.014(6)	0.186±0.016(3)	0.175±0.015(5)	0.178±0.026(4)	0.266±0.026(1)
Voice	0.998±0.002(1)	0.996±0.005(2)	0.995±0.005(3)	0.984±0.006(6)	0.995±0.005(3)	0.985±0.011(5)
Thyroid	0.982±0.004(3)	0.983±0.004(1)	0.983±0.004(1)	0.982±0.004(3)	0.982±0.004(3)	0.982±0.005(3)
TIC2000	0.981±0.004(1)	0.979±0.004(2)	0.977±0.005(4)	0.964±0.006(6)	0.976±0.005(5)	0.978±0.003(3)
Adult	0.658±0.008(2)	0.657±0.010(3)	0.651±0.010(4)	0.587±0.011(5)	0.586±0.011(6)	0.669±0.008(1)

shown in the parentheses. Moreover, to show whether M³MDC achieves statistically superior performance against compared approaches, we employ *Wilcoxon signed-ranks test* [12] (at 0.05 significance level) whose statistical test results are sum-

TABLE VI
WILCOXON SIGNED-RANKS TEST FOR THE PROPOSED M³MDC APPROACH AGAINST EACH COMPARED APPROACH IN TERMS OF *Hamming Score*, *Exact Match* AND *Sub-Exact Match* RESPECTIVELY WHERE THE *p*-VALUES AT 0.05 SIGNIFICANCE LEVEL ARE ALSO SHOWN IN THE BRACKETS.

(a) Linear kernel for M ³ MDC, BR, ECC, ECP and ESC					
Evaluation Metric	M ³ MDC vs BR	M ³ MDC vs ECC	M ³ MDC vs ECP	M ³ MDC vs ESC	M ³ MDC vs gMML
<i>Hamming Score</i>	win [6.10e-05]	win [1.53e-03]	win [6.10e-05]	win [1.22e-03]	win [6.10e-04]
<i>Exact Match</i>	win [2.44e-04]	tie [4.21e-01]	tie [8.47e-01]	tie [1.00e+00]	win [6.71e-04]
<i>Sub-Exact Match</i>	win [6.10e-03]	tie [5.74e-02]	win [8.54e-03]	tie [8.03e-02]	win [5.25e-03]

(b) RBF kernel for M ³ MDC, BR, ECC, ECP and ESC					
Evaluation Metric	M ³ MDC vs BR	M ³ MDC vs ECC	M ³ MDC vs ECP	M ³ MDC vs ESC	M ³ MDC vs gMML
<i>Hamming Score</i>	win [6.10e-04]	win [8.54e-04]	win [1.22e-04]	win [1.22e-04]	tie [9.78e-01]
<i>Exact Match</i>	win [3.30e-03]	tie [1.58e-01]	tie [2.22e-01]	tie [8.13e-02]	tie [8.58e-01]
<i>Sub-Exact Match</i>	win [3.98e-02]	tie [7.42e-02]	win [8.06e-03]	win [2.44e-03]	tie [6.37e-01]

marized in Table VI.⁶

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

- Among M³MDC and five compared approaches, M³MDC ranks first in 51 cases (56.7%), ranks second in 21 cases (23.3%) across all the 90 cases (15 data sets × 3 evaluation metrics × 2 kernel types).
- As shown in Table VI, M³MDC achieves statistically superior performance against BR, ECC, ECP, ESC in terms of *Hamming Score* on both kernel types.
- ECP works by conducting class powerset transformation in output space and then training a multi-class classifier, which actually can be viewed as optimizing *Exact Match*. It is impressive to notice that M³MDC still achieves comparable performance against ECP and ranks first in 15 out of 30 cases in term of this metric.
- It is worth noting that M³MDC achieves statistically superior performance against BR in terms of all three metrics on both kernel types. This result clearly validates the necessity of considering class dependencies in learning from MDC examples and also the effectiveness of M³MDC’s dependency modeling strategy.

C. Further Analysis

1) *Correlation Analysis*: In this paper, M³MDC makes use of covariance matrix \mathbf{C} in Eq.(2) to model the dependencies among class spaces. Here, we normalize each element in \mathbf{C} with its corresponding two diagonal elements as follows and then obtain the correlation matrix \mathbf{R} :

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} \times C_{jj}}} \quad (24)$$

where R_{ij} (C_{ij}) denotes the element in the i th row and j th column of \mathbf{R} (\mathbf{C}). Specifically, the value of R_{ij} represents

⁶In this paper, both the mean metric value and standard deviation are represented by three decimal digits which will make some experimental results with tiny difference looking to be exactly the same. For example, the mean metric values of *Hamming Score* for M³MDC, BR, ECC, ECP and ESC over *Thyroid* in Table IV are 0.965470, 0.964956, 0.964753, 0.964862 and 0.964675 by keeping six decimal digits. The performance rank is based on the rounded mean metric values with three decimal digits, while the *Wilcoxon signed-ranks test* is based on the accurate mean metric values.

the degree of correlation between the i th and j th pair of class labels, where $R_{ij} = +1$ (-1) indicates fully positive (negative) correlation while $R_{ij} = 0$ indicates null correlation.

To show whether the proposed M³MDC approach can recover the ground-truth dependencies among class spaces, we generate a synthetic MDC data set with known class dependencies. Specifically, a total of 1000 examples in two-dimensional feature space within the unit square are generated. Three class variables (denoted as y_1 , y_2 and y_3) are considered whose values are set as:

$$y_1 = \begin{cases} 1, & \text{when } x_1 + x_2 < 1 \\ 2, & \text{otherwise} \end{cases}$$

$$y_2 = \begin{cases} 1, & \text{when } x_1 + x_2 < 0.5 \\ 2, & \text{when } 0.5 < x_1 + x_2 < 1 \\ 3, & \text{when } 1 < x_1 + x_2 < 1.5 \\ 4, & \text{otherwise} \end{cases}$$

$$y_3 = \begin{cases} 1, & \text{when } x_1 > x_2 \\ 2, & \text{otherwise} \end{cases}$$

In this case, it is not difficult to show that y_1 and y_2 are positively related while both y_1 and y_2 are independent of y_3 . For the proposed M³MDC approach, a total of 8 binary classification models will be generated accordingly based on the one-vs-one decomposition. We depict the correlation matrix \mathbf{R} learned by M³MDC (with linear kernel) in Figure 1(a). Note that for the correlation value, red color corresponds to $+1$ and blue color corresponds to -1 , while green color corresponds to 0. It is shown that, excluding diagonal elements, elements in the 8-th row and the 8-th column almost equal to zero (at most 0.0168) while the remaining elements almost equal to one (at least 0.9998). These results indicate that the M³MDC is capable of recovering the ground-truth dependencies among class spaces.

Moreover, we also depict the correlation matrix \mathbf{R} learned by M³MDC (with linear kernel) on two real-world MDC data sets *Song* and *WaterQuality* in Figure 1(b)-(c). It is shown that, excluding diagonal elements, some red or blue blocks do exist which indicate dependencies between classes, while more blocks are green which indicate independencies between classes. Generally, we should consider class dependencies when designing MDC approaches but also need to be

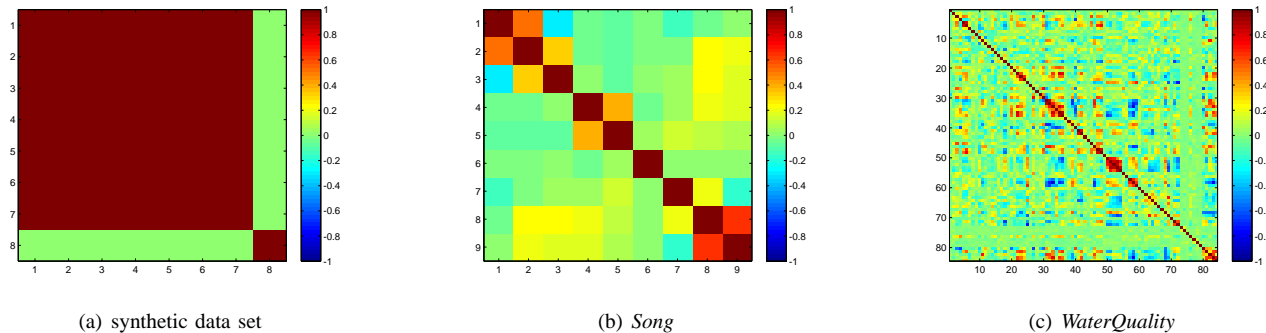


Fig. 1. Correlation matrix on synthetic data set and two real-world MDC data sets *Song*, *WaterQuality*.

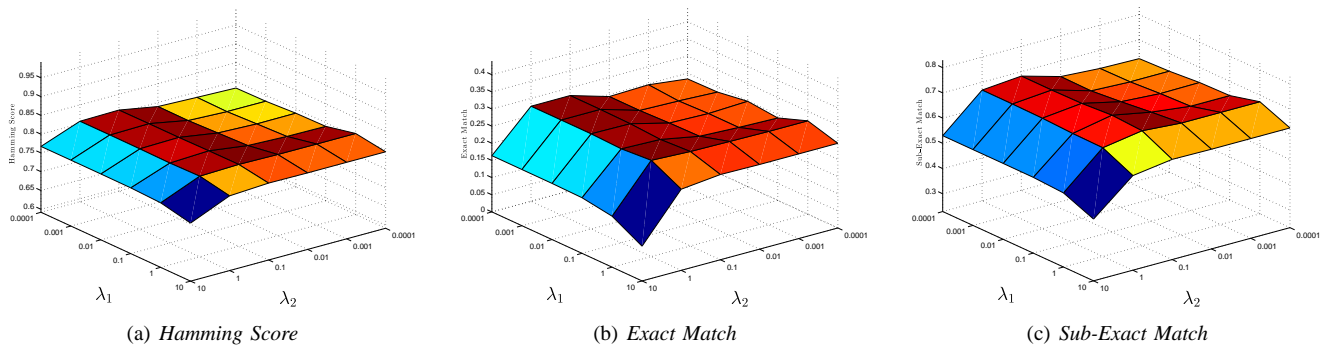


Fig. 2. Performance of M^3MDC changes as λ_1, λ_2 range in $\{10, 1, 0.1, 0.01, 0.001, 0.0001\}$ (on *Music*).

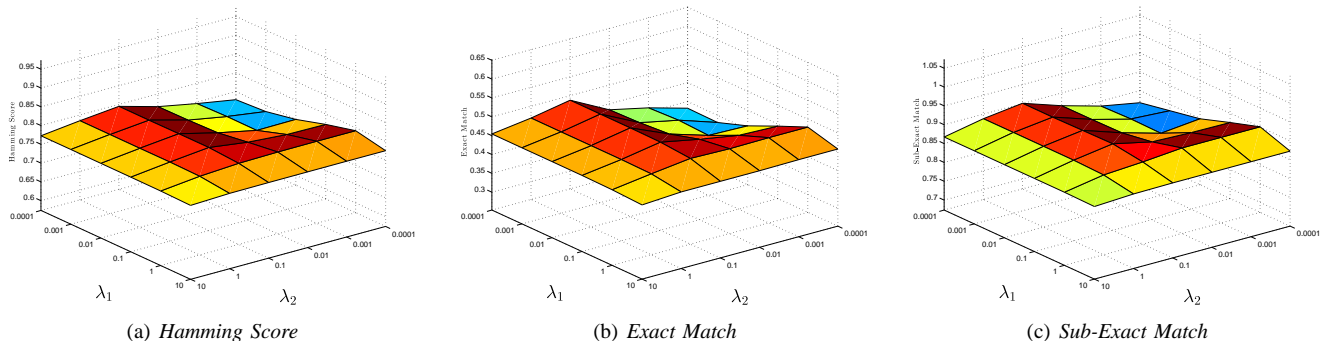


Fig. 3. Performance of M^3MDC changes as λ_1, λ_2 range in $\{10, 1, 0.1, 0.01, 0.001, 0.0001\}$ (on *Song*).

careful whether the dependencies captured by the model are ground-truth ones. As per the favorable experimental results reported in Subsection IV-B, we hypothesize that the class dependencies captured by M^3MDC are beneficial in helping generate MDC classification models with strong generalization performance.

2) *Sensitivity Analysis*: As shown in Eq.(2), M^3MDC makes a balance among empirical risk, structural risk and relationship regularization by the two trade-off parameters λ_1, λ_2 . In this subsection, we study the sensitivity of M^3MDC on how its classification performance changes under different parameter settings. Accordingly, Figures 2 and 3 illustrate the experimental results of M^3MDC on benchmark data sets *Music* and *Song* respectively. Similar results can be observed on other data sets as well. As shown in Figure 2, M^3MDC achieves relatively better performance when $\lambda_1 \leq 1$ and $\lambda_2 = 0.1$ in

terms of each evaluation metric. Therefore, the two parameters are fixed as $\lambda_1 = 0.001, \lambda_2 = 0.1$ in this paper, which can serve as the default parameter setting of M^3MDC for ease of use.

3) *Convergence Analysis*: As shown in Table I, M^3MDC solves the main optimization problem in Eq.(2) in an alternating manner. Let $\Omega(\mathbf{W}, \mathbf{C})$ be the objective function of Eq.(2), and \mathbf{W}^t and \mathbf{C}^t be the obtained values of \mathbf{W} and \mathbf{C} in the t -th iteration respectively, it is obvious that the following inequalities hold:

$$\Omega(\mathbf{W}^t, \mathbf{C}^t) \geq \Omega(\mathbf{W}^{t+1}, \mathbf{C}^t) \geq \Omega(\mathbf{W}^{t+1}, \mathbf{C}^{t+1}) \geq \dots$$

which ensures the convergence of M^3MDC . In this subsection, we further analyze the convergence rate of M^3MDC via experiments. Specifically, we record \mathbf{W} after each iteration of alternating optimization (with linear kernel) and then compute the Frobenius norm of the difference between two adjacent

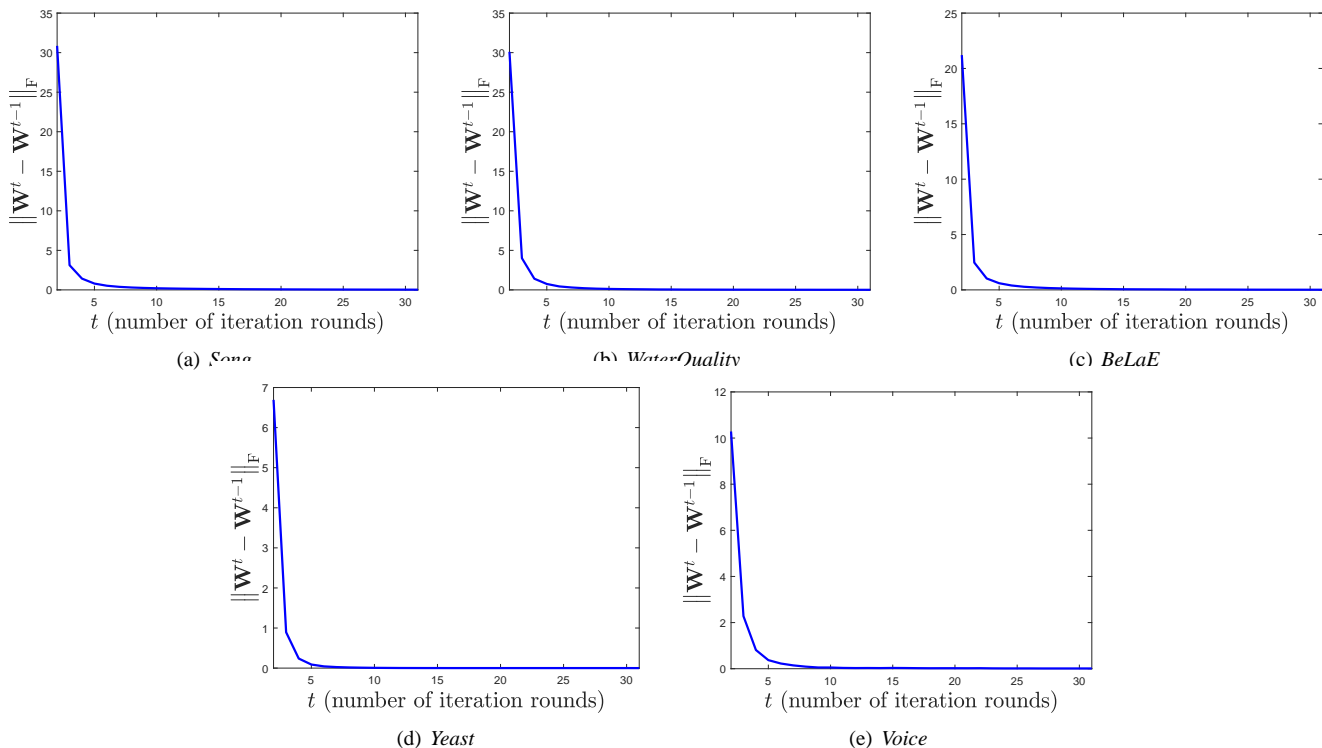


Fig. 4. Convergence curves on benchmark data sets *Song*, *WaterQuality*, *BeLaE*, *Yeast* and *Voice*.

W. Figure 4 illustrates the convergence curve of M^3MDC on benchmark data sets *Song*, *WaterQuality*, *BeLaE*, *Yeast* and *Voice*, where the vertical axis represents the resulting Frobenius norm and the horizontal axis represents the number of iterations. We can observe that the value of the Frobenius norm decreases quickly in the first few iterations, which demonstrates the fast convergence property of M^3MDC . To be more specific, for *Song*, *WaterQuality*, *BeLaE*, *Yeast* and *Voice*, the Frobenius norm $\|\mathbf{W}^t - \mathbf{W}^{t-1}\|_F$ will be less than 0.1 in 16, 11, 11, 4, 7 iteration rounds, and less than 0.05 in 23, 15, 16, 5, 10 iteration rounds, respectively.

4) *Effects of Initialization:* As shown in Table I, the procedure of M^3MDC relies on the initialization of the covariance matrix \mathbf{C} and the Lagrangian variables α . In this subsection, we perform some studies specifically to show how the proposed approach would be affected by the initialization step. Table VII reports the detailed experimental results of four different initialization strategies, including the initialization strategy of M^3MDC (denoted as Random-None), randomly initializing \mathbf{C} (denoted as Random-C), randomly initializing α (denoted as Random- α) and randomly initializing both \mathbf{C} and α (denoted as Random-Both). It is shown that different initialization strategies affect the proposed approach slightly. To be more specific, the initialization of Lagrangian variables α has little effect on the experimental results, while the initialization of covariance matrix \mathbf{C} has relatively more effect on the experimental results. When there is no prior knowledge, it is more reasonable to assume independent relationship among class spaces, i.e., initializing $\mathbf{C} = \frac{1}{m} \mathbf{I}_m$.

V. CONCLUSION

This paper extends our preliminary work [22] which focuses on designing margin-based MDC approach. Specifically, we propose a novel approach named M^3MDC which not only maximizes the margins between each pair of class labels via OvO decomposition, but also considers the class dependencies via covariance regularization. The resulting convex formulation of M^3MDC is solved with alternating optimization admitting QP or closed-form solution in either step. To validate the effectiveness of M^3MDC , extensive comparative studies over fifteen benchmark data sets have been conducted. Experimental results show that M^3MDC achieves favorable generalization performance against state-of-the-art MDC approaches.

The M^3MDC approach serves as the first attempt towards adapting margin-based techniques for learning from MDC examples. In the future, it is interesting to investigate other ways of instantiating margin-based MDC approaches, such as exploring margins based on one-vs-rest (OvR) decomposition, powerset transformation, etc.

REFERENCES

- [1] M. Benjumbeda, C. Bielza, and P. Larrañaaga, “Tractability of most probable explanations in multidimensional Bayesian network classifiers,” *International Journal of Approximate Reasoning*, vol. 93, pp. 74–87, 2018.
- [2] C. Bielza, G. Li, and P. Larrañaaga, “Multi-dimensional classification with Bayesian networks,” *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.
- [3] Å. Björck and S. Hammarling, “A Schur method for the square root of a matrix,” *Linear Algebra and its Applications*, vol. 52-53, pp. 127–140, 1983.
- [4] 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.

TABLE VII
 PREDICTIVE PERFORMANCE (MEAN \pm STD. DEVIATION) OF M³MDC (LINEAR KERNEL) WITH DIFFERENT INITIALIZATION STRATEGIES ON FIVE BENCHMARK DATA SETS. HERE, ‘RANDOM-NONE’ DENOTES INITIALIZING $\mathbf{C} = \frac{1}{n}\mathbf{I}_m$ AND $\alpha = \mathbf{0}$, ‘RANDOM-C’ AND ‘RANDOM- α ’ DENOTE RANDOMLY INITIALIZING \mathbf{C} AND α RESPECTIVELY, AND ‘RANDOM-BOTH’ DENOTES RANDOMLY INITIALIZING BOTH \mathbf{C} AND α .

(a) Hamming Score					
Initialization	Song	WaterQuality	BeLaE	Yeast	Voice
Random-None	0.796 \pm 0.028	0.646 \pm 0.012	0.454 \pm 0.021	0.802 \pm 0.006	0.970 \pm 0.008
Random-C	0.794 \pm 0.031	0.638 \pm 0.012	0.447 \pm 0.015	0.802 \pm 0.006	0.971 \pm 0.008
Random- α	0.796 \pm 0.028	0.646 \pm 0.012	0.454 \pm 0.021	0.802 \pm 0.006	0.970 \pm 0.008
Random-Both	0.795 \pm 0.031	0.638 \pm 0.012	0.441 \pm 0.022	0.802 \pm 0.006	0.970 \pm 0.008

(b) Exact Match					
Initialization	Song	WaterQuality	BeLaE	Yeast	Voice
Random-None	0.486 \pm 0.069	0.008 \pm 0.008	0.033 \pm 0.016	0.157 \pm 0.018	0.941 \pm 0.017
Random-C	0.484 \pm 0.068	0.006 \pm 0.008	0.032 \pm 0.012	0.157 \pm 0.018	0.942 \pm 0.016
Random- α	0.486 \pm 0.069	0.008 \pm 0.008	0.033 \pm 0.016	0.157 \pm 0.018	0.941 \pm 0.017
Random-Both	0.485 \pm 0.070	0.006 \pm 0.008	0.030 \pm 0.011	0.156 \pm 0.018	0.941 \pm 0.016

(c) Sub-Exact Match					
Initialization	Song	WaterQuality	BeLaE	Yeast	Voice
Random-None	0.905 \pm 0.039	0.051 \pm 0.025	0.162 \pm 0.029	0.273 \pm 0.028	0.999 \pm 0.001
Random-C	0.902 \pm 0.043	0.045 \pm 0.023	0.151 \pm 0.030	0.273 \pm 0.028	0.999 \pm 0.001
Random- α	0.905 \pm 0.039	0.051 \pm 0.025	0.162 \pm 0.029	0.273 \pm 0.028	0.999 \pm 0.001
Random-Both	0.904 \pm 0.040	0.045 \pm 0.023	0.146 \pm 0.025	0.273 \pm 0.027	0.999 \pm 0.001

- [5] H. Borchani, C. Bielza, P. Martínez-Martín, and P. Larrañaga, “Markov blanket-based approach for learning multi-dimensional Bayesian network classifiers: An application to predict the European quality of Life-5 dimensions (EQ-5D) from the 39-item Parkinsons disease questionnaire (PDQ-39),” *Journal of Biomedical Informatics*, vol. 45, no. 6, pp. 1175–1184, 2012.
- [6] H. Borchani, C. Bielza, P. Martinez-Martin, and P. Larrañaga, “Predicting the EQ-5D from the Parkinson’s disease questionnaire PDQ-8 using multi-dimensional Bayesian network classifiers,” *Biomedical Engineering: Applications, Basis and Communications*, vol. 26, no. 1, pp. 1–11, 2014.
- [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] J. Chai, I. W. Tsang, and W. Chen, “Large margin partial label machine,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2594–2608, 2020.
- [9] 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>.
- [10] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [11] 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.
- [12] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [13] C. Deng, X. Liu, C. Li, and D. Tao, “Active multi-kernel domain adaptation for hyperspectral image classification,” *Pattern Recognition*, vol. 77, pp. 306–315, 2018.
- [14] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, “Active transfer learning network: A unified deep joint spectral-spatial feature learning model for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1741–1754, 2019.
- [15] A. Elisseeff and J. Weston, “A kernel method for multi-labelled classification,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, pp. 681–687.
- [16] P. Fernandez-Gonzalez, C. Bielza, and P. Larrañaga, “Multidimensional classifiers for neuroanatomical data,” in *ICML Workshop on Statistics, Machine Learning and Neuroscience*, Lille, France, 2015.
- [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, P. Larrañaga, and C. Bielza, “Multi-dimensional Bayesian network classifier trees,” in *Proceedings of the 19th International Conference on Intelligent Data Engineering and Automated Learning*, Madrid, Spain, 2018, pp. 354–363.
- [19] C. Gong, T. Liu, J. Yang, and D. Tao, “Large-margin label-calibrated support vector machines for positive and unlabeled learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3471–3483, 2019.
- [20] O. C. Hamsici and A. M. Martinez, “Multiple ordinal regression by maximizing the sum of margins,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2072–2083, 2016.
- [21] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [22] 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, 2020, pp. 4312–4319.
- [23] B.-B. Jia and M.-L. Zhang, “Multi-dimensional classification via k NN feature augmentation,” *Pattern Recognition*, vol. 106, 2020, Article 107423.
- [24] 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.
- [25] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, “Exploiting feature and class relationships in video categorization with regularized deep neural networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 352–364, 2018.
- [26] C. Liu, P. Zhao, S.-J. Huang, Y. Jiang, and Z.-H. Zhou, “Dual set multi-label learning,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018, pp. 3635–3642.
- [27] 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.
- [28] 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, 2015, pp. 2800–2806.
- [29] 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.
- [30] Z. Ma and S. Chen, “Multi-dimensional classification via a metric approach,” *Neurocomputing*, vol. 275, pp. 1121–1131, 2018.
- [31] Z. Ma and S. Chen, “A convex formulation for multiple ordinal output classification,” *Pattern Recognition*, vol. 86, pp. 73–84, 2019.
- [32] B. Mihaljević, C. Bielza, R. Benavides-Piccione, J. DeFelipe, and P. Larrañaga, “Multi-dimensional classification of GABAergic interneurons with Bayesian network-modeled label uncertainty,” *Frontiers in Computational Neuroscience*, vol. 8, 2014, Article 150.
- [33] R. D. Monteiro and I. Adler, “Interior path following primal-dual algorithms. Part II: Convex quadratic programming,” *Mathematical Programming*, vol. 44, pp. 43–46, 1989.
- [34] A. H. A. Muktaadir, 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.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] R. Sagarna, A. Mendiburu, I. Inza, and J. A. Lozano, “Assisting in search heuristics selection through multidimensional supervised classification: A case study on software testing,” *Information Sciences*, vol. 258, pp. 122–139, 2014.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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, 2020, pp. 6178–6185.
- [45] Y. Xiao, B. Liu, and Z. Hao, “A maximum margin approach for semisupervised ordinal regression clustering,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 1003–1019, 2016.
- [46] 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.
- [47] M. Xu, Y.-F. Li, and Z.-H. Zhou, “Robust multi-label learning with PRO loss,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1610–1624, 2020.
- [48] M. Yang, C. Deng, and F. Nie, “Adaptive-weighting discriminative regression for multi-view classification,” *Pattern Recognition*, vol. 88, pp. 236–245, 2019.
- [49] F. Yu and M.-L. Zhang, “Maximum margin partial label learning,” *Machine Learning*, vol. 106, no. 4, pp. 573–593, 2017.
- [50] J. H. Zaragoza, L. E. Sucar, and E. F. Morales, “A two-step method to learn multidimensional Bayesian network classifiers based on mutual information measures,” in *Proceedings of the 34th International Florida Artificial Intelligence Research Society Conference*, Palm Beach, FL, 2011, pp. 644–649.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] Q. Zhang, B. Gu, C. Deng, and H. Huang, “Secure bilevel asynchronous vertical federated learning with backward updating,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, Virtual Event, 2021, in press.
- [55] Q. Zhang, F. Huang, C. Deng, and H. Huang, “Faster stochastic quasi-newton methods,” *IEEE Transactions on Neural Networks and Learning Systems*, 2021, in press.
- [56] Y. Zhang and D.-Y. Yeung, “A regularization approach to learning task relationships in multi-task learning,” *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 3, 2014, Article 12.
- [57] 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.