Few-Shot Classification Using Tensor Completion

Anastasia Aidini*†, Grigorios Tsagkatakis*†, Nicholas D. Sidiropoulos‡, and Panagiotis Tsakalides*†
*Department of Computer Science, University of Crete, Greece
†Institute of Computer Science, Foundation for Research and Technology Hellas (FORTH), Greece
‡Department of Electrical and Computer Engineering, University of Virginia, VA, USA

***Abstract*—Machine learning is increasingly applied to tackle classification challenges in various real-world contexts. Recently, there has been a growing interest in tensor-based modeling techniques within the machine learning community. In this work, we combine classification and tensor decomposition methods to reformulate the classification problem as a tensor completion task. Specifically, a tensor of scores of the samples is learned, where the sample values correspond to indices in the tensor, containing the scores for each class. Subsequently, when we encounter new data points, we can classify them by extracting the score values from the learned tensor at the indicated sample position, assigning them to the class with the highest score. Given that only a fraction of tensor entries can be obtained from a limited training set of samples, we utilize Tucker decomposition in conjunction with the hinge loss function to complete the score tensor, considering the discrete nature of the predicted class variable. Our experimental results across various real-world classification tasks reveal that this proposed tensor-based learning approach enhances classification performance, especially when dealing with a constrained number of training samples, outperforming state-of-the-art methods.**

***Index Terms*—Classification, tensor completion, Tucker decomposition, hinge-loss function, learning**

I. Introduction

In many practical applications, supervised classification is crucial for analyzing observations [1]. The main objective of classification is to predict a discrete variable that represents the class label assigned to the observation. Various machine learning techniques [2] have been developed to address the classification problem by detecting patterns and similar features in the data and separating them into different classes. Kernel methods [3], random forests [4], and neural networks [5] are powerful machine learning approaches that have proven effective in classification tasks. However, the performance of each method depends on the characteristics of the data, and it is impossible to determine in advance which method will perform best for a given problem. Despite the tremendous success of current neural network-based approaches, particularly in deep learning, their outstanding performance necessitates a large amount of training data.

In machine learning and deep learning, tensor decomposition techniques have been increasingly utilized in a variety of applications, such as regression, classification, and data pre-processing tasks [6]. Among them, Tucker and CANDECOMP/PARAFAC (CP) models [7] play a critical role in feature extraction for classification problems by capturing multi-linear and multi-aspect structures in various high-dimensional datasets [8]–[13]. These models are also useful in deep learning scenarios for accelerating and compressing neural networks [14], [15]. However, the above tensor-based approaches use tensor decomposition techniques for feature extraction, and subsequently, a separate classifier like Support Vector Machines (SVM) or a neural network must be designed for the classification task given the extracted features.

In this work, we introduce a new machine learning approach that employs tensor decomposition for modeling and addressing classification problems. Specifically, our method learns a tensor of scores for the samples, where each sample value serves as an index pointing to "fibers" in the tensor that contain scores for each class. In this setting, training examples correspond to entries of the tensor while inference-time examples are typically empty or missing entries. When presented with new data points, the corresponding tensor cells are queried, and the highest score values determine their class labels, as depicted in Figure 1. While our approach is well-suited for handling discrete input variables, it can also be applied to continuous-valued variables by adopting a suitable basis such as a multivariate Fourier series [16].

Since only a small portion of the tensor entries can be derived from a given training set, we approach the classification problem as a tensor completion task. The idea to model a general nonlinear function using a single high-order tensor with missing values was first introduced in [17] for the regression problem. However, in the classification problem, we want to recover the discrete-valued class labels, and the existing tensor completion methods are intended for recovering real-valued entries [18]–[20]. To address this issue, we propose a new tensor model that captures the interactions between the predictors and recovers the unlabeled samples while considering the discrete nature of the predicted class variable. We achieve this by solving an optimization problem that combines the hinge loss function, which measures the performance of our classification model, and the Tucker decomposition, which imposes a low multi-linear rank constraint to complete the score tensor by extracting correlations from the training data. Experimental results on several real-world classification tasks demonstrate that the proposed tensor-based approach can improve the classification performance of competing methods.

In summary, the main contributions of this work are:

- The formulation of the classification problem as a tensor completion problem, where a tensor of scores for all classes and possible samples is learned.
- The introduction of a new learning method that combines tensor decomposition and machine learning techniques

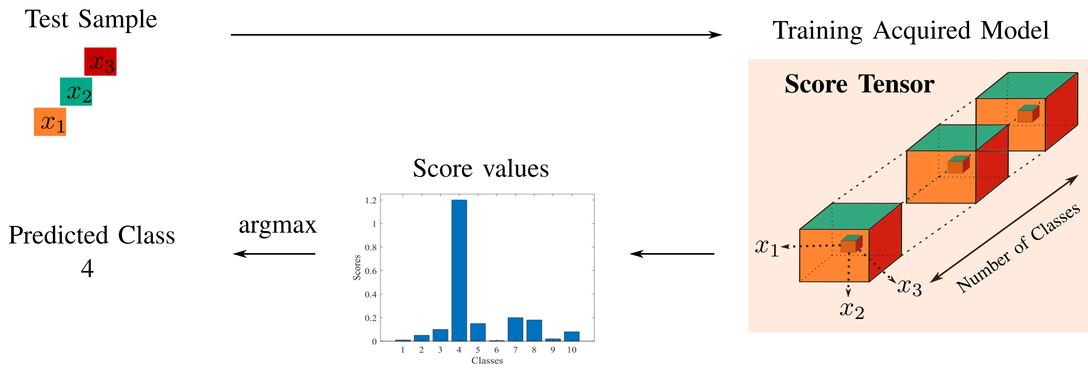

Fig. 1. Flowchart depicting the steps involved in the proposed classification method. To classify a new sample, we extract the scores that correpsond to the last dimension of the learned tensor at the position specified by the sample value. The predicted class for the sample is the one associated with the highest score.

leveraging their strengths in a unified framework.

- The elegance and generality of the proposed approach, which uses the score values of the learned tensor in the position corresponding to the sample values for classifying new samples.
- The ability to achieve satisfactory classification results with a limited number of samples by imposing a low-rank constraint to recover unlabeled data.
- The demonstration of the effectiveness of the proposed method on various real-world classification tasks.

II. Tensor Preliminaries

An N-way or Nth-order *tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as a multidimensional array, whereby the order of a tensor is the number of its dimensions. Extending beyond the matrix format, tensors offer a mathematical approach for representing and handling multi-modal and multi-relational data using tensor decomposition methods [21].

Specifically, the *Tucker decomposition* involves expressing a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ as a core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ multiplied by a matrix $\mathbf{D}_n \in \mathbb{R}^{I_n \times R_n}$ along each mode, i.e.,

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \cdots \times_N \mathbf{D}_N, \tag{1}$$

where the mode-n product $\times_n$ denotes the tensor-times-matrix operation, with elements

$$(\mathcal{G} \times_n \mathbf{D}_n)_{r_1 \ldots r_{n-1} i_n r_{n+1} \ldots r_N} = \sum_{r_n=1}^{R_n} g_{r_1 r_2 \ldots r_n \ldots r_N} \cdot d_{n_{i_n r_n}}.$$

This product can also be expressed as

$$\mathcal{Y} = (\mathcal{G} \times_n \mathbf{D}_n) \Leftrightarrow \mathbf{Y}_{(n)} = \mathbf{D}_n \cdot \mathbf{G}_{(n)}, \tag{2}$$

where $\mathbf{G}_{(n)} \in \mathbb{R}^{R_n \times \prod_{i \neq n} R_i}$ is the mode-n matricization or unfolding of $\mathcal{G}$ and it corresponds to a matrix with columns being the vectors obtained by fixing all indices of $\mathcal{G}$ except the n-th index. Note that the order of the multiplication in a series of distinct mode-n multiplications is irrelevant, i.e., $\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n \mathbf{B} \times_m \mathbf{A}$ $(m \neq n)$, but if the modes are the same, then $\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{B} \cdot \mathbf{A})$.

The N-tuple $(R_1, R_2, ..., R_N)$ that corresponds to the dimensions of the core tensor in the exact Tucker decomposition with orthogonal factor matrices is called the multi-linear rank of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Specifically, the multi-linear rank actually indicates the rank of each mode matricization $\mathbf{X}_{(n)}$ of the tensor, for $n = 1, .., N$ with $R_n \leq I_n$.

III. Proposed Method

To solve the classification problem, we have to predict a discrete integer variable denoting the class label $y \in \mathcal{C} = \{1, .., C\}$ of each sample $\mathbf{x}$, using a number of training pairs $D = \{(\mathbf{x}^s, y^s), s = 1, .., S\}$, where C is the number of different classes. Let us assume that the samples $\mathbf{x} \in \mathcal{I}_1 \times \cdots \times \mathcal{I}_N$ take discrete values from an alphabet $\mathcal{I}_n$ for each variable $n = 1, .., N$.

Inspired by [17], the label y^s of each sample $\mathbf{x}^s$ can be considered as the output of a nonlinear function $f : \mathcal{I}_1 \times \cdots \times \mathcal{I}_N \rightarrow \mathcal{C}$ of the input $\mathbf{x}^s = (x_1^s, .., x_N^s)$, i.e.,

$$f(x_1^s, .., x_N^s) = y^s, \;\; s = 1, .., S \tag{3}$$

that can be modeled as an N-th order tensor $\mathcal{L} \in \mathcal{C}^{I_1 \times \cdots \times I_N}$ that consists of the labels of all samples. Specifically, the tuple of the input variables $(x_1^s, .., x_N^s)$ can be viewed as a cell multi-index of $\mathcal{L}$ and the cell content could be the response y^s of the system. Therefore, the number of dimensions of the label tensor $\mathcal{L}$ is the input size N, and the size of each dimension is the range of input values in the corresponding dimension, i.e., $I_n = |\mathcal{I}_n|$ for $n = 1, .., N$. Then, the label of a new testing sample will be the cell content of the label tensor $\mathcal{L}$ in the corresponding cell depending on the sample values.

However, only a small fraction of the tensor entries are observed during training. Hence, we have to complete the label tensor $\mathcal{L}$ in order to predict the labels of the new samples. To solve this problem, we propose a novel low-rank tensor model, which captures the interactions between the predictors and recovers the unobserved labels by combining a classification loss function with tensor decomposition techniques.

Specifically, to measure the performance of our classification model, we use the hinge loss function defined by

$$l_s = \sum_{c=1, c\neq y_s}^{C} \max(0, a_c - a_{y_s} + \Delta), \tag{4}$$

for each sample $s = 1, .., S$, where Δ is the desired margin between our classes, and a_c, a_{y_s} are the predicted scores of the classification model corresponding to the class c and the true class y_s of the sample, respectively. In our case, the predicted scores are encoded in a $(N{+}1)$-th order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N\times C}$ that has an additional dimension for the scores of each class. In more detail, for each sample $\mathbf{x}_s = (i_1, .., i_N)$ with label $y_s = \mathcal{L}_{i_1..i_N}$, we estimate the scores for each class in a vector of size C, $(\mathcal{A}_{i_1..i_N 1}, .., \mathcal{A}_{i_1..i_N C})$, that corresponds to the last dimension of the score tensor $\mathcal{A}$. Note that the highest score value for each sample indicates the predicted class label.

Therefore, to predict the class labels of new samples, we learn an $(N{+}1)$-th order tensor $\mathcal{A}$ that comprises the score values of all possible samples for the different classes. However, only a small fraction of the score tensor can be easily obtained from the training pairs. To learn the score values of the unobserved samples, we complete the score tensor $\mathcal{A}$ by imposing a low-rank constraint on it to extract the correlations in the training data. Specifically, we use the Tucker decomposition of the desired tensor $\mathcal{A}$ to impose its low multi-linear rank retaining the structure of the score tensor. Therefore, we seek to minimize the hinge loss of the score tensor $\mathcal{A}$ and its Tucker decomposition $\mathcal{A} = \mathcal{G}\times_1\mathbf{D}_1\times_2\cdots\times_N\mathbf{D}_N\times_{N+1}\mathbf{D}_{N+1} \in \mathbb{R}^{I_1\times\cdots\times I_N\times C}$ on the given training data, i.e.,

$$\begin{gathered}\min_{\mathcal{A},\mathcal{G},\mathbf{D}_n} \frac{1}{S} \sum_{i=1, i\in\Omega}^{I_1\cdots I_N} \sum_{j=1, j\neq y_i}^{C} \max(0, \mathbf{A}_{(N+1)_{ji}} - \mathbf{A}_{(N+1)_{y_i i}} + \Delta) \\ +\tfrac{\lambda}{2}\|\mathcal{A} - \mathcal{G}\times_1 \mathbf{D}_1 \times_2 \cdots \times_{N+1} \mathbf{D}_{N+1}\|_F^2 \\ \text{subject to } \mathbf{D}_n^T \cdot \mathbf{D}_n = \mathbf{I}_{R_n},\ n = 1, .., N{+}1 \end{gathered} \tag{5}$$

where $\mathbf{A}_{(N+1)} \in \mathbb{R}^{C\times(I_1\cdots I_N)}$ is the mode-$(N{+}1)$ matricization of the score tensor $\mathcal{A}$, with elements $\mathbf{A}_{(N+1)_{ji}}$ for $j = 1, .., C$ and $i = 1, .., (I_1\cdots I_N)$, and Ω indicates the vectorized indices of the given training samples that correspond to the first N dimensions of the score tensor. Additionally, $\mathcal{G} \in \mathbb{R}^{R_1\times\cdots\times R_N\times R_{N+1}}$ is the core tensor of the Tucker decomposition, $\mathbf{D}_n \in \mathbb{R}^{I_n\times R_n}$ with $R_n \leq I_n$ for $n{=}1, .., N{+}1$ are the factor matrices that are restricted to be orthogonal, $\mathbf{I}_{R_n}$ is the identity matrix with dimensions $R_n\times R_n$, where $(R_1, R_2, .., R_N, R_{N+1})$ is the multi-linear rank of $\mathcal{A}$, and $\|\cdot\|_F$ is the Frobenius norm, i.e., the square root of the sum of squares of the elements. We also introduce the parameter $\lambda > 0$ that acts as a balancing weight between the terms of our optimization problem.

To address the given problem, we employ an alternating optimization approach by iteratively minimizing the objective function in (5) with respect to each variable, while keeping the others constant. As a result, we create the following sequence of update rules at each outer iteration, until a maximum number of iterations is reached or the decrease in the objective function between consecutive iterations is smaller than a predefined threshold.

- For the score tensor $\mathcal{A}$, we apply gradient descent following the update rule

$$\mathcal{A}^l \leftarrow \mathcal{A}^{l-1} - \gamma\nabla_{\mathcal{A}}F(\mathcal{A}^{l-1}, \mathcal{G}, \mathbf{D}_n) \tag{6}$$

  at each inner iteration l of the gradient step, until a maximum number of iterations is reached, where γ is the step size parameter and the partial derivative $\nabla_{\mathcal{A}}F$ of the objective function is given by

$$\nabla_{\mathcal{A}}F = \mathcal{Z} + \lambda(\mathcal{A} - \mathcal{G}\times_1\mathbf{D}_1\times_2\cdots\times_{N+1}\mathbf{D}_{N+1}), \tag{7}$$

  where

$$\mathbf{Z}_{(N+1)_{ji}} = \begin{cases} \mathbb{1}(\mathbf{A}_{(N+1)_{ji}} - \mathbf{A}_{(N+1)_{y_i i}} + \Delta > 0) & , j\neq y_i, i\in\Omega \\ -\sum_{c=1, c\neq y_i}^{C} \mathbb{1}(\mathbf{A}_{(N+1)_{ci}} - \mathbf{A}_{(N+1)_{y_i i}} + \Delta > 0), & j = y_i, i\in\Omega \\ 0 & , i\in\bar{\Omega} \end{cases}$$

  by taking into consideration the true class y_i of each sample i, for $i = 1, .., (I_1\cdots I_N)$, $j = 1, .., C$, and $\mathbb{1}(x)$ equals to 1 if x is true and 0 otherwise.
- For the factor matrices $\mathbf{D}_n, n = 1, .., N{+}1$, we first update each $\mathbf{D}_n$ without regard for its orthogonal constraint through a least square estimation by

$$\hat{\mathbf{D}}_n \leftarrow \mathbf{A}_{(n)} \cdot \mathbf{C}_{n_{(n)}}^{-1}, \tag{8}$$

  where $\mathcal{C}_n = \mathcal{G}\times_1\mathbf{D}_1\times_2\cdots\times_{n-1}\mathbf{D}_{n-1}\times_{n+1}\mathbf{D}_{n+1}\times_{n+2} \cdots \times_{N+1}\mathbf{D}_{N+1}$, for $n = 1, .., N{+}1$, and $\mathbf{B}^{-1}$ denotes the pseudoinverse of a matrix $\mathbf{B}$. Subsequently, we apply a *QR* factorization on each $\hat{\mathbf{D}}_n$ in order to impose the orthogonality constraint on them. Then, the factor matrix $\mathbf{D}_n$ is constructed as the R_n columns of the unitary matrix $\mathbf{Q}$ of the corresponding *QR* factorization, i.e.,

$$\begin{aligned} (\mathbf{Q}, \mathbf{R}) &\leftarrow QR(\hat{\mathbf{D}}_n) \\ \mathbf{D}_n &\leftarrow \mathbf{Q}(:, 1:R_n) \end{aligned}$$

  for $n = 1, .., N{+}1$.
- For the coefficients of the updated factor matrices, we update

$$\mathcal{G} \leftarrow \mathcal{A}\times_1\mathbf{D}_1^T\times_2\cdots\times_{N+1}\mathbf{D}_{N+1}^T, \tag{9}$$

  using the least squares method.

IV. Experimental Results

We evaluate the proposed classification method on several real-world classification tasks using datasets obtained from the UCI machine learning repository [22] with a limited number of samples. To assess the performance of our algorithm, we use the *classification accuracy* that indicates the percentage of

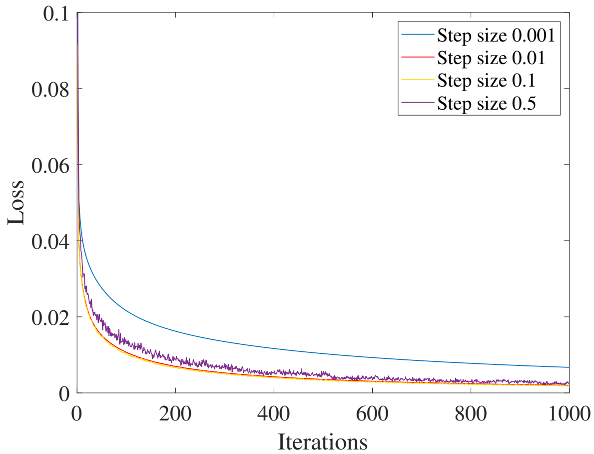

Fig. 2. Convergence behavior of the proposed method for varying gradient step sizes.

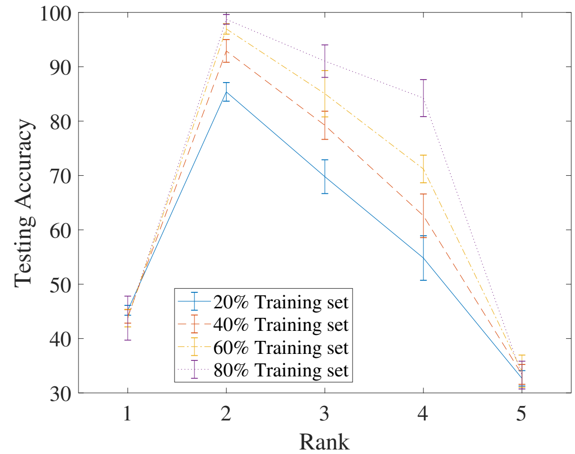

Fig. 3. Evaluation of testing accuracy across various multi-linear ranks of the score tensor and different training set sizes.

the correct predictions over the total number of predictions. Specifically, for each experiment, we split the dataset into two sets, 80% used for training and 20% for testing. We also use a small part of the training set to perform cross-validation for parameter tuning. Finally, we run 10 Monte-Carlo simulations for each experiment and we report the average accuracy and the corresponding standard deviation (SD).

Firstly, we investigate the empirical convergence of the proposed classification tensor completion method on the dataset Balance Scale, which consists of 4 attributes, each with 5 possible values, and encompasses 3 distinct classes. Therefore, in this case, we learn a score tensor of size $5 \times 5 \times 5 \times 5 \times 3$. As we can see in Fig. 2, the convergence behavior is depends heavily on the step size parameter of the gradient step of the score tensor update. Specifically, the convergence is monotone when the step size is small, while there are small spikes in the loss using a larger step size. As a result, we opted for a step size of 0.01 in our experiments to attain rapid and monotonous convergence.

Next, we explore how the selection of the multi-linear rank of the score tensor, which aligns with the dimensions of the core tensor in Tucker decomposition, influences the outcome. This examination is carried out across a range of training set sizes. Given that each dimension of the score tensor is 5 in this dataset, the rank of each mode matricization can be at most 5, as we can see in Fig. 3. It is important to emphasize that our primary concern is the rank of the score tensor for all dimensions, except for the final dimension that corresponds to the classes. We always take the full rank for this dimension in our model. As anticipated, the accuracy of our proposed method rises as the size of the training set expands, as more data becomes available and fewer entries are missing in the score tensor. Moreover, our model consistently performs better when employing a multi-linear rank of 2 in each instance, which aligns with the actual rank of the score tensor.

Furthermore, we assess the performance of our classification method across various datasets with distinct characteristics, all of which are detailed in Table I. It is worth noting that the performance of tensor completion for the score tensor is contingent on the percentage of available samples within the tensor. As the number of available measurements decreases, the performance of our algorithm deteriorates, as indicated by the results, which is an expected outcome. However, as demonstrated in Table II, our algorithm surpasses competing alternatives in scenarios with a scarcity of samples.

Another factor that can impact the effectiveness of the proposed approach is the potential presence of samples that share identical attribute values but have different labels. To address these conflicting samples in our approach, we resolve them by considering the majority vote as the correct label. However, the outcomes reported in Table I for the Breast Cancer and Hayes-Roth datasets are less favorable compared to the results obtained with datasets like Balloons and Balance Scale, which have an equal number of classes. Notably, the loss becomes more visible when dealing with a larger percentage of non-unique-labeled samples.

Finally, we conducted a comparative analysis of our algorithm with other classification methods. Specifically, we utilized Random Forests, Support Vector Machines (SVM), and the Multilayer Perceptron (MLP) with 3 hidden layers, as baseline models. We performed cross-validation for parameter tuning across 10 Monte-Carlo simulations for each method. The mean testing accuracies, along with the corresponding standard deviation values, for all methods on various datasets are presented in Table II, highlighting the effectiveness of our approach. The small standard deviation values associated with our method also indicate the reliability and consistency of the results across different simulations. Additionally, our findings verify that it is not possible to predict in advance which of the other classification methods will perform the best for a given problem, whereas our proposed method outperforms the baseline methods across all the datasets.

TABLE I
THE DATA CHARACTERISTICS OF VARIOUS DATASETS ALONGSIDE THE CORRESPONDING PERFORMANCE OF OUR PROPOSED METHOD.

Dataset	**Number of Attributes**	**Number of Classes**	**Precentage of Training Samples in the Label Tensor**	**Percentage of Not-unique-labeled Samples**	**Number of Samples**	**Testing Accuracy**
Balance Scale	4	3	80%	0%	625	98.7%
Balloons	4	2	81.25%	0%	16	70%
Breast Cancer	9	2	0.0453%	4.55%	286	68.9%
Car Evaluation	6	4	79.98%	0%	1728	98.9%
Hayes Roth	4	3	66.67%	20.63%	160	84%

TABLE II
ASSESSMENT OF THE PROPOSED APPROACH IN RELATION TO OTHER CLASSIFICATION METHODS ACROSS MULTIPLE DATASETS.

Dataset	**Classification Tensor Completion**	**Random Forest**	**SVM**	**MLP**
Balance Scale	**98.7 (0.89)**	83.6 (2.18)	96 (2.99)	98.4 (1.34)
Balloons	**70 (27.69)**	66.7 (33.3)	66.7 (29.8)	66.7 (25.8)
Breast Cancer	**68.9 (5.77)**	**68.9 (4.78)**	68.4 (5.66)	67.4 (4.45)
Car Evaluation	**98.9 (0.78)**	98.1 (0.79)	91.6 (2.38)	98.6 (0.4)
Hayes-Roth	**84 (4.06)**	82.8 (6.59)	55 (5.8)	79.1 (6.71)

V. CONCLUSION

In this study, we introduced a general machine learning alternative for the classification problem, leveraging tensor decomposition techniques. More precisely, we reformulated the problem as a tensor completion task by learning a tensor of scores for all potential samples and classes. To predict scores for unlabeled samples, we employed a combination of the hinge loss function and Tucker decomposition to complete the score tensor. Additionally, we conducted an evaluation of our method across various real-world classification tasks, demonstrating its effectiveness in comparison to other competing methods.

ACKNOWLEDGMENT

This work was supported by the TITAN ERA Chair project (contract no. 101086741) within the Horizon Europe Framework Program of the European Commission.

REFERENCES

[1] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Emerging technology in modelling and graphics*. Springer, 2020, pp. 99–111.

[2] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[3] M. Awad and R. Khanna, "Support vector machines for classification," in *Efficient learning machines*. Springer, 2015, pp. 39–66.

[4] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[5] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.

[6] Y. Ji, Q. Wang, X. Li, and J. Liu, "A survey on tensor techniques and applications in machine learning," *IEEE Access*, vol. 7, pp. 162 950–162 990, 2019.

[7] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[8] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear theory and its applications, IEICE*, vol. 1, no. 1, pp. 37–68, 2010.

[9] A. Fargeas, L. Albera, A. Kachenoura, G. Dréan, J.-D. Ospina, J. Coloigner, C. Lafond, J.-B. Delobel, R. De Crevoisier, and O. Acosta, "On feature extraction and classification in prostate cancer radiotherapy using tensor decompositions," *Medical engineering & physics*, vol. 37, no. 1, pp. 126–131, 2015.

[10] G. Zhao, B. Tu, H. Fei, N. Li, and X. Yang, "Spatial-spectral classification of hyperspectral image via group tensor decomposition," *Neurocomputing*, vol. 316, pp. 68–77, 2018.

[11] M. Jouni, M. Dalla Mura, and P. Comon, "Hyperspectral image classification based on mathematical morphology and tensor decomposition," *Mathematical Morphology-Theory and Applications*, vol. 4, no. 1, pp. 1–30, 2020.

[12] K. Maruhashi, M. Todoriki, T. Ohwa, K. Goto, Y. Hasegawa, H. Inakoshi, and H. Anai, "Learning multi-way relations via tensor decomposition with neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[13] R. K. Kaliyar, A. Goswami, and P. Narang, "Deepfake: improving fake news detection using tensor decomposition-based deep neural network," *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1015–1037, 2021.

[14] A. Tjandra, S. Sakti, and S. Nakamura, "Tensor decomposition for compressing recurrent neural network," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[15] A.-H. Phan, K. Sobolev, K. Sozykin, D. Ermilov, J. Gusak, P. Tichavskỳ, V. Glukhov, I. Oseledets, and A. Cichocki, "Stable low-rank tensor decomposition for compression of convolutional neural network," in *European Conference on Computer Vision*. Springer, 2020, pp. 522–539.

[16] N. Kargas and N. D. Sidiropoulos, "Supervised learning and canonical decomposition of multivariate functions," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1097–1107, 2021.

[17] ——, "Nonlinear system identification via tensor completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4420–4427.

[18] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional systems and signal processing*, vol. 26, no. 3, pp. 677–692, 2015.

[19] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2016.

[20] L. Karlsson, D. Kressner, and A. Uschmajew, "Parallel algorithms for tensor completion in the cp format," *Parallel Computing*, vol. 57, pp. 222–234, 2016.

[21] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[22] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml