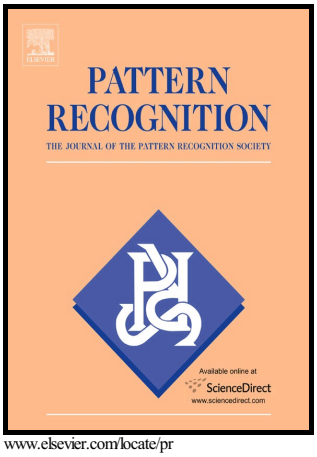
Author's Accepted Manuscript

Multi-instance dictionary learning via multivariate performance measure optimization

Jim Jing-Yan Wang, Ivor Wai-Hung Tsang, Xuefeng Cui, Zhiwu Lu, Xin Gao



www.elsevier.com/locate/pr

PII: S0031-3203(16)30443-5
DOI: http://dx.doi.org/10.1016/j.patcog.2016.12.023
Reference: PR5995

To appear in: *Pattern Recognition*

Received date: 31 May 2016
Revised date: 3 December 2016
Accepted date: 21 December 2016

Cite this article as: Jim Jing-Yan Wang, Ivor Wai-Hung Tsang, Xuefeng Cui, Zhiwu Lu and Xin Gao, Multi-instance dictionary learning via multivariat performance measure optimization, *Pattern Recognitior* http://dx.doi.org/10.1016/j.patcog.2016.12.023

This is a PDF file of an unedited manuscript that has been accepted fo publication. As a service to our customers we are providing this early version o the manuscript. The manuscript will undergo copyediting, typesetting, an review of the resulting galley proof before it is published in its final citable form Please note that during the production process errors may be discovered whic could affect the content, and all legal disclaimers that apply to the journal pertain

ACCEPTED MANUSCRIPT

Multi-instance dictionary learning via multivariate performance measure optimization

Jim Jing-Yan Wang[a], Ivor Wai-Hung Tsang[b], Xuefeng Cui[a], Zhiwu Lu[c], Xin Gao[a,*]

[a] *King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE), Computational Bioscience Research Center (CBRC), Thuwal 23955, Saudi Arabia*
[b] *Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Australia*
[c] *Beijing Key Laboratory of Big Data Management and Analysis Methods, School of Information, Renmin University of China, Beijing, 100872, China*

Abstract

The multi-instance dictionary plays a critical role in multi-instance data representation. Meanwhile, different multi-instance learning applications are evaluated by specific multivariate performance measures. For example, multi-instance ranking reports the precision and recall. It is not difficult to see that to obtain different optimal performance measures, different dictionaries are needed. This observation motives us to learn performance-optimal dictionaries for this problem. In this paper, we propose a novel joint framework for learning the multi-instance dictionary and the classifier to optimize a given multivariate performance measure, such as the F_1 score and precision at rank k. We propose to represent the bags as bag-level features via the bag-instance similarity, and learn a classifier in the bag-level feature space to optimize the given performance measure. We propose to minimize the upper bound of a multivariate loss corresponding to the performance measure, the complexity of the classifier, and the complexity of the dictionary, simultaneously, with regard to both the dictionary and the classifier parameters. In this way, the dictionary learning is regularized by the performance optimization, and a performance-optimal dictionary is ob-

Accepted manuscript

*Corresponding author: Xin Gao. E-mail: xin.gao@kaust.edu.sa. Tel: +966-12-8080323.
Email addresses: jimjywang@gmail.com (Jim Jing-Yan Wang), xin.gao@kaust.edu.sa (Xin Gao)

ACCEPTED MANUSCRIPT

tained. We develop an iterative algorithm to solve this minimization problem efficiently using a cutting-plane algorithm and a coordinate descent method. Experiments on multi-instance benchmark data sets show its advantage over both traditional multi-instance learning and performance optimization methods.

Keywords: Multi-instance learning, dictionary, multivariate performance measures, cutting-plane algorithm

1. Introduction

In many machine learning applications, the data is given as multi-instance data. Meanwhile, different specific multivariate performance measures are often used to evaluate the performance of machine learning models in the test process. For example, in the problem of image classification, the most popular image representation method is bag-of-features, which treats an image as a bag of local instances, while precision at rank k (Prec@k) is the most popular performance measure for the evaluation of image classification methods. To obtain a good performance in the test process, the problem of learning a classifier to directly optimize the same multivariate performance measure over the training set is proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Meanwhile, the key of representing multi-instance data is the learning of a multi-instance dictionary, which is used to map a bag to a bag-level feature vector [14, 15, 16, 17, 18, 19, 20, 16, 21, 22, 23, 24]. We observe that different performance measures require different optimal multi-instance dictionaries, i.e., the optimal performance is dependent on the optimal dictionary. This observation strongly motives us to learn a performance-optimal multi-instance dictionary for the problem of optimizing a specific multivariate performance measure.

To our knowledge, none of the existing multi-instance dictionary learning methods considers a specific multivariate performance measure, while most of the existing performance measure optimization methods try to learn performance-optimal classifier parameters, but ignore the learning of performance-optimal dictionaries. For example, to learn the multi-instance dictionary, Lazebnik and

Accepted manuscript

ACCEPTED MANUSCRIPT

Raginsky [25] proposed to learn a dictionary and the posterior distributions of class labels of the instances simultaneously. Chen et al. [23] proposed to use the training instance set as an initial dictionary and then reduce its size by feature selection. Zhou and Zhang [26] constructed a multi-instance dictionary by clustering the instances of all the training bags and representing the bags by binary features according to the clustering results. Different classifiers are trained and combined for features of different numbers of clusters. Zhang and Zhou [27] used a multi-instance clustering algorithm named k-MEDOIDS to construct a multi-instance dictionary, and represented each bag by the distance between the bag and the dictionary instances. Fu et al. [24] suggested to learn a dictionary by selecting one instance from each training bag and training a classifier jointly. Wang et al. [22] proposed to learn a multi-instance dictionary and maximize classification margins in the bag-level feature space simultaneously. Recently, Shrivastava et al. [19] developed a noisy-OR method to generalize the mean-based optimization framework to learn the multi-instance dictionary in the feature space. However, the classification performance measures of these models are empirical information loss [25], hinge loss [23, 24], classification margin [22], and negative log loss [19], which do not necessarily lead to a good target specific multivariate performance measure, such as Prec@k.

To optimize a specific multivariate performance measure, Joachims [13] proposed to learn a classifier to optimize the upper bound of a multivariate loss function corresponding to a target multivariate performance measure. Mao and Tsang [12] proposed to select features and learn the classifier simultaneously to optimize multivariate performance measures from high-dimensional data. Li et al. [10] adapted a common classifier to a target multivariate performance measure by classifier adaptation. Parambath et, al. [8] optimized a multivariate performance measure, F_1 score, by solving a series of cost-sensitive classification problems. Recently, Wang and Gao [4] proposed to use partially labeled data tuple to optimize multivariate performance measures. Narasimhan et al. [5] expressed two families of multivariate performance measures as functions of true positive/negative rates, and optimized them by point-based stochastic

Accepted manuscript

ACCEPTED MANUSCRIPT

updates. Gao and Zhou [6] proposed to optimize a multivariate performance measure, the area under receiver operating characteristic curve (AUC), by optimizing surrogate loss functions which are consistent with AUC, and provided a sufficient condition for the asymptotic consistency of learning methods which optimize these surrogate loss functions. Recently, Li et al. [2] proposed a performance-safe semi-supervised learning method to optimize multivariate performance measures by integrating multiple semi-supervised learners and maximizing the worst-case performance gain. Multi-instance data sets were used to evaluate these methods, but the process to construct the multi-instance dictionary ignores the target performance measure, and a performance-optimal dictionary is thus not guaranteed. In [28], the feature selection method for multivariate performance measures [12] was extended to multi-instance learning. All the instances of the training bags are combined to construct a large dictionary to generate the bag-level features for the performance-optimal feature selection method [12]. This could result in a large initial feature space even for moderately large data sets, requiring high costs of both computation and storage. Meanwhile, their method assumes that the performance-optimal dictionary lies on the training instance set, and thus is limited to selecting instances from the training bags only.

To fill this gap, in this paper, we propose the problem of performance-optimal multi-instance dictionary learning. A novel joint learning framework for learning the multi-instance dictionary and classifier parameters to optimize a specific multivariate performance measure is proposed to solve this problem. Instead of selecting instances from the training bags to construct the performance-optimal dictionary, we propose to learn it directly, use it to represent each bag using the bag-instance similarities, and learn a classifier for the bag-level representations. The dictionary instances and the classifier parameters are jointly learned by simultaneous minimization of an upper bound of a multivariate loss function corresponding to the multivariate performance measure, the complexity of the classifier, and the complexity of the dictionary. To solve this minimization problem, we propose an efficient iterative algorithm to update the dictionary

Accepted manuscript

ACCEPTED MANUSCRIPT

instances and the classifier parameters alternately. The dictionary is updated by a sub-gradient coordinate descent method in a fixed-point algorithm, and the classifier parameters are updated by a cutting-plane algorithm. Moreover, we also propose an efficient algorithm to find the most validated constraint with a linear complexity for the cutting-plane algorithm.

2. Proposed method

2.1. Problem formulation

Suppose we have a training tuple of n bags, denoted as $\overline{B} = (B_1, \cdots, B_n)$, where $B_i = \{\mathbf{x}_{i,j}\}|_{j=1}^{m_i}$ is the i-th bag which contains m_i instances, and $\mathbf{x}_{i,j} \in \mathbb{R}^d$ is the d-dimensional feature vector of the j-th instance of the i-th bag. To represent the bags, we learn a dictionary $D = \{\mathbf{d}_k\}|_{k=1}^{m}$, which is a set of m dictionary instances, where $\mathbf{d}_k \in \mathbb{R}^d$ is the k-th dictionary instance. With the dictionary, the bag-level representation of B_i is given as the bag-to-instance similarities,

$$g(B_i, D) = [s(B_i, \mathbf{d}_1), \cdots, s(B_i, \mathbf{d}_m)]^\top, \tag{1}$$

where

$$\begin{aligned} s(B_i, \mathbf{d}_k) &= \max_{j=1}^{m_i} \exp\left(-\gamma \|\mathbf{x}_{i,j} - \mathbf{d}_k\|_2^2\right) \\ &= \exp\left(-\gamma \|\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k\|_2^2\right) \end{aligned} \tag{2}$$

is the maximum similarity measure between instances of the bag B_i and the k-th dictionary instance $\mathbf{d}_k$, γ is a parameter of the similarity function, and

$$\phi_i^k = \arg\max_{j=1}^{m_i} \exp\left(-\gamma \|\mathbf{x}_{i,j} - \mathbf{d}_k\|_2^2\right) \tag{3}$$

is the index of the nearest instance of the i-th bag which gives the maximum similarity to $\mathbf{d}_k$.

To predict the label tuple of the training bag tuple from their bag-level representations, we design a linear classification function,

$$\overline{y}^* = \arg\max_{\overline{y}' \in \overline{\mathcal{Y}}} \mathbf{w}^\top \sum_{i=1}^{n} g(B_i, D) y_i', \tag{4}$$

Accepted manuscript

ACCEPTED MANUSCRIPT

where $\mathbf{w} = [w_1, \cdots, w_m]^\top \in \mathbb{R}^m$ is the parameter vector of the classifier, w_k is its k-th element, $\overline{y}^*$ is the output label tuple of the classifier, and $\overline{\mathcal{Y}} = \{+1, -1\}^n$. To optimize a given multivariate performance measure, we propose to minimize its corresponding multivariate loss function, $\Delta(\overline{y}, \overline{y}^*)$, with regard to both $\mathbf{w}$ and D. As an example of how $\Delta(\overline{y}, \overline{y}^*)$ is defined, we give the multivariate loss function of the F_1 score as follows

$$\begin{aligned}
&\Delta(\overline{y}, \overline{y}^*) = 100 \times (1 - F_1(\overline{y}, \overline{y}^*)), \ \textit{and} \\
&F_1(\overline{y}, \overline{y}^*) = \frac{2 \times \sum_{i=1}^n I(y_i = +1, y_i^* = +1)}{\left(\begin{array}{c} 2 \times \sum_{i=1}^n I(y_i = +1, y_i^* = 1) \\ + \sum_{i=1}^n I(y_i = 1, y_i^* = -1) + \sum_{i=1}^n I(y_i = -1, y_i^* = +1) \end{array} \right)},
\end{aligned} \tag{5}$$

where $F_1(\overline{y}, \overline{y}^*)$ is the F_1 score given the true label tuple $\overline{y}$ and the predicted label tuple $\overline{y}^*$, and $I(x) = 1$ if x is true, and 0 otherwise. An upper bound of the loss function is given as follows in [13],

$$\begin{aligned}
\Delta(\overline{y}, \overline{y}^*) \leq \max \Bigg(0, \max_{\overline{y}' \in \overline{\mathcal{Y}}/\overline{y}} \Bigg(& \Delta(\overline{y}, \overline{y}') \\
& - \mathbf{w}^\top \sum_{i=1}^n g(B_i, D)(y_i - y_i') \Bigg) \Bigg).
\end{aligned} \tag{6}$$

Besides minimizing the upper bound of the multivariate loss function to seek an optimal multivariate performance measure, we also impose the classifier and the dictionary to be as simple as possible to prevent the over-fitting problem. We have a joint optimization problem to learn both the classifier parameter $\mathbf{w}$ and the dictionary D,

$$\begin{aligned}
&\min_D \min_{\mathbf{w}, \xi \geq 0} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{C_1}{2} \sum_{k=1}^m \|\mathbf{d}_k\|_2^2 + C_2 \xi \right\}, \\
&\quad s.t.\ \overline{y}' \in \mathcal{Y}/\overline{y} : \ \mathbf{w}^\top \sum_{i=1}^n g(B_i, D)(y_i - y_i') \geq \Delta(\overline{y}, \overline{y}') - \xi,
\end{aligned} \tag{7}$$

where $\frac{1}{2}\|\mathbf{w}\|_2^2$ and $\frac{1}{2}\sum_{k=1}^m \|\mathbf{d}_k\|_2^2$ are the regularization terms to control the complexities of the classifier and the dictionary respectively, ξ is a slack variable to represent the upper bound of the multivariate loss function, and C_1 and C_2 are the tradeoff parameters.

Accepted manuscript

ACCEPTED MANUSCRIPT

2.2. Problem optimization

The Lagrange function of the problem in (7) is obtained as follows,

$$\begin{aligned}\mathcal{L}(\mathbf{w},\xi,D,\alpha,\tau) &= \frac{1}{2}\|\mathbf{w}\|_2^2 + C\xi \\ &- \tau\xi - \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}\left(\mathbf{w}^\top \sum_{i=1}^{n} g(B_i,D)(y_i - y_i') - \Delta(\overline{y},\overline{y}') + \xi\right),\end{aligned} \tag{8}$$

where τ is the Lagrange multiplier variable for the constraint of $\xi \geq 0$, and $\alpha = \{\alpha_{\overline{y}'}\}|_{\overline{y}'\in\mathcal{Y}/\overline{y}}$ is a set of Lagrange multiplier variables for the constraints in $\mathcal{Y}/\overline{y}$. To obtain the KKT conditions of this problem, we set the gradients of the Lagrange function with regard to $\mathbf{w}$ and ξ to zeros,

$$\begin{aligned}&\nabla\mathcal{L}_{\mathbf{w}} = \mathbf{w} - \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}\left(\sum_{i=1}^{n} g(B_i,D)(y_i - y_i')\right) = 0 \\ &\Rightarrow \mathbf{w} = \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}\left(\sum_{i=1}^{n} g(B_i,D)(y_i - y_i')\right), \textit{ and} \\ &\nabla\mathcal{L}_{\xi} = C - \tau - \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'} = 0 \\ &\Rightarrow \tau = C - \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'} \geq 0 \Rightarrow C_2 \geq \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}.\end{aligned} \tag{9}$$

We substitute the KKT conditions back to the Lagrange function, and have the dual form of the problem,

$$\begin{aligned}\min_D \max_\alpha \Bigg\{ &-\frac{1}{2}\sum_{\overline{y}',\overline{y}''\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'} Q^D_{\overline{y}'\overline{y}''} \alpha_{\overline{y}''} \\ &+ \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}\Delta(\overline{y},\overline{y}') + \frac{C_1}{2}\sum_{k=1}^{m}\|\mathbf{d}_k\|_2^2 \Bigg\}, \\ &s.t.\ \overline{y}' \in \mathcal{Y}/\overline{y} : \alpha_{\overline{y}'} \geq 0,\ C_2 \geq \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'},\end{aligned} \tag{10}$$

where $\alpha = \{\alpha_{\overline{y}'}\}|_{\overline{y}'\in\mathcal{Y}/\overline{y}}$ is a set of Lagrange multiplier variables for the constraints in $\mathcal{Y}/\overline{y}$, and

$$Q^D_{\overline{y}'\overline{y}''} = \left(\sum_{i=1}^{n} g(B_i,D)(y_i - y_i')\right)^\top \left(\sum_{i=1}^{n} g(B_i,D)(y_i - y_i'')\right). \tag{11}$$

Accepted manuscript

ACCEPTED MANUSCRIPT

To solve the joint optimization problem in (10), we use the alternate optimization strategy in an iterative algorithm. In each iteration, we first fix α to optimize D, and then fix D to optimize α.

2.2.1. Optimizing α while fixing D

When D is fixed, we first calculate $Q^D_{\overline{y}'\overline{y}''}$ from D according to (11), and then fix it to update α by solving the problem in (10). We remove the term irrelevant to α from (10), and reduce it to the following problem with regard to α only,

$$\begin{aligned}\max_{\alpha=\{\alpha_{\overline{y}'}\}|_{\overline{y}'\in\mathcal{Y}/\overline{y}}} \quad & \left\{ -\frac{1}{2}\sum_{\overline{y}',\overline{y}''\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'} Q^D_{\overline{y}'\overline{y}''}\alpha_{\overline{y}''} \right. \\ & \left. + \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}\Delta(\overline{y},\overline{y}') \right\}, \\ & s.t.\ \overline{y}'\in\mathcal{Y}/\overline{y} : \alpha_{\overline{y}'}\geq 0,\ C_2 \geq \sum_{\overline{y}'\in\mathcal{Y}/\overline{y}} \alpha_{\overline{y}'}.\end{aligned} \tag{12}$$

To solve this problem, we use a cutting-plane algorithm. This algorithm is an iterative algorithm which updates the constraints in $\mathcal{Y}/\overline{y}$ and the corresponding Lagrange multiplier variables in α alternately. We maintain an active set of constraints, $\mathcal{W}$, in the algorithm. In each iteration, given the current $\mathcal{W}$, we use it to replace the constraint set $\mathcal{Y}/\overline{y}$ in (12), and then update α by solving the problem in (12). Then with the current α, we update $\mathcal{W}$ by finding the most violated constraint and adding it to $\mathcal{W}$.

- **Updating** α: Using the current $\mathcal{W}$ as the constraint set, we update α by solving the problem in (12),

$$\begin{aligned}\max_{\alpha=\{\alpha_{\overline{y}}\}|_{y\in\mathcal{W}}} \quad & \left\{ -\frac{1}{2}\sum_{\overline{y}',\overline{y}''\in\mathcal{W}} \alpha_{\overline{y}'} Q^D_{\overline{y}'\overline{y}''}\alpha_{\overline{y}''} \right. \\ & \left. + \sum_{\overline{y}'\in\mathcal{W}} \alpha_{\overline{y}'}\Delta(\overline{y},\overline{y}') \right\} \\ & s.t.\ \overline{y}'\in\mathcal{W} : \alpha_{\overline{y}'}\geq 0,\ C_2 \geq \sum_{\overline{y}'\in\mathcal{W}} \alpha_{\overline{y}'}.\end{aligned} \tag{13}$$

Accepted manuscript

ACCEPTED MANUSCRIPT

This problem is a quadratic programming (QP) problem with linear constrains. We solve it by using an active set algorithm. With the updated α, $\mathbf{w}$ can be recovered as follows,

$$\mathbf{w} = \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} \left(\sum_{i=1}^{n} g(B_i, D)(y_i - y_i') \right). \tag{14}$$

- **Updating** $\mathcal{W}$: Since the constraint set is used to approximate the upper bound of the multivariate loss, $\Delta(\overline{y}, \overline{y}^*)$, as in (6), we seek the most violated constraint, $\overline{y}^\diamond$, to achieve the maximization of the right hand side of (6),

$$\overline{y}^\diamond = \arg\max_{\overline{y}' \in \overline{\mathcal{Y}}/\overline{y}} \left\{ \Delta(\overline{y}, \overline{y}') + \mathbf{w}^\top \sum_{i=1}^{n} g(B_i, D) y_i' \right\}. \tag{15}$$

After obtaining the most violated constraint $\overline{y}^\diamond$, we update $\mathcal{W}$ by adding it to $\mathcal{W}$, $\mathcal{W} \leftarrow \mathcal{W} \cup \{\overline{y}^\diamond\}$. The problem of finding the most validated constraint of (15) is dependent on the definition of the multivariate loss. Following [12], we also focus on the multivariate performance measures based on a contingency table. Given true class labels y and the candidate labels y', the contingency table is defined in Table 1. The elements of this table satisfy the following two basic rules, $a + c = \#pos$ and $b + d = \#neg$, where $\#pos$ is the number of positive data points, and $\#neg$ is the number of negative data points. Based on the contingency table, some most popular multivariate performance measures with their corresponding multivariate losses are given in Table 2. Some performance measures have an additional rule over the contingency table, as listed in Table 2.

Table 1: Contingency table.

	$y = +1$	$y = -1$
$y' = +1$	a	b
$y' = -1$	c	d

To obtain a candidate constraint from a given candidate contingency table, we first sort the classification responses of the positive data points

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 2: Definitions of typically used multivariate losses based on contingency table.

Performance	Multivariate measure	Multivariate loss $\Delta(\overline{y}, \overline{y}')$	Additional rule
PRBEP	$\frac{a}{a+b}$	$100 \cdot (1 - PRBEP)$	$b = c$
Prec@k	$\frac{a}{a+b}$	$100 \cdot (1 - Prec@k)$	$a + b = k$
Rec@k	$\frac{a}{a+c}$	$100 \cdot (1 - Rec@k)$	$a + b = k$
F_1	$\frac{a}{a+b+c}$	$100 \cdot (1 - F_1)$	None
ACC	$\frac{a+d}{a+b+c+d}$	$100 \cdot (1 - ACC)$	None

$f_i|_{i:y_i=+1}$, and the negative data points $f_i|_{i:y_i=-1}$, respectively and descendingly. The indexes of the sorted responses of the positive data points are denoted as $i^p_1, \cdots, i^p_{\#pos}$, and the indexes of the sorted negative data points are denoted as $i^n_1, \cdots, i^n_{\#neg}$. Then, to generate a candidate label tuple, $\overline{y}'$, according to a given contingency table, we set the candidate labels of the first $a = \#pos - c$ sorted positive data points, and the first b sorted negative data points to $+1$, while set the labels of the remaining c positive data points, and the remaining $d = \#neg - b$ to -1,

$$\begin{aligned}
&(y'_{i^p_1}, \cdots, y'_{i^p_{\#pos-c}}) = (+1, \cdots, +1), \\
&(y'_{i^n_1}, \cdots, y'_{i^n_b}) = (+1, \cdots, +1) \\
&(y'_{i^p_{\#pos-c+1}}, \cdots, y'_{i^p_{\#pos}}) = (-1, \cdots, -1), and \\
&(y'_{i^n_{b+1}}, \cdots, y'_{i^n_{\#neg}}) = (-1, \cdots, -1).
\end{aligned} \tag{16}$$

According to Table 2, the multivariate loss function is a function of the elements of the corresponding contingency table, $\Delta(\overline{y}, \overline{y}') = \Delta(a, b, c, d) = \Delta(\#pos - c, b, c, \#neg - b)$, thus a function of (b, c). According to (16), $\overline{y}'$ is also a function of (b, c). Submitting the definitions of $\Delta(\overline{y}, \overline{y}')$ and $\overline{y}'$ to (15), we transfer it to a maximization problem with regard to only two

Accepted manuscript

ACCEPTED MANUSCRIPT

free parameters, $(b, c) \in \{0, \cdots, \#neg\} \times \{0, \cdots, \#pos\}$,

$$(b^\diamond, c^\diamond) = \underset{(b,c)\in\{0,\cdots,\#neg\}\times\{0,\cdots,\#pos\}}{\arg\max} \Bigg\{ \Theta(b, c) = \Delta(\#pos - c, b, c, \#neg - b) + \sum_{i'=1}^{\#pos-c} f_{i^p_{i'}} + \sum_{i'=1}^{b} f_{i^n_{i'}} - \sum_{i'=\#pos-c+1}^{\#pos} f_{i^p_{i'}} - \sum_{i'=b+1}^{\#neg} f_{i^n_{i'}} \Bigg\}, \tag{17}$$

where $\Theta(b, c)$ is the objective function, and $(b^\diamond, c^\diamond)$ is the most violated (b,c). To solve this problem, Joachims (2015) [13] proposed an algorithm to explore the entire admissible space of (b, c) of size $\#pos \times \#neg$. According to (17), the complexity of calculating the objective $\Theta(b, c)$ of each candidate (b, c) is $\mathcal{O}(n)$, thus the complexity of the algorithm is $\mathcal{O}(n^3)$. This complexity is not affordable for even moderately large data set.

We discuss how to solve this problem more efficiently by classifying the performance measures to two different types, and proposing efficient algorithms for the two types of measures respectively. From Table 2, we can see that the performance measures can be classified to two types, defined as Type I measures and Type II measures. Type I measures impose an additional rule over the contingency table, and typical examples of this type include performance measures of Prec@k, Rec@k, and PRBEP. For example, PRBEP imposes $b = c$, while Prec@k and Rec@k imposes $a+b = k$. On the other hand, Type II measures impose no additional rules other than basic rules over the contingency table, and its typical examples include F_1 and ACC.

- For the **Type I measures**, we can use the additional rule to reduce the free parameters from (b, c) to b, and design an efficient algorithm to solve it with complexity of $\mathcal{O}(n)$. According to the additional rules of Type I measures in Table 2, c is also a function of b. For PRBEP, $c = b$, while for Prec@k and Rec@k, $c = \#pos - a = \#pos - k + b$,

Accepted manuscript

ACCEPTED MANUSCRIPT

thus $\Theta(b, c)$ is reduced to a function with only one free parameter, $b \in \{0, \cdots, \#neg\}$. We give the new form of (17) of PRBEP as an example with $c = b$,

$$\begin{aligned} b^{\diamond} = \underset{b\in\{0,\cdots,\#neg\}}{\arg\max} \Bigg\{ & \Theta(b) \\ & = \frac{100 \cdot b}{\#pos} + \sum_{i'=1}^{\#pos-b} f_{i^p_{i'}} + \sum_{i'=1}^{b} f_{i^n_{i'}} \\ & - \sum_{i'=\#pos-b+1}^{\#pos} f_{i^p_{i'}} - \sum_{i'=b+1}^{\#neg} f_{i^n_{i'}} \Bigg\}. \end{aligned} \tag{18}$$

To solve this problem, we conduct a linear search in the admissible space of b. b is initialized as the smaller one of $\#pos$ and $\#neg$, so that the corresponding a and d are positive. b is reduced to 0 in a WHILE loop sequentially to generate to test its response to the objective $\Theta(b)$. In each iteration of the WHILE loop, b is reduced by 1, and its corresponding objective response is,

$$\begin{aligned} \Theta(b-1) = & \left(\frac{100 \cdot b}{\#pos} - \frac{100}{\#pos} \right) \\ & + \left(\sum_{i'=1}^{\#pos-b} f_{i^p_{i'}} + f_{i^p_{\#pos-b+1}} \right) + \left(\sum_{i'=1}^{b} f_{i^n_{i'}} - f_{i^n_b} \right) \\ & - \left(\sum_{i'=\#pos-b+1}^{\#pos} f_{i^p_{i'}} - f_{i^p_{\#pos-b+1}} \right) \\ & - \left(\sum_{i'=b+1}^{\#neg} f_{i^n_{i'}} + f_{i^n_b} \right) \\ = & \Theta(b) + \frac{100}{\#pos} + 2\left(f_{i^p_{\#pos-b+1}} - f_{i^n_b} \right). \end{aligned} \tag{19}$$

The complexity of direct calculating $\Theta(b)$ is $\mathcal{O}(n)$ with a given b according to (18). However, using this incremental property of $\Theta(b)$ shown in (19), when b is changed to $b-1$, we can update the objective in $\mathcal{O}(1)$ time. The algorithm for finding the most violated constraint for PRBEP is given in Algorithm 1. The complexity of this algorithm

Accepted manuscript

ACCEPTED MANUSCRIPT

is $\mathcal{O}(n)$. Note that the outputs of this algorithm not only contain the most violated constraints themselves, but also the corresponding multivariate losses, which will be used as the inputs of the next step of updating α. For other Type I measures, similar reasoning and algorithms can be straightforward derived.

Algorithm 1 Algorithm for finding the most violated constraint for PRBEP.

Input: $f_i|_{i=1}^n$, α, $\mathcal{W}$, and $\overline{y} = (y_1, \cdots, y_n)$;

$(i_1^p, \cdots, i_{\#pos}^p) \leftarrow$ sort $\{i : y_i = +1\}$ by f_i;

$(i_1^n, \cdots, i_{\#neg}^n) \leftarrow$ sort $\{i : y_i = -1\}$ by f_i;

$b = min(\#pos, \#neg)$;

$\Theta = \frac{100 \cdot b}{\#pos} + \sum_{i'=1}^{\#pos-b} f_{i_{i'}^p} + \sum_{i'=1}^{b} f_{i_{i'}^n} - \sum_{i'=\#pos-b+1}^{\#pos} f_{i_{i'}^p} - \sum_{i'=b+1}^{\#neg} f_{i_{i'}^n}$;

$\Theta^* = \Theta$, $b^* = b$;

while $b > 0$ **do**

  $\Theta = \Theta + \frac{100}{\#pos} + 2\left(f_{i_{\#pos-b+1}^p} - f_{i_b^n}\right)$;

  $b--$;

  if $\Theta > \Theta^*$ **then**

    $\Theta^* = \Theta$, $b^* = b$;

  end if

end while

Set $y_{i_1^p}^\diamond, \cdots, y_{i_{\#pos-b^*}^p}^\diamond$, and $y_{i_1^n}^\diamond, \cdots, y_{i_{b^*}^n}^\diamond$ to $+1$.

Set $y_{i_{\#pos-b^*+1}^p}^\diamond, \cdots, y_{i_{\#pos}^p}^\diamond$ and $y_{i_{b^*+1}^p}^\diamond, \cdots, y_{i_{\#neg}^p}^\diamond$ to -1.

Output: $\overline{y}^\diamond = (y_1^\diamond, \cdots, y_n^\diamond)$, and $\Delta(\overline{y}, \overline{y}^\diamond) = \frac{100 \cdot b^*}{\#pos}$.

– For the **Type II measures**, we solve the problem in (19) by developing a linear search algorithm to find the maximum response of $\Theta(b, c)$ in the admissible space of (b, c). We use ACC as an example to develop an algorithm to find the most violated constraint for Type II measurements. According to the definition of the loss function of ACC in Table 2, we rewrite the loss function $\Delta(\#pos - c, b, c, \#neg - b) = \frac{100 \cdot (b+c)}{\#pos + \#neg}$. Substituting it to $\Theta(b, c)$ of (17), we have the objective for finding the most violated constraint

Accepted manuscript

ACCEPTED MANUSCRIPT

for ACC,

$$\begin{aligned}\Theta(b,c) =& \frac{100\cdot(b+c)}{\#pos+\#neg} + \sum_{i'=1}^{\#pos-c} f_{i^p_{i'}} \\ &+ \sum_{i'=1}^{b} f_{i^n_{i'}} - \sum_{i'=\#pos-c+1}^{\#pos} f_{i^p_{i'}} - \sum_{i'=b+1}^{\#neg} f_{i^n_{i'}} \\ =& \Phi(b)+\Omega(c),\ where \end{aligned} \tag{20}$$

$$\Phi(b) = \frac{100\cdot b}{\#pos+\#neg} + \sum_{i'=1}^{b} f_{i^n_{i'}} - \sum_{i'=b+1}^{\#neg} f_{i^n_{i'}},\ and$$

$$\Omega(c) = \frac{100\cdot c}{\#pos+\#neg} + \sum_{i'=1}^{\#pos-c} f_{i^p_{i'}} - \sum_{i'=\#pos-c+1}^{\#pos} f_{i^p_{i'}}.$$

According to (20), $\Theta(b,c)$ of ACC can be decomposed to a summation of two sub-functions of b and c respectively, $\Phi(b)$ and $\Omega(c)$. Thus the maximization of $\Theta(b,c)$ can be decomposed to two independent maximization problem of $\Phi(b)$ and $\Omega(c)$ with regard to b and c respectively. Thus we can use two WHILE loops to search the maximum responses of $\Phi(b)$ and $\Omega(c)$ over the admissible spaces of b and c. b is initialized to $\#pos$, and in each iteration of the WHILE loop, b is reduced by one, until b is reduced to 0. The response of the objective $\Phi(b)$ of $b-1$ is,

$$\begin{aligned}\Phi(b-1) =& \frac{100\cdot(b-1)}{\#pos+\#neg} + \sum_{i'=1}^{b-1} f_{i^n_{i'}} - \sum_{i'=b}^{\#neg} f_{i^n_{i'}} \\ =& \Phi(b) - \frac{100}{\#pos+\#neg} + 2f_{i^n_b}.\end{aligned} \tag{21}$$

Similarly, we have $\Omega(c-1)$

$$\Omega(c-1) = \Omega(c) - \frac{100}{\#pos+\#neg} - 2f_{i^p_{\#pos-c+1}}. \tag{22}$$

Using the incremental property of the calculation of $\Phi(b)$ and $\Omega(c)$, we can also reduce the complexities of their calculation from $\mathcal{O}(n)$ to $\mathcal{O}(1)$. The algorithm for finding the most voilated constraint for ACC is given in Algorithm 2, and its complexity is also $\mathcal{O}(n)$.

Accepted manuscript

ACCEPTED MANUSCRIPT

Algorithm 2 Algorithm for finding the most violated constraint for ACC.

Input: $f_i|_{i=1}^n$, α, $\mathcal{W}$, $\overline{y} = (y_1, \cdots, y_n)$, and κ;

$(i_1^p, \cdots, i_{\#pos}^p) \leftarrow$ sort $\{i : y_i = +1\}$ by f_i;

$(i_1^n, \cdots, i_{\#neg}^n) \leftarrow$ sort $\{i : y_i = -1\}$ by f_i;

$b = \#neg$, $c = \#pos$;

$\Phi = \frac{100 \cdot b}{\#pos + \#neg} + \sum_{i'=1}^{b} f_{i_{i'}^n} - \sum_{i'=b+1}^{\#neg} f_{i_{i'}^n}$;

$\Omega = \frac{100 \cdot c}{\#pos + \#neg} + \sum_{i'=1}^{\#pos-c} f_{i_{i'}^p} - \sum_{i'=\#pos-c+1}^{\#pos} f_{i_{i'}^p}$;

$\Phi^* = \Phi$, $\Omega^* = \Omega$, $b^* = b$, $c^* = c$;

while $b > 0$ **do**

  $\Phi = \Phi - \frac{100}{\#pos + \#neg} + 2 f_{i_b^n}$;

  $b - -$;

  if $\Phi \geq \Phi^*$ **then**

    $\Phi^* = \Phi$, $b^* = b$;

  end if

end while

while $c > 0$ **do**

  $\Omega = \Omega - \frac{100}{\#pos + \#neg} - 2 f_{i_{\#pos-c+1}^p}$;

  $c - -$;

  if $\Omega \geq \Omega^*$ **then**

    $\Omega^* = \Omega$, $c^* = c$;

  end if

end while

Set $y_{i_1^p}^\diamond, \cdots, y_{i_{\#pos-c^*}^p}^\diamond$, and $y_{i_1^n}^\diamond, \cdots, y_{i_{b^*}^n}^\diamond$ to $+1$.

Set $y_{i_{\#pos-c^*+1}^p}^\diamond, \cdots, y_{i_{\#pos}^p}^\diamond$ and $y_{i_{b^*+1}^p}^\diamond, \cdots, y_{i_{\#neg}^p}^\diamond$ to -1.

Output: $\overline{y}^\diamond = (y_1^\diamond, \cdots, y_n^\diamond)$, and $\Delta(\overline{y}, \overline{y}^\diamond) = \frac{100 \cdot (b^* + c^*)}{\#pos + \#neg}$.

2.2.2. Optimizing D while fixing α

With the current α and $\mathcal{W}$ updated from the cutting-plane algorithm, we fix them to optimize D. Since $\mathcal{W}$ obtained from the cutting-plane algorithm is a good approximation of $\mathcal{Y}/\overline{y}$, we also use it to replace $\mathcal{Y}/\overline{y}$ when we optimize D. By removing the objective terms irrelevant to D, the problem in (10) is reduced

Accepted manuscript

ACCEPTED MANUSCRIPT

to

$$\min_{D} \left\{ -\frac{1}{2} \sum_{\overline{y}', \overline{y}'' \in \mathcal{W}} \alpha_{\overline{y}'} Q^{D}_{\overline{y}'\overline{y}''} \alpha_{\overline{y}''} + \frac{C_1}{2} \sum_{k=1}^{m} \|\mathbf{d}_k\|_2^2 \right\}. \tag{23}$$

Substituting $Q^{D}_{\overline{y}'\overline{y}''}$ of (11), and $\mathbf{w}$ of (14) to (23), we can rewrite it as

$$\begin{aligned} \min_{D} \Bigg\{ & -\frac{1}{2} \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} \sum_{i=1}^{n} \left(\mathbf{w}^{\top} g(B_i, D)(y_i - y_i') \right) + \frac{C_1}{2} \sum_{k=1}^{m} \|\mathbf{d}_k\|_2^2 \\ & = \sum_{k=1}^{m} \Bigg[-\frac{1}{2} \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} \sum_{i=1}^{n} \Big(w_k \exp\left(-\gamma \|\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k\|_2^2 \right) \\ & (y_i - y_i') \Big) + \frac{C_1}{2} \|\mathbf{d}_k\|_2^2 \Bigg] = \sum_{k=1}^{m} \Psi(\mathbf{d}_k) \Bigg\}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} \Psi(\mathbf{d}_k) = & -\frac{1}{2} \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} \sum_{i=1}^{n} \Big(w_k \exp\left(-\gamma \|\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k\|_2^2 \right) \\ & (y_i - y_i') \Big) + \frac{C_1}{2} \|\mathbf{d}_k\|_2^2 \end{aligned} \tag{25}$$

is a function of the individual k-th dictionary instance $\mathbf{d}_k$, independent from other dictionary instances, $\mathbf{d}_{k'}|_{k' \neq k}$. Thus we can update each dictionary instance $\mathbf{d}_k$ independently using the coordinate descent method, by updating $\mathbf{d}_k$ as follows,

$$\mathbf{d}_k \leftarrow \mathbf{d}_k - \eta \nabla \Psi(\mathbf{d}_k), \tag{26}$$

where η is the descent step, and $\nabla \Psi(\mathbf{d}_k)$ is the gradient function of $\Psi(\mathbf{d}_k)$. According to (3) and (14), both ϕ_i^k and w_k are functions of $\mathbf{d}_k$, which makes it difficult to obtain the gradient function $\nabla \Psi(\mathbf{d}_k)$ directly. To solve this problem, we use the fixed-point strategy. In an iterative algorithm, we first fix $\mathbf{d}_k$ to update ϕ_i^k and w_k as in (3) and (14), and then fix them to obtain a sub-gradient function, $\nabla \Psi(\mathbf{d}_k)$,

$$\begin{aligned} \nabla \Psi(\mathbf{d}_k) = & -\gamma \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} \sum_{i=1}^{n} \Big(w_k \exp\left(-\gamma \|\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k\|_2^2 \right) \\ & (\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k)(y_i - y_i') \Big) + C_1 \mathbf{d}_k. \end{aligned} \tag{27}$$

Accepted manuscript

ACCEPTED MANUSCRIPT

2.3. Algorithm

Based on the optimization results obtained in Section 2.2, we develop an iterative algorithm, Algorithm 3, to learn D and $\mathbf{w}$ jointly to optimize a given multivariate performance measure. In the cutting-plane algorithm we incrementally update the elements of parameters of the QP problem in (13), $Q^D_{\overline{y}'\overline{y}''}|_{\overline{y}',\overline{y}''\in\mathcal{W}}$ and $\Delta(\overline{y},\overline{y}')|_{\overline{y}'\in\mathcal{W}}$. When a new most violated constraint $\overline{y}^\diamond$ is found, we only calculate the elements related to $\overline{y}^\diamond$, which are $Q^D_{\overline{y}^\diamond\overline{y}''}|_{\overline{y}''\in\mathcal{W}}$, $Q^D_{\overline{y}',\overline{y}^\diamond}|_{\overline{y}'\in\mathcal{W}}$ and $\Delta(\overline{y},\overline{y}^\diamond)$. This process can avoid most of the time-consuming re-calculation of these elements.

2.3.1. Algorithm complexity analysis

Most steps of the algorithm have linear complexities with regard to the number of the training bags, n. To explain it, we define some $\overline{y}'$-specific vectors, which can be calculated as summations of responses of individual bags and labels with a complexity of $\mathcal{O}(n)$,

$$\begin{aligned}
&\mathbf{u}_{\overline{y}'} = \sum_{i=1}^{n} g(B_i, D)(y_i - y_i'),\ \mathbf{v}_{\overline{y}'} = \sum_{i=1}^{n} g(B_i, D) y_i',\ \textit{and} \\
&\boldsymbol{\beta}_{\overline{y}'} = \sum_{i=1}^{n} \left(w_k \exp\left(-\gamma \|\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k\|_2^2 \right) (\mathbf{x}_{i,\phi_i^k} - \mathbf{d}_k)(y_i - y_i') \right).
\end{aligned} \tag{28}$$

According to (11), (14), (15), and (27), $Q^D_{\overline{y}'\overline{y}''} = \mathbf{u}_{\overline{y}'}^\top \mathbf{u}_{\overline{y}''}$, $\mathbf{w} = \sum_{\overline{y}'\in\mathcal{W}} \alpha_{\overline{y}'} \mathbf{u}_{\overline{y}'}$, the second term of the objective of (15) $\mathbf{w}^\top \sum_{i=1}^n g(B_i, D) y_i' = \mathbf{w}^\top \mathbf{v}_{\overline{y}'}$, and $\nabla\Psi(\mathbf{d}_k) = -\gamma \sum_{\overline{y}'\in\mathcal{W}} \alpha_{\overline{y}'} \boldsymbol{\beta}_{\overline{y}'} + C_1 \mathbf{d}_k$ are all functions of these $\overline{y}'$-specific vectors, thus the complexities of their calculations are $\mathcal{O}(n)$, $\mathcal{O}(n \times |\mathcal{W}|)$, $\mathcal{O}(n \times m)$, and $\mathcal{O}(n \times |\mathcal{W}|)$ respectively, where $|\mathcal{W}|$ is the size of $\mathcal{W}$. For the process of finding the most violated constraint for typical performance measures such as the F_1 score, PRBEP, and Prec@k, Joachims (2005) proposed a linear search algorithm. This algorithm searches in a space of two free parameters, i.e., the true positive number (denoted as a) and the true negative number (denoted as b), thus its complexity is $\mathcal{O}(n^2)$. However, Prec@k requires $b = k - a$, and PRBEP requires $b = \#pos - a$, where $\#pos$ is the number of positive bags. Thus we can reduce

Accepted manuscript

ACCEPTED MANUSCRIPT

Algorithm 3 Iterative algorithm of learning Performance-Optimal multi-instance Dictionary (POD).

Input: $\overline{B}$, $\overline{y}$, C_1, C_2, and m.

Initialize D by randomly selecting m instances from training bags;

repeat

Update the nearest instance indexes, $\phi_i^k|_{k=1}^m$, according to (3), and the bag-level feature vectors, $g(B_i, D)$, according to (2) by fixing D for each $i = 1, \cdots, n$;

Initialize an active constraint set $\mathcal{W} = \emptyset$, the corresponding multiplier set $\alpha = \emptyset$, the classifier parameter vector $\mathbf{w} = [0, \cdots, 0]^\top$;

repeat

Find the most violated constraint $\overline{y}^\diamond$, and its corresponding multivariate loss $\Delta(\overline{y}, \overline{y}^\diamond)$ according to (15), by fixing $\mathbf{w}$ and $g(B_i, D)|_{i=1}^n$;

$\mathcal{W} \leftarrow \mathcal{W} \cup \{\overline{y}^\diamond\}$;

Calculate $Q^D_{\overline{y}'\overline{y}^\diamond}$ and $Q^D_{\overline{y}^\diamond\overline{y}'}|_{\overline{y}' \in \mathcal{W}}$ according to (11);

Update α by solving the problem in (13) with C_2, the updated $Q^D_{\overline{y}'\overline{y}''}|_{\overline{y}', \overline{y}'' \in \mathcal{W}}$, and the updated $\Delta(\overline{y}, \overline{y}')|_{\overline{y}' \in \mathcal{W}}$;

Update $\mathbf{w}$ according to (14) by fixing $\mathcal{W}$, α, and $g(B_i, D)|_{i=1}^n$;

until The maximum iteration number is reached or convergence.

Update $\mathbf{d}_k$ for each $k = 1, \cdots, m$ according to (26) by fixing $\phi_i^k|_{i=1}^n$, $\mathbf{w}$, α, and $\mathcal{W}$;

until The maximum iteration number is reached or convergence.

$\mathbf{w} = \sum_{\overline{y}' \in \mathcal{W}} \alpha_{\overline{y}'} (\sum_{i=1}^n g(B_i, D)(y_i - y_i'))$;

Output: $D = \{\mathbf{d}_k\}|_{k=1}^m$, $\mathbf{w}$.

free parameters of the search algorithms of both PRBEP and Prec@k to only a, and the complexity is correspondingly reduced to $\mathcal{O}(n)$.

Accepted manuscript

ACCEPTED MANUSCRIPT

3. Experiments

3.1. Data sets and experimental setup

To evaluate the proposed method, we conducted experiments on eight widely-used benchmark multi-instance data sets, including the data sets of MUSK1 and MUSK2[29], the image data sets of Elephant, Fox, and Tiger data [30], the course data set [31], the MIML data set [32], and the data set of spoken Arabic digit [33]. Some statistical information of these data sets are given in Table 3.

Table 3: Statistical information of benchmark data sets.

Data set	#Bags	#Classes	#Instances
MUSK1	92	2	476
MUSK2	102	2	6,598
Elephant	200	2	1,391
Fox	200	2	1,320
Tiger	200	2	1,220
Course	1,348	2	3,528
MIML	2,000	5	18,000
Spoken Arabic Digit	8,800	10	350,319

We used the ten-fold cross-validation protocol to conduct the experiments. Each data set was split to ten folds, and each fold was used as the test set while the remaining nine sets were combined to a training set. Given a target performance measure, the POD algorithm was performed to the training set to learn a performance-optimal dictionary and the corresponding classifier. The maximum number of iterations of the proposed algorithm was determined via cross-validation over the training set. The learned dictionary was used to represent the test bags, and the classifier was used to classify them. The classification performance was evaluated by the target performance measure. We reported the average performance measure over the test sets. In our experiments, we considered four typical multivariate performance measures, F_1 score, precision recall

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 4: P-values of paired-sample T-test of scores of POD and compared dictionary learning algorithms over the MUSK1 and Elephant data sets.

MUSK1 data set

Performance	miFV	MILBD	MMDL	MILIS	MILES
F1	0.0484	0.0158	0.0040	0.0059	0.0006
PRBEP	0.0314	0.0121	0.0231	0.0657	0.0003
PrecAtK	0.0120	0.0162	0.0074	0.0200	0.0031
ACC	0.1295	0.0044	0.0024	0.0011	0.0032

Elephant data set

Performance	miFV	MILBD	MMDL	MILIS	MILES
F1	0.0292	0.0127	0.0042	0.0024	0.0283
PRBEP	0.2257	0.0133	0.0206	0.0026	0.0557
PrecAtK	0.0150	0.0346	0.0884	0.0582	0.0431
ACC	0.0337	0.0212	0.0550	0.0122	0.0468

break even point (PRBEP), Prec@k, and classification accuracy (ACC). Note that the multi-instance learning algorithms usually deal with binary classification problems, but the MIML and Spoken Arabic Digit data sets are multi-class data sets. To evaluate the performance of each comparing algorithm over these two data sets, we used the one-vs-rest strategy. We treated the problem of distinguishing the data of each class from other classes as a binary classification problem, and conducted the experiments. The performance measures of all the classes were averaged and reported as the final results.

3.2. Experimental results

3.2.1. Comparison with multi-instance dictionary learning methods

We compared the proposed performance-optimal dictionary learning method, POD, against the state-of-the-art dictionary learning algorithms, including multiple-instance learning via embedded instance selection (MILES) [23], multiple instance learning with instance selection (MILIS) [24], max-margin multiple-instance dictionary learning (MMDL) [22], multi-instance learning based on the Fisher

Accepted manuscript

ACCEPTED MANUSCRIPT

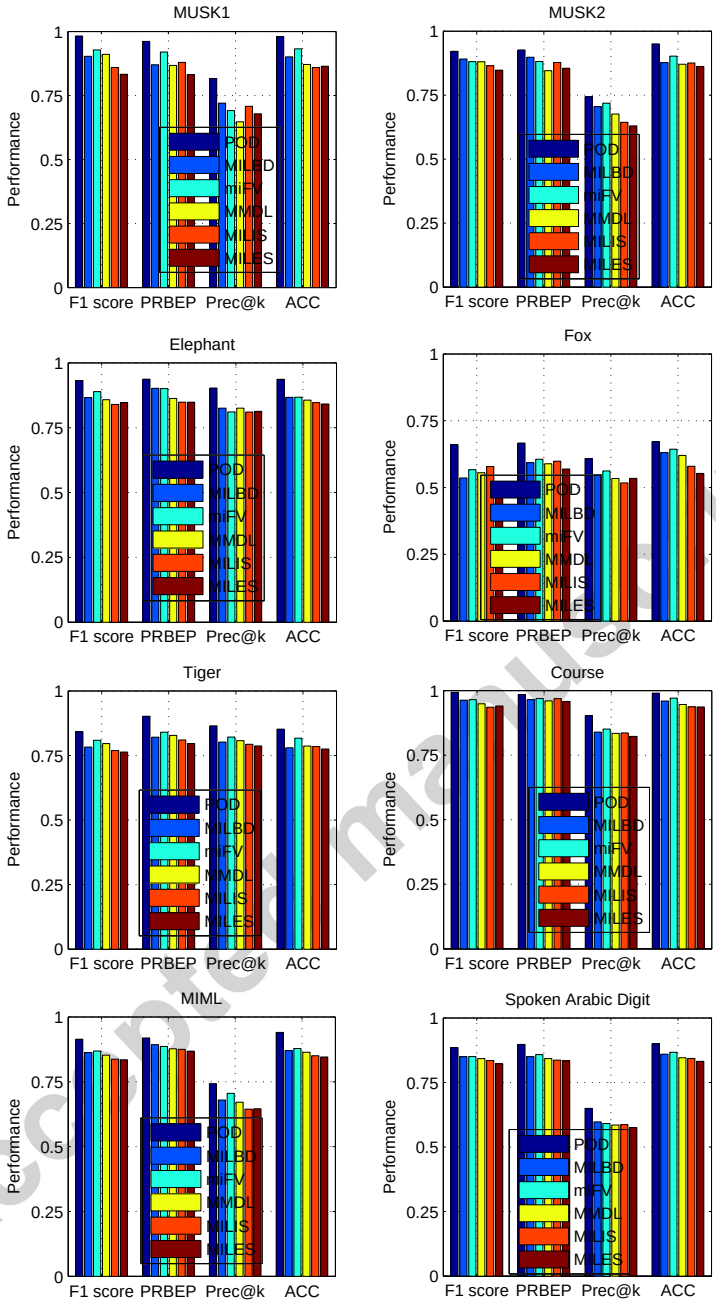


Figure 1: Results of comparison with multi-instance dictionary learning methods.

ACCEPTED MANUSCRIPT

Vector representation (miFV) [16], and multiple instance learning with bag dissimilarities (MILBD) [34]. The parameters of the compared methods were tuned and optimized over the training set by cross validation. The results of the optimized performance measures of F_1 score, PRBEP, Prec@k, and ACC over different data sets are given in Figure 1. Clearly, over all eight data sets and four performance measures, the proposed POD method outperforms the compared traditional dictionary learning methods significantly. This is not surprising, because POD is the only method which considers the target performance measure to learn the dictionary, while other methods learn dictionaries to optimize different performance measures other than the considered one, such as a soft margin [22], a hinge loss [23, 24], and log-likelihood [16]. Comparing the optimization results of F_1 score and ACC, we can observe that the scores of F_1 and ACC are comparable. To further verify if the improvements are significant, we perform the paired-sample T-test to compare the optimized scores of POD and a compared algorithm of the ten splittings over a data set. The null hypothesis of the test is that the pairwise difference between the results of POD and a compared algorithm has a mean equal to zero. The P-values of the T-test of the scores of POD and compared dictionary learning algorithms over the MUSK1 and Elephant data sets are shown in Table 4. As we can observe from the table, for most of cases, the null hypothesises are rejected, and the improvements of POD algorithm compared to the other dictionary learning methods are significant at the significance level of 0.05.

3.2.2. Comparison with performance optimization methods

We also compared POD against the state-of-the-art performance optimization methods, including support vector machine for performance measures (SVM^{perf}) [13], feature selection for performance measures (FS^{perf}) [12], and classifier adaptation for performance optimization (CAPO) [10]. For both SVM^{perf} and CAPO, we pre-learned dictionaries by using the k-means clustering algorithm to present the bags, and the dictionaries were fixed during the learning process. Other dictionary learning methods can also be adopted to pre-learn the dictio-

Accepted manuscript

ACCEPTED MANUSCRIPT

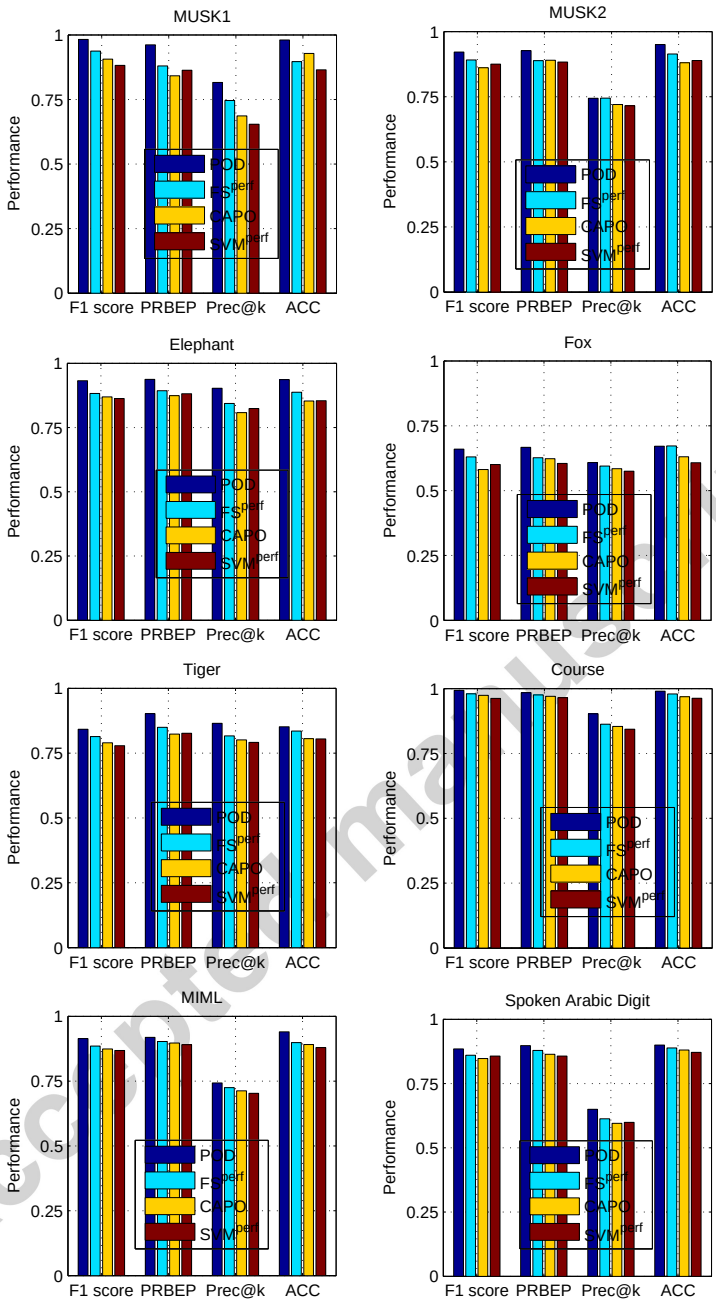


Figure 2: Results of comparison with performance optimization learning methods.

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 5: P-values of paired-sample T-test of scores of POD and compared performance optimization algorithms over the MUSK1 and Elephant data sets.

MUSK1 data set

Performance	FS^{perf}	CAPO	SVM^{perf}
F1	0.0355	0.0024	0.0042
PRBEP	0.0390	0.0012	0.0026
Prec@k	0.0840	0.0375	0.0011
ACC	0.0050	0.1140	0.0036

Elephant data set

Performance	FS^{perf}	CAPO	SVM^{perf}
F1	0.0379	0.1138	0.0038
PRBEP	0.0878	0.0228	0.0379
Prec@k	0.2112	0.0234	0.1347
ACC	0.1326	0.0152	0.0196

Accepted manuscript

nary, and we observed that the different choices of the pre-learned dictionaries influenced the performance of the comparing algorithms. This is the motivation to develop our performance-optimal dictionary. From the results reported in Figure 2, we observed that in most cases, the proposed method, POD, obtains the best results. For some cases, the results of POD and FS^{perf} are comparable, such as the optimized Prec@k over the MUSK2 data set. For the task of optimizing ACC over the Fox data set, FS^{perf} achieves slightly higher Prec@k than POD. A possible reason is that both FS^{perf} and POD try to learn performance-optimal dictionaries. But in all other cases, POD still performs better than FS^{perf}, possibly because FS^{perf} is limited to learn a dictionary by selecting instances from the training bags, while POD does not have such a limitation, and the learned dictionary is more optimal to the target performance measure. This means a performance-optimal dictionary does not necessarily lay in the set of training instances, and our motivation to learn a performance-optimal dictionary directly beyond the training instances is justified. Moreover, by comparing the results of Figure 2 and Figure 1, we observed that the results of performance

ACCEPTED MANUSCRIPT

optimization methods are generally better than those of the dictionary learning methods. This is an evidence of the importance of considering the target performance measure in the learning process. To verify the significance of the improvements achieved by POD algorithm compared to the other performance measure optimization methods, we also conducted the paired sample T-test, and the P-values are reported in Table 5. According to the P-values in the table, the improvements are significant in most cases at the significance level of 0.05.

3.2.3. Parameter sensitivity analysis

There are three important parameters of our algorithm, C_1, C_2, and m. We evaluated the sensitivity of the proposed POD algorithm against these parameters. Due to the page limit, here we show the parameter sensitivity analysis on the MUSK1 and Elephant data sets only. The performance curves of different measures with varying parameter values are given in Figure 3. For performance measures of F_1 score, PRBEP, Prec@k, and ACC over both data sets, a smaller C_1 seems to achieve better results than a larger one. The only exception is the result of optimizing Prec@k over the Elephant data set. Since C_1 is the weight of the term of reducing the complexity of the dictionary, this indicates that reducing the dictionary complexity cannot boost the optimization of a specific performance measure. From the second line of Figure 3, we can see that the performance is very stable over the changes of C_2 for both data sets with different performance measures, especially when C_2 is small. For the Elephant data set, F_1 score and PRBEP are improved slightly when C_2 is larger than 1. Besides these two cases, the algorithm seems to be insensitive to the changes of C_2. From the third line of Figure 3, we cannot see a clear and consistent trend of the changes of performance when m is changing. When F_1 score, PRBEP and ACC are considered, it seems that a larger m gives a better result over the MUSK1 data set, and a smaller m works better for the Elephant data set, while for Prec@k, the case is the opposite. However, the changes of the results according to changes of m are minor, e.g., a dictionary of a small size does not give a significantly worse result than a large one. This indicates that a

Accepted manuscript

ACCEPTED MANUSCRIPT

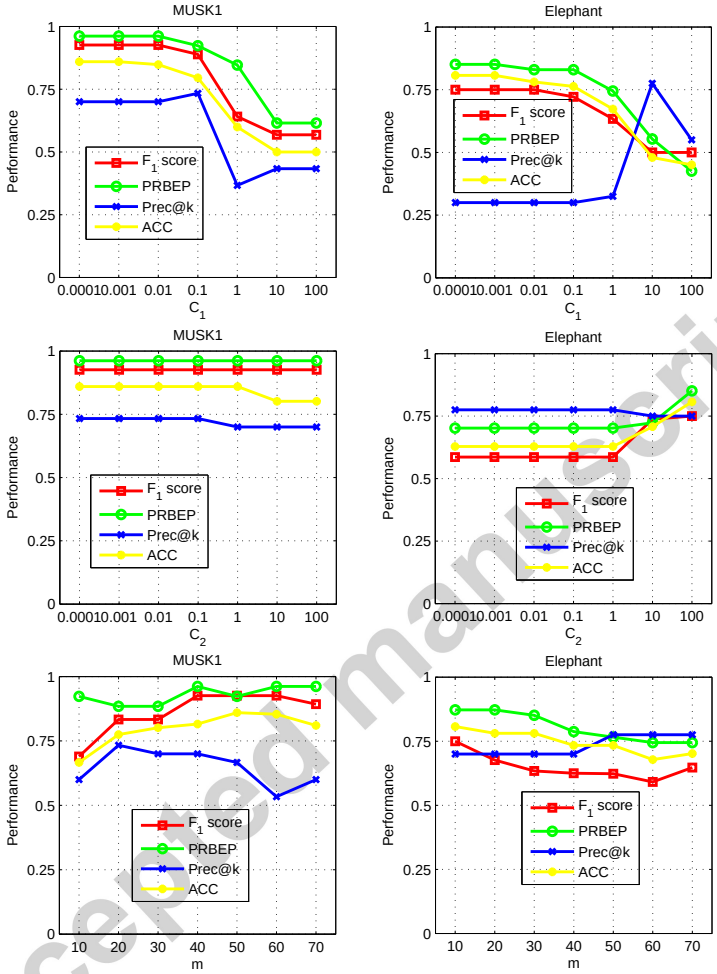


Figure 3: Optimized multivariate performance measures with varying parameters over the MUSK1 and Elephant data sets.

Accepted manuscript

ACCEPTED MANUSCRIPT

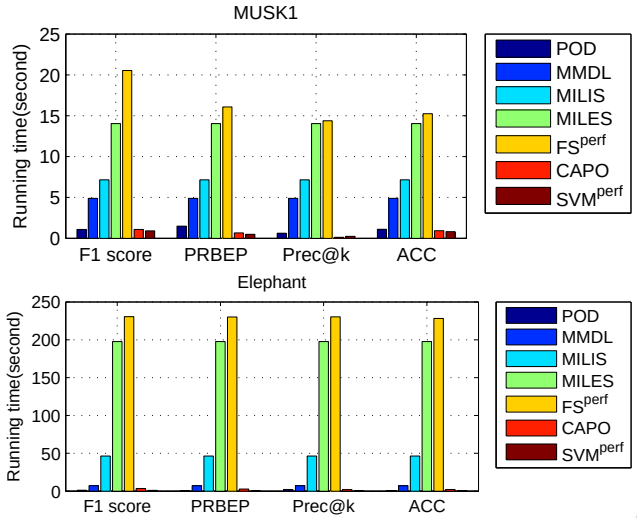


Figure 4: Running time of different algorithms over the MUSK1 and Elephant data sets.

performance-optimal dictionary is not necessarily a large dictionary.

3.2.4. Comparison of running time

We compared the running time of POD to that of the competing algorithms over the MUSK1 and Elephant data sets, and the results are shown in Figure 4. For both data sets, MILIS, MILES, and FS^{perf} consume long running time, because they all explore the entire set of training instances to construct the dictionary. This is even more obvious over the Elephant data set, which has more training instances than the MUSK1 data set. However, POD consumes much less running time than these algorithms because it updates each dictionary instance by the sub-gradient descent algorithm, while MILIS, MILES, and FS^{perf} update each dictionary instance by conducting a linear search over a set of training instances. The least time consuming algorithms are SVM^{perf} and CAPO, which ignore the process of dictionary learning, yet the running time of POD is favorably comparable to that of these two methods.

Accepted manuscript

ACCEPTED MANUSCRIPT

3.2.5. Convergence analysis

The proposed algorithm is an iterative algorithm. Thus we also studied the convergence property of the algorithm. The curves of the objective values of 100 iterations over two data sets are plotted in Figure 5. From the curves, we can observe that the algorithm converges well over two data sets, especially over the Elephant data set. For example, when the PRBEP is optimized over the Elephant data set, the objective value decreases from over 4,000 to a very small value within a few iterations, and then stays stable in the following iterations. It is interesting to see when Prec@k is optimized over the Elephant data set, the objective value increases to a large amount and then decreases rapidly within 10 iterations. The algorithm seems to converge slower over the MUSK1 data set than over the Elephant data set. However, the objective values still keep decreasing stably over the MUSK1 data set. This is a solid evidence of the convergence of the proposed iterative algorithm.

4. Conclusions and future work

In this paper, we proposed the problem of learning multivariate performance-optimal multi-instance dictionary, and a novel algorithm to solve it. The learning of the dictionary and a bag-level linear classifier are modeled jointly to optimize a specific multivariate performance measure, such as Prec@k, and the dictionary is learned by a coordinate sub-gradient descent method, instead of selecting instances from the training set as the traditional time-consuming multi-instance dictionary learning methods. The experimental results show that our performance-optimal dictionary learning algorithm not only outperforms the traditional performance-insensitive multi-instance learning methods, but also outperforms the state-of-the-art performance optimization methods when multi-instance data are used. In the future we will use the Map-Reduce distributed data processing protocol to parallelize the algorithm to scale it to the big data scenario [35]. As discussed in Section 2.3, most steps of the algorithm are calculating summations over responses of individual bags and labels. We may first

Accepted manuscript

ACCEPTED MANUSCRIPT

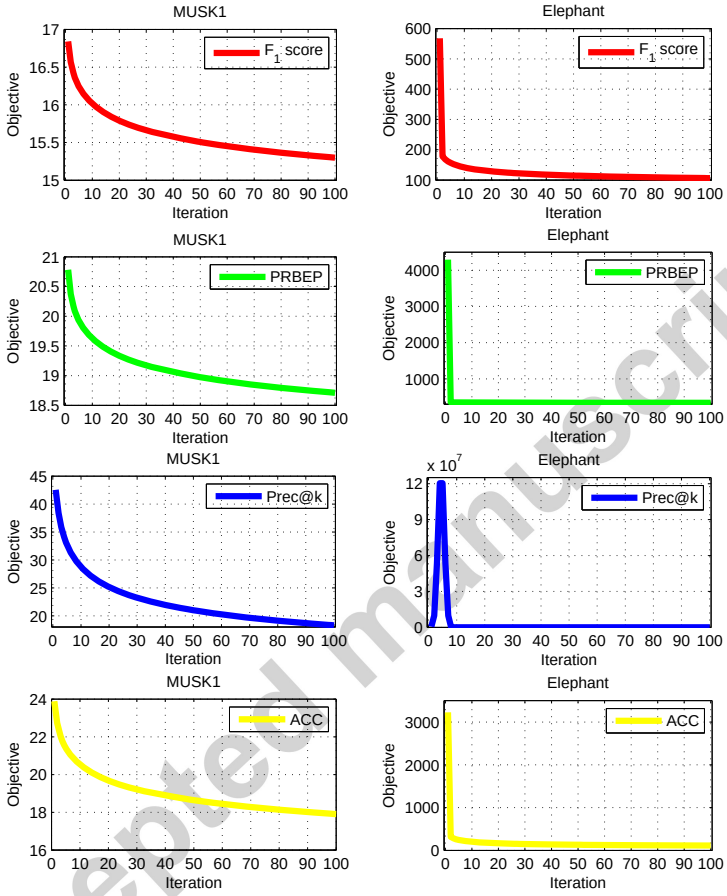


Figure 5: Convergence curves of the proposed algorithm over the MUSK1 and Elephant data sets.

Accepted manuscript

ACCEPTED MANUSCRIPT

calculate the summations of the responses over the subsets simultaneously using Map programs, and then combine them to obtain the final output using Reduce programs.

5. Acknowledgements

We are grateful to Prof. Zhi-Hua Zhou for the fruitful discussions. The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST).

[1] J. J.-Y. Wang, I. W.-H. Tsang, X. Gao, Optimizing multivariate performance measures from multi-view data, in: AAAI, 2016, pp. 2152–2158.

[2] Y.-F. Li, J. T. Kwok, Z.-H. Zhou, Towards safe semi-supervised learning for multivariate performance measures, in: AAAI, 2016, pp. 1816 – 1822.

[3] L. Liu, T. G. Dietterich, N. Li, Z.-H. Zhou, Transductive optimization of top k precision, in: IJCAI, 2016.

[4] J. J.-Y. Wang, X. Gao, Partially labeled data tuple can optimize multivariate performance measures, in: CIKM, 2015, pp. 1915–1918.

[5] H. Narasimhan, P. Kar, P. Jain, Optimizing non-decomposable performance measures: A tale of two classes, in: ICML, 2015.

[6] W. Gao, Z.-H. Zhou, On the consistency of auc pairwise optimization, in: IJCAI, 2015, pp. 939 –945.

[7] N. Li, R. Jin, Z.-H. Zhou, Top rank optimization in linear time, in: NIPS, 2014, pp. 1502–1510.

[8] S. P. Parambath, N. Usunier, Y. Grandvalet, Optimizing f-measures by cost-sensitive classification, in: NIPS, 2014, pp. 2123–2131.

[9] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, I. S. Dhillon, Consistent binary classification with generalized performance metrics, in: NIPS, 2014, pp. 2744–2752.

Accepted manuscript

ACCEPTED MANUSCRIPT

[10] N. Li, I. W. Tsang, Z.-H. Zhou, Efficient optimization of performance measures by classifier adaptation, TPAMI 35 (6) (2013) 1370–1382.

[11] W. Gao, R. Jin, S. Zhu, Z.-H. Zhou, One-pass auc optimization, in: ICML, 2013, pp. 906–914.

[12] Q. Mao, I. W.-H. Tsang, Optimizing performance measures for feature selection, in: ICDM, IEEE, 2011, pp. 1170–1175.

[13] T. Joachims, A support vector method for multivariate performance measures, in: ICML, ACM, 2005, pp. 377–384.

[14] X. Wei, J. Wu, Z. Zhou, Scalable algorithms for multi-instance learning, IEEE Transactions on Neural Networks and Learning Systems-doi:10.1109/TNNLS.2016.2519102.

[15] X. Wang, Z. Zhu, C. Yao, X. Bai, Relaxed multiple-instance svm with application to object discovery, in: ICCV, 2015, pp. 1224–1232.

[16] X.-S. Wei, J. Wu, Z.-H. Zhou, Scalable multi-instance learning, in: ICDM, 2014, pp. 1037–1042.

[17] W. Shen, X. Bai, Z. Hu, Z. Zhang, Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images, Pattern Recognitiondoi:10.1016/j.patcog.2015.10.015.

[18] A. Shrivastava, J. K. Pillai, V. M. Patel, Multiple kernel-based dictionary learning for weakly supervised classification, Pattern Recognition 48 (8) (2015) 2667–2675.

[19] A. Shrivastava, V. M. Patel, J. K. Pillai, R. Chellappa, Generalized dictionaries for multiple instance learning, IJCV 114 (2-3) (2015) 288–305.

[20] H. H. Mohamed, S. Belaid, Algorithm boss (bag-of-salient local spectrums) for non-rigid and partial 3d object retrieval, Neurocomputing 168 (2015) 790–798.

Accepted manuscript

ACCEPTED MANUSCRIPT

[21] A. Shrivastava, J. K. Pillai, V. M. Patel, R. Chellappa, Dictionary-based multiple instance learning, in: ICIP, IEEE, 2014, pp. 160–164.

[22] X. Wang, B. Wang, X. Bai, W. Liu, Z. Tu, Max-margin multiple-instance dictionary learning, in: ICML, 2013, pp. 846–854.

[23] Y. Chen, J. Bi, J. Z. Wang, Miles: Multiple-instance learning via embedded instance selection, TPAMI 28 (12) (2006) 1931–1947.

[24] Z. Fu, A. Robles-Kelly, J. Zhou, Milis: Multiple instance learning with instance selection, TPAMI 33 (5) (2011) 958–977.

[25] S. Lazebnik, M. Raginsky, Supervised learning of quantizer codebooks by information loss minimization, TPAMI 31 (7) (2009) 1294–1309.

[26] Z.-H. Zhou, M.-L. Zhang, Solving multi-instance problems with classifier ensemble based on constructive clustering, Knowledge and Information Systems 11 (2) (2007) 155–170.

[27] M.-L. Zhang, Z.-H. Zhou, Multi-instance clustering with applications to multi-instance prediction, Applied Intelligence 31 (1) (2009) 47–68.

[28] Q. Mao, I. W.-H. Tsang, A feature selection method for multivariate performance measures, TPAMI 35 (9) (2013) 2051–2063.

[29] T. G. Dietterich, R. H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1) (1997) 31–71.

[30] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, J. Malik, Blobworld: A system for region-based image indexing and retrieval, in: Visual Information and Information Systems, Springer, 1999, pp. 509–517.

[31] D. Zhang, J. He, R. Lawrence, Mi2ls: multi-instance learning from multiple information sources, in: KDD, 2013, pp. 149–157.

Accepted manuscript

ACCEPTED MANUSCRIPT

[32] Z.-H. Zhou, Y.-Y. Sun, Y.-F. Li, Multi-instance learning by treating instances as non-iid samples, in: ICML, ACM, 2009, pp. 1249–1256.

[33] N. Hammami, M. Sellam, Tree distribution classifier for automatic spoken arabic digit recognition, in: Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, IEEE, 2009, pp. 1–4.

[34] V. Cheplygina, D. M. Tax, M. Loog, Multiple instance learning with bag dissimilarities, Pattern Recognition 48 (1) (2015) 264 – 275.

[35] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.

Accepted manuscript