SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2020, Vol. 63 222102:1–222102:14
https://doi.org/10.1007/s11432-019-2905-3

Multi-dimensional classification via stacked dependency exploitation

Bin-Bin JIA[1,2,3] & Min-Ling ZHANG[1,3,4*]

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

Received 14 November 2019/Revised 15 February 2020/Accepted 21 March 2020/Published online 9 November 2020

Abstract Multi-dimensional classification (MDC) aims to build classification models for multiple heterogenous class spaces simultaneously, where each class space characterizes the semantics of an object w.r.t. one specific dimension. Modeling dependencies among class spaces plays a key role in solving MDC tasks, where most approaches work by assuming directed acyclic graph (DAG) structure or random chaining structure over class spaces. Different from existing probabilistic strategies, a deterministic strategy named SEEM for dependency modeling is proposed in this paper via stacked dependency exploitation. In the first-level, pairwise dependencies are considered which can be modeled more reliably than modeling full dependencies among all class spaces by DAG or chaining structure. In the second-level, the class label of unseen instance w.r.t. each class space is determined by adaptively stacking predictive outputs from first-level pairwise classifiers. Experimental results show that stacked dependency exploitation leads to superior performance against state-of-the-art MDC approaches.

Keywords machine learning, multi-dimensional classification, class dependencies, deterministic strategy, stacked dependency exploitation

Citation Jia B-B, Zhang M-L. Multi-dimensional classification via stacked dependency exploitation. Sci China Inf Sci, 2020, 63(12): 222102, https://doi.org/10.1007/s11432-019-2905-3

1 Introduction

Multi-class classification is an important learning task in traditional supervised learning. Sometimes, however, we need to classify the same object from different dimensions. For example, when conducting demographic census, the Census Bureau needs to classify people from the `occupation` dimension (with possible classes teacher, lawyer, farmer, salesman, etc.), from the `marital-status` dimension (with possible classes unmarried, married, divorced, etc.), and from the `education` dimension (with possible classes bachelor, master, doctor, etc.). This particular problem can be naturally formalized under multi-dimensional classification framework [1–3]. Specifically, multi-dimensional classification deals with the problem where each training example is represented by a single instance while associated with multiple class variables. Here, each class variable corresponds to one specific class space which characterizes the object's semantics from one dimension. Multi-dimensional classification (MDC) techniques have been widely utilized in real-world applications involving objects with rich semantics [4–12].

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional input 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 consists

* Corresponding author (email: zhangml@seu.edu.cn)

of K_j possible class labels, i.e., $C_j = \{c_1^j, c_2^j, \ldots, c_{K_j}^j\}$. Given an MDC training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant m\}$, the task of MDC is to induce a multi-dimensional classification model $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$. For each MDC training example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^{\mathrm{T}} \in \mathcal{X}$ corresponds to a d-dimensional feature vector, and $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^{\mathrm{T}} \in \mathcal{Y}$ corresponds to the class vector associated with $\boldsymbol{x}_i$. Here, each class variable y_{ij} in $\boldsymbol{y}_i$ takes one possible value in C_j, i.e., $y_{ij} \in C_j$. For unseen instance $\boldsymbol{x}$, a class vector $f(\boldsymbol{x}) \in \mathcal{Y}$ is expected to be properly assigned by the induced MDC model.

It is widely acknowledged that modeling dependencies among class spaces is one of the core ways for designing better MDC approaches. Most existing approaches solve MDC problems by fitting probabilistic graphical model to MDC data, which can explicitly model dependencies among class spaces via assuming directed acyclic graph (DAG) structure over class spaces [13–20]. Dependencies also can be modeled by transforming MDC problem into a chain of multi-class classification problems [21,22], where chaining order over class spaces is critical but difficult to be determined. Therefore, random chaining order is usually used. Other attempt also includes partitioning class spaces into groups via a heuristic algorithm [1].

However, these existing MDC approaches aim at estimating the conditional joint probability $P(\boldsymbol{y} \mid \boldsymbol{x})$, which is challenging due to huge number of possible values of $\boldsymbol{y}$ (i.e., $\prod_{j=1}^{q} K_j$) to be modeled given limited training examples. Moreover, there is randomness in the learning process (e.g., DAG structure learning, random chaining order, heuristic partitions) of these approaches where the induced models might be unstable even based on identical training set.

Different from existing probabilistic strategies, we focus on modeling dependencies among class spaces via deterministic strategy for multi-dimensional classification. Accordingly, a novel multi-dimensional classification approach named Seem, i.e., stacked dependency exploitation for multi-dimensional classification, is proposed. Specifically, Seem models dependencies in a stacked way, where the whole process is divided into two levels. In the first-level, pairwise dependencies are considered via training a total of $\binom{q}{2}$ classifiers, one per a pair of class spaces. In the second-level, with the help of kNN techniques, the class label of unseen instance w.r.t. each class space is predicted via adaptively stacking $q-1$ predictive outputs, which correspond to the class space from first-level pairwise classifiers. Overall, Seem aims at modeling pairwise dependencies in the first-level and then considering full-order dependencies in the second-level instead of directly estimating $P(\boldsymbol{y} \mid \boldsymbol{x})$ as existing approaches do. By doing this, we expect that Seem is able to achieve better performance, and extensive experiments clearly validate the superiority of Seem over state-of-the-art MDC approaches.

The rest of this paper is organized as follows. Firstly, existing studies related to multi-dimensional classification are briefly discussed. Secondly, technical details of the proposed Seem approach are introduced. Thirdly, experimental results of comparative studies are reported. Finally, we conclude this paper.

2 Related work

In multi-dimensional classification, the output space includes multiple class spaces, where each class space consists of multiple class labels. Obviously, if we independently consider each class space one by one, then MDC can be regarded as a set of traditional multi-class classification problems. If we restrict the number of class labels in each class space to be two, then MDC will degenerate to multi-label classification (MLC) [23–25] as per the mathematical definition of MLC. However, class labels of MLC problem are generally assumed to belong to the same class space, which differs from the heterogeneous class spaces assumption made by MDC. Take news documents classification as an example, labels such as sports, politics, economics, Sci&Tech are from the `topic` class space while labels such as good news, neutral news, bad news are from the `mood` class space, where an MLC problem is unlikely to consider class labels sports and good news simultaneously which are from different class spaces.

Intuitively, MDC problems can be solved in two basic ways. The first one converts the original MDC problem into multiple multi-class classification problems by training an independent multi-class classifier for each class space, while the second one converts the original MDC problem into a single multi-class

classification problem by treating each class combination in training set as a new class. The first one does not consider potential dependencies among class spaces and may lead to suboptimal MDC model, while the second one cannot predict class combinations which do not appear in training set due to limited training examples. Therefore, modeling dependencies among class spaces appropriately plays a key role in designing good MDC learning approaches.

Because of powerful modeling capabilities of probabilistic graph model (PGM), most existing MDC approaches model class dependencies by assuming different DAG structures over class spaces. These models give rise to a family of PGMs for MDC called multi-dimensional Bayesian network classifiers [13, 15,18,20]. Nonetheless, learning and inference in PGMs are computationally demanding. Accordingly, the resulting MDC learning approaches can only deal with tasks with discrete-valued features [14, 16, 17, 19]. Following the idea of classifier chain (CC) for MLC [26], the MDC problem can be transformed into a chain of multi-class classification problems, where the feature space is augmented with predictions of preceding classifiers when training subsequent ones in the chain [21, 22]. Dependencies among class spaces are considered by the chain and selecting a better chaining order is critical. However, it is difficult to find an optimal chaining order and random ones are usually employed. By grouping class spaces as super-classes, the MDC problem can be transformed into a new MDC problem with less number of class spaces [1], where dependencies among class spaces will be modeled by super-class structure. However, the super-classes partition process is heuristic which introduces randomness to the approach. Moreover, this approach should be regarded as a meta-strategy which also needs an MDC learner to help accomplish learning.

3 The Seem approach

In this section, we present technical details of the Seem approach which models dependencies among class spaces via deterministic strategy.

Following the same notations given in previous section, let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant m\}$ be the MDC training set where $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^{\mathrm{T}} \in \mathcal{Y}$ corresponds to the class vector associated with $\boldsymbol{x}_i$. Generally speaking, dependencies among fewer class spaces can be modeled more reliably than dependencies among many class spaces due to limited examples in training set. Therefore, Seem only considers pairwise dependencies in the first-level by training a total of $\binom{q}{2}$ classifiers, one per a pair of class spaces, which are denoted as h_{rs} $(1 \leqslant r, s \leqslant q, r < s)$ respectively. Specifically, classifier h_{rs} is trained over the following data set:

$$\mathcal{D}^{rs} = \{(\boldsymbol{x}_i, \phi_{rs}(y_{ir}, y_{is})) \mid 1 \leqslant i \leqslant m\}, \tag{1}$$

i.e., $h_{rs} = \mathfrak{L}(\mathcal{D}^{rs})$, where $\mathfrak{L}$ corresponds to the employed multi-class training algorithm. Here, $\phi_{rs}(\cdot,\cdot)$ denotes some injective function from Cartesian product of C_r and C_s to natural numbers and $\phi_{rs}^{-1}(\cdot)$ is the corresponding inverse function. In other words, the set of new classes in $\mathcal{D}^{rs}$ corresponds to $\Phi(\mathcal{D}^{rs}) = \{\phi_{rs}(y_{ir}, y_{is}) \mid 1 \leqslant i \leqslant m\}$. Accordingly, for any instance $\boldsymbol{x}_i$, its class labels w.r.t. the rth and sth class space can be recovered by h_{rs} and $\phi_{rs}^{-1}(\cdot)$, i.e., $[\hat{y}_{ir}^{rs}, \hat{y}_{is}^{rs}] = \phi_{rs}^{-1}(h_{rs}(\boldsymbol{x}_i))$, where $\hat{y}_{ir}^{rs}$ $(\hat{y}_{is}^{rs})$ denotes the recovered class label in the rth (sth) class space for $\boldsymbol{x}_i$.

Example 1. Assume that the output space of $\mathcal{D}$ is $\mathcal{Y} = C_1 \times C_2 \times C_3 \times C_4$ (i.e., $q = 4$), where $C_1 = \{c_1^1, c_2^1, c_3^1\}$, $C_2 = \{c_1^2, c_2^2\}$, $C_3 = \{c_1^3, c_2^3, c_3^3, c_4^3\}$, $C_4 = \{c_1^4, c_2^4, c_3^4\}$. Then, there are a total of six different $\mathcal{D}^{rs}$, i.e., $\mathcal{D}^{12}$, $\mathcal{D}^{13}$, $\mathcal{D}^{14}$, $\mathcal{D}^{23}$, $\mathcal{D}^{24}$, $\mathcal{D}^{34}$. Take $\mathcal{D}^{12}$ as an example, the Cartesian product $C_1 \times C_2 = \{(c_1^1, c_1^2), (c_1^1, c_2^2), (c_2^1, c_1^2), (c_2^1, c_2^2), (c_3^1, c_1^2), (c_3^1, c_2^2)\}$, then $\phi_{12}(\cdot,\cdot)$ maps it into $\Phi(\mathcal{D}^{12}) = \{1, 2, 3, 4, 5, 6\}$ (i.e., $\phi_{12}(c_1^1, c_1^2) = 1$, $\phi_{12}(c_1^1, c_2^2) = 2$). Moreover, for instance $\boldsymbol{x}_i$, if $h_{12}(\boldsymbol{x}_i) = 6$, then the recovered class labels w.r.t. C_1 and C_2 are c_3^1 and c_2^2, respectively.

It is easy to know that there are $q-1$ out of $\binom{q}{2}$ pairwise classifiers which are related to one specific class space. Without loss of generality, for instance $\boldsymbol{x}_i$, its $q-1$ predictive outputs from first-level pairwise

classifiers w.r.t. the jth class space ($1 \leqslant j \leqslant q$) are denoted as $\hat{\boldsymbol{y}}_{ij}$:

$$\hat{\boldsymbol{y}}_{ij} = \left[\hat{y}_{ij}^{1j}, \ldots, \hat{y}_{ij}^{(j-1)j}, \hat{y}_{ij}^{j(j+1)}, \ldots, \hat{y}_{ij}^{jq}\right]^{\mathrm{T}}. \tag{2}$$

Here, each component in $\hat{\boldsymbol{y}}_{ij}$ corresponds to the predictive output of one pairwise classifier which considers the dependency between the jth class space and one of the other class spaces. In the second-level, these outputs will be further stacked to consider high-order dependencies among class spaces in an adaptive manner. Intuitively, the importance of one classifier might vary when it is used to classify different examples. Therefore, SEEM chooses to weight the outputs of first-level classifiers for second-level stacking based on their abilities in classifying different examples.

To prepare for the following steps, the multi-class predictive output is transformed into a binary-valued vector. Specifically, the predictive output of h_{rs} on the rth class space of $\boldsymbol{x}_i$, i.e., $\hat{y}_{ir}^{rs}$, is transformed into binary-valued vector $\boldsymbol{\delta}_{ir}^{rs}$ with length K_r as follows:

$$\boldsymbol{\delta}_{ir}^{rs} = \left[\delta_{ir}^{rs}(1), \delta_{ir}^{rs}(2), \ldots, \delta_{ir}^{rs}(K_r)\right], \tag{3}$$

where $\delta_{ir}^{rs}(a)$ equals $+1$ if $\hat{y}_{ir}^{rs} = c_a^r$, and -1 otherwise. At the same time, we identify $\boldsymbol{x}_i$'s k nearest neighbors in $\mathcal{D}$ and store them in $\mathcal{N}(\boldsymbol{x}_i)$. Let n_{ir}^{rs} and n_{is}^{rs} be the number of examples in $\mathcal{N}(\boldsymbol{x}_i)$ which are correctly predicted by h_{rs} in the rth and sth class space respectively. Then, the accuracies of h_{rs} in classifying examples in $\mathcal{N}(\boldsymbol{x}_i)$ correspond to

$$\eta_{ir}^{rs} = \frac{n_{ir}^{rs}}{k}, \quad \eta_{is}^{rs} = \frac{n_{is}^{rs}}{k}. \tag{4}$$

Conceptually, η_{ir}^{rs} and η_{is}^{rs} can be approximately regarded as the generalization ability of classifier h_{rs} w.r.t. the rth and sth class space respectively. After that, we re-scale $\boldsymbol{\delta}_{ir}^{rs}$ with its corresponding accuracy η_{ir}^{rs} to yield the following vector:

$$\boldsymbol{\zeta}_{ir}^{rs} = \eta_{ir}^{rs} \cdot \boldsymbol{\delta}_{ir}^{rs}. \tag{5}$$

Then, the multi-class predictive vector for the jth class space, i.e., $\hat{\boldsymbol{y}}_{ij}$ in Eq. (2), is transformed into a new vector $\boldsymbol{Z}_{ij}$ with length $(q-1) \cdot K_j$:

$$\boldsymbol{Z}_{ij} = \left[\boldsymbol{\zeta}_{ij}^{1j}, \ldots, \boldsymbol{\zeta}_{ij}^{(j-1)j}, \boldsymbol{\zeta}_{ij}^{j(j+1)}, \ldots, \boldsymbol{\zeta}_{ij}^{jq}\right]^{\mathrm{T}}. \tag{6}$$

After traversing all training examples in $\mathcal{D}$, we can finally get the following data set for each class space, based on which a second-level classifier g_j can be trained:

$$\mathcal{D}^j = \{(\boldsymbol{Z}_{ij}, y_{ij}) \mid 1 \leqslant i \leqslant m\} \quad (1 \leqslant j \leqslant q), \tag{7}$$

i.e., $g_j = \mathfrak{L}(\mathcal{D}^j)$. Here, g_j can adaptively model dependencies among class spaces by considering the classification abilities of first-level classifiers with kNN estimation.

Example 2. Following the setting in Example 1, for instance $\boldsymbol{x}_i$, assume its 3 predictive outputs from first-level pairwise classifiers w.r.t. C_1 are $\hat{\boldsymbol{y}}_{i1} = [c_1^1, c_3^1, c_1^1]^{\mathrm{T}}$ and the corresponding 3 accuracies are 0.8, 0.9, 0.6, then $\boldsymbol{Z}_{i1} = [+0.8, -0.8, -0.8, -0.9, -0.9, +0.9, +0.6, -0.6, -0.6]^{\mathrm{T}}$.

For unseen instance $\boldsymbol{x}_*$, its input features $\boldsymbol{Z}_{*j}$ for the jth class space of second-level classifiers can be obtained similarly according to Eqs. (2)–(6). Then, its class label w.r.t. the jth class space can be predicted by classifier g_j, i.e., $y_{*j} = g_j(\boldsymbol{Z}_{*j})$. After traversing all class spaces, finally we can obtain $\boldsymbol{x}_*$'s predicted class vector $\boldsymbol{y}_* = [y_{*1}, y_{*2}, \ldots, y_{*q}]^{\mathrm{T}}$.

In summary, Algorithm 1 presents the complete procedure of SEEM. Firstly, $\binom{q}{2}$ first-level classifiers are trained (steps 1–6). Then, training sets which will be used to train the second-level classifiers are constructed (steps 7–22). After that, the second-level classifier for each class space is induced one by one (steps 23–25). Finally, class vector for unseen instance is predicted based on the stacked classifiers (steps 26–31).

Algorithm 1 The pseudo-code of SEEM.

Input: $\mathcal{D}$: the MDC training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant m\}$;
k: the number of nearest neighbors considered;
$\mathfrak{L}$: the employed multi-class training algorithm;
$\boldsymbol{x}_*$: the unseen instance.
Output: $\boldsymbol{y}_*$: the predicted class vector for $\boldsymbol{x}_*$.

for $r = 1$ to $q - 1$ **do**
  for $s = r + 1$ to q **do**
    Construct training set $\mathcal{D}^{rs}$ according to Eq. (1);
    Train pairwise classifier h_{rs} over $\mathcal{D}^{rs}$, i.e., $h_{rs} = \mathfrak{L}(\mathcal{D}^{rs})$;
  end for
end for
Initialize $\mathcal{D}^j$ $(j = 1, 2, \ldots, q)$ as empty set;
for $i = 1$ to m **do**
  Identify k nearest neighbors of $\boldsymbol{x}_i$ in training set $\mathcal{D}$ and store them in $\mathcal{N}(\boldsymbol{x}_i)$;
  for $r = 1$ to $q - 1$ **do**
    for $s = r + 1$ to q **do**
      $[\hat{y}_{ir}^{rs}, \hat{y}_{is}^{rs}] = \phi_{rs}^{-1}(h_{rs}(\boldsymbol{x}_i))$;
      Obtain $\boldsymbol{\delta}_{ir}^{rs}$ and $\boldsymbol{\delta}_{is}^{rs}$ according to Eq. (3);
      Obtain η_{ir}^{rs} and η_{is}^{rs} according to Eq. (4);
      Obtain $\boldsymbol{\zeta}_{ir}^{rs}$ and $\boldsymbol{\zeta}_{is}^{rs}$ according to Eq. (5);
    end for
  end for
  for $j = 1$ to q **do**
    Obtain $\boldsymbol{Z}_{ij}$ according to Eq. (6);
    $\mathcal{D}^j = \mathcal{D}^j \cup (\boldsymbol{Z}_{ij}, y_{ij})$;
  end for
end for
for $j = 1$ to q **do**
  Train classifier g_j over $\mathcal{D}^j$, i.e., $g_j = \mathfrak{L}(\mathcal{D}^j)$;
end for
Identify k nearest neighbors of $\boldsymbol{x}_*$ in training set $\mathcal{D}$ and store them in $\mathcal{N}(\boldsymbol{x}_*)$;
for $j = 1$ to q **do**
  Obtain $\boldsymbol{Z}_{*j}$ according to Eqs.(3)–(6);
  $y_{*j} = g_j(\boldsymbol{Z}_{*j})$;
end for
Return $\boldsymbol{y}_* = [y_{*1}, y_{*2}, \ldots, y_{*q}]^{\mathrm{T}}$.

Generally speaking, SEEM embodies three major merits that any practically useful algorithm is desirable to have: (1) There is only one parameter (i.e., k) while most existing MDC approaches usually have many parameters to be set; (2) As our approach is a deterministic strategy for dependency modeling, SEEM has no randomness in learning process while most existing MDC approaches have; (3) As to be reported in the following experimental section, SEEM achieves highly competitive performance against state-of-the-art MDC approaches.

4 Experiments

4.1 Experimental setup

4.1.1 *Data sets*

A total of twelve benchmark MDC data sets are employed for performance evaluation[1)]. Table 1 summarizes detailed characteristics of these data sets, including number of examples (#Exam.), number of class spaces (#Dim.), number of class labels per class space (#Labels/Dim.)[2)], and number of features (#Features). Generally, it is rather costly to collect labeled MDC examples which have to be anno-

1) Here, only MDC data sets which have no less than 4 class spaces (12 pairs of class spaces) are employed.
2) If all class spaces have the same number of class labels, then only this number is recorded; Otherwise, the number of class labels in each class space is recorded in turn.

Table 1 Characteristics of the experimental data sets

Data set	#Exam.	#Dim.	#Labels/Dim.	#Features[a)]
WQplants	1060	7	4	$16n$
WQanimals	1060	7	4	$16n$
WaterQuality	1060	14	4	$16n$
Scm20d	8966	16	4	$61n$
Rf1	8987	8	4, 4, 3, 4, 4, 3, 4, 3	$64n$
Thyroid	9172	7	5, 5, 3, 2, 4, 4, 3	$7n$, $20b$, $2x$
Pain	9734	10	2, 5, 4, 2, 2, 5, 2, 5, 2, 2	$136n$
Scm1d	9803	16	4	$280n$
CoIL2000	9822	5	6, 10, 10, 4, 2	$81x$
Disfa	13095	12	5,5,6,3,4,4,5,4,4,4,6,4	$136n$
Adult	18419	4	7,7,5,2	$5n$, $5x$
Default	28779	4	2,7,4,2	$14n$,$6x$

a) n, b, and x denote numeric, binary, and nominal features, respectively.

tated from several dimensions (class spaces). To the best of our knowledge, we have employed the most comprehensive as well as largest publicly-available MDC data sets for experimental studies [1–3,15,27,28].

Specifically, `WaterQuality` (including its two divisions `WQplants` and `WQanimals`), `Scm20d`, `Rf1`, `Scm1d` are adapted from multi-target regression tasks[3)], `Thyroid`, `CoIL2000`, `Adult`, `Default` are adapted from UCI data sets[4)], and `Pain`, `Disfa` are adapted from copula ordinal regression tasks [29,30].

4.1.2 *Evaluation metrics*

Let $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant p\}$ be the test set and $f : \mathcal{X} \mapsto \mathcal{Y}$ be the induced MDC model which is to be evaluated. For each test example $\boldsymbol{x}_i$, let $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}]^{\mathrm{T}} \in \mathcal{Y}$ be its ground-truth class vector and $\hat{\boldsymbol{y}}_i = f(\boldsymbol{x}_i) = [\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iq}]^{\mathrm{T}}$ be its predicted one by f. Then, the number of class labels of $\boldsymbol{x}_i$ which are correctly predicted can be defined as $r^{(i)} = \sum_{j=1}^{q} \mathbf{1}_{y_{ij}=\hat{y}_{ij}}$, where predicate $\mathbf{1}_{\pi}$ returns 1 if π holds and 0 otherwise. Based on these notations, the following three metrics which are employed in this paper to measure the generalization abilities of MDC approaches can be given as follows:

- Hamming score:

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

- Exact match:

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

- Sub-exact match:

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

In a nutshell, hamming score measures the average accuracy of all class spaces, while exact match measures the accuracy when considering all class spaces as a single one. Sub-exact match serves as a relaxed version of exact match that allows at most one incorrectly predicted class space, because exact match might be rather low when the number of class spaces is large. For all three metrics, the larger the values the better the performance. Ten-fold cross-validation is conducted over all the benchmark data sets, where both mean metric value and standard deviation are recorded for performance comparison[5)].

3) http://mulan.sourceforge.net/datasets-mtr.html.

4) http://archive.ics.uci.edu/ml/index.php.

5) In some literatures, hamming score and exact match are also termed as class accuracy and example accuracy [1], or mean accuracy and global accuracy [15], respectively.

4.1.3 *Comparing approaches*

In this paper, we compare the performance of SEEM with five state-of-the-art MDC approaches [1, 2]:

• Binary relevance (BR). This approach learns from MDC examples by training a number of independent multi-class classifiers, one per class space. In other words, dependencies among class spaces are completely ignored by BR.

• Class powerset (CP). This approach learns from MDC examples by training a single multi-class classifier, where each distinct class combination in training set is treated as a new class. In other words, any dependencies among class spaces existing in training set are considered by CP.

• Ensembles of classifier chains (ECC). This approach learns from MDC examples by training a chain of multi-class classifiers, where the feature space is augmented with predictions of preceding classifiers when training subsequent ones in the chain. Specifically, different random chaining orders are considered to constitute an ensemble of classifier chains.

• Ensembles of super class classifiers (ESC). This approach learns from MDC examples by grouping class spaces into super-classes according to the conditional dependencies among class spaces. Specifically, random samples of training set are considered to constitute an ensemble of super-class classifiers.

• A metric learning approach for MDC (gMML). This approach learns from MDC examples by alternately learning linear regression models for each class label as well as a Mahalanobis distance metric to solve MDC problem effectively [2]. gMML is not dependent on one multi-class classifier, so the experimental results in the following subsections are identical when different multi-class classifier is employed.

In this paper, support vector machine (SVM), logistic regression (LR) and classification & regression trees (CART) are investigated as the multi-class classifier $\mathfrak{L}$ to instantiate each MDC approach except gMML[6]. Specifically, SVM is implemented by LIBSVM [31] with linear kernel, LR is implemented by LIBLINEAR [32] with L2-regularized logistic regression (primal), and CART is implemented by MATLAB built-in function `fitctree` with default parameters. For ensemble approaches ECC and ESC, a total of 10 base classifiers are used and predictive outputs from these 10 base models are combined via majority voting. For gMML, recommended parameters in [2] are used. As shown in Algorithm 1, the only parameter k (number of nearest neighbors considered) for SEEM is set to be 10.

4.2 Experimental results

Tables 2–4 report the detailed experimental results. Additionally, pairwise t-test based on ten-fold cross-validation (at 0.05 significance level) is conducted to show whether the performance of SEEM is significantly different to the five comparing MDC approaches respectively. Accordingly, Table 5 summarizes the resulting win/tie/loss counts over 12 data sets and 3 evaluation metrics.

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

• Among the 525 configurations[7] (12 data sets × 5 comparing approaches × 3 multi-class classifiers × 3 metrics), SEEM achieves superior or at least comparable performance against the five comparing approaches in 83.6% cases.

• BR solves MDC problems by independent decomposition, where dependencies among class spaces are completely ignored by this approach. As shown in Table 5, SEEM achieves superior or at least comparable performance against BR in all configurations. These results clearly suggest that class dependencies should be considered when inducing MDC models.

• ECC solves MDC problems by modeling class dependencies via specifying a random chaining order over class spaces. It is interesting to notice that SEEM also achieves superior or at least comparable performance against ECC in all configurations when SVM and LR are utilized as the multi-class classifier.

6) Similar to BR, CP, ECC and ESC, the proposed SEEM can also be regarded as a problem transformation approach, which works by transforming the MDC problem into some well-established learning problem (e.g., multi-class classification problem for SEEM). Accordingly, SEEM needs to be instantiated by employing certain multi-class classifier.

7) A total of 15 configurations are excluded where the experimental results of CP and ESC (multi-class classifier: SVM) over `Scm1d`, `Scm20d` and `Disfa` (only for CP) cannot be returned due to "out of memory" error caused by the combinatorial nature of CP and ESC.

Table 2 Experimental results (mean±std. deviation) of each comparing MDC approach (multi-class classifier: SVM). In addition, •/○ indicates whether SEEM is significantly superior/inferior to other comparing MDC approaches on each data set (pairwise t-test at 0.05 significance level). (a) Hamming score; (b) exact match; (c) sub-exact match

(a)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.666±.014	.634±.013	.647±.011	.783±.006	.962±.001	.967±.003	.962±.003	.861±.002	.955±.004	.925±.002	.714±.002	.669±.003
BR	.657±.016•	.630±.014•	.644±.013	.666±.006•	.891±.002•	.965±.002•	.953±.003•	.829±.004•	.944±.004•	.901±.002•	.710±.004•	.666±.003•
CP	.647±.015•	.629±.013•	.626±.012•	–	.928±.003•	.965±.002•	.954±.003•	–	.935±.005•	–	.707±.005•	.665±.003•
ECC	.654±.016•	.630±.014•	.643±.013•	.665±.005•	.888±.004•	.965±.002•	.952±.004•	.824±.003•	.944±.003•	.900±.002•	.710±.004•	.668±.003
ESC	.651±.017•	.631±.014•	.641±.013•	–	.919±.003•	.965±.002•	.954±.003•	–	.957±.004○	.904±.003•	.708±.004•	.667±.003•
gMML	.655±.015•	.630±.015•	.643±.013•	.600±.007•	.730±.007•	.960±.002•	.948±.004•	.697±.007•	.894±.004•	.884±.003•	.705±.004•	.666±.004•

(b)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.102±.035	.061±.023	.006±.007	.129±.011	.736±.009	.791±.016	.788±.013	.224±.009	.815±.015	.495±.013	.262±.007	.188±.006
BR	.097±.033	.058±.022	.007±.008	.065±.008•	.428±.014•	.773±.015•	.759±.015•	.175±.010•	.767±.016•	.401±.009•	.247±.009•	.179±.007•
CP	.093±.028	.063±.018	.000±.000•	–	.612±.013•	.776±.014•	.771±.016•	–	.757±.016•	–	.307±.012○	.186±.006•
ECC	.093±.037	.061±.023	.006±.008	.101±.010•	.438±.017•	.772±.014•	.761±.016•	.197±.013•	.770±.016•	.402±.010•	.260±.008	.181±.008•
ESC	.093±.037	.064±.024	.006±.008	–	.580±.011•	.771±.014•	.769±.015•	–	.832±.011○	.427±.011•	.310±.009○	.182±.008•
gMML	.092±.035	.062±.023	.006±.008	.052±.007•	.138±.011•	.741±.015•	.750±.018•	.102±.009•	.576±.015•	.379±.011•	.230±.009•	.177±.007•

(c)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.294±.044	.239±.029	.058±.024	.266±.011	.964±.005	.982±.004	.895±.010	.438±.016	.964±.006	.752±.009	.679±.007	.595±.007
BR	.287±.055	.229±.034•	.051±.024•	.131±.008•	.785±.006•	.982±.004	.863±.009•	.348±.018•	.956±.006•	.652±.012•	.669±.009•	.593±.008
CP	.281±.049	.230±.031	.034±.017•	–	.867±.012•	.981±.005	.867±.008•	–	.934±.011•	–	.637±.007•	.589±.006•
ECC	.283±.049	.229±.032•	.050±.023•	.171±.007•	.769±.010•	.981±.004	.859±.010•	.358±.014•	.956±.007•	.652±.011•	.662±.009•	.596±.007
ESC	.282±.049	.232±.032	.046±.022•	–	.842±.012•	.982±.004	.864±.008•	–	.960±.007•	.668±.013•	.638±.008•	.595±.007
gMML	.286±.053	.227±.033•	.049±.024•	.100±.009•	.375±.014•	.982±.005	.846±.010•	.198±.015•	.903±.010•	.590±.009•	.669±.008•	.593±.008

Table 3 Experimental results (mean±std. deviation) of each comparing MDC approach (multi-class classifier: LR). In addition, •/○ indicates whether SEEM is significantly superior/inferior to other comparing MDC approaches on each data set (pairwise t-test at 0.05 significance level). (a) Hamming score; (b) exact match; (c) sub-exact match

(a)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.661±.023	.635±.015	.646±.014	.749±.006	.954±.001	.967±.002	.962±.003	.835±.005	.955±.004	.919±.001	.721±.004	.672±.003
BR	.658±.014	.631±.013	.644±.011	.649±.005•	.835±.004•	.965±.002•	.953±.003•	.762±.004•	.924±.005•	.896±.002•	.721±.004	.669±.003•
CP	.649±.016•	.628±.013•	.625±.011•	.618±.010•	.885±.004•	.963±.003•	.952±.004•	.703±.009•	.936±.005•	.886±.003•	.709±.004•	.669±.004•
ECC	.654±.016	.629±.013	.642±.012	.635±.006•	.834±.003•	.964±.002•	.952±.003•	.741±.005•	.923±.004•	.893±.002•	.720±.003	.670±.003•
ESC	.653±.016	.631±.014	.642±.014	.637±.008•	.864±.005•	.963±.002•	.952±.004•	.733±.005•	.952±.003•	.890±.003•	.710±.005•	.672±.004
gMML	.655±.015	.630±.015	.643±.013	.600±.007•	.730±.007•	.960±.002•	.948±.004•	.697±.007•	.894±.004•	.884±.003•	.705±.004•	.666±.004•

(b)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.096±.034	.049±.022	.009±.006	.105±.006	.700±.008	.791±.015	.788±.013	.192±.010	.817±.014	.469±.010	.289±.010	.190±.009
BR	.092±.033	.058±.017	.005±.008	.058±.008•	.288±.015•	.769±.014•	.755±.015•	.121±.009•	.686±.018•	.396±.009•	.275±.008•	.181±.007•
CP	.093±.031	.065±.018	.000±.000•	.132±.010○	.509±.011•	.762±.014•	.760±.017•	.180±.015•	.767±.016•	.403±.012•	.317±.010○	.194±.008
ECC	.092±.034	.059±.017	.005±.008	.076±.009•	.288±.012•	.764±.013•	.757±.015•	.126±.008•	.685±.017•	.392±.011•	.287±.007	.185±.006•
ESC	.093±.036	.064±.019	.005±.008	.095±.009•	.425±.017•	.761±.013•	.758±.017•	.170±.016•	.821±.010	.393±.012•	.312±.011○	.187±.007
gMML	.092±.035	.062±.023	.006±.008	.052±.007•	.138±.011•	.741±.015•	.750±.018•	.102±.009•	.576±.015•	.379±.011•	.230±.009•	.177±.007•

(c)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.287±.042	.241±.029	.050±.025	.224±.011	.942±.005	.981±.003	.894±.011	.375±.014	.963±.007	.724±.008	.680±.006	.604±.007
BR	.286±.044	.229±.030	.047±.019	.115±.009•	.624±.011•	.983±.004	.862±.011•	.234±.015•	.937±.008•	.632±.012•	.685±.009	.601±.006•
CP	.285±.052	.232±.032	.034±.017•	.209±.011•	.758±.013•	.982±.005	.858±.010•	.284±.017•	.934±.008•	.605±.010•	.637±.007•	.594±.008•
ECC	.285±.053	.226±.026	.048±.022	.135±.007•	.632±.011•	.982±.004	.859±.010•	.230±.014•	.936±.009•	.625±.011•	.679±.008	.600±.007•
ESC	.282±.049	.231±.029	.048±.019	.162±.010•	.703±.014•	.982±.004	.858±.011•	.275±.016•	.953±.005•	.614±.011•	.644±.007•	.604±.008
gMML	.286±.053	.227±.033	.049±.024	.100±.009•	.375±.014•	.982±.005	.846±.010•	.198±.015•	.903±.010•	.590±.009•	.669±.008•	.593±.008•

These results show that the stacked dependency exploitation strategy serves as a promising dependency modeling mechanism for MDC tasks.

Table 4 Experimental results (mean±std. deviation) of each comparing MDC approach (multi-class classifier: CART). In addition, •/◦ indicates whether SEEM is significantly superior/inferior to other comparing MDC approaches on each data set (pairwise t-test at 0.05 significance level). (a) Hamming score; (b) exact match; (c) sub-exact match

(a)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.569±.019	.554±.020	.557±.020	.826±.003	.978±.002	.991±.001	.950±.003	.851±.004	.946±.003	.901±.004	.672±.007	.619±.003
BR	.561±.026	.561±.014	.561±.015	.769±.006•	.975±.002•	.990±.001•	.940±.003•	.814±.003•	.946±.004	.888±.003•	.669±.006	.592±.004•
CP	.586±.024◦	.556±.018	.559±.015	.663±.013•	.971±.002•	.990±.002•	.938±.006•	.706±.009•	.921±.005•	.882±.004•	.663±.006•	.605±.004•
ECC	.634±.026◦	.622±.023◦	.634±.017◦	.831±.004◦	.980±.001◦	.991±.002	.960±.004◦	.864±.003◦	.957±.004◦	.922±.002◦	.711±.003◦	.649±.004◦
ESC	.645±.015◦	.630±.022◦	.641±.012◦	.793±.006•	.982±.001◦	.991±.001	.959±.003◦	.835±.005•	.956±.004◦	.918±.003◦	.707±.005◦	.649±.004◦
gMML	.655±.015◦	.630±.015◦	.643±.013◦	.600±.007•	.730±.007•	.960±.002•	.948±.004	.697±.007•	.894±.004•	.884±.003•	.705±.004◦	.666±.004◦

(b)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.039±.021	.033±.020	.002±.004	.139±.005	.835±.009	.947±.007	.700±.012	.194±.016	.781±.011	.416±.019	.221±.009	.137±.005
BR	.033±.023	.029±.011	.003±.005	.062±.008•	.816±.014•	.941±.007•	.636±.018•	.120±.008•	.781±.014	.352±.014•	.212±.007•	.118±.005•
CP	.048±.028	.025±.016	.000±.000	.156±.007◦	.826±.008•	.946±.012	.705±.018	.175±.014•	.725±.013•	.411±.011	.244±.012◦	.132±.004
ECC	.069±.033◦	.046±.023◦	.005±.005	.193±.013◦	.851±.006◦	.950±.009	.782±.018◦	.233±.012◦	.819±.014◦	.494±.009◦	.293±.009◦	.172±.008◦
ESC	.075±.026◦	.050±.026◦	.005±.007	.200±.016◦	.859±.008◦	.951±.007	.787±.017◦	.230±.012◦	.818±.016◦	.496±.012◦	.288±.009◦	.169±.006◦
gMML	.092±.035◦	.062±.023◦	.006±.008	.052±.007•	.138±.011•	.741±.015•	.750±.018◦	.102±.009•	.576±.015•	.379±.011•	.230±.009◦	.177±.007◦

(c)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	.146±.026	.140±.030	.019±.015	.326±.012	.990±.004	.991±.004	.872±.012	.406±.014	.952±.004	.675±.013	.608±.013	.503±.006
BR	.141±.042	.119±.025•	.012±.012	.178±.010•	.986±.004•	.991±.003	.851±.012•	.280±.012•	.952±.007	.630±.010•	.605±.012	.460±.006•
CP	.177±.052	.133±.017	.022±.017	.279±.014•	.970±.005•	.983±.004•	.824±.018•	.299±.017•	.899±.009•	.616±.016•	.573±.009•	.480±.008•
ECC	.224±.044◦	.213±.055◦	.041±.020◦	.375±.008◦	.991±.003	.991±.003	.888±.011◦	.444±.013◦	.966±.007◦	.743±.007◦	.656±.007◦	.559±.003◦
ESC	.262±.038◦	.225±.045◦	.044±.026◦	.364±.014◦	.993±.002◦	.988±.004	.884±.008◦	.417±.011◦	.964±.007◦	.727±.011◦	.645±.011◦	.561±.011◦
gMML	.286±.053◦	.227±.033◦	.049±.024◦	.100±.009•	.375±.014•	.982±.005•	.846±.010•	.198±.015•	.903±.010•	.590±.009•	.669±.008◦	.593±.008◦

Table 5 Win/tie/loss counts of pairwise t-test (at 0.05 significance level) between SEEM and each MDC approach in terms of hamming score (HScore), exact match (EMatch), and sub-exact match (SEMatch)

SEEM	Multi-class classifier: SVM			Multi-class classifier: LR			Multi-class classifier: CART			In total
against	HScore	EMatch	SEMatch	HScore	EMatch	SEMatch	HScore	EMatch	SEMatch	
BR	11/1/0	9/3/0	9/3/0	8/4/0	9/3/0	7/5/0	7/5/0	8/4/0	7/5/0	**75/33/0**
CP	9/0/0	6/2/1	6/3/0	12/0/0	7/3/2	9/3/0	9/2/1	3/7/2	9/3/0	**70/23/6**
ECC	11/1/0	8/4/0	9/3/0	8/4/0	8/4/0	7/5/0	0/1/11	0/2/10	0/2/10	**51/26/31**
ESC	9/0/1	5/3/2	6/4/0	8/4/0	6/5/1	7/5/0	2/1/9	0/2/10	0/1/11	**43/25/34**
gMML	12/0/0	9/3/0	9/3/0	9/3/0	9/3/0	8/4/0	6/1/5	6/1/5	7/0/5	**75/18/15**
In total	**52/2/1**	**37/15/3**	**39/16/0**	**45/15/0**	**39/18/3**	**38/22/0**	**24/10/26**	**17/16/27**	**23/11/26**	**314/125/86**

• As shown in Table 5, when SVM and LR are utilized as the multi-class classifier, SEEM significantly outperforms CP in 49 out of 63 configurations and outperforms ESC in 41 out of 66 configurations respectively. Among the 7 configurations that SEEM is inferior to CP or ESC, 6 of them are in terms of exact match. This might be attributed to the class space powerset transformation made by CP and ESC, which makes them suitable for maximizing the exact match metric (though computationally demanding).

• When CART is utilized as the multi-class classifier, the prominent advantage of SEEM over BR and CP (with only 3 loss cases with CP) clearly validate the effectiveness of SEEM, while there are 61 out of 72 configurations where SEEM is inferior to ECC and ESC. This might be due to that CART is not a suitable classifier for adaptive stacking in the second-level, which will be further analyzed in Subsection 4.3.1.

• Note that there are less significant differences between the baselines and SEEM for exact match and sub-exact match over `WaterQuality`. For this data set, there are 989 distinct class combinations appearing within the 1060 examples where the number of examples w.r.t. one specific class combinations is rather small. Specifically, there are 942 class combinations each with only one example appearing in the data set. This might lead to the fact that the value of exact match is very low for all MDC approaches, and thus it is hard to achieve statistically superior performance for one approach than other approaches based on pairwise t-test. Sub-exact match is a relaxed version of exact match, and `WQplants`

and `WQanimals` are two divisions of `WaterQuality` [33], so similar observations can be made on these cases.

4.3 Further analysis

4.3.1 *Effectiveness of* Seem*'s design*

We also compare the performance of Seem with its three simplified versions to verify the effectiveness of Seem's design. The three variants are denoted as Pair, Vote, and Stack, respectively.

• Pair. This variant simply partitions all the q class spaces into $\lfloor \frac{q}{2} \rfloor$ pairs (and a single one when q is odd) according to the value of Cramér's V [34] between each pair of class spaces. Specifically, the pair of class spaces which has the largest Cramér's V is selected out as a pair, and then the same process is done in the remaining class spaces repeatedly until all class spaces have been selected. For each pair of class spaces, a data set is formed according to Eq. (1) and then a multi-class classifier is trained over it.

• Vote. This variant simply makes majority voting based on Eq. (2) instead of training a second-level classifier for each class space.

• Stack. This variant simply substitutes the following vector $\boldsymbol{\Delta}_{ij}$:

$$\boldsymbol{\Delta}_{ij} = \left[\boldsymbol{\delta}_{ij}^{1j}, \ldots, \boldsymbol{\delta}_{ij}^{(j-1)j}, \boldsymbol{\delta}_{ij}^{j(j+1)}, \ldots, \boldsymbol{\delta}_{ij}^{jq}\right]^{\mathrm{T}}$$

for $\boldsymbol{Z}_{ij}$ in Eq. (7), which will be employed as input features by the second-level classifiers.

Detailed experimental results are shown in Figure 1. Besides, to show overall statistical relationships among Pair, Vote, Stack, and Seem over all data sets, Wilcoxon signed-ranks test [35] is employed to serve this purpose. Table 6 summarizes the statistical test results where the p-values for the corresponding tests are also shown in the brackets.

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

• It is no doubt that Pair achieves the worst performance no matter which multi-class classifier is used, which validates the effectiveness of our two levels dependency modeling strategy.

• Stack can achieve superior performance against Vote in terms of hamming score when SVM is utilized as the multi-class classifier and all evaluation metrics when LR is utilized as the multi-class classifier. Furthermore, when SVM and LR are utilized as the multi-class classifier, Seem can achieve superior performance against Vote in terms of all evaluation metrics, and Stack over hamming score and sub-exact match. These results suggest that it is effective to improve classification performance by adaptively stacking predictive outputs from first-level pairwise classifiers as Seem does.

• When CART is utilized as the multi-class classifier, Vote achieves superior performance against Stack and Seem which might lie in the fact that CART is more suitable for handling discrete features, while the features used in the second-level of Seem are numeric ones (i.e., the vector in Eq. (6)). Therefore, multi-class classifiers like SVM, LR are more recommended for Seem.

4.3.2 *Sensitivity analysis*

As shown in Algorithm 1, the only parameter k, i.e., the number of nearest neighbors considered, is used by Seem for adaptive stacking. Figure 2 illustrates how the performance of Seem (with SVM as multi-class classifier) changes as the value of k increases from 5 to 15. In terms of each evaluation metric, it is shown that the performance of Seem is relatively stable with varying values of k. Insensitivity w.r.t. the only parameter serves as a desirable property. In this paper, the value of k is moderately fixed to be 10.

4.3.3 *Computational complexity*

Given a multi-class classification algorithm $\mathfrak{L}$, let $\mathcal{F}(m, d, N)$ and $\mathcal{F}'(m, d, N)$ be the training and testing complexity of $\mathfrak{L}$, where m, d, N corresponds to the number of examples, number of features and number of class labels, respectively. Furthermore, the complexity of identifying k nearest neighbors is $\mathcal{O}(m(d + \ln(m)))$. Then, the time complexity of the proposed Seem approach corresponds to $\mathcal{O}(q(q\mathcal{F}(m, d, K^2) +$

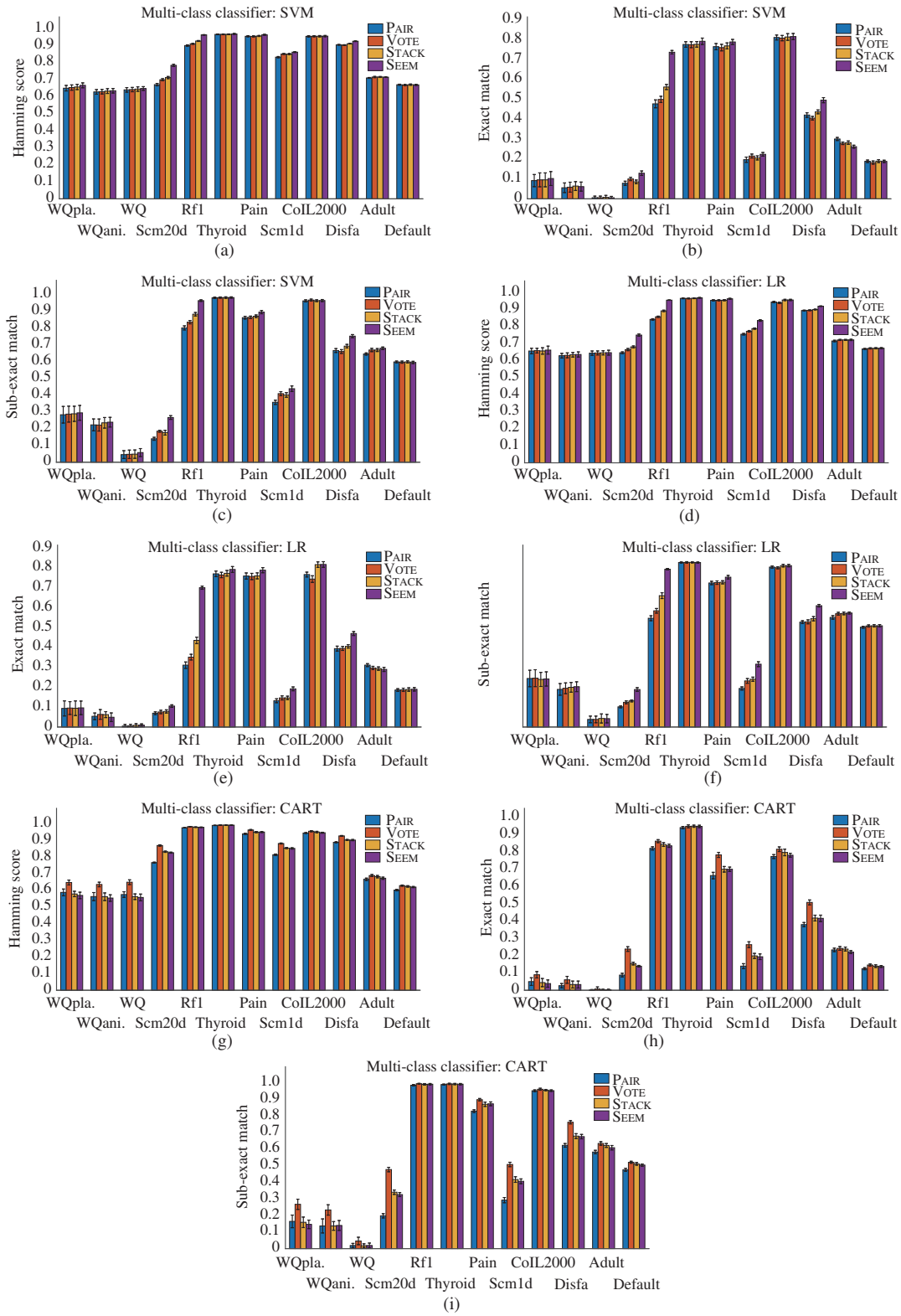

Figure 1 (Color online) Performance comparison of PAIR, VOTE, STACK, and SEEM. (a) Hamming score (SVM); (b) exact match (SVM); (c) sub-exact match (SVM); (d) hamming score (LR); (e) exact match (LR); (f) sub-exact match (LR); (g) hamming score (CART); (h) exact match (CART); (i) sub-exact match (CART).

$\mathcal{F}(m,(q-1)K,K))+m^2(d+\ln(m)+q^2\mathcal{F}'(m,d,K^2)))$. Here, q represents the number of class spaces (dimensions) and K represents the maximum number of class labels in each class space. Conceptually,

Table 6 Wilcoxon signed-ranks test among PAIR, VOTE, STACK, and SEEM in terms of hamming score (HScore), exact match (EMatch), and sub-exact match (SEMatch) (significance level $\alpha = 0.05$; p-values shown in the brackets)

Multi-class classifier	Evaluation metric	VOTE against PAIR	STACK against PAIR	STACK against VOTE	SEEM against PAIR	SEEM against VOTE	SEEM against STACK
SVM	HScore	**tie**[6.40e−2]	**win**[4.88e−4]	**win**[3.42e−3]	**win**[9.77e−4]	**win**[9.77e−4]	**win**[4.88e−3]
	EMatch	**tie**[8.98e−1]	**win**[2.69e−2]	**tie**[1.75e−1]	**win**[3.22e−2]	**win**[1.86e−2]	**tie**[7.71e−2]
	SEMatch	**win**[3.42e−2]	**win**[4.88e−4]	**tie**[3.39e−1]	**win**[2.44e−3]	**win**[4.88e−3]	**win**[2.44e−3]
LR	HScore	**win**[2.00e−2]	**win**[4.88e−4]	**win**[6.84e−3]	**win**[4.88e−4]	**win**[4.88e−4]	**win**[4.88e−4]
	EMatch	**tie**[5.69e−1]	**win**[2.10e−2]	**win**[4.00e−2]	**win**[1.27e−2]	**win**[1.61e−2]	**win**[2.69e−2]
	SEMatch	**win**[9.77e−3]	**win**[2.93e−3]	**win**[1.12e−2]	**win**[2.44e−3]	**win**[7.32e−3]	**win**[9.28e−3]
CART	HScore	**win**[4.88e−4]	**win**[4.25e−2]	**loss**[4.88e−4]	**tie**[2.04e−1]	**loss**[4.88e−4]	**loss**[1.46e−3]
	EMatch	**win**[4.88e−4]	**win**[2.44e−3]	**loss**[9.77e−4]	**win**[2.69e−2]	**loss**[4.88e−4]	**loss**[3.91e−3]
	SEMatch	**win**[4.88e−4]	**win**[1.37e−2]	**loss**[4.88e−4]	**win**[9.77e−3]	**loss**[4.88e−4]	**tie**[5.22e−2]

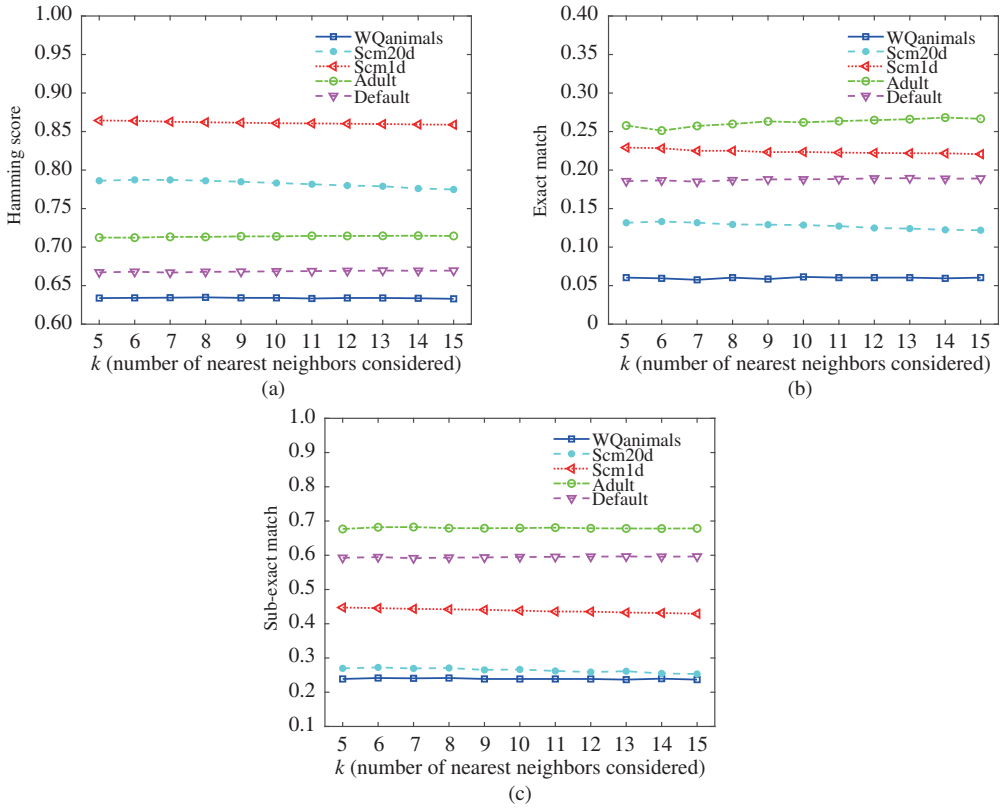

Figure 2 (Color online) Performance of SEEM changes as k ranges from 5 to 15 in terms of each evaluation metric. (a) Hamming score; (b) exact match; (c) sub-exact match.

each class space in MDC problem corresponds to one heterogeneous semantic space. Therefore, it is generally impractical to assume too many semantic spaces and the number of class spaces handled by MDC is at moderate size. Table 7 shows the time costs of SEEM and all comparing approaches over each data set. Among BR, CP, ECC, ESC and SEEM who are dependent on certain multi-class classifier, BR is undoubtedly the most efficient approach while ESC is usually the most computationally demanding one. SEEM usually consumes less execution time than ESC and comparable execution time to CP (ECC) when LR (CART) is utilized as the multi-class classifier.

5 Conclusion

In this paper, a novel MDC approach named SEEM is proposed by focusing on deterministic strategy

Table 7 The time costs (unit: s) of SEEM and all comparing approaches over each data set. Multi-class classifier: (a) SVM; (b) LR; (c) CART

(a)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	31	27	135	24299	2241	548	5896	56719	10607	35685	4677	15694
BR	3	3	5	1254	262	27	372	2986	1173	1545	480	2486
CP	10	12	33	–	74	28	432	–	777	–	348	1179
ECC	39	46	168	5658	1251	176	1703	12474	7017	8077	2236	10477
ESC	202	746	7557	–	4742	622	8702	–	23463	37242	7824	36479
gMML	6	6	9	105	44	44	48	118	69	115	96	110

(b)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	8	7	20	2630	634	138	1515	10328	855	4400	326	834
BR	1	1	1	68	28	6	61	468	66	160	11	21
CP	8	9	16	4750	295	6	338	19917	896	3649	203	229
ECC	5	5	10	658	279	58	490	2298	377	1001	112	165
ESC	33	36	61	9684	1148	74	1650	51570	2861	10784	661	263
gMML	6	6	9	105	44	44	48	118	69	115	96	110

(c)

Algo.	WQpla.	WQani.	WQ	Scm20d	Rf1	Thyroid	Pain	Scm1d	CoIL2000	Disfa	Adult	Default
SEEM	16	16	58	2051	238	137	1173	11288	429	4288	378	967
BR	4	4	8	244	40	7	212	1386	60	553	31	119
CP	6	8	12	4773	37	4	224	34750	2509	3014	230	122
ECC	32	32	72	2175	359	66	1322	9481	636	3415	286	1038
ESC	65	70	150	35591	438	101	2054	120133	2820	10997	1180	1191
gMML	6	6	9	105	44	44	48	118	69	115	96	110

while most existing approaches focus on probabilistic strategy. Specifically, SEEM works in a stacked way, where pairwise dependencies are considered in the first level, and high-order dependencies are further considered by adaptively stacking the predictive outputs from first-level pairwise classifiers. The effectiveness of SEEM is thoroughly validated via comprehensive experiments on ten real-world MDC data sets.

SEEM needs to train a total of $\binom{q}{2}$ pairwise classifiers which leads to quadratic (i.e., $\mathcal{O}(q^2)$) computational complexity. On the other hand, better generalization performance is expected to be achieved in the second-level with more predictive outputs from first-level classifiers. In the future, for MDC tasks with large number of class spaces, a compromising solution to computation and performance is to just select part of the class space pairs according to some well-designed criteria or the exploitation of domain knowledge [36].

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2018YFB1004300), China University S&T Innovation Plan Guided by the Ministry of Education, and partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. The authors wish to thank the associate editor and anonymous reviewers for their helpful comments and suggestions.

References

1 Read J, Bielza C, Larranaga P. Multi-dimensional classification with super-classes. IEEE Trans Knowl Data Eng, 2014, 26: 1720–1733
2 Ma Z C, Chen S C. Multi-dimensional classification via a metric approach. Neurocomputing, 2018, 275: 1121–1131
3 Jia B B, Zhang M L. Multi-dimensional classification via KNN feature augmentation. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3975–3982
4 Theeramunkong T, Lertnattee V. Multi-dimensional text classification. In: Proceedings of the 19th International Conference on Computational Linguistics, 2002

5 Shatkay H, Pan F X, Rzhetsky A, et al. Multi-dimensional classification of biomedical text: toward automated, practical provision of high-utility text to diverse users. Bioinformatics, 2008, 24: 2086–2093
6 Rodriguez J D, Perez A, Arteta D, et al. Using multidimensional bayesian network classifiers to assist the treatment of multiple sclerosis. IEEE Trans Syst Man Cybern C, 2012, 42: 1705–1715
7 Borchani H, Bielza C, Toro C, et al. Predicting human immunodeficiency virus inhibitors using multi-dimensional Bayesian network classifiers. Artif Intell Med, 2013, 57: 219–229
8 Borchani H, Bielza C, Martinez-Martin P, et al. Predicting the EQ-5D from the Parkinson's disease questionnaire PDQ-8 using multi-dimensional bayesian network classifiers. Biomed Eng Appl Basis Commun, 2014, 26: 1450015
9 Mihaljevic B, Bielza C, Benavides-Piccione R, et al. Multi-dimensional classification of GABAergic interneurons with Bayesian network-modeled label uncertainty. Front Comput Neurosci, 2014, 8: 150
10 Sagarna R, Mendiburu A, Inza I, et al. Assisting in search heuristics selection through multidimensional supervised classification: a case study on software testing. Inf Sci, 2014, 258: 122–139
11 Fernandez-Gonzalez P, Bielza C, Larrañaga P. Multidimensional classifiers for neuroanatomical data. In: Proceedings of ICML Workshop on Statistics, Machine Learning and Neuroscience, 2015
12 Muktadir A H A, Miyazawa T, Martinez-julia P, et al. Multi-target classification based automatic virtual resource allocation scheme. IEICE Trans Inf Syst, 2019, 102: 898–909
13 van der Gaag L C, de Waal P R. Multi-dimensional Bayesian network classifiers. In: Proceedings of the 3rd European Workshop in Probabilistic Graphical Models, 2006. 107–114
14 Rodríguez J D, Lozano J A. Multi-objective learning of multi-dimensional Bayesian classifiers. In: Proceedings of the 8th International Conference Hybrid Intelligent Systems, Barcelona, 2008. 501–506
15 Bielza C, Li G D, Larrañaga P. Multi-dimensional classification with Bayesian networks. Int J Approx Reason, 2011, 52: 705–727
16 Batal I, Hong C, Hauskrecht M. An efficient probabilistic framework for multi-dimensional classification. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, San Francisco, 2013. 2417–2422
17 Zhu M M, Liu S Y, Jiang J W. A hybrid method for learning multi-dimensional Bayesian network classifiers based on an optimization model. Appl Intell, 2016, 44: 123–148
18 Bolt J H, van der Gaag L C. Balanced sensitivity functions for tuning multi-dimensional Bayesian network classifiers. Int J Approx Reason, 2017, 80: 361–376
19 Benjumeda M, Bielza C, Larrañaga P. Tractability of most probable explanations in multidimensional Bayesian network classifiers. Int J Approx Reason, 2018, 93: 74–87
20 Gil-Begue S, Larrañaga P, Bielza C. Multi-dimensional Bayesian network classifier trees. In: Proceedings of International Conference on Intelligent Data Engineering and Automated Learning, 2018. 354–363
21 Zaragoza J H, Sucar L E, Morales E F, et al. Bayesian chain classifiers for multidimensional classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, 2011. 2192–2197
22 Read J, Martino L, Luengo D. Efficient monte carlo methods for multi-dimensional learning with classifier chains. Pattern Recogn, 2014, 47: 1535–1546
23 Zhang M L, Zhou Z H. A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng, 2014, 26: 1819–1837
24 Gibaja E, Ventura S. A tutorial on multilabel learning. ACM Comput Surv, 2015, 47: 1–38
25 Zhang M L, Li Y K, Liu X Y, et al. Binary relevance for multi-label learning: an overview. Front Comput Sci, 2018, 12: 191–202
26 Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification. Mach Learn, 2011, 85: 333–359
27 Jia B B, Zhang M L. Maximum margin multi-dimensional classification. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 4312–4319
28 Wang H, Chen C, Liu W, et al. Incorporating label embedding and feature augmentation for multi-dimensional classification. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 6178–6185
29 Walecki R, Rudovic O, Pavlovic V, et al. Copula ordinal regression for joint estimation of facial action unit intensity. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 4902–4910
30 Ma Z C, Chen S C. A convex formulation for multiple ordinal output classification. Pattern Recogn, 2019, 86: 73–84
31 Chang C C, Lin C J. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol, 2011, 2: 1–27
32 Fan R E, Chang K W, Hsieh C, et al. LIBLINEAR: a library for large linear classification. J Mach Learn Res, 2008, 9: 1871–1874
33 Kocev D, Vens C, Struyf J, et al. Ensembles of multi-objective decision trees. In: Proceedings of the 18th European Conference on Machine Learning, Warsaw, 2007. 624–631
34 Cramér H. Mathematical Methods of Statistics. Princeton: Princeton University Press, 1999
35 Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res, 2006, 7: 1–30
36 Zhou Z H. Abductive learning: towards bridging machine learning and logical reasoning. Sci China Inf Sci, 2019, 62: 076101