Information Sciences 375 (2017) 234–245

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Supervised aggregated feature learning for multiple instance classification

Rocco Langone*, Johan A.K. Suykens

Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

ARTICLE INFO

Article history:
Received 2 May 2016
Revised 7 September 2016
Accepted 28 September 2016
Available online 1 October 2016

Keywords:
Multi-instance classification
Kernel PCA
Kernel spectral clustering

ABSTRACT

This paper introduces a novel algorithm, called Supervised Aggregated FEature learning or SAFE, which combines both (local) instance level and (global) bag level information in a joint framework to address the multiple instance classification task. In this realm, the *collective assumption* is used to express the relationship between the instance labels and the bag labels, by means of taking the sum as aggregation rule. The proposed model is formulated within a least squares support vector machine setting, where an unsupervised core model (either kernel PCA or kernel spectral clustering) at the instance level is combined with a classification loss function at the bag level. The corresponding dual problem consists of solving a linear system, and the bag classifier is obtained by aggregating the instance scores. Synthetic experiments suggest that SAFE is advantageous when the instances from both positive and negative bags can be naturally grouped in the same cluster. Moreover, real-life experiments indicate that SAFE is competitive with the best state-of-the-art methods.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Multiple instance classification (MIC) refers to a learning task where a training set of bags has associated class labels. Each bag contains multiple instances (i.e. the data vectors), but the label of the individual instances constituting a bag is unknown[1]. This learning setting arises in several domains such as pharmacy, text mining, computer vision and many others. For example, in the drug activity prediction problem chemical molecules which can take different shapes must be classified as good or bad drugs, depending on their ability to bind to a target site. In the image classification problem, an image either belongs or not to a given class, based on a collection of regions defining its visual content.

MIC algorithms can be categorized into two families [2]: instance-space and bag-space methods. In the first paradigm the bag classifier is obtained by aggregating local instance-level decisions, under the assumptions that either only few instances determine the positive label of a bag *(standard hypothesis)* or all instances contribute equally to the bag's label *(collective hypothesis)*. Examples of techniques based on the standard hypothesis are the Diverse Density algorithm (DD) [22], the Expectation Maximization Diverse Density method (EM-DD) [40] and the Multi Instance Support Vector Machine technique (MI-SVM) [4], where the bag-level classifier is defined as the maximum between the instance-level scores. Alternatively, the Wrapper MI method [12] and the SbMil method [5] are based on the collective hypothesis: each instance inherits the label

* Corresponding author.
E-mail address: rocco.langone@esat.kuleuven.be (R. Langone).

[1] Although extensions have also been explored, the MIC problem is generally defined for binary cases.

http://dx.doi.org/10.1016/j.ins.2016.09.060
0020-0255/© 2016 Elsevier Inc. All rights reserved.

of the bag it belongs to, and a (weighted) sum instead of the maximum is used as aggregation rule. The methods being part of the bag-space family consider the bags as a whole and learn a decision function based on the global information conveyed by each bag. This information is extracted either implicitly through the definition of a distance function as in the Earth Movers Distance (EMD) + SVM algorithm [39], or explicitly by means of a mapping from the bag to a feature vector. The mapping can be performed by defining a bag as the average of the instances inside, like in the Simple MI method [10], by representing each bag as an indirected graph [41], by means of vocabularies like in the Bag-of-Words (BoW) algorithms [28] and the methods [6,27], where each bag is mapped into a histogram.

In this paper we introduce a new MIC model formulated within a least squares support vector machine (LS-SVM) setting [30], where clustering of the instances is performed jointly with the classification of the bags. The clustering is done by means of the kernel spectral clustering (KSC) algorithm [1], where the binary clustering model is expressed by a hyperplane in a high dimensional space induced by a kernel (just as in a standard SVM classification setting). This allows to readily build the bag classifier by summing the clustering scores. Furthermore, if we replace KSC with kernel PCA, the classification accuracy of the algorithm remains practically unaltered. Therefore the first two terms in the optimization objective act as a feature extraction process (through either clustering or PCA) which is exploited in the third term to perform the final classification of the bags. For these reasons we call the new method SAFE, i.e. Supervised Aggregated FEature learning. The main characteristics of the proposed approach are the following:

- *one-stage model*: in the vocabulary-based methods like the BoW algorithms, the classification of the bags is performed in two steps. In a first phase the instances are clustered and each bag is represented by a histogram that counts how many instances from the bag belong to each cluster. Afterwards, a standard classifier like SVM [8] or AdaBoost [25] is used to classify the bags. Differently, in the proposed method the clustering and the classification steps are done jointly. Experiments on two synthetic datasets discussed in Section 4.2 suggest that, compared to an alternative two-stage model, the proposed algorithm can achieve a higher accuracy when the instances belonging to either positive or negative bags cannot be naturally clustered together.
- *combining local and global information*: the local information related to the distribution of the instances is learned through either kernel PCA or kernel spectral clustering. The local instance scores are then combined to represent a bag by making use of the collective assumption. This allows to jointly learn the instance clustering model and the bag classifier. In this sense the proposed approach considers at the same time both the characteristics of individual instances (local information) and the characteristics of the bags determined by their labels (global information).
- *computational efficiency*: the bag classification model is obtained by solving a linear system, for which efficient linear algebra libraries, such as *LAPACK*, exist.

The remainder of this paper is organized as follows. In Section 2 some related methods are summarized. Section 3 presents the SAFE algorithm, in Section 4 we delve into the peculiarities of our method by conducting experiments on two synthetic datasets. Section 5 presents the results on real-world datasets. In Section 6 an analysis of the computational complexity is discussed. Finally, Section 7 concludes the article.

2. Related work

2.1. The collective assumption

As mentioned in the previous section, the proposed approach makes use of the collective assumption [2,11]. Under the standard multi-instance assumption only a few positive instances can have any influence on the class label. In contrast, in the collective assumption all instances in a bag contribute equally to the bag's label. The collective assumption is motivated by probability theory considerations. In this view, a bag is not a finite collection of fixed elements as in case of the standard assumption, but instead is modeled as a probability distribution over the instance space, where the observed instances were generated by random sampling from that distribution. Instances are assumed to be assigned class labels according to some unknown probability function and under the collective hypothesis, the bag-level class probability function is determined by the expected class value of the population of that bag.

Although the majority of the methods present in literature are based on the standard hypothesis, a number of algorithms grounded on the collective assumption exist. Examples include the Wrapper MI method [12], the multiple instance learning for sparse positive bags (SbMil) [5], multiple instance learning via successive linear programming [21], logistic regression and boosting for labeled lags of instances [36].

2.2. Joint learning

To the best of our knowledge, methods that try to exploit at the same time both local instance-level information and global bag-level information have been investigated by few researchers and only very recently [15,38] and [3]. In [38] the MILEAGE framework is introduced, where a large margin method is formulated to adaptively tune the importance given to the local and global feature representations when training the bag classifier [15] presented a large-margin formulation which treats the instance labels as hidden latent variables, and simultaneously infers the instance labels as well as the

instance-level classification model. This framework, which learns from label proportions, assumes positive bags to have a large number of positive instances while negative bags to have the fewest ones. This assumption, although reasonable, is less general than the collective hypothesis used in our approach. In fact, in our framework a bag can be classified as positive also when a few positive instances, but with a high clustering score, are present[2]. Furthermore, the proposed algorithm is more efficient than [15], where an alternating optimization procedure is employed. In case of MILDE [3] the authors in a first phase define a discriminative embedding of the original space based on the responses of cluster-adapted instance classifiers, which turns out to be a generalization of the multiple instance learning via embedded instance selection (MILES) method [6]. Afterwards, the classification is performed in a separate stage.

Other related methods are supervised dictionary learning [18] and the algorithm introduced in [13]. In supervised dictionary learning a generative model adapted to the classification task and a standard classifier are learned jointly by means of a block coordinate descent algorithm. In [13], kernel principal component analysis is used to define a projection constraint for each positive bag, with the aim to classify its constituent instances far away from the separating hyper-plane while placing positive instances and negative instances at opposite sides.

3. Proposed approach

3.1. Model

Given a training set of N_B bags and their corresponding labels $\mathcal{D}_B = \{(\mathcal{X}_1, \mathbf{y}_1), \ldots, (\mathcal{X}_{N_B}, \mathbf{y}_{N_B})\}$, the multiple instance classification task is defined as learning a model that is able to predict the class labels of unseen bags. A bag $\mathcal{X}_k$ is a collection of N_k elements $\mathcal{X}_k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_{N_k}^k\}$, where $\mathbf{x}_j^k \in \mathbb{R}^d$ are the data vectors or instances constituting the bag, whose label is unknown. The proposed model is based on the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{w},e_j^k,b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} - \frac{\gamma}{2}\sum_{k=1}^{N_B}\sum_{j=1}^{N_k} v_j^k (e_j^k)^2 + \frac{\rho}{2}\sum_{k=1}^{N_B}\left(\sum_{j=1}^{N_k} e_j^k - y_k\right)^2 \\ \text{subject to} \quad & e_j^k = \mathbf{w}^T\varphi(\mathbf{x}_j^k) + b, \quad j = 1, \ldots, N_k \;\; k = 1, \ldots, N_B \end{aligned} \tag{1}$$

where:

- $\mathbf{w} \in \mathbb{R}^{d_h}$ and $b \in \mathbb{R}$ are the model parameters
- v_j^k are properly chosen weights for the instances (see Eq. (2) in the next paragraph and discussion therein)
- $y_k \in \{-1, 1\}$ is the label associated with the bag $\mathcal{X}_k$
- $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the feature map
- $\gamma \in \mathbb{R}$ and $\rho \in \mathbb{R}$ are positive regularization constants.

In the case $\rho = 0$, objective (1) results into the (binary) weighted kernel PCA problem related to the instances, which can be re-written in a simplified form as:

$$\begin{aligned} \min_{\mathbf{w},e_i,b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} - \frac{\gamma}{2}\sum_{i=1}^{N} v_i e_i^2 \\ \text{subject to} \quad & e_i = \mathbf{w}^T\varphi(\mathbf{x}_i) + b, i = 1, \ldots, N \end{aligned} \tag{2}$$

where index i runs from 1 to the total number of instances across all bags, indicated by $N = \sum_{k=1}^{N_B} N_k$. Objective (2) means that one is seeking for the direction $\mathbf{w}$ with a small L_2 norm such that the variance of each mapped instance $\varphi(x_i)$ projected along this direction $\mathbf{w}$, i.e. $e_i = \mathbf{w}^T\varphi(x_i) + b$, is maximized[3]).

We have two main choices regarding the weights v_i:

- $v_i = 1, \forall i$. In this case (2) becomes the kernel PCA formulation given in [31].
- $v_i = \frac{1}{d_i} = \frac{1}{\sum_{r=1}^{N}\Omega_{ir}}, \forall i$, where $\Omega \in \mathbb{R}^{N\times N}$ denotes the kernel matrix, with $\Omega_{ir} = \varphi(x_i)^T\varphi(x_r) = K(x_i, x_r)$. This leads to the kernel spectral clustering (KSC) model [1], which is a spectral clustering algorithm [7,24,33] formulated as a weighted kernel PCA model.

The KSC part present in the objective (1) allows to divide the instances into two clusters. Somehow, the clusters can be considered the words of the vocabulary used to describe the bags, similarly to the BoW methods. The third term in the objective (1), that is $\sum_{k=1}^{N_B}(\sum_{j=1}^{N_k} e_j^k - y_k)^2$, permits to find the combination of cluster 1 and cluster 2 instances that determine a bag to be classified as positive or negative. Here we make the assumption that the (global) bag classification model can be expressed as the sum of the (local) instance clustering scores. In other words, we use the collective hypothesis

[2] This is due to the fact that these high clustering scores can drive the classifier latent variable towards a positive value.

[3] This means that minus the variance should be minimized, hence the "-" sign in the second term of (2).

[2,11] according to which "*all instances in a bag contribute equally to the bag label*". In particular, the latent variable E_k corresponding to bag $\mathcal{X}_k$ is:

$$E_k = \sum_{j=1}^{N_k} e_j^k = \sum_{j=1}^{N_k} \mathbf{w}^T \varphi(\mathbf{x}_j^k) + b. \tag{3}$$

Thus bag $\mathcal{X}_k$ will be classified as positive if $\text{sign}(E_k) > 0$ and negative otherwise. In order to derive the dual solution of problem (1), it is convenient to rewrite it using matrix notation:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{e},b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} - \frac{\gamma}{2}\mathbf{e}^T V \mathbf{e} + \frac{\rho}{2}(J_B^T\mathbf{e} - \mathbf{y_B})^T(J_B^T\mathbf{e} - \mathbf{y_B}) \\ \text{subject to} \quad & \mathbf{e} = \Phi\mathbf{w} + b\mathbf{1_N}, \end{aligned} \tag{4}$$

where:

- $J_B \in \mathbb{R}^{N \times N_B}$ is a bag indicator matrix and $J_{B_{ik}} = 1$ if instance i belongs to bag k and $J_{B_{ik}} = 0$ otherwise
- $\mathbf{e} \in \mathbb{R}^N$ is a column vector containing all the instance scores, that is $\mathbf{e} = [e_1; \ldots; e_i; \ldots; e_N]$
- $\mathbf{y_B} \in \mathbb{R}^{N_B}$ is a column vector containing all bag labels, i.e. $\mathbf{y_B} = [y_1; \ldots; y_k; \ldots; y_{N_B}]$
- $\mathbf{1_N} \in \mathbb{R}^N$ is a column vector of ones
- $\Phi = [\varphi(\mathbf{x}_1)^T; \ldots; \varphi(\mathbf{x}_i)^T; \ldots; \varphi(\mathbf{x}_N)^T]$ denotes the $N \times d_h$ feature matrix.

Objective (4) can be re-written as:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{e},b} \quad & \frac{1}{2}[\mathbf{w}^T \ \mathbf{e}^T]\begin{bmatrix} I & 0 \\ 0 & -\gamma V + \rho J_B J_B^T \end{bmatrix}\begin{bmatrix} \mathbf{w} \\ \mathbf{e} \end{bmatrix} - \frac{\rho}{2}(\mathbf{e}J_B^T\mathbf{y}_B + \mathbf{y}_B^T J_B^T\mathbf{e}) \\ \text{subject to} \quad & \mathbf{e} = \Phi\mathbf{w} + b\mathbf{1_N}. \end{aligned}$$

This problem is convex if the quadratic form in $\mathbf{w}$ and $\mathbf{e}$ is positive definite, which occurs when $-\gamma V + \rho J_B J_B^T$ is a positive definite matrix. This condition is exploited in the model selection to reduce the search space for the tuning parameters[4] γ, ρ and σ. The dual solution to problem (4) is formalized in the following Lemma:

Lemma 1. *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, with $K(\mathbf{x}_i, \mathbf{x}_r) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_r) = \Omega_{ir}$, non-negative regularization constants γ and ρ, $V = D^{-1} = \text{diag}(\frac{1}{d_1}, \ldots, \frac{1}{d_N})$ in case of the KSC based model or $V = I_N$ in case of the KPCA based model, the matrix $B = J_B J_B^T$, the matrix $G = \rho B - \gamma V$, the Karush–Kuhn–Tucker (KKT) optimality conditions of the Lagrangian of (4) result into the following linear system:*

$$\left[\left(I_N - \frac{G\mathbf{1_N}\mathbf{1_N}^T}{\mathbf{1_N}^T G\mathbf{1_N}}\right)G\Omega\right]\alpha = \rho\left(I_N - \frac{G\mathbf{1_N}\mathbf{1_N}^T}{\mathbf{1_N}^T G\mathbf{1_N}}\right)J_B\mathbf{y_B}, \tag{5}$$

where

- $I_N \in \mathbb{R}^{N \times N}$ means the identity matrix
- $\Omega \in \mathbb{R}^{N \times N}$ is the kernel matrix
- $\alpha \in \mathbb{R}^N$ denotes the dual solution vector

Proof. The Lagrangian of problem (4) is:

$$\mathcal{L}(\mathbf{w}, \mathbf{e}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \frac{\gamma}{2}\mathbf{e}^T V\mathbf{e} + \frac{\rho}{2}(J_B^T\mathbf{e} - \mathbf{y_B})^T(J_B^T\mathbf{e} - \mathbf{y_B}) + \alpha^T(\mathbf{e} - \Phi\mathbf{w} - b\mathbf{1_N}).$$

The KKT optimality conditions are:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 \to \mathbf{w} = \Phi^T\alpha,$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{e}} = 0 \to -\gamma V\mathbf{e} + \rho B\mathbf{e} - \rho J_B\mathbf{y_B} + \alpha = 0,$$
$$\frac{\partial\mathcal{L}}{\partial b} = 0 \to \mathbf{1_N}^T\alpha = 0,$$
$$\frac{\partial\mathcal{L}}{\partial\alpha} = 0 \to \mathbf{e} = \Phi\mathbf{w} + b\mathbf{1_N}.$$

From $\mathbf{w} = \Phi^T\alpha$ and $1_N^T\alpha = 0$, the bias term becomes:

$$b = \frac{\rho\mathbf{1_N}^T J_B\mathbf{y_B} - \mathbf{1_N}^T(\rho B - \gamma V)\Omega\alpha}{\mathbf{1_N}^T(\rho B - \gamma V)\mathbf{1_N}}. \tag{6}$$

[4] We have observed that the proposed algorithm has low sensitivity w.r.t. γ.

Eliminating the primal variables $\mathbf{e}$, $\mathbf{w}$, b leads to (5). □

Finally, the estimated class label for all the N_B training bags can be computed as:

$$\hat{\mathbf{y}}_B = \text{sign}(J_B^T(\Omega\alpha + \mathbf{1_N}b)) \tag{7}$$

where $\hat{\mathbf{y}}_B = [\hat{y}_1; \ldots; \hat{y}_{N_B}]$.

3.2. Out-of-sample extension

Eq. (7) allows to calculate the class labels on training data. Concerning the test stage, suppose we have a test set given by $\mathcal{D}_{\text{B}}^{\text{test}} = \{(\mathcal{X}_1^{\text{test}}, \mathbf{y}_1^{\text{test}}), \ldots, (\mathcal{X}_{N_B}^{\text{test}}, \mathbf{y}_{N_B}^{\text{test}})\}$. The class labels for the unseen N_B^{test} test bags can be predicted by projecting the N_{test} test instances into the solution vector $\boldsymbol{\alpha}$ and aggregating (i.e. summing) the scores over each bag. In matrix notation this means:

$$\hat{\mathbf{y}}_B^{\text{test}} = \text{sign}(J_B^{\text{test}^T}(\Omega^{\text{test}}\alpha + \mathbf{1}_{\mathbf{N}^{\text{test}}}b)) \tag{8}$$

where:

- $J_B^{\text{test}} \in \mathbb{R}^{N_{\text{test}} \times N_B^{\text{test}}}$ represents the bag indicator matrix for test instances and $J_{B_{ik}}^{\text{test}} = 1$ if test instance i belongs to test bag k and $J_{B_{ik}}^{\text{test}} = 0$ otherwise
- $\Omega_{\text{test}} \in \mathbb{R}^{N_{\text{test}} \times N}$ is the test kernel matrix and represents the similarity between each pair of test and training instances. In particular $\Omega_{ir}^{\text{test}} = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_r^{\text{test}})$
- $\mathbf{1_N}^{\text{test}} \in \mathbb{R}_{\text{test}}^N$ is a column vector of N_{test} ones.

The entire approach is summarized in Algorithm 1 and the related Matlab package can be downloaded from: http://www.esat.kuleuven.be/stadius/ADB/langone/SAFElab.php

Algorithm 1: SAFE algorithm.

Data: Training set $\mathcal{D}_{\text{B}} = \{(\mathcal{X}_1, \mathbf{y}_1), \ldots, (\mathcal{X}_{N_B}, \mathbf{y}_{N_B})\}$, test set $\mathcal{D}_{\text{B}}^{\text{test}} = \{(\mathcal{X}_1^{\text{test}}, \mathbf{y}_1^{\text{test}}), \ldots, (\mathcal{X}_{N_B}^{\text{test}}, \mathbf{y}_{N_B}^{\text{test}})\}$, positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, kernel parameters (e.g. RBF bandwidth σ), regularization constants γ and ρ.
Compute bag indicator matrix J_B
Calculate kernel matrix Ω
Set either $V = I_N$ or $V = D^{-1}$
Compute matrices $B = J_B J_B^T$ and $G = \rho B - \gamma V$
Compute vector α, solution of problem (5)
Compute bias term b, using eq. (6)
Estimate class labels $\hat{\mathbf{y}}_B$ for training bags, using eq. (7)
Compute bag indicator matrix for test instances J_B^{test}
Calculate test kernel matrix Ω^{test}
Predict class labels $\hat{\mathbf{y}}_B^{\text{test}}$ for test bags, using eq. (8)
Result: Class labels for training and test bags.

3.3. Model selection

In the proposed method, there can be in total three tuning parameters, namely the kernel parameter (for instance the bandwidth σ of the RBF kernel) and the two regularization constants γ and ρ. As in the case of any learning method, in order to obtain meaningful results it is important to do a careful search for these parameters. Throughout the experiments described later we used grid search to tune σ, γ and ρ, where the search space is reduced by exploiting the condition for the convexity of problem (4) given in Section 3.1. An example of tuning related to *toy dataset 2* is shown in Fig. 1. Table 1 reports the selected parameters in case of the real-life datasets described in Section 5.1. From the table it can be noticed how the tuning of γ and ρ allows to automatically give more or less emphasis to the unsupervised learning (first two terms in objective (4)), depending on how much the information provided by either KPCA or KSC affects the final bag classification accuracy.

4. Understanding the algorithm

To gain more insights into the working mechanism of the proposed approach based on solving Eq. (5), we now address the following questions:

- is it essential that in (4) the first two-terms are related to clustering (namely KSC)? Or can we use a general feature extraction method like PCA?

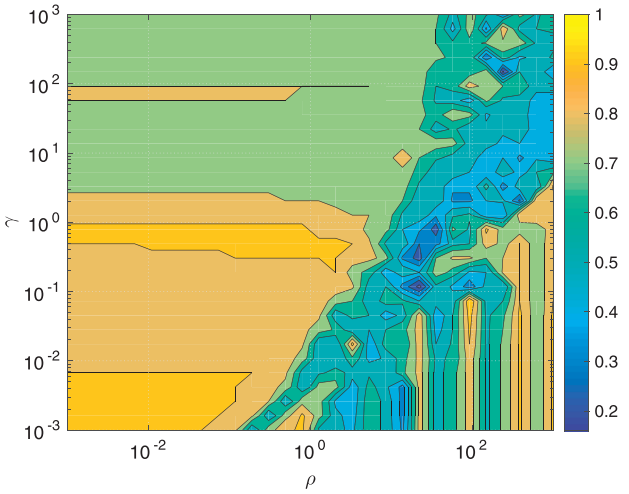


Fig. 1. Tuning example Sensitivity of the proposed approach with respect to the hyper-parameters ρ and γ in case of *toy dataset 2*.

Table 1
Selected hyper-parameters. Selected tuning parameters and runtime (train + test) of the proposed method on the datasets described in Section 5.1. In case of *Text1* and *Text2* datasets the normalized linear kernel has been used, thus σ^2 is not present.

Dataset	γ	ρ	σ^2	**Runtime (s)**
Musk1	20.86	28.57	22.08	0.035
Musk2	0.67	0.09	45.72	20.25
Text1	0.68	0.77	NA	4.52
Text2	0.001	1.22	NA	4.76
Elephant	129.00	14.73	284.36	0.37
Fox	186.65	10.05	348.34	0.23
Tiger	169.44	1.66	681.52	0.41

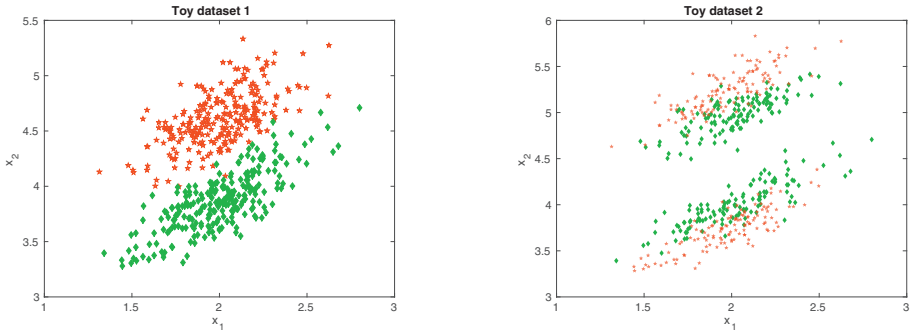


Fig. 2. Toy datasets (Left) Each cluster contains instances from either positive or negative bags **(Right)** Each cluster contains instances from positive and negative bags.

- how does the performance of the proposed method compares to an alternative two-stage algorithm?
- what is the quality of the instance clusters?

To understand these issues we created two toy datasets illustrated in Fig. 2. In both cases, the instances are randomly generated from 2 Gaussian distributions, one for the positive bags (green points) and one for the negative bags (red points). Furthermore, the instances are 2D vectors and each bag is formed by 5 instances. In our notation introduced at the beginning of Section 3.1, $d = 2$, $k = 5$, $N_B = 100$, $N = 500$. The difference between the two datasets lies in the fact that in the first one the instances belonging to either positive or negative bags belong to different clusters, while this is not the case for the second dataset.

Table 2
One-stage versus two-stages algorithm. Comparison of the proposed method (using either KSC or KPCA as core model) against the alternative two-stage algorithm soft KSC + LS-SVM, concerning clustering and classification quality, as measured in terms of adjusted rand index (ARI) and the test accuracy, respectively.

Algorithm	Synthetic dataset **1**		Synthetic dataset **2**	
	ARI	Acc	ARI	Acc
SAFE (KSC)	0.94	100	0.0016	99.1
SAFE (KPCA)	0.94	100	0.0017	99.0
SKSC + LS-SVM	0.67	99.1	0.97	66.2

Table 3
Performance comparison on real-life datasets. Comparison of the proposed method against other state-of-the-art techniques in terms of mean 10-fold cross-validation accuracy.

Dataset	N	N_B	d	SAFE	miFV [34]	miVLAD [35]	Simple MI [10]	Wrapper MI [12]	EM-DD [40]	MI-SVM [4]	MIBoosting [37]
Musk1	476	92	166	**0.92 ± 0.09**	0.91 ± 0.08	0.87 ± 0.09	0.83 ± 0.12	0.85 ± 0.10	0.85 ± 0.09	0.87 ± 0.12	0.84 ± 0.12
Musk2	6598	102	166	**0.89 ± 0.10**	0.88 ± 0.09	0.87 ± 0.09	0.85 ± 0.11	0.79 ± 0.10	0.87 ± 0.11	0.84 ± 0.09	0.79 ± 0.09
Text1	3224	400	66 552	0.94 ± 0.03	0.93 ± 0.04	0.80 ± 0.08	**0.95 ± 0.03**	0.88 ± 0.02	0.87 ± 0.07	0.71 ± 0.04	0.90 ± 0.04
Text2	3344	400	66 553	0.75 ± 0.07	0.79 ± 0.06	0.71 ± 0.05	0.82 ± 0.06	**0.88 ± 0.05**	0.85 ± 0.04	0.72 ± 0.04	0.82 ± 0.05
Elephant	1220	200	230	0.84 ± 0.08	**0.85 ± 0.08**	**0.85 ± 0.08**	0.80 ± 0.09	0.82 ± 0.09	0.77 ± 0.10	0.82 ± 0.07	0.83 ± 0.07
Fox	1320	200	230	0.60 ± 0.10	0.62 ± 0.11	0.62 ± 0.11	0.54 ± 0.09	0.58 ± 0.10	0.61 ± 0.10	0.58 ± 0.10	**0.64 ± 0.10**
Tiger	1391	200	230	**0.84 ± 0.08**	0.81 ± 0.08	0.81 ± 0.08	0.78 ± 0.09	0.77 ± 0.09	0.73 ± 0.09	0.79 ± 0.09	0.80 ± 0.08

4.1. SAFE based on KPCA

To answer the first question we considered an alternative model by having $V = I_N$ in the second term of (4). As a consequence, this method uses kernel PCA [23,26] to model the distribution of the instances instead of kernel spectral clustering. The classification accuracy, which is depicted in Table 2, is the same for both versions of the algorithm. These findings are confirmed also for the real-life datasets (see Table 3).

4.2. Comparison with a two-stage algorithm

Concerning the answer to the second question, we constructed a novel vocabulary-based method that we call SKSC + LS-SVM. In the first stage, a soft KSC (SKSC) algorithm [17] is used to form the N_c concepts of a vocabulary $\mathcal{V}$, that is the N_c clusters of instances and corresponding prototypes. After building-up the vocabulary, we design a mapping function $M(\mathcal{X}, \mathcal{V})$ which, given a bag $\mathcal{X}$ and the vocabulary $\mathcal{V}$, outputs a N_c-dimensional feature vector $\mathbf{v} = (v_1, \ldots, v_{N_c})$. The mapping function is built by computing the average degree of belonging to each cluster for all the instances in a bag. In the second stage, an LS-SVM classifier [30] is trained using as inputs the feature vectors $\mathbf{v_1}, \ldots, \mathbf{v_{N_B}}$ associated to each bag. As shown in Table 2 this method reaches a much lower classification accuracy in the study of the toy dataset 2. This is due to the fact that the SKSC clustering algorithm correctly identifies (as expected) the two clusters of instances (ARI = 0.97), which contains feature vectors from both positive and negative bags. This negatively affects the LS-SVM bag classifier at the next stage. On the other hand, in case of the toy dataset 1, both algorithms achieve a high classification accuracy. This fact can be explained by considering that the instances belonging to either positive or negative bags can naturally be grouped into two distinct clusters.

4.3. Quality of the instance clusters

In Table 2 the adjusted rand index (ARI [14]) is used as metric to evaluate the cluster quality related to the instance clusters provided by the SAFE algorithm and the SKSC method (representing the first stage of the SKSC + LS-SVM algorithm). Moreover, the clustering outcomes provided by the SAFE algorithm and the soft KSC method are visualized at the top and bottom of Fig. 3, respectively. The results indicate that SAFE produces clusters of better/worse quality (i.e. higher/lower ARI) than a standalone clustering algorithm depending on weather the instance clusters reflect/do not reflect the distinction between positive and negative bags[5].

[5] These findings can be explained by considering that the γ constant present in (4) is tuned to maximize the final bag classification accuracy and not the intermediate instance cluster quality.

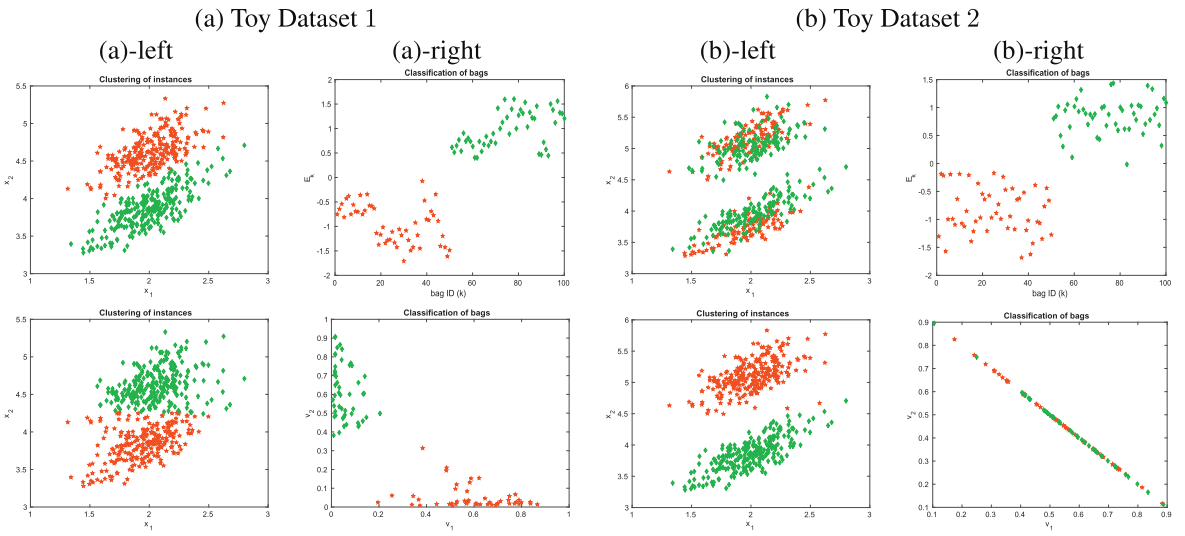


Fig. 3. Synthetic experiments. **(Top)** *a-(left)*, *b-(left)* Clustering results provided by the SAFE algorithm *a-(right)*, *b-(right)* representation of the bags in the space of the classification latent variable of the SAFE algorithm. **(Bottom)** *a-(left)*, *b-(left)* Clustering outcomes produced by the soft KSC method *a-(right)*, *b-(right)* representation of the bags in the embedding provided by soft KSC + LS-SVM algorithm. *Best visible in colors.*

5. Experimental results

5.1. Setup

The performance of the proposed algorithm is compared with the following state-of-the-art methods[6]: miFV [34], miVLAD [35], Simple MI [10], Wrapper MI [12], EM-DD [40], MI-SVM [4], MIBoosting [37]. The evaluation is based on running ten times 10-fold cross validation and reporting the average accuracy. The dataset that have been used belong to three different fields (drug discovery, information retrieval and computer vision) and are standard benchmarks in the MIC literature:

- *Musk1* and *Musk2* [9] relate to classifying molecules as positive or negative. Each molecule can adopt multiple conformations and represents a bag in the MIC setting. Every instance in the bags is represented by 166 features, and an average number of 5 and 65 instances form a bag in *Musk1* and *Musk2* respectively. *Musk1* has 92 molecules, of which 47 are labelled positive, *Musk2* has 102 molecules, of which 39 are positive.
- *Text1* and *Text2* [4] concern classifying a series of documents consisting of *Medline* articles written in 1987. Each article is annotated with Medical Subject Headings (Mesh) terms, each defining a binary concept. A total of 200 positive bags and 200 negative bags are present.
- *Fox*, *Elephant* and *Tiger* [4] regard the classification of 100 positive and 100 negative example images.

5.2. Discussion

In Table 3 the average 10-fold classification accuracy of the various algorithms is reported. We can notice how SAFE is very competitive with the other state-of-the-art methods, although in some cases it reaches a lower performance. On one hand, in many MIL databases there might be a few instances that are especially relevant, but all the instances inside the bag have characteristics that convey information about the fact that the bag is positive. In this cases the collective hypothesis that is used in our approach is more appropriate than the standard hypothesis to take a decision. In particular, our approach is designed in such a way to exploit both global bag level information and local instance level information. As explained in Section 3.1, this is obtained by means of Eq. (1) which allows every bag to be mapped into a proper embedding and classified in one shot. Furthermore, since we solve the dual problem (5), large dimensional data can be readily handled, in contrast to other methods like miFV, miVLAD, EM-DD where dimensionality reduction must be performed first (which may affect the final results). On the contrary, the methods based on the standard hypothesis tend to discard a big part of the information, because only one instance per positive bag is considered in the learning stage. On the other hand, when the composition of a bag is noisy (i.e. a large number of instances do not provide relevant information about the bag), the methods based on the standard hypothesis may be more beneficial because focus on the most positive instance.

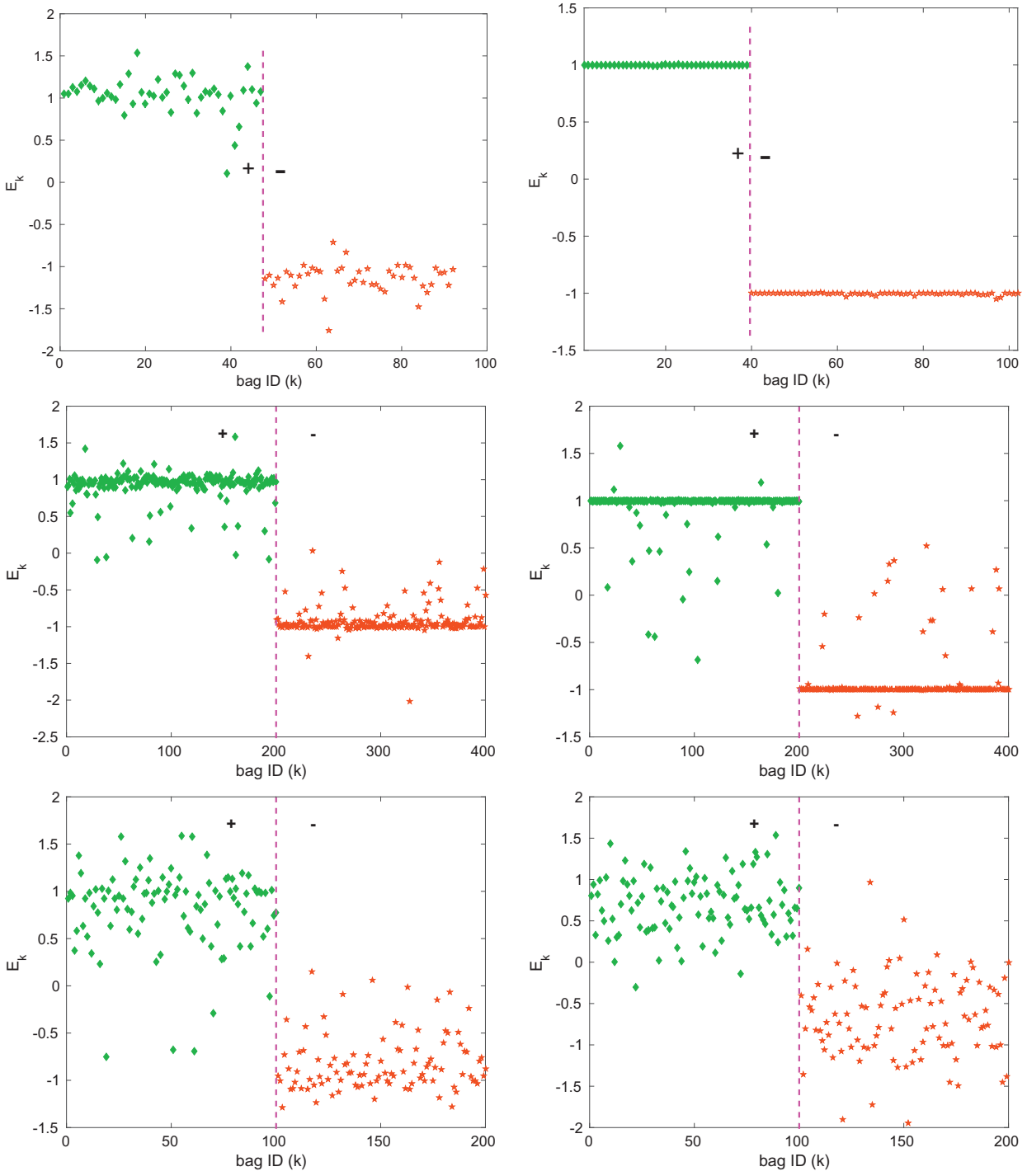


Fig. 4. Representation of the bags in the SAFE classifier latent variable space. (Top) *Musk1* (left) and *Musk2* (right), **(Middle)** *Text1* (left) and *Text2* (right), **(Bottom)** *Elephant* (left) and *Tiger* (right) The different spread around $+1$ and -1 reflects the clustering scores assigned by the algorithm to the instances forming the various bags. The vertical dashed line indicates the true separation, the green color denotes the labels predicted as positive and the red color the labels predicted as negative. Thus, in the ideal case of 100 % accuracy like in *Musk1* and *Musk2* datasets, all the green points should be on the positive side and all the red points on the negative side. The results are related to the run of the 10-fold cross-validation scheme where the best validation accuracy is reached. *Best visible in colors.*

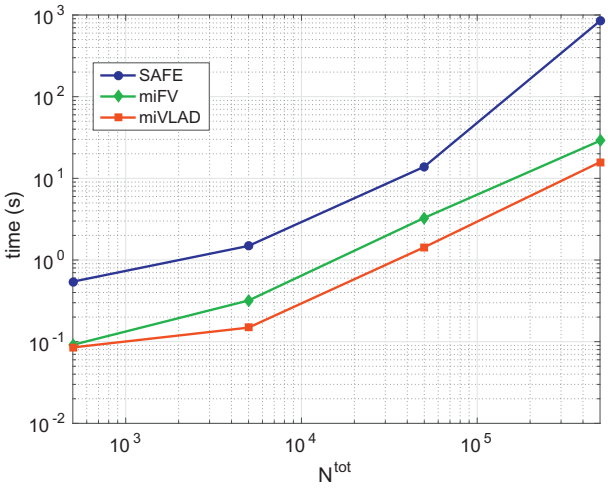


Fig. 5. Computational complexity analysis. Runtime Versus number of instances in the classification of toy dataset 2. The proposed algorithm, although slower than the state-of-the-art algorithms miFV and miVLAD, allows to classify $N^{tot} = 500,000$ instances ($N_B^{tot} = 100\,000$) in about 850 s. *Best visible in colors.*

Table 4
Computational complexity analysis. Average test accuracy of the proposed SAFE algorithm compared to miFV and miVLAD methods related to the analysis of toy dataset 2 shown in Fig. 5.

N^{tot}	SAFE	miFV	miVLAD
500	0.97	**0.98**	0.74
5000	**0.98**	0.95	0.74
50,000	**0.98**	0.95	0.75
500,000	**0.98**	0.94	0.75

In Fig. 4 we depict, for all the datasets, the representation of the bags in the space of the latent variable E_k (see Eq. (3) in Section 3.1). The points belong to both training and validation sets concerning the best run of the 10-fold cross-validation procedure. Only in case of *Musk1* and *Musk2* datasets a perfect separation between the two classes can be achieved. In Fig. 4 we can also notice how the bags are mapped at different distances from the decision boundary $E_k = 0$. This is due to their distinct composition in terms of positive and negative instances.

6. Computational complexity

The training complexity of Algorithm 1 is dominated by time needed to solve linear system (5), while the construction of the $N \times N$ matrices Ω, B, G requires the most memory, leading to a space complexity of $O(N^2)$. Given a total number of data instances denoted by N^{tot}, the total runtime of Algorithm 1 can be decomposed as: $O(N^2) + O(N * N^{test})$. The first part is related to solving Eqs. (5) and (7), and the second part is due to the computation of Eq. (8). Thus, given $N^{tot} = N + N^{test}$ data instances, there are two scenarios: (i) $N \approx N^{test} \approx N^{tot}$; the complexity is quadratic in the total number of instances N^{tot} (ii) $N \ll N^{test} \approx N^{tot}$; the complexity is linear in the total number of instances. The second case corresponds to using a small training set to construct the MIC model and the out-of-sample extension property to predict the labels for the remaining data. This methodology has already been shown to be successful in dealing with large datasets in LS-SVM regression, classification [20] and clustering [19] problems. To have a better understanding of the efficiency and scalability of the proposed method, in Fig. 5 an analysis of the relationship between runtime and the number of instances N^{tot} is illustrated. The toy dataset 2 has been used for the study. The training set size[7] has been set as $N = \min(0.1 * N^{tot}, 5000)$, and stratified random sampling has been used to select training and test sets. Fig. 5 shows that SAFE can be scaled to large datasets and is competitive with the most efficient MIL algorithms such as miFV [34] and miVLAD [35]. Finally, Table 4 reports the average test accuracy (over 10 randomizations) of SAFE, miFV and miVLAD algorithms corresponding to the simulations depicted in Fig. 5.

[6] The Matlab implementation of these algorithms can be downloaded at: http://lamda.nju.edu.cn/Data.ashx

[7] This choice is due to the amount of available RAM in the PC used for the experiments.

Fig. 5 suggests also that the runtime is in line with, but not better, than the other leading techniques. In order to make the proposed approach more competitive, the Nyström method could be used. The Nyström method has been shown to be a useful tool to scale-up kernel-based algorithms such as SVM, Gaussian processes, kernel principal component analysis [29]. When N is large, replacing the kernel matrix with its low-rank approximation can remarkably reduce the computational cost of computing the dual solution. Furthermore, an approximate explicit feature map corresponding to a given kernel can serve as a basis for reducing the cost of learning nonlinear classification models with large datasets, as shown in [32] and in case of fixed-size methods [16,30].

7. Conclusion

In this paper we have proposed a new multiple instance classification model called SAFE, which combines in a joint framework both local instance level and global bag level information. The model is formulated as a constrained optimization problem, and the algorithm consists of solving a linear system at the dual level. In a number of experiments we have shown how this simple design allowed to obtain high classification accuracy at a low computational cost. Future work may be related to (i) improving the accuracy of the proposed method by either considering a robust loss function instead of the L_2 loss in the primal optimization objective or by using the *max* instead of the *sum* as aggregation rule (ii) decreasing the computational cost by means of the Nyström method, as discussed at the end of Section 6 (iii) extending the approach to deal with multi-instance clustering and multi-class classification.

Acknowledgments

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017).

References

[1] C. Alzate, J.A.K. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA, TPAMI 32 (2) (2010) 335–347.
[2] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105.
[3] J. Amores, Milde: Multiple instance learning by discriminative embedding, Knowl. Inf. Syst. 42 (2) (2015) 381–407.
[4] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: NIPS, 2003, pp. 561–568.
[5] R.C. Bunescu, R.J. Mooney, Multiple instance learning for sparse positive bags, in: ICML, 2007, pp. 105–112.
[6] Y. Chen, J. Bi, J. Wang, Miles: Multiple-instance learning via embedded instance selection, TPAMI 28 (12) (2006) 1931–1947.
[7] F.R.K. Chung, Spectral Graph Theory, 1997.
[8] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
[9] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1–2) (1997) 31–71.
[10] L. Dong, A comparison of multi-instance learning algorithms, Master's thesis, University of Waikato (2006).
[11] J. Foulds, E. Frank, A review of multi-instance learning assumptions, Knowl. Eng. Rev. 25 (01) (2010) 1–25.
[12] E. Frank, X. Xu, Applying propositional learning algorithms to multi-instance data, Technical report, University of Waikato (2003).
[13] Y. Han, Q. Tao, J. Wang, Avoiding false positive in multi-instance learning, in: Advances in Neural Information Processing Systems 23, 2010, pp. 811–819.
[14] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 1 (2) (1985) 193–218.
[15] K.-T. Lai, F. Yu, M.-S. Chen, S.-F. Chang, Video event detection by inferring temporal instance labels, in: CVPR, 2014, pp. 2251–2258.
[16] R. Langone, R. Mall, V. Jumutc, J.A.K. Suykens, Fast in-memory spectral clustering using a fixed-size approach, in: Proceedings of the 24 European Symposium on Artificial Neural Networks, Computational Intelligence and machine Learning (ESANN), 2016, pp. 557–562.
[17] R. Langone, R. Mall, J.A.K. Suykens, Soft kernel spectral clustering., in: IJCNN 2013, 2013, pp. 1–8.
[18] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F.R. Bach, Supervised dictionary learning, in: Advances in Neural Information Processing Systems 21, 2009, pp. 1033–1040.
[19] R. Mall, R. Langone, J.A.K. Suykens, Multilevel hierarchical kernel spectral clustering for real-life large scale complex networks, PLoS ONE 9 (6) (2014) e99966.
[20] R. Mall, J.A.K. Suykens, Very sparse LSSVM reductions for large scale data, Trans. Neural Netw. Learn. Syst. 26 (5) (2015) 1086–1097.
[21] O.L. Mangasarian, E.W. Wild, Multiple instance classification viaâ successiveâ linearâ programming, J. Optim. Theory Appl. 137 (3) (2007) 555–568.
[22] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: NIPS, 1998, pp. 570–576.
[23] S. Mika, B. Schölkopf, A.J. Smola, K.R. Müller, M. Scholz, G. Rätsch, Kernel PCA and de-noising in feature spaces, NIPS, 1999.
[24] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), NIPS, Cambridge, MA, 2002, pp. 849–856.
[25] R. Rojas, Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting, Tech. Rep. FUB (2009) 1–6.
[26] B. Schölkopf, A.J. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[27] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms, TPAMI 29 (3) (2007) 411–426.
[28] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: ICCV, vol. 2, 2003, pp. 1470–1477.
[29] S. Sun, J. Zhao, J. Zhu, A review of nyström methods for large-scale machine learning, Inf. Fus. 26 (C) (2015) 36–48.
[30] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.
[31] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, A support vector machine formulation to PCA analysis and its kernel version, IEEE TNN 14 (2) (2003) 447–450.
[32] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 480–492.
[33] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[34] X.S. Wei, J. Wu, Z.H. Zhou, Scalable multi-instance learning, in: 2014 IEEE International Conference on Data Mining, 2014, pp. 1037–1042.

[35] J.W. Xiu-Shen Wei, Z.-H. Zhou, Scalable algorithms for multi-instance learning, Trans. Neural Netw. Learn. Syst. (TNNLS) in press (2016).
[36] X. Xu, E. Frank, Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26–28, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 272–281.
[37] X. Xu, E. Frank, Logistic regression and boosting for labeled bags of instances, in: Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining, 2004, pp. 272–281.
[38] D. Zhang, J. He, L. Si, R.D. Lawrence, Mileage: Multiple instance learning with global embedding., in: ICML, vol. 28, 2013, pp. 82–90.
[39] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, CVPRW, 2006.
[40] Q. Zhang, S.A. Goldman, EM-DD: an improved multiple-instance learning technique, in: NIPS, 2001, pp. 1073–1080.
[41] Z.-H. Zhou, Y.-Y. Sun, Y.-F. Li, Multi-instance learning by treating instances as non-i.i.d. samples, in: ICML, 2009, pp. 1249–1256.