Maximum Margin Multi-Dimensional Classification

Bin-Bin Jia, Min-Ling Zhang

1School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
2College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
3Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
4Collaborative Innovation Center of Wireless Communications Technology, China

jiabb@seu.edu.cn, zhangml@seu.edu.cn

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. Due to the heterogeneity of class spaces, the major difficulty in designing margin-based MDC techniques lies in that the modeling outputs from different class spaces are not comparable to each other. In this paper, a first attempt towards maximum margin multi-dimensional classification is investigated. Following the one-vs-one decomposition within each class space, the resulting models are optimized by leveraging classification margin maximization on individual class variable and model relationship regularization across class variables. We derive convex formulation for the maximum margin MDC problem, which can be tackled with alternating optimization admitting QP or closed-form solution in either alternating step. Experimental studies over real-world MDC data sets clearly validate effectiveness of the proposed maximum margin MDC techniques.

Introduction

In multi-dimensional classification, each training example is represented by a single instance while associated with multiple class variables (Read, Bielza, and Larrañaga 2014; Ma and Chen 2018; Jia and Zhang 2019). Here, each class variable corresponds to one specific class space which characterizes the semantics of an object along one dimension. Many real-world problems can be naturally formalized under MDC frameworks (Theeramunkong and Lertnatee 2002; Rodríguez et al. 2012; Borchani et al. 2013; Sagarna et al. 2014; Serafino et al. 2015). For example, a news document can be characterized from the topic dimension (with possible classes sports, politics, social, SciTech, etc.), from the mood dimension (with possible classes good news, neutral news, bad news), and from the zone dimension (with possible classes domestic, intracontinental, etc.).

Formally speaking, let \( \mathcal{X} = \mathbb{R}^d \) be the d-dimensional input (feature) space and \( \mathcal{Y} = C_1 \times C_2 \times \cdots \times C_q \) be the output space which corresponds to the Cartesian product of q class spaces. Here, each class space \( C_j \) (1 \( j \leq q \)) consists of \( K_j \) possible class labels, i.e., \( C_j = \{ c_{j1}, c_{j2}, \ldots, c_{jK_j} \} \), among which only one is relevant to the example. Furthermore, let \( D = \{ (x_i, y_i) \mid 1 \leq i \leq N \} \) be the MDC training set with \( N \) training examples, where \( x_i \in \mathcal{X} \) is a d-dimensional feature vector and \( y_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^\top \in \mathcal{Y} \) is the associated class vector, each of which is one possible value in the corresponding class space, i.e., \( y_{ij} \in C_j \).

Then, the task of multi-dimensional classification is to induce a predictive function \( f : \mathcal{X} \to \mathcal{Y} \) from \( D \) which can assign a proper class vector \( f(x) \in \mathcal{Y} \) for the unseen instance \( x \).

To accomplish the task of learning from MDC examples, the most intuitive strategy is to induce a number of independent multi-class classifiers, one per class space. However, this strategy completely ignores potential dependencies among class variables which would impact the generalization performance of induced predictive model. Therefore, most existing approaches try to model class dependencies in different ways, such as specifying chaining order over class variables (Zaragoza et al. 2011; Read, Martino, and Luengo 2014), assuming directed acyclic graph (DAG) structure over class variables (Bielza, Li, and Larrañaga 2011; Batal, Hong, and Hauskrecht 2013; Zhu, Liu, and Jiang 2016; Bolt and van der Gaag 2017; Gil-Begue, Larrañaga, and Bielza 2018; Benjumeda, Bielza, and Larrañaga 2018), and partitioning class variables into groups (Read, Bielza, and Larrañaga 2014), etc.

To derive margin-based techniques for multi-dimensional classification, the major difficulty lies in that the modeling outputs from different class spaces are not directly comparable. In this paper, we make a first attempt to adapt maximum margin technique for multi-dimensional classification, and propose a novel approach named M^3MDC, i.e., \( \text{MaxMargin for Multi-Dimensional Classification} \). Specifically, based on one-vs-one decomposition within each class space, the multi-dimensional classification models are optimized by maximizing classification margin on individual class variable and regularizing model relationship across class variables. The resulting convex formulation is solved with alternating optimization admitting QP or closed-form solu-
The Maximum Margin MDC Approach

The common premise of margin-based approaches is that different modeling outputs are comparable. However, due to MDC’s inherent property that each class variable corresponds to one heterogeneous class space, the modeling outputs from different class spaces are not directly comparable. In this section, we present technical details of the M3MDC approach which considers the margins between each pair of class labels in the same class space.

Following the same notations given in previous section, it is easy to know that there are totally \( m = \sum_{j=1}^{K} K_j (K_j - 1) / 2 \) pairs of class labels across all class spaces. To obtain margins between each pair of class labels, one-vs-one (OvO) decomposition is made accordingly. Without loss of generality, for the \( ij \)th pair of class labels \( l_i^+ \) and \( l_j^- \), let \( D^{ij} = \{ (x_i^j, y_i^j) \mid 1 \leq j \leq n_i \} \) be the corresponding OvO decomposition training set. Here, \( x_i^j \in X \), \( y_i^j \) equals +1 when \( l_i^+ \) is relevant and -1 when \( l_j^- \) is relevant. \( n_i \) is the number of training examples in \( D \) for which either \( l_i^+ \) or \( l_j^- \) is relevant. Assuming that hyperplane \( \langle w_i, b_i \rangle \) can perfectly classify examples in \( D^{ij} \), the margin of \( \langle w_i, b_i \rangle \) can be defined as \( \frac{1}{||w_i||} \) by appropriately normalizing \( \langle w_i, b_i \rangle \) (Cortes and Vapnik 1995), where \( ||\cdot|| \) denotes the vector norm. We can get the maximum margin hyperplane by maximizing \( 2/||w_i|| \) which is equivalent to minimizing \( ||w_i||^2 / 2 \). Considering all pairs of class labels, let \( W = [w_1, \ldots, w_m] \in \mathbb{R}^{d \times m} \) and \( b = (b_1, \ldots, b_m) \), and for a more general case that training examples in \( D^{ij} \) can’t be perfectly slack, variables \( \xi = (\xi_1^1, \ldots, \xi_m^1, \ldots, \xi_1^m, \ldots, \xi_m^m)^\top \) can be introduced to model the empirical risk. Then, we can get the following maximum margin formulation for MDC:

\[
\min_{w,b,\xi} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(WW^\top) \tag{1}
\]

\[
\text{s.t. } y_j^i(\langle w_i, x_j^i \rangle + b_i) \geq 1 - \xi_j^i, \\
\quad \xi_j^i \geq 0, i = 1, \ldots, m, j = 1, \ldots, n_i
\]

where \( \langle \cdot, \cdot \rangle \) denotes inner product of two vectors and \( \lambda_1 \) is a regularization parameter. The formulation in Eq.(1) just independently deals with each pair of class labels, i.e., dependencies among class spaces are ignored. Following the idea in (Zhang and Yeung 2014; Liu et al. 2016; Jiang et al. 2018; Ma and Chen 2019), we model the relationships among all \( w_i \)s in \( W \) with the column covariance matrix of \( W \). Therefore, the above optimization problem turns out to be:

\[
\min_{w,b,\xi,C} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(WW^\top) + \frac{\lambda_2}{2} \text{tr}(WC^{-1}W^\top) \tag{2}
\]

\[
\text{s.t. } C \succeq 0, \text{tr}(C) \leq 1, \\
\quad y_j^i(\langle w_i, x_j^i \rangle + b_i) > 1 - \xi_j^i, \\
\quad \xi_j^i \geq 0, i = 1, \ldots, m, j = 1, \ldots, n_i
\]

Here, \( \text{tr} (\cdot) \) denotes the trace of a square matrix. \( C \succeq 0 \) means that \( C \) is positive semi-definite which corresponds to a covariance matrix, and \( \text{tr}(C) \leq 1 \) is used to penalize its complexity. \( \lambda_2 \) is another regularization parameter.

Obviously, the first two terms in objective function are convex with respect to \( W \) and \( b \), and it has been proved in (Zhang and Yeung 2014) that the third term in the objective function is also a convex function with respect to \( W \), \( b \) and \( C \). So the optimization problem in Eq.(2) is jointly convex.

However, it is not easy to solve this optimization problem directly because of the non-linear and non-smooth constraint \( C \succeq 0 \). Here, we use an alternating method to solve it efficiently. Specifically, the objective function with respect to \( W \) and \( b \) is firstly optimized when \( C \) is fixed, and then it is optimized with respect to \( C \) when \( W \) and \( b \) are fixed. These two steps are repeated until convergence. Technical details of the two alternating steps are introduced as follows.

Optimizing with respect to \( W \) and \( b \) when \( C \) is fixed

When \( C \) is fixed, we can reformulate the optimization problem in Eq.(2) as follows:

\[
\min_{w,b,\xi} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(WW^\top) + \frac{\lambda_2}{2} \text{tr}(WC^{-1}W^\top) \tag{3}
\]

\[
\text{s.t. } y_j^i(\langle w_i, x_j^i \rangle + b_i) > 1 - \xi_j^i, \\
\quad \xi_j^i \geq 0, i = 1, \ldots, m, j = 1, \ldots, n_i
\]

The Lagrangian of the above problem is given by:

\[
\mathcal{L}(W,b,\xi,\alpha,\beta) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \xi_j^i + \frac{\lambda_1}{2} \text{tr}(WW^\top) + \frac{\lambda_2}{2} \text{tr}(WC^{-1}W^\top) - \sum_{i=1}^{m} \sum_{j=1}^{n_i} \alpha_j^i \left[ y_j^i(\langle w_i, x_j^i \rangle + b_i) - 1 + \xi_j^i \right] - \sum_{i=1}^{m} \sum_{j=1}^{n_i} \beta_j^i \xi_j^i
\]

where \( \alpha_j^i, \beta_j^i \geq 0 \). Then, the gradients of \( \mathcal{L} \) are calculated with respect to \( W, b_i \) and \( \xi_j^i \), and we can obtain the fol-
lowing equations by setting them to be 0:

\[ \frac{\partial L}{\partial W} = 0 \Rightarrow W = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{j} y_i^{j} x_i^{j} x_i^{j \top} C (\lambda_1 I_m + \lambda_2 C)^{-1} \]  

(5)

\[ \frac{\partial L}{\partial b_i} = 0 \Rightarrow \sum_{j=1}^{n} \alpha_i^{j} y_i^{j} = 0, \; (1 \leq i \leq m) \]  

(6)

\[ \frac{\partial L}{\partial \alpha_i^{j}} = 0 \Rightarrow \alpha_i^{j} + \beta_i^{j} = 1 \]  

(7)

where \( e_i \) is the \( i \)th column vector of identity matrix \( I_m \). Plugging Eq.(5)\textendash Eq.(7) into Eq.(4), the dual problem, i.e.,

\[ \max_{\alpha} \min_{W,b} L(W,b), \]  

(4)

can be equivalently formulated as:

\[ \min \; \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{j} M(i,j) \]  

(8)

\[ \text{s.t.} \; \sum_{j=1}^{n} \alpha_i^{j} y_i^{j} = 0 \; (1 \leq i \leq m) \]  

where \( M = (\lambda_1 I_m + \lambda_2 C)^{-1} \) and \( M(i,j) \) denotes the element in \( i \)th row and \( j \)th column of \( M \). \( \alpha = (\alpha_1^{1}, \ldots, \alpha_n^{1}, \ldots, \alpha_1^{m}, \ldots, \alpha_n^{m}) \top \in \mathbb{R}^{n \times n} \). Eq.(8) is a standard quadratic programming (QP) problem with \( m \) equality constraints, but the number of \( \alpha_i^{j} \), i.e., \( \sum_{j=1}^{m} n_j \), is usually too large making this QP problem difficult to be solved directly. Here, we decompose it into \( m \) sub-QP problems with only one equality constraint as follows:

\[ \min \; \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{j} M(i,j) \]  

(9)

\[ \text{s.t.} \; \sum_{j=1}^{n} \alpha_i^{j} y_i^{j} = 0, \; 0 \leq \alpha_i^{j} \leq 1 \]  

where \( 1 \leq i \leq m, S_j^{i} = y_j^{i} \sum_{i=1}^{n} \alpha_i^{j} \), and \( \alpha = (\alpha_1^{1}, \ldots, \alpha_n^{1}) \top \in \mathbb{R}^{n \times n} \). To solve the problem in Eq.(8), we can initialize \( \alpha = 0 \), and then repeatedly solve the \( m \) sub-QP problems in Eq.(9) until all \( \alpha_i^{j} \) meet Karush-Kuhn-Tucker (KKT) conditions.

Here, the values of \( W \) and \( b \) need to be obtained for validating KKT conditions. For \( W \), it just needs to plug \( \alpha \) into Eq.(5). But for \( b \), the situation is somewhat complicated. When there are \( \alpha_i^{j} \)s in \((0, 1)\), we have \( y_j^{i} \langle (w_i, x_j^{i}) + b_i \rangle = 1 \), it’s easy to know that \( b_i = y_j^{i} \langle (w_i, x_j^{i}) \rangle - \langle w_i, x_j^{i} \rangle \). However, when there aren’t \( \alpha_i^{j} \)s in \((0, 1)\), i.e., either \( \alpha_i^{j} = 0 \) or \( \alpha_i^{j} = 1 \), we need to solve the inequalities. In this case, when \( \alpha_i^{j} = 0, b_i \) should meet \( y_j^{i} \langle (w_i, x_j^{i}) + b_i \rangle \geq 1 \), while when \( \alpha_i^{j} = 1, b_i \) should meet \( y_j^{i} \langle (w_i, x_j^{i}) + b_i \rangle \leq 1 \). Then we can get many upper and lower limits of \( b_i \), in which we select the moderate one for \( b_i \).

Optimizing with respect to \( C \) when \( W \) and \( b \) are fixed. When \( W \) and \( b \) are fixed, the optimization problem in Eq.(2) for finding \( C \) becomes:

\[ \min_{C} \; \text{tr}(C^{-1} W^{\top} W), \; \text{s.t.} \; C \geq 0, \text{tr}(C) \leq 1 \]  

(10)

As per the property of \( \text{tr}(XYZ) = \text{tr}(YXZ) \) and the constraint \( \text{tr}(C) \leq 1 \), we can lower-bound the objective in Eq.(10) as:

\[ \text{tr}(C^{-1} W^{\top} W) \geq \text{tr}(C^{-1} W^{\top} W) \; \text{tr}(C) \]  

(11)

\[ = \text{tr}(C^{-1} W^{\top} W) \; \text{tr}(C) \geq (\text{tr}(C^{-1} W^{\top} W))^{2} = (\text{tr}(A^{\top} A))^{2} \]  

where \( A = W^{\top} W = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{j} y_j^{i} y_j^{i} M_{ij}(x_i^{j}, x_j^{j}) \). The last inequality in Eq.(11) holds based on the property that both \( A \) and \( C \) are symmetric as well as the following Lemma:

**Lemma 1.** Given \( U, V \in \mathbb{R}^{ \ell \times \ell} \), then \( \text{tr}(U^{\top}U) \text{tr}(V^{\top}V) \geq (\text{tr}(U^{\top}V))^{2} \) holds. The minimum can be got when \( U = \mu \cdot V \) where \( \mu \) is a constant.

**Proof.** It is easy to know,

\[ \text{tr}(U^{\top}U) = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} U_{ij}^{2} = \langle \text{vec}U, \text{vec}U \rangle = \| \text{vec}U \|^{2} \]  

\[ \text{tr}(V^{\top}V) = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} V_{ij}^{2} = \langle \text{vec}V, \text{vec}V \rangle = \| \text{vec}V \|^{2} \]  

\[ \text{tr}(U^{\top}V) = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} U_{ij} V_{ij} = \langle \text{vec}U, \text{vec}V \rangle \]  

Here, \( \text{vec}U, \text{vec}V \) denote the results of vectorization for \( U, V \). As per the property of inner product \( \| \text{vec}U \| \cdot \| \text{vec}V \| \geq |\langle \text{vec}U, \text{vec}V \rangle| \), and take the square over both sides of this inequality, then we have \( \| \text{vec}U \|^{2} \cdot \| \text{vec}V \|^{2} \geq (\langle \text{vec}U, \text{vec}V \rangle)^{2} \). The equality relationship holds only when \( \text{vec}U = \mu \cdot \text{vec}V \), i.e., \( U = \mu \cdot V \). \( \square \)

According to Eq.(11), \( \text{tr}(C^{-1} W^{\top} W) \) attains its minimum value \( (\text{tr}(A^{\top} A))^{2} \) when \( C = 1 \) and \( A^{\top} A = \mu^{2} C^{2} \). Therefore, the closed-form solution of \( C \) can be obtained (Zhang and Yeung 2014) as follows:

\[ C = \frac{(W^{\top} W)^{\frac{1}{2}}}{\text{tr}((W^{\top} W)^{\frac{1}{2}})} \]  

(12)

As the above two alternating optimizing steps converge, we can get the optimal values of \( W, b \) and \( C \). Then, predictions
Table 1: The pseudo-code of M^3MDC.

<table>
<thead>
<tr>
<th>Inputs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{D}$: MDC training set ${(x_i, y_i) \mid 1 \leq i \leq N}$</td>
</tr>
<tr>
<td>$\lambda_1, \lambda_2$: regularization parameters</td>
</tr>
<tr>
<td>$x_i^*$: unseen instance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i^<em>$: predicted class vector for $x_i^</em>$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Transform $\mathcal{D}$ to a total of $m = \sum_{j=1}^{q}(K_j(K_j - 1))/2$ binary classification data sets via OvO decomposition w.r.t. each class space;</td>
</tr>
<tr>
<td>2: Initialize $C = \frac{1}{m}I_m$ and $\alpha = 0$;</td>
</tr>
<tr>
<td>3: repeat</td>
</tr>
<tr>
<td>4: while not all $\alpha$ meet KKT conditions do</td>
</tr>
<tr>
<td>5: for $i = 1$ to $m$ do</td>
</tr>
<tr>
<td>6: Solve sub-QP problem in Eq.(9);</td>
</tr>
<tr>
<td>7: end for</td>
</tr>
<tr>
<td>8: end while</td>
</tr>
<tr>
<td>9: Calculate $C$ according to Eq.(12);</td>
</tr>
<tr>
<td>10: until convergence</td>
</tr>
<tr>
<td>11: Calculate $m$ binary predictions $y_i^b$ for $x_i^*$ according to Eq.(13);</td>
</tr>
<tr>
<td>12: Return $y_i^*$ via OvO decoding rule based on $y_i^b$.</td>
</tr>
</tbody>
</table>

for unseen instances can be made accordingly. Specifically, for test instance $x_i^*$, we can get its binary prediction vector $y_i^b$ with a total of $m$ elements as follows:

$$y_i^b = \text{sign}(W^T x_i^* + b)$$

$$= \text{sign}(\sum_{i=1}^{m} \sum_{j=1}^{n_i} \alpha_j^i y_j^i \text{Me}(x_j^i, x_i^*) + b)$$

where $\text{sign}(\cdot)$ is the (element-wise) signed function. The first $\frac{K_1(K_1 - 1)}{2}$ elements in $y_i^b$ belong to the first class space, the $\frac{K_1(K_1 - 1)}{2} + 1 \sim \frac{K_2(K_2 - 1)}{2}$ elements belong to the second class space, and so on. Finally, we can make predictions of each class space for $x_i^*$ via OvO decoding rule based on these binary predictions.

In summary, Table 1 presents the complete procedure of the proposed M^3MDC approach. Firstly, we employ OvO decomposition for the original MDC problem per class space (Step 1). After that, an alternating optimizing process is used to solve the problem in Eq.(2) (Steps 2-10). Finally, the class vector for unseen instance is predicted based on its m binary predictions (Steps 11-12).

**Computational complexity.** Let $F_{QP}(r)$ denote the time complexity to solve Eq.(9) with $r$ variables, and $F_{SR}(s)$ denote the time complexity to solve matrix square root operation in Eq.(12) with $s \times s$ elements. M^3MDC has computational complexity $O(T_1 \cdot T_2 \cdot m \cdot F_{QP}(N) + T_1 \cdot F_{SR}(m))$ for training phase, where $T_1$ denotes the number of iterations of the whole alternating optimizing process and $T_2$ denotes the number of iterations of $m$ sub-QP problems in Eq.(9). Moreover, $F_{QP}(N)$ is actually the maximum complexity of each Eq.(9) because the number of examples in each OvO decomposition is always less than $N$.

**Related Work**

Intuitively, MDC corresponds to a set of traditional multi-class classification (MCC), one per class space. However, it is better to solve the set of MCC together rather than one by one independently, because dependencies among class variables usually exist due to the fact that all these MCC problems share the same input space. Therefore, most existing MDC approaches try to model class dependencies in different ways, such as capturing pairwise interactions between class variables (Arias et al. 2016), specifying chaining order over class variables (Zaragoza et al. 2011; Read, Martino, and Luengo 2014), assuming directed acyclic graph (DAG) structure over class variables (Bielza, Li, and Larrañaga 2011; Batal, Hong, and Hauskrecht 2013; Zhu, Liu, and Jiang 2016; Bolt and van der Gaag 2017; Gil-Begue, Larrañaga, and Bielza 2018; Benjumeda, Bielza, and Larrañaga 2018), and partitioning class variables into groups (Read, Bielza, and Larrañaga 2014, etc.).

Furthermore, MDC can also be regarded as a generalized version of multi-label classification (MLC) (Zhang and Zhou 2014; Gibaja and Ventura 2015) by not restricting binary-valued class variable in each class space. However, the key difference between MDC and MLC is whether the class space is heterogeneous or homogeneous. Generally, MDC assumes heterogeneous class spaces which characterize objects’ semantics along different dimensions, while MLC assumes homogeneous class space which characterizes the relevancy of specific concepts along one dimension. In other words, the relationship between a pair of class labels from different class spaces in MDC is different from the relationship between a pair of class labels in MLC. Therefore, it is unreasonable and will get suboptimal solutions to directly align class labels from different class spaces when trying to design MDC approaches.

Maximum margin techniques have been widely used to solve MCC and MLC problems. For MCC with single-label assignment, one can derive margin-based classification models by transforming the MCC problem into a number of binary classification problems via one-vs-one, one-vs-all, and many-vs-many decomposition, or directly maximizing multi-class margins. For MLC with multi-label assignment, one can also derive margin-based classification models via binary decomposition, or by maximizing margins between relevant-irrelevant label pairs (Elisseeff and Weston 2002), or relevant-relevant label pairs with different importance degrees (Xu, Li, and Zhou 2019), or output coding margins (Liu and Tsang 2015; Liu et al. 2019), etc.

It is worth noting that we adopt the same strategy in (Zhang and Yeung 2014; Liu et al. 2016; Ma and Chen 2019) by employing the regularization term $\text{tr}(W^{-1}W^T)$ to help induce a set of learners jointly. Nonetheless, M^3MDC

---

1MOSEK optimization software (https://www.mosek.com/) is used to solve Eq.(9), and built-in function $\text{sqrt}(\cdot)$ in Matlab is used to solve Eq.(12).
Concrete metric definitions can be given as follows: base MDC models are combined via majority voting. Sup-base classifiers is set to be 10. Furthermore, predictions of used to generate the base MDC model and the number of cut of 67% examples from the original MDC training set is among class spaces by specifying a chaining order over class spaces, conducting powerset transformation, and grouping the MDC model dependencies per class space, while ECC, ECP, ESC model dependencies training a number of independent multi-class classifiers, one per Class classifiers (ESC). BR solves MDC problem by enforcing the characteristics of all MDC data sets, including Benchmark data sets. A total of ten benchmark data sets is compared with four well-established MDC approaches (Read, Bielza, and Larrañaga 2014) including Binary Relevance (BR), Ensembles of Classifier Chains (ECC), Ensembles of Class Powerset (ECP), and Ensembles of Super Class classifiers (ESC). BR solves MDC problem by training a number of independent multi-class classifiers, one per class space, while ECC, ECP, ESC model dependencies among class spaces by specifying a chaining order over class spaces, conducting powerset transformation, and grouping the MDC class variables into super-classes respectively.

For ensemble approaches ECC, ECP and ESC, a random cut of 67% examples from the original MDC training set is used to generate the base MDC model and the number of base classifiers is set to be 10. Furthermore, predictions of base MDC models are combined via majority voting. Support vector machine (SVM) is used to instantiate BR, ECC, ECP, ESC as base classifier. Specifically, LIBSVM (Chang and Lin 2011) with linear kernel is used. As shown in Table 1, the two regularization parameters for M$^3$MDC are set to be $\lambda_1 = 0.1$, $\lambda_2 = 0.001$ respectively.

Evaluation metrics. In this paper, a total of three metrics, i.e., Hamming Score, Exact Match and Sub-Exact Match, are utilized to measure the generalization abilities of MDC approaches. Specifically, let $\mathcal{S} = \{ (x_i, y_i) \mid 1 \leq i \leq p \}$ denote the test set, where $y_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ is the ground-truth class vector associated with $x_i$. To evaluate the performance of the MDC predictive function $f$, let $\hat{y}_i = f(x_i) = [\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{in}]$ denote the predicted class vector of $x_i$, and then we can get the number of class spaces which $f$ predicts correctly, i.e., $\rho(i) = \sum_{j=1}^{n} [y_{ij} = \hat{y}_{ij}]$. Here, the predicate $[\pi]$ returns 1 if $\pi$ holds and 0 otherwise. Concrete metric definitions can be given as follows:

- **Hamming Score**: 

  $$\text{HScore}_f = \frac{1}{p} \sum_{i=1}^{p} \frac{1}{q} \cdot r(i)$$

- **Exact Match**: 

  $$\text{EMatch}_f = \frac{1}{p} \sum_{i=1}^{p} [r(i) = q]$$

- **Sub-Exact Match**: 

  $$\text{SEMatch}_f = \frac{1}{p} \sum_{i=1}^{p} [r(i) \geq q - 1]$$

In a nutshell, Hamming Score is the average accuracy over all class spaces, while Exact Match is the accuracy when considering all class spaces as a single one by conducting powerset transformation. Sub-Exact Match is a relaxed version of Exact Match where at most one incorrect prediction can be made over all class spaces for each test example. For all three metrics, the larger the values the better the performance. Ten-fold cross-validation is performed on the benchmark data sets, where the mean metric value as well as standard deviation are recorded for each comparing approach.

### Table 2: Characteristics of the experimental data sets.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>#Exam.</th>
<th>#Dim.</th>
<th>#Labels/Dim.</th>
<th>#Features$^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edm</td>
<td>154</td>
<td>2</td>
<td>3</td>
<td>16n</td>
</tr>
<tr>
<td>Flare1</td>
<td>323</td>
<td>3</td>
<td>3,4,2</td>
<td>10x</td>
</tr>
<tr>
<td>Cal500</td>
<td>502</td>
<td>10</td>
<td>2</td>
<td>68n</td>
</tr>
<tr>
<td>Music</td>
<td>591</td>
<td>6</td>
<td>2</td>
<td>71n</td>
</tr>
<tr>
<td>Song</td>
<td>785</td>
<td>3</td>
<td>3</td>
<td>98n</td>
</tr>
<tr>
<td>WQplants</td>
<td>1060</td>
<td>7</td>
<td>4</td>
<td>16n</td>
</tr>
<tr>
<td>WQanimals</td>
<td>1060</td>
<td>7</td>
<td>4</td>
<td>16n</td>
</tr>
<tr>
<td>WaterQuality</td>
<td>1060</td>
<td>14</td>
<td>4</td>
<td>16n</td>
</tr>
<tr>
<td>Yeast</td>
<td>2417</td>
<td>14</td>
<td>2</td>
<td>103n</td>
</tr>
<tr>
<td>Voice</td>
<td>3136</td>
<td>2</td>
<td>4,2</td>
<td>19n</td>
</tr>
</tbody>
</table>

$^1$ n and x denote numeric and nominal features respectively.

### Experimental Results

Table 3 reports the detailed experimental results of five comparing approaches in terms of each evaluation metric, where the best performance among all comparing approaches is shown in boldface. Moreover, Wilcoxon signed-ranks test (Demšar 2006) is used as the statistical test to show whether M$^3$MDC performs significantly better than BR, ECC, ECP, ESC respectively. Table 4 summarizes the statistical test results and the p-values for the corresponding tests are also shown in the brackets. Here, the significance level is set to be 0.05. Based on the reported experimental results, the following observations can be made:

- Across all the 30 cases (10 data sets × 3 evaluation metrics), M$^3$MDC ranks first in 21 cases, ranks second in 3 cases, and never ranks last.
- In terms of Hamming Score, M$^3$MDC is statistically better than BR, ECC, ECP, ESC.
- ECP can be regarded as an approach which works by maximizing Exact Match via class powerset transformation. It is worth noting that M$^3$MDC still ranks first in 5 out of 10 cases in term of this metric and can achieve comparable performance against ECP.
Table 3: Predictive performance of each comparing approach (mean±std. deviation) on experimental data sets. Moreover, the best performance among all comparing approaches is shown in boldface.

(a) Hamming Score

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Edm</th>
<th>Flare</th>
<th>Cal500</th>
<th>Music</th>
<th>Song</th>
<th>WQpla</th>
<th>WQani</th>
<th>WQ</th>
<th>Yeast</th>
<th>Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>M³Mdc</td>
<td>.728±.083</td>
<td>.923±.033</td>
<td>.630±.010</td>
<td>.811±.022</td>
<td>.795±.029</td>
<td>.660±.013</td>
<td>.632±.014</td>
<td>.647±.012</td>
<td>.802±.006</td>
<td>.971±.009</td>
</tr>
<tr>
<td>BR</td>
<td>.689±.070</td>
<td>.922±.034</td>
<td>.628±.011</td>
<td>.808±.023</td>
<td>.793±.023</td>
<td>.657±.016</td>
<td>.630±.014</td>
<td>.644±.013</td>
<td>.801±.006</td>
<td>.964±.007</td>
</tr>
<tr>
<td>ECC</td>
<td>.695±.065</td>
<td>.922±.034</td>
<td>.625±.015</td>
<td>.814±.025</td>
<td>.790±.024</td>
<td>.654±.016</td>
<td>.630±.014</td>
<td>.643±.013</td>
<td>.797±.007</td>
<td>.961±.008</td>
</tr>
<tr>
<td>ECP</td>
<td>.721±.082</td>
<td>.921±.036</td>
<td>.616±.015</td>
<td>.799±.032</td>
<td>.786±.029</td>
<td>.647±.015</td>
<td>.629±.013</td>
<td>.628±.015</td>
<td>.795±.007</td>
<td>.955±.013</td>
</tr>
<tr>
<td>ESC</td>
<td>.701±.079</td>
<td>.923±.033</td>
<td>.616±.019</td>
<td>.809±.022</td>
<td>.790±.030</td>
<td>.651±.016</td>
<td>.630±.014</td>
<td>.641±.013</td>
<td>.800±.006</td>
<td>.961±.008</td>
</tr>
</tbody>
</table>

(b) Exact Match

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Edm</th>
<th>Flare</th>
<th>Cal500</th>
<th>Music</th>
<th>Song</th>
<th>WQpla</th>
<th>WQani</th>
<th>WQ</th>
<th>Yeast</th>
<th>Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>M³Mdc</td>
<td>.501±.139</td>
<td>.821±.073</td>
<td>.016±.016</td>
<td>.281±.074</td>
<td>.488±.065</td>
<td>.102±.035</td>
<td>.059±.022</td>
<td>.008±.008</td>
<td>.157±.018</td>
<td>.942±.017</td>
</tr>
<tr>
<td>BR</td>
<td>.442±.125</td>
<td>.821±.073</td>
<td>.016±.016</td>
<td>.272±.075</td>
<td>.479±.059</td>
<td>.097±.033</td>
<td>.058±.022</td>
<td>.007±.008</td>
<td>.151±.017</td>
<td>.929±.014</td>
</tr>
<tr>
<td>ECC</td>
<td>.454±.123</td>
<td>.817±.078</td>
<td>.020±.016</td>
<td>.346±.079</td>
<td>.481±.057</td>
<td>.093±.037</td>
<td>.061±.023</td>
<td>.006±.008</td>
<td>.207±.014</td>
<td>.923±.016</td>
</tr>
<tr>
<td>ECP</td>
<td>.559±.136</td>
<td>.817±.078</td>
<td>.026±.028</td>
<td>.343±.076</td>
<td>.484±.054</td>
<td>.093±.028</td>
<td>.065±.018</td>
<td>.001±.003</td>
<td>.252±.012</td>
<td>.912±.025</td>
</tr>
<tr>
<td>ESC</td>
<td>.513±.122</td>
<td>.821±.073</td>
<td>.014±.013</td>
<td>.330±.069</td>
<td>.480±.067</td>
<td>.094±.038</td>
<td>.062±.021</td>
<td>.006±.008</td>
<td>.236±.019</td>
<td>.924±.016</td>
</tr>
</tbody>
</table>

(c) Sub-Exact Match

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Edm</th>
<th>Flare</th>
<th>Cal500</th>
<th>Music</th>
<th>Song</th>
<th>WQpla</th>
<th>WQani</th>
<th>WQ</th>
<th>Yeast</th>
<th>Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>M³Mdc</td>
<td>.955±.053</td>
<td>.951±.036</td>
<td>.082±.046</td>
<td>.687±.067</td>
<td>.901±.042</td>
<td>.287±.051</td>
<td>.237±.028</td>
<td>.051±.025</td>
<td>.273±.028</td>
<td>.999±.001</td>
</tr>
<tr>
<td>BR</td>
<td>.935±.061</td>
<td>.947±.039</td>
<td>.074±.037</td>
<td>.674±.067</td>
<td>.903±.033</td>
<td>.287±.055</td>
<td>.229±.034</td>
<td>.051±.024</td>
<td>.269±.029</td>
<td>.999±.002</td>
</tr>
<tr>
<td>ECC</td>
<td>.935±.069</td>
<td>.951±.036</td>
<td>.080±.031</td>
<td>.676±.064</td>
<td>.891±.036</td>
<td>.283±.049</td>
<td>.229±.032</td>
<td>.050±.023</td>
<td>.288±.023</td>
<td>.998±.002</td>
</tr>
<tr>
<td>ECP</td>
<td>.883±.074</td>
<td>.947±.039</td>
<td>.078±.036</td>
<td>.640±.064</td>
<td>.878±.040</td>
<td>.281±.049</td>
<td>.230±.032</td>
<td>.035±.018</td>
<td>.304±.020</td>
<td>.998±.003</td>
</tr>
<tr>
<td>ESC</td>
<td>.890±.076</td>
<td>.951±.036</td>
<td>.086±.038</td>
<td>.669±.062</td>
<td>.893±.038</td>
<td>.284±.050</td>
<td>.232±.033</td>
<td>.046±.022</td>
<td>.309±.028</td>
<td>.998±.002</td>
</tr>
</tbody>
</table>

Table 4: Wilcoxon signed-ranks test for M³Mdc against BR, ECC, ECP, ESC in terms of each evaluation metric (significance level $\alpha = 0.05$; $p$-values shown in the brackets).

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>M³Mdc vs BR</th>
<th>M³Mdc vs ECC</th>
<th>M³Mdc vs ECP</th>
<th>M³Mdc vs ESC</th>
</tr>
</thead>
</table>

- It is impressive to notice that M³Mdc is statistically better than BR in terms of all evaluation metrics, which clearly validates the effectiveness of M³Mdc in modeling relationships among class spaces.

**Further Analysis**

**Sensitivity analysis.** As shown in Eq.(2), $\lambda_1$, $\lambda_2$ are used to make a tradeoff among empirical risk, structural risk and relationship regularizer. Figure 1 shows how the performance of M³Mdc changes w.r.t. $\lambda_1$, $\lambda_2$ on data sets Music and Song respectively. Similar results can be obtained on other data sets which are not reported here due to page limit. In terms of each evaluation metric, M³Mdc can achieve relatively better performance when $\lambda_1 = 0.1$ and $\lambda_2 \leq 1$. In this paper, we fix $\lambda_1 = 0.1$, $\lambda_2 = 0.001$ respectively, which are also the recommended default parameter settings for ease of use.

**Correlation analysis.** By normalizing matrix $C$ in Eq.(2) with its diagonal elements, we can get correlation matrix $R$ which represents the relationships among all pairs of class labels, i.e., $R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}$, where $R_{ij}$ ($C_{ij}$) denotes element in ith row and jth column of $R$ (C). We depict the correlation matrix $R$ on data sets Song, WaterQuality and Yeast in Figure 2. Here, +1 indicates absolutely positive correlation (i.e., red color) while −1 indicates absolutely negative correlation (i.e., blue color). As shown in Figure 2, there are indeed some red or blue squares (excluding diagonal ones), which indicate that dependencies among classes do exist. However, there are many squares in green which indicate independencies between classes. These observations show that class dependencies should indeed be taken into account but with great care when designing MDC approaches. M³Mdc can model class dependencies automatically as long as dependencies exist which is a desirable property when inducing predictive models.

**Convergence analysis.** The optimization problem in Eq.(3) is solved in an alternating way. Although the objective function is jointly convex, here we also analyze its convergent characteristics. Specifically, Frobenius norm of the difference between each pair of $W$s in two adjacent it-
Figure 1: Performance of M$^3$Mdc changes as $\lambda_1, \lambda_2$ range in $\{10, 1, 0.1, 0.01, 0.001, 0.0001\}$

Figure 2: Correlation matrix on data sets Song, WaterQuality, and Yeast

Figure 3: Convergence curves on data sets Song and Voice.

In this paper, the problem of margin-based multi-dimensional classification is investigated. Specifically, a novel approach named M$^3$Mdc is proposed which considers the margin over MDC examples via OvO decomposition and models the dependencies among class spaces with covariance regularization. The resulting convex formulation is solved via alternating optimization admitting QP or closed-form solution in either alternating step. Experimental studies on benchmark data sets clearly validate the effectiveness of the derived M$^3$Mdc approach.
References


