TensMIL2: Improved Multiple Instance Classification Through Tensor Decomposition and Instance Selection

Thomas Papastergiou
Computer Engineering and Informatics Department
University of Patras
Rio, Achaia, Greece
papastergiou@ceid.upatras.gr

Evangelia I. Zacharaki
Department of Electrical and Computer Engineering
University of Patras
Rio, Achaia, Greece
ezachar@upatras.gr

Vasileios Megalooikonomou
Computer Engineering and Informatics Department
University of Patras
Rio, Achaia, Greece
vasilis@ceid.upatras.gr

Abstract— **Multiple instance learning (MIL) has shown great potential in addressing weakly supervised problems in which class labels are provided for sets (bags) of instances. The main challenge in MIL comes from the lack of knowledge on the pertinence of each individual instance in class discrimination. In this paper we propose TensMIL2, a generic unsupervised feature extraction procedure based on non-negative PARAFAC (CP) decomposition, combined with instance selection and MIL classification, that is efficient also for partially observed datasets. Evaluation of our algorithm in standard MIL benchmark datasets showed that TensMIL2 is performing better than state-of-the-art algorithms in most of the cases. Moreover, the comparison of the proposed feature representation via CP decomposition to the previously used features, showed an increase in performance in most of the cases, in both full and partially observed (90% missing values) datasets.**

Keywords— multiple instance learning, constrained PARAFAC (CP) tensor decomposition, image classification

I. Introduction

In many real-life applications data tend to be complex, incorporating different concepts, represented as a collection of features vectors, each one covering an aspect of the sample (e.g. patches of an image or paragraphs of a text). This fact, led to the introduction of Multiple Instance Learning (MIL) and was first applied by Dietterich *et al.* in [1] for drug activity prediction. MIL is a form of weakly supervised learning, where each observation (called a bag) contains several feature vectors (called instances). In the MIL setting labels are provided only for the bags, while class labels of the individual instances are unknown. Furthermore, some of the feature vectors could provide none or sometimes even misleading information about the respective bag's class.

Many algorithms have been proposed for the MIL setting and according to the taxonomy proposed by Amores [2] MIL algorithms can be classified into three paradigms: instance-space, bag-space and embedded- or vocabulary-based. In the instance-space paradigm inference about the bag-labels is drawn by considering the predictions of the instances [1, 3], while in the bag space paradigm, a similarity function is defined and the unknown bag labels are inferred using only bag information. In the embedded-space paradigm the instances are mapped to a new concepts' space, where each feature is a compact representation of the bag [4].

Multidimensional data are very common in the signal processing field like physiological signals [5], signals from gyroscopes and accelerometers for activity recognition [6], signals from EEG recordings for neurophysiological monitoring [7] or color images and video [8]. Often, malfunction of the recording devices or corrupted measurements due to high noise levels can result in partially observed data sets. In such cases, inference of the missing values is often necessary, which can be achieved e.g. with tensor completion techniques [9], or feature extraction can be based only on the observed values [5].

In this work we propose as extension of our previous work [5], a weakly supervised feature extraction and MIL classification method, called TensMIL2, that relies on tensor decomposition with non-negativity constraints and instance selection, and that can handle full or partially observed multidimensional data, like color images. In this approach, applicable for ordered classes, a high dimensional dictionary is constructed from a set of unlabeled observations using the PARAFAC decomposition and used for feature extraction. Robust regression is then applied to map the obtained feature vectors to class probability scores, while the estimated confidence intervals for the predicted instance responses are exploited for instance selection. Furthermore, a fusion process is applied to obtain a bag representation followed by Quadratic Discriminant Analysis for final label prediction.

The main contributions of this work are summarized as follows:

- Unsupervised feature extraction using PARAFAC decomposition with non-negativity constraints
- Incorporation to the basic TensMIL algorithm [5] of an instance selection phase for selecting the most informative instances inside each bag
- Improvement of classification accuracy over previously used features on classical MIL algorithms
- Outperformance of TensMIL2 in the majority of the cases over classical state-of-the-art MIL algorithms on common benchmark datasets
- Comparable performance of TensMIL2 using partially observed data (90% of missing values) with competitive algorithms using full values

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-9-0827-9703-9/19/$31.00 ©2019 IEEE

II. Related Work

MIL algorithms have been widely studied in the past years with characteristic examples the embedded space algorithms MILES [10], JC2MIL [4], MILBoost [11] and MCILBoost [12] or other weakly supervised algorithms like in [13]. In MILES [10], bags are embedded in features space using an instance similarity measure and then relevant features are selected using 1-norm SVM. In JC2MIL [4] bags are embedded in a concepts' space via instance clustering while the problem of the embedding in the new space is jointly solved with the problem of the classification. Cost functions from the MIL literature are combined with the AnyBoost framework in MILBoost [11], whereas MCILBoost [12] performs image (i.e. bag)-level classification, medical image segmentation and patch (i.e. instance)-level clustering in one framework. Recently two MIL architectures based on neural networks [14] and on attention-based deep neural networks [15] have been proposed for end-to-end training and to gain insight into each instance's contribution to the bag label, respectively.

In the field of feature extraction and classification based on tensor decomposition, High Order Discriminant Analysis (HODA) [16] is proposed for image and motor imagery classification. Recently, a framework from common and individual features extraction using non-negative PARAFAC and LL1 tensor decompositions is proposed in [8]. For a comprehensive review on tensor decompositions and machine learning we refer to the extensive review paper [17].

III. Notation and preliminaries

A. The PARAFAC decomposition

A tensor of order *N* is a *N*-dimensional array. We denote tensors by boldface Euler letters $(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Z}})$, matrices by boldface capital letters (**A**, **B**, **C**), arrays by boldface lowercase letters (**x**, **y**, **z**) and scalars by lowercase italic letters (*a, b, c*). We refer to an element of a tensor of order *N* using *N* indices $x_{i_1,i_2,\dots,i_N}$. Columns of a matrix are denoted by a boldface capital letter indexed by a star and a number (e.g. $\boldsymbol{A}_{*,1}$ refers to the first column of matrix $\boldsymbol{A}$). We briefly outline the CANDECOMP/PARAFAC decomposition referred also as CP decomposition. For a comprehensive review on tensor decompositions the interested reader is referred to [18]. Without loss of generality we will refer to tensors of order 3 for the sake of simplicity. Let $\boldsymbol{\mathcal{X}}$ be a 3rd order tensor of size $I \times J \times K$. The CP decomposition of $\boldsymbol{\mathcal{X}}$ consists in expressing the tensor as a sum of *R* rank-1 tensors

$$\boldsymbol{\mathcal{X}} \approx \sum_{\mathrm{r}=1}^{\mathrm{R}} \mathbf{U}_{*,\mathrm{r}} \circ \mathbf{V}_{*,\mathrm{r}} \circ \mathbf{W}_{*,\mathrm{r}} \qquad (1),$$

where matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ are of sizes $I \times R, J \times R, K \times R$, respectively. Each matrix corresponds to one of the tensor's dimensions having as columns all the components of the corresponding dimension. R is the rank of the CP decomposition and "°" refers to the outer product of vectors.

In order to compute a CP decomposition, the following optimization problem must be solved:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{W}} \left\| \boldsymbol{\mathcal{W}} \circledast \left(\boldsymbol{\mathcal{X}} - \sum_{\mathrm{r}=1}^{\mathrm{R}} \mathbf{U}_{*,\mathrm{r}} \circ \mathbf{V}_{*,\mathrm{r}} \circ \mathbf{W}_{*,\mathrm{r}} \right) \right\|_{\mathrm{F}}^{2} \qquad (2)$$

where $\boldsymbol{\mathcal{W}}$ is an indicator tensor of size $I \times J \times K$ such that $\boldsymbol{\mathcal{W}}(i,j,k) = 1$, $\forall (i,j,k) \in \Omega$ and 0 otherwise, $\Omega \subseteq \{1,2,\dots,I\} \times \{1,2,\dots,J\} \times \{1,2,\dots,K\}$ is the set of the observed indices of tensor $\boldsymbol{\mathcal{X}}$ and the symbol $\circledast$ denotes the Hadamard (element-wise) product. When Ω is equal to the set of indices of tensor $\boldsymbol{\mathcal{X}}$, we have a full values problem, otherwise we have a missing values decomposition problem. We can additionally add constraints to the optimization problem (2), such as non-negativity, sparsity or orthogonality constraints to the factors.

For calculating the CP decomposition, we exploit the well-known Alternating Least Squares (ALS) algorithm [18] for the case of full values problem, the Proximal methods proposed in [19] for the case of missing values, and the Multiplicative Update (MU) algorithm [20] in the case of full values problem with non-negativity constraint imposed on all factors. Finally for the case of non-negativity constrains in the missing values setting we exploit the GenProxSGD algorithm [19] using a non-negativity projection on each update of the SGD algorithm, used for computing a proximal step of the algorithm. The update rule we used is:

$$\vartheta^{t+1} = \Pi(\vartheta^{t} - \eta_t |\Omega| \nabla \mathcal{L}_{x_{ijl}}(\theta)) \qquad (3),$$

where Π is the non-negativity projection expressed by the rectifier activation function. Further details on the GenProxSGD algorithm are included in [19].

B. MIL problem statement

We define formally the MIL problem. A bag $B_i = \{\boldsymbol{x}_{i,j}, j = 1, \dots, n_i\}, i = 1, \dots, n$ is a set of n_i instances in the *d*-dimensional space and *n* is the number of bags. The cardinality of B_i can vary across the bags. The provided labels $Y_i \in \mathcal{Y} = \{1, 2, \dots, C\}$ live in the bag space, where C is the number of classes. The labels of the individual instances are not known; in our approach, we make the weak assumption that the instances inherit the label of the bag. The objective of a MIL problem is given a collection of n bags and their corresponding labels $\{(B_i, Y_i), i = 1, \dots, n\}$ to learn a model for predicting the labels of new unseen bags.

IV. TensMIL2

In this paper, we propose a feature extraction and instance selection mechanism along with a multiple instance classification method as an extension of our previous work [5], and show that it can improve the accuracy in the image classification problem. TensMIL2 consists of three components: (1) feature extraction from raw data based on the CP decomposition, (2) instance-level responses' prediction and instance-selection and (3) the instance responses' fusion and bag label prediction. In addition, an external Bayesian optimization procedure [21] is implemented for estimating the set of hyperparameters.

A. Feature Extraction

In order to extract MIL features from a collection of multi-channel (color) images, we first divide the image in $p \times p$ equal-sized patches of size $k \times m \times l$ pixels, where $l = 3$ corresponds to the 3 RGB color channels. Each of the $k \times m$ patches is subsequently vectorized in each different color channel to form a $\boldsymbol{M} = (k \cdot m) \times 3$ matrix representing each instance. All matrices $\boldsymbol{M}$ corresponding to the training and testing instances are concatenated to form a 3rd-order tensor $\boldsymbol{\mathcal{X}}$ of dimensionality $I \times J \times K$, where $I = p^2 \cdot n$, $J = k \cdot m$, $K = 3$, with n being the number of images. In that arrangement the first dimension is dedicated to the instances, the second dimension to the spatial information of the image (pixels) and the third dimension to the RGB channels.

We can write the CP decomposition of (1) in mode-1 slices corresponding to each instance as follows:

$$x_{i,*,*} \approx \sum_{r=1}^{R} u_{ir}\left(\boldsymbol{V}_{*,r} \circ \boldsymbol{W}_{*,r}\right), i = 1,2,\dots,I \qquad (4)$$

Each instance is approximated by a linear combination of R 2-dimensional components that form a high-dimensional dictionary. Thus, we can choose as instance-level features the i^{th} row of the factor $\mathbf{U}$ corresponding to the instances' dimension. This feature extraction procedure can be employed to tensors of order $N \geq 3$, producing dictionaries of order $N-1$.

After having computed the CP decomposition of rank R we perform Principal Component Analysis (PCA) on the extracted features, i.e. on the matrix $\mathbf{U}$ corresponding to the instances' dimension. The retained variance (percentage p) is a crucial parameter that must be adjusted during the hyperparameter tuning phase discussed in section IV-D. After PCA the truncated training and testing matrices are obtained and used to fit an instance-level regression model as discussed next.

B. MIL regression on the instance-level responses

At the second phase we build a full quadratic regression model $f: \mathbb{R}^d \rightarrow \mathbb{R}$ (containing squared terms, interactions, linear terms and an intercept) to calculate the instances' responses using the squared error loss function. In order to train the instance-level regression model, we assume that all instances inherit the label from the bag they correspond to. To overcome this weak assumption, since many of the instances will behave as outliers, we employ robust regression which uses iteratively reweighted least squares with a logistic weighting function [22]. Details on the weighting function and the rest of the model parameters can be found in [5].

After constructing the regression model, instance selection is performed based on the certainty of the predictions. As prior knowledge does not exist, we obtain a certainty measure by examining the distribution of the residual intervals, of the instance level responses, $\gamma_{ij} = b_{ij} - a_{ij}$, where $[a_{ij}, b_{ij}]$ is the 95% confidence interval of the true mean response of instance j inside bag i. A large value of γ_{ij} suggests that the true response value lies on a broader interval, meaning that the confidence on the predicted response is low. On the other hand, a small value of γ_{ij} suggests that the regression model predicted the response with higher confidence. Thus, when selecting the instances for which γ_{ij} is larger than a threshold (thr), we can discard ambiguous instances. By tuning the threshold, the amount of retained information can be adjusted. In practice this threshold depends on the dataset and the amount of irrelevant (background) information it contains. Therefore, instead of fixing this value, it forms one of the hyperparameters of the method and is optimized using the validation set. Since the distribution of γ might vary significantly for every experiment, it is more stable to use as cutting point the value corresponding to the q-quantile of the cumulative distribution of γ, and optimize over q instead of thr.

The proposed instance selection criterion was visually confirmed by experiments on the natural images classification problem, which showed that the majority of the discarded instances corresponded to the image background. An example is shown in Fig. 1 where three bags from the positive class of the Tiger are depicted. We observe that the discarded patches correspond mainly to the background of the image. Based on these empirical results it is expected that instance selection can increase the efficiency of classification.

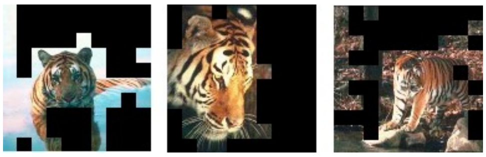

Figure 1 Instance selection: The discarded patches with low confidence on the prediction are shown in black

C. MIL classification: Bag labels predictions

After training the robust instance-level regression model, a fusion of the individual predictions is required in order to extract bag-level information from the selected instances. We treat the instance-level responses as random variables and approximate their cumulative density function as the cumulative histogram of the responses inside each bag $\mathcal{H}_i$ using ϑ_H bins of equal size ($\vartheta_H = 8$).

Having computed the cumulative histograms of the instance label predictions, a bag-level classifier $F: \mathbb{R}^{\vartheta_H} \rightarrow \{1, 2, \dots, C\}$ can be trained using as feature representation the histograms $\mathcal{H}_i$. Assuming that the observations of each class are drawn from a multivariate Gaussian distribution, a pseudo-Quadratic Discriminant Analysis classifier can be trained assigning each bag to the class with the maximum discriminant score, as described in [5].

D. Hyperparameter tuning phase

As mentioned earlier, the proposed method has two hyperparameters: q (the quantile defining the threshold for the instance selection) and p (the retained variance in the dimensionality reduction and decorrelation phase). In order to tune these parameters we employ the Bayesian optimizer [21] using as objective function the mean two-fold Cross Validation (CV) error. After computing the hyperparameters we use them for training our classifier and asses it on a separate test set.

V. Results and Discussion

For the evaluation of the proposed method we used the classical MIL benchmark data sets: Tiger, Fox, Elephant introduced by Andrews et al. [3]. Each of these datasets consists of 100 target animals (Tiger, Fox, Elephant) from the COREL dataset, that forms the positive class. The negative class consists of 100 animals randomly selected from the COREL database. There are multiple challenges in these datasets: (1) the animals are in different poses and in different backgrounds, (2) especially the Fox dataset includes animals with different phenotype (white and orange foxes), (3) the negative class is a mixture of different animals, (4) the background in the negative class often is similar to the background of the positive class.

We evaluate TensMIL2 with full and missing values against state-of-the-art MIL algorithms including classical MIL algorithms like MILES [10], JC2MIL [4], MILBoost[11], MCILBoost [12], as well as MIL approaches that are based on neural networks [14] or on deep neural networks [15]. Furthermore, we evaluate the extracted features via the CP decomposition when exploited by state-of-the-art classification algorithms with full or partially observed values. In all experiments we partitioned the raw images in 10×10 equal-sized patches. The rank for all the CP decompositions is set to $R = 60$. For computing the CP decomposition with full values, we employed the ALS algorithm from the Tensor Toolbox for MATLAB[1], while for computing the non-negative CP decomposition with full values we employed the Multiplicative Update (MU) algorithm [20]. In the case of missing values, in all our experiments we discarded uniformly at random 90% of the values of the tensor. For computing the CP decomposition with missing values, we employed the GenProxSGD [19] while for calculating the non-negative CP decomposition we employed the GenProxSGD with non-negativity projection, as described in section III-A.

For tuning the hyperparameters of TensMIL2 algorithm we employed the Bayesian optimizer [21] with 30 repetitions using 2-fold cross-validation on the training folds. Similarly, for tuning the hyperparameters of the classical MIL algorithms (MILES, JC2MIL, MILBoost and MCILBoost) we performed a grid search on the hyperparameter space with 30 repetitions using 2-fold cross-validation on the training set. The results of the mi-NET, MI-NET and attention-based algorithms were not reproduced but are derived from [15]. In all experiments we report the 10-fold CV accuracy with the standard error of mean indicated inside the parentheses.

A. *Evaluation of TensMIL2 vs state-of-the-art MIL algorithms*

In Table 1 we report the 10-fold CV accuracy for the Tiger-Fox-Elephant datasets. We observe that overall TensMIL2 is achieving better test accuracy using non-negative CP decomposition than the other MIL approaches. Furthermore, we observe that the feature selection phase introduced in TensMIL2 improves the performance of TensMIL by 6%. Regarding the missing values problem, the accuracy of the proposed TensMIL2 algorithm is slightly better with unconstrained CP decomposition than with non-negative constrained decomposition. On the other hand, TensMIL2 with missing values achieves slightly better performance than MILES algorithm with full values.

TABLE I. 10-FOLD CROSS VALIDATION ACCURACY

	10-fold CV accuracy		
	Tiger	*Fox*	*Elephant*
MILES [10]	0.77(0.028)	0.67(0.044)	0.77(0.022)
JC2MIL [4]	0.85(0.019)	0.66(0.025)	0.85(0.028)
MILBoost [11]	0.80(0.022)	0.61(0.051)	0.84(0.022)
MCILBoost [12]	0.78(0.022)	0.59(0.044)	0.81(0.025)
mi-Net [14]	0.82(0.034)	0.62(0.035)	0.86(0.037)
MI-NET [14]	0.83(0.032)	0.63(0.038)	0.87(0.037)
MI-Net with DS [14]	0.84(0.039)	0.64(0.037)	**0.88(0.032)**
MI-Net with RC [14]	0.84(0.037)	0.62(0.047)	0.86(0.040)
Attention [15]	0.84(0.022)	0.62(0.043)	0.87(0.022)
Gated-Attention [15]	0.85(0.018)	0.60(0.029)	0.86(0.027)
TensMIL [5]	0.80(0.038)	0.70(0.038)	0.76(0.022)
TensMIL2	**0.86(0.025)**	0.68(0.028)	0.76(0.032)
TensMIL2-nng	**0.85(0.028)**	**0.71(0.035)**	0.81(0.022)
TensMIL missing 90% [5]	**0.77(0.028)**	**0.68(0.032)**	0.79(0.025)
TensMIL2 missing 90%	**0.78(0.032)**	**0.65(0.032)**	0.75(0.025)
TensMIL2-nng missing 90%	**0.77(0.038)**	**0.62(0.028)**	0.76(0.013)

Specifically, for the Tiger dataset TensMIL2 with unconstrainted CP decomposition achieves the best accuracy and TensMIL2 with non-negativity constraint is as accurate as JC2MIL algorithm. For the Fox dataset, the proposed TensMIL2 algorithm with non-negative CP decomposition performs 1-12% better than all other state-of-the-art algorithms. Regarding the instance selection step introduced in TensMIL2, it improves the accuracy of TensMIL by 1%. For the case of 90% missing values, we see that TensMIL and TensMIL2 perform better than all other investigated algorithms with full values. Finally, for the case of the Elephant dataset the proposed algorithm performs similar or worse than the other algorithms.

In general, we observe that the feature selection scheme with the non-negativity constraint on the CP decomposition, improves the performance of TensMIL algorithm in complete datasets and performs overall better than most of the other algorithms. In the case of incomplete data, we observe that the instance selection step as well as the introduced non-negativity constrains in the CP decomposition are not beneficial. The later fact could be interpreted as follows: The missing values problem is already a constrained CP decomposition problem, since part of the full tensor is observed, and introducing additional non-negativity constraints does not improve the classification accuracy, despite that the non-negative factors make the components more sparse and easier for interpretation.

B. *Comparison of features extraction techniques*

In this set of experiments, we compare the performance of classical MIL classifiers, when using standard features versus our proposed features from unconstrained CP decomposition, in the case of full and incomplete data. In the Tables that follow (II, III, IV) boldface results indicate the best performance along rows.

We observe that for full values (first 2 columns) CP decomposition features improve the classification accuracy in half of the cases for Tiger, in all cases for Fox, and in 3 (out of 4) cases for Elephant. It is also important to note that the proposed features are robust to incomplete data, since results are comparable or even better when only 10% of the data are used for feature extraction. Finally, in case of the Fox dataset we observe that in all cases the CP features improve the classification accuracy of all algorithms from 1.5% to 12%.

TABLE II. TIGER

Tiger	10-fold CV accuracy		
	Andrews [3]	***ALS R=60 10×10***	***GenProSGD R=60 (90% missing values)***
MILES [10]	**0.77(0.028)**	0.75(0.025)	0.76(0.025)
JC2MIL [4]	**0.85(0.019)**	0.74(0.022)	0.79(0.019)
MILBoost [11]	0.80(0.022)	**0.81(0.025)**	0.78(0.035)
MCILBoost [12]	0.78(0.022)	0.76(0.032)	**0.79(0.032)**

[1] Available on line at https://www.tensortoolbox.org/

TABLE III. FOX

Fox	**10-fold CV accuracy**		
	Andrews [3]	***ALS R=60 10x10***	***GenProSGD R=60 (90% missing values)***
MILES [10]	0.67(0.044)	**0.68(0.038)**	0.60(0.032)
JC2MIL [4]	0.66(0.025)	**0.67(0.032)**	0.59(0.041)
MILBoost [11]	0.61(0.051)	**0.70(0.028)**	**0.71(0.022)**
MCILBoost [12]	0.59(0.044)	**0.71(0.035)**	**0.70(0.025)**

In the case of the Elephant with missing values we observe that MILES performs slightly better (0.5%) than with the original full values, while the other algorithms perform from 1.5% to 11.5% poorer, which is a significant finding since we compare here the performance on the full data set against the incomplete dataset.

TABLE IV. ELEPHANT

Elephant	**10-fold CV accuracy**		
	Andrews [3]	***ALS R=60 10x10***	***GenProSGD R=60 (90% missing values)***
MILES [10]	0.77(0.022)	**0.84(0.035)**	**0.77(0.028)**
JC2MIL [4]	0.85(0.028)	**0.86(0.019)**	0.78(0.028)
MILBoost [11]	**0.84(0.022)**	0.80(0.032)	0.73(0.032)
MCILBoost [12]	0.81(0.025)	**0.82(0.025)**	0.80(0.035)

VI. DISCUSSION AND CONCLUSIONS

In this paper we present TensMIL2, an improved version of the TensMIL algorithm [5], in which we introduce instance selection and non-negativity constraints on the feature extraction phase, improving its performance from 1% to 6% in the natural images classification problem. Furthermore, TensMIL2 outperforms many other state-of-the-art MIL algorithms including MIL neural networks and MIL deep networks. Moreover, the assessment of the algorithm on benchmark datasets using only 10% of the values, showed that in some cases it performs even better than the classical MIL algorithms with full values. Furthermore, it was observed that in most of the examined cases (9 out of 12) the CP decomposition feature extraction scheme overall improves the performance of the classifiers even when the proposed features are extracted from incomplete data. In the future, we plan to investigate higher order ($N > 3$) representations of multiple instance datasets and incorporate additional constraints on the CP decomposition factors, as for example sparsity and orthogonality constraints.

ACKNOWLEDGMENT

The first author is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme “Human Resources Development, Education and Lifelong Learning” in the context of the project “Strengthening Human Resources Research Potential via Doctorate Research” (MIS-5000432), implemented by the State Scholarships Foundation (IKY). The second and third authors were supported by the FrailSafe Project [H2020-PHC-21-2015-690140] “Sensing and predictive treatment of frailty and associated co-morbidities using advanced personalized models and advanced interventions”, co-funded by the European Commission under the Horizon 2020 research and innovation program.

REFERENCES

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

[2] Amores, J., Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence, 2013. 201(Supplement C): p. 81-105.

[3] Andrews, S., I. Tsochantaridis, and T. Hofmann, Support vector machines for multiple-instance learning, in Proceedings of the 15th International Conference on Neural Information Processing Systems. 2002, MIT Press. p. 577-584.

[4] Sikka, K., R. Giri, and M.S. Bartlett. Joint Clustering and Classification for Multiple Instance Learning, Proc. BMCV p. 1-12, 2015.

[5] Papastergiou, T., E.I. Zacharaki, and V. Megalooikonomou, Tensor Decomposition for Multiple-Instance Classification of High-Order Medical Data. Complexity, 2018. 2018: p. 13.

[6] Papagiannaki, A., et al., Recognizing Physical Activity of Older People from Wearable Sensors and Inconsistent Data. Sensors, 2019. 19(4): p. 880.

[7] Evangelia, P., et al., EEG-based Classification of Epileptic and Non-Epileptic Events using Multi-Array Decomposition. International Journal of Monitoring and Surveillance Technologies Research (IJMSTR), 2016. 4(2): p. 1-15.

[8] Kisil, I., et al. Common and Individual Feature Extraction Using Tensor Decompositions: a Remedy for the Curse of Dimensionality? in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018.

[9] Porro-Muñoz, D., R.P.W. Duin, and I. Talavera. Missing Values in Dissimilarity-Based Classification of Multi-way Data. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.

[10] Chen, Y., J. Bi, and J.Z. Wang, MILES: Multiple-instance learning via embedded instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. 28(12): p. 1931-1947.

[11] Viola, P., J.C. Platt, and C. Zhang, Multiple instance boosting for object detection, in Proceedings of the 18th International Conference on Neural Information Processing Systems. 2005, MIT Press: Vancouver, British Columbia, Canada. p. 1417-1424.

[12] Xu, Y., et al., Weakly supervised histopathology cancer image segmentation and classification. Medical Image Analysis, 2014. 18(3): p. 591-604.

[13] Karlos, S., et al. Self-Train LogitBoost for Semi-supervised Learning. 2015. Cham: Springer International Publishing.

[14] Wang, X., et al., Revisiting multiple instance neural networks. Pattern Recognition, 2018. 74: p. 15-24.

[15] Ilse, M., J. Tomczak, and M. Welling, Attention-based Deep Multiple Instance Learning, in Proceedings of the 35th International Conference on Machine Learning, D. Jennifer and K. Andreas, Editors. 2018, PMLR: Proceedings of Machine Learning Research. p. 2127--2136.

[16] Phan, A.H. and A. Cichocki, Tensor decompositions for feature extraction and classification of high dimensional datasets. Nonlinear Theory and Its Applications, IEICE, 2010. 1(1): p. 37-68.

[17] Sidiropoulos, N.D., et al., Tensor Decomposition for Signal Processing and Machine Learning. IEEE Transactions on Signal Processing, 2017. 65(13): p. 3551-3582.

[18] Kolda, T.G. and B.W. Bader, Tensor Decompositions and Applications. SIAM Review, 2009. 51(3): p. 455-500.

[19] Papastergiou, T. and V. Megalooikonomou. A distributed proximal gradient descent method for tensor completion. in 2017 IEEE International Conference on Big Data (Big Data). 2017.

[20] Welling, M. and M. Weber, Positive tensor factorization. Pattern Recognition Letters, 2001. 22(12): p. 1255-1261.

[21] Gelbart, M.A., J. Snoek, and R.P. Adams, Bayesian optimization with unknown constraints, in Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence. 2014, AUAI Press: Quebec City, Quebec, Canada. p. 250-259.

[22] Dumouchel, W. and F. O'Brien, Integrating a robust option into a multiple regression computing environment, in Computing and graphics in statistics. 1991, Springer-Verlag New York, Inc. p. 41-48.