

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Optimal Feature Extraction and Classification of Tensors via Matrix Product State Decomposition

Johann A. Bengua, Ho N. Phien, Hoang D. Tuan

Centre for Health Technologies,

University of Technology, Sydney, Ultimo 2007, Australia

Email: Johann.A.Bengua@student.uts.edu.au, NgocPhien.Ho@uts.edu.au, Tuan.Hoang@uts.edu.au

Abstract—Big data consists of large multidimensional datasets that would often be difficult to analyze if working with the original tensor. There is a rising interest in the use of tensor decompositions for feature extraction due to the ability to extract necessary features from a large dimensional feature space. In this paper the matrix product state (MPS) decomposition is used for feature extraction of large tensors. The novelty of the paper is the introduction of a single *core tensor* obtained from the MPS that not only contains a significantly reduced feature space, but can perform classification with high accuracy without the need of feature selection methods.

Keywords—Matrix product state, tensor, feature extraction, classification, pattern classification

I. INTRODUCTION

There is an increasing need to handle large multidimensional datasets that cannot easily be analyzed or processed using modern day computers. Due to the curse of dimensionality researchers need to investigate mathematical tools which can evaluate information beyond the properties of large matrices. The essential goal is to reduce the dimensionality of big data with minimal information loss. One such method is to approximate multidimensional datasets in terms of tensor decompositions [1]. This method has been of great interest within the last decade and successfully applied in a diverse range of research areas such as data classification [2], computer vision [3], quantum many-body physics [4], [5] and signal processing [6].

In this paper we focus on the dimensionality reduction of higher-order tensors using the MPS decomposition [7] (also known as the tensor-train (TT) decomposition [8]). Specifically, we introduce the concept of a *core tensor*, similarly used in the Tucker decomposition (TD) [1], in the context of MPS. We demonstrate that this core tensor contains a greatly reduced feature space from the original tensor, and can be used sufficiently in supervised learning problems. To our knowledge, the concept of a core tensor in MPS has not been introduced in previous research.

II. MATRIX PRODUCT STATE DECOMPOSITION AND MULTILINEAR CLASSIFICATION

A. Mathematical notation

We first review some mathematical notation frequently used in multilinear algebra [1]. A *tensor* is a multidimensional array and its *order* is the number of dimensions it contains. Zero-order tensors are scalars and denoted by lowercase letters, e.g., x . A first-order tensor is a vector, which we denote by boldface lowercase letters, e.g., \mathbf{x} . A matrix is a tensor of order two and is defined by boldface capital letters, e.g., \mathbf{X} . A higher-order tensor (tensors of order three and above) are denoted by boldface calligraphic letters, e.g., \mathcal{X} . Therefore a general N th-order tensor can be defined as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, where each I_i is the dimension of the local subspace i . We denote a_i as the i th entry of a vector \mathbf{a} and a_{ij} as an element of a matrix \mathbf{A} . The element of a third-order tensor \mathcal{X} is denoted as x_{ijk} and thus defined similarly for a general N th-order tensor. Indices will range from 1 to their capital version, e.g., $i = 1, \dots, I$ or $\delta = 1, \dots, \Delta$ for Greek letters. The n th element in a sequence of tensors is denoted with a superscript in parentheses, e.g., $\mathbf{A}^{(n)}$ is the n th matrix.

B. Matrix product state decomposition for multilinear classification

The general problem for multilinear classification is as follows [9]:

Problem 1: Given a set of K training samples represented by N th-order tensors $\mathcal{X}^{(k)} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ($k = 1, 2, \dots, K$) corresponding to Q categories, and a set of T test data $\mathcal{X}^{(t)} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ($t = 1, 2, \dots, T$), classify the test data into the categories Q with high accuracy. The classification problem is a supervised problem where the categories Q are defined according to the problem and the data provided.

This problem has been solved using the TD [9], where a single core tensor and common factors to training and test data were obtained. The main issue for the TD is that the feature space of the core tensor for classification can be quite large, thus further feature selection is always needed.

Alternatively we propose the MPS solution to *Problem 1* similar to the TD, in which we obtain a main core tensor¹ and common factors. Firstly concatenate the set of training data $\{\mathcal{X}^{(k)}\}$ such that

$$\mathcal{Y} = [\mathcal{X}^{(1)} \mathcal{X}^{(2)} \dots \mathcal{X}^{(K)}] \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N \times K_{N+1}}, \quad (1)$$

with elements $y_{i_1 \dots i_N k_{N+1}}$. In the tensor representation in Eq. (1), a new index k_{N+1} is added to represent the number of samples in the training data. We can now introduce the MPS decomposition of \mathcal{Y} , where each element is given by

$$y_{i_1 \dots k_{N+1}} = \mathbf{a}_{i_1}^{(1)} \mathbf{A}_{i_2}^{(2)} \dots \mathbf{A}_{i_N}^{(N)} \mathbf{a}_{k_{N+1}}^{(N+1)}, \quad (2)$$

For each i_j ($j \neq 1$), the corresponding matrix $\mathbf{A}_{i_j}^{(j)}$ has the size $\Delta_{(j-1)} \times \Delta_j$, and for each i_1, k_{N+1} , $\mathbf{a}_{i_1}^{(1)}$ and $\mathbf{a}_{k_{N+1}}^{(N+1)}$ correspond to row and column vectors of size Δ_1 and Δ_N , respectively. This MPS decomposition can be obtained by applying a sequence of singular value decompositions (SVD) to \mathcal{Y} [10].

The sample index k_p can be at an arbitrary position $p = \{1, \dots, N+1\}$, e.g. in Eq. (2) $p = N+1$. To obtain a unique core tensor \mathcal{C}^{tr} from the MPS decomposition we need to ensure the decomposition is in a *mixed-canonical* form [10]. Rearranging Eq. (2)

$$y_{i_1 \dots k_p \dots i_{N+1}} = \mathbf{a}_{i_1}^{(1)} \dots \mathbf{A}_{i_{p-1}}^{(p-1)} \mathbf{A}_{k_p}^{(p)} \mathbf{A}_{i_{p+1}}^{(p+1)} \dots \mathbf{a}_{i_{N+1}}^{(N+1)}, \quad (3)$$

then the tensors to the left and right of $\mathcal{A}^{(p)}$ in Eq. (3) must satisfy *left- and right-canonical* constraints

$$\sum_{i_1=1}^{I_1} \mathbf{a}_{i_1}^{\dagger(1)} \mathbf{a}_{i_1}^{(1)} = I \quad (4)$$

$$\sum_{i_n=1}^{I_n} \mathbf{A}_{i_n}^{\dagger(n)} \mathbf{A}_{i_n}^{(n)} = I, \quad (5)$$

and

$$\sum_{i_{N+1}=1}^{I_{N+1}} \mathbf{a}_{i_{N+1}}^{(N+1)} \mathbf{a}_{i_{N+1}}^{\dagger(N+1)} = I \quad (6)$$

$$\sum_{i_{n'}=1}^{I_{n'}} \mathbf{A}_{i_{n'}}^{\dagger(n')} \mathbf{A}_{i_{n'}}^{(n')} = I, \quad (7)$$

to be considered mixed-canonical. With I as the identity matrix, $n = \{1, \dots, p-1\}$ and $n' = \{p+1, \dots, N+1\}$. More specifically, each tensor from positions $(1) \rightarrow (p-1)$ and $(p+1) \rightarrow (N+1)$ are left- and right-orthogonal, respectively. The core tensor can now be extracted from the MPS in the following way,

$$\begin{aligned} \mathcal{C}^{tr} &= (\dots \mathbf{A}_{i_{p-1}}^{\dagger(p-1)} \mathbf{A}_{i_{p-1}}^{(p-1)}) \mathbf{A}_{k_p}^{(p)} (\mathbf{A}_{i_{p+1}}^{(p+1)} \mathbf{A}_{i_{p+1}}^{\dagger(p+1)} \dots) \\ &= \mathbf{A}_{k_p}^{(p)}, \end{aligned} \quad (8)$$

¹In [8] each tensor in the MPS is defined as a core tensor, however the core tensor defined in this paper follows from the TD, and is a unique tensor from the MPS that is used for classification.

where for each training sample k_p , there are $\Delta_{p-1} \times \Delta_p$ features. Subsequently, for each k_p of the core tensor, the remaining tensors are unchanged. Therefore all tensors but the core tensor with index k_p can be regarded as common factors to all training samples. In the context of supervised learning we only need to use the tensor with the index k_p for classification because it contains the necessary features for each training sample k_p .

Furthermore the position $p = \frac{(N+1)}{2}$ or $p = \frac{(N+1)+1}{2}$ for even or odd order \mathcal{Y} is the optimal position of the core tensor because it can represent the entire feature space of the original tensor! [7]. This may sound like a huge setback, but what can be exploited here is that the size of the core tensor can be significantly reduced to a fixed small size Δ , which correlates to decreasing the number of features for classification. Then for fixed $\Delta \ll \min(\Delta_{p-1}, \Delta_p)$ the size of \mathcal{C}^{tr} is $\Delta \times K_p \times \Delta$. We define Δ as a *bond dimension* of the training core tensor, then for each training sample k_p there are Δ^2 features. We have applied this smaller core tensor directly to pattern recognition problems and have obtained high classification rates with only a small number of features.

The next step is to obtain the test data core tensor that can be used for classification with the training data core tensor. Let the concatenation of test data $\{\mathcal{X}^{(t)}\}$ result in a tensor

$$\mathcal{W} = [\mathcal{X}^{(1)} \mathcal{X}^{(2)} \dots \mathcal{X}^{(T)}] \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N \times T}. \quad (9)$$

This is necessary to be compatible with the concatenation of the training tensor. As we moved the index k_p describing the number of samples into position p . The test tensor should have index t_p in the position p as well. The common factors obtained from the mixed-canonical MPS decomposition of the training tensor \mathcal{Y} should be common to all test data, then the test core tensor can be extracted using these common factors. Assuming the position of the training and test indices k_p and t_p are at an optimal p , the resultant test core tensor \mathcal{C}^{ts} in element form is

$$\begin{aligned} c_{\delta_{p-1} t_p \delta_{p+1}}^{ts} &= \sum_{\{I_l\}, \{\Delta_m\}} w_{i_1 \dots t_p \dots i_{N+1}} \\ &= \sum_{\{i_l\}, \{\delta_m\}=1} a_{i_1 \delta_1}^* \dots a_{\delta_{p-2} i_{p-1} \delta_{p-1}}^* \\ &\quad a_{\delta_p i_{p+1} \delta_{p+1}}^* \dots a_{\delta_N i_{N+1}}^* \end{aligned} \quad (10)$$

where $l = \{1, \dots, N+1\} \setminus p$ and $m = \{1, \dots, N\} \setminus (\delta_{p-1}, \delta_p)$. Note that if \mathcal{C}^{tr} has fixed Δ , then \mathcal{C}^{ts} will have equivalent bond dimensions. We can now apply classifiers such as support vector machine (SVM) or k -nearest neighbors (k -NN) for classification using \mathcal{C}^{tr} and \mathcal{C}^{ts} .

III. EXPERIMENTS AND RESULTS

The MPS training and test core tensor have been tested with three datasets: The Columbia University Image Libraries COIL-20 [11] and COIL-100 [12], [13], and the Extended Yale B dataset [14].

A. COIL-20

The database consists of 1440 grayscale images of 20 objects (72 images per object). All images used were initially 128×128 pixels and then downsampled to 32×32 grayscale (0-255). Images were randomly divided into two partitions according to a hold/out ratio. For our experiment we tested with hold/out ratios of 10%, 30%, 50%, 70%, 90% and 95%, e.g. a 10% hold/out ratio accounts to 10% test data and 90% training data. The training and test data were structured as third-order tensors of dimensions $K \times 32 \times 32$ and $T \times 32 \times 32$, respectively. The training and core tensors were at $p = 2$ and for each k_2, t_2 , the bond dimension $\Delta_1 = \Delta_2 = \Delta$, with $\Delta = \{2, \dots, 32\}$. The results were averaged over 10 trials.

The classification algorithm used was k -NN with correlational distance. The classification accuracy was plotted versus the bond dimension (Δ^2 features) in Fig. 1. The highest classification accuracy was from a 10% hold/out ratio as expected, with a 100% classification accuracy for a several number of features. With 30% hold/out ratio the highest accuracy was 99.79%. At 50% hold/out ratio the highest accuracy is 99.35%. The remaining hold/out ratios 70%, 90% and 95% had maximum classification accuracies of 97.64%, 89.95% and 81.18%, respectively.

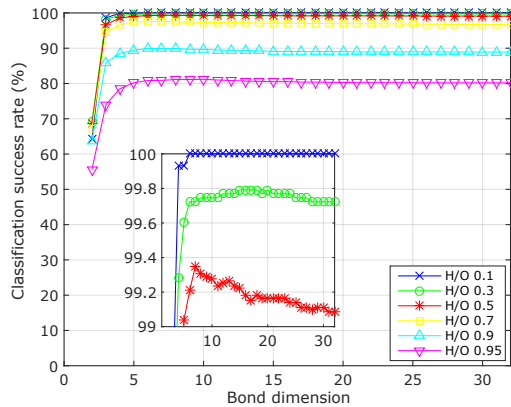


Figure 1. Classification accuracy on COIL-20 database.

B. COIL-100

This database has 7200 *color* images of 100 objects (72 images per object). The hold/out ratios for training and test data were also tested with 10%, 30%, 50%, 75%, 90% and 95%. Images were originally $128 \times 128 \times 3$ pixels and downsampled to $32 \times 32 \times 3$ pixels. The training and test data were constructed as fourth-order tensors of dimensions $K \times 32 \times 32 \times 3$ and $T \times 32 \times 32 \times 3$, respectively. The training and core tensors had the same conditions from COIL-20. The results were averaged over 10 trials.

The classification algorithm used was k -NN with correlational distance. Classification rate is plotted in Fig. 2. The highest classification accuracy was from a 10% hold/out

ratio with 99.85%. A 30% hold/out ratio had 99.74%. At 50% hold/out ratio the highest accuracy is 99.22%. The remaining hold/out ratios 75%, 90% and 95% had accuracies of 96.65%, 89.14% and 80.59%, respectively.

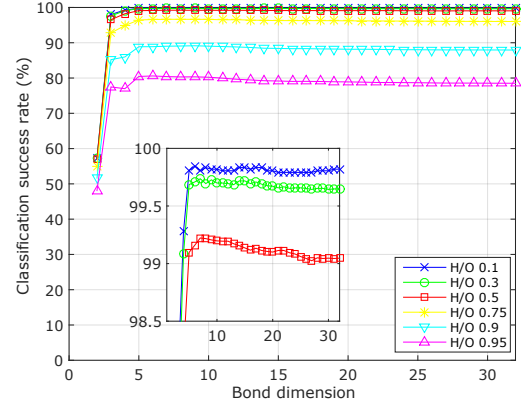


Figure 2. Classification accuracy on COIL-100 database.

C. Extended Yale Face database B

The database contains 16128 grayscale images with 28 human subjects under 9 poses, where for each pose there is 64 illumination conditions. To improve computational time each image was cropped to keep only the center area containing the face, then resized to 73×55 . Training data and test data was but partitioned according to poses. For training and test data we selected poses 0, 2, 4, 6 and 8 and 1, 3, 5, and 7, respectively. For a single subject the training tensor was size $5 \times 73 \times 55 \times 64$ and the test tensor was size $4 \times 73 \times 55 \times 64$. Hence for all 28 subjects we had a training fourth-order tensors $140 \times 73 \times 55 \times 64$ and $112 \times 73 \times 55 \times 64$ for training and test data, respectively. The training and core tensors had the same conditions from COIL-20 and COIL-100 except $\Delta = \{2 \dots, 55\}$.

Four classification methods were used to compare performance: Support vector machine (SVM) one-against-one (1v1), SVM one-against-all (1vall), 1-NN and 2-NN algorithms. Classification accuracy was plotted versus bond dimension (Δ^2 features) in Fig. 3. The 1v1 SVM algorithm obtained the highest accuracy with 93.75% compared to the 1vall SVM with 88.39%. Using k -NN, we see that 1-NN and 2-NN have classification accuracies of 91.07% and 89.29%, respectively.

D. Analysis

The fixed bond dimension Δ for the training and test core tensors had a direct affect on the classification of the test data. In all experiments a small number of features was only needed to achieve high classification accuracies, however the accuracy does not necessarily increase with more features. This is because a larger number of features

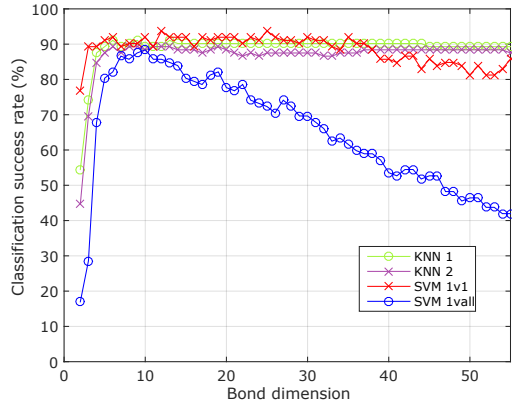


Figure 3. Classification accuracy on extended Yale face database B.

does not necessarily correlate to increased classification performance.

In the COIL-20 and COIL-100 results it was expected that larger hold/out ratios decreases the classification performance and this can be seen by the decreased accuracies at a hold/out ratio of 95%. For COIL-100 our algorithm outperformed several methods [15]–[17] at the 75% hold/out ratio. This is also using the full color image as opposed to the converted grayscale image for these experiments.

The classification performance for the Extended Yale Face database B (EYFB) is much more stable, has lower computational complexity, and has higher classification accuracies at small δ compared to multilinear discriminant analysis (MDA) algorithms known as Direct General Tensor Discriminant Analysis (DGTDA) and Constrained Multilinear Discriminant Analysis (CMDA) proposed recently by Li & Schonfeld [2] as well as other methods mentioned in their paper.

IV. CONCLUSION

The classification performance has shown that the MPS decomposition can be used as an efficient and simple mechanism in representing multidimensional datasets for feature extraction and classification with supervised learning. Furthermore the core tensor required to sufficiently classify data was of a reduced order and size, showing that only a small number of features was needed to classify higher-order tensors in pattern recognition problems. In future work we need to compare the efficiency in detail of our method against other higher-order classification algorithms such as multilinear principal component analysis (MPCA) and the TD as well as test our method with higher-order data.

REFERENCES

- [1] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

- [2] Q. Li and D. Schonfeld, “Multilinear discriminant analysis for higher-order tensor data classification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 12, pp. 2524–2537, Dec 2014.
- [3] M. A. O. Vasilescu and D. Terzopoulos, “Multilinear analysis of image ensembles: Tensorfaces,” *Proceedings of the European Conference on Computer Vision (ECCV ’02)*, vol. 2350, pp. 447–460, 2002.
- [4] G. Vidal, “Efficient classical simulation of slightly entangled quantum computation,” *Phys. Rev. Lett.*, vol. 91, no. 14, p. 147902, Oct 2003.
- [5] F. Verstraete, D. Porras, and J. I. Cirac, “Density matrix renormalization group and periodic boundary conditions: A quantum information perspective,” *Phys. Rev. Lett.*, vol. 93, no. 22, p. 227205, Nov 2004.
- [6] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *Signal Processing Magazine, IEEE*, vol. 32, no. 2, pp. 145–163, March 2015.
- [7] D. Pérez-García, F. Verstraete, M. Wolf, and J. Cirac, “Matrix product state representations,” *Quantum Information and Computation*, vol. 7, no. 5, pp. 401–430, 2007.
- [8] I. V. Oseledets, “Tensor-train decomposition,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [9] 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.
- [10] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Annals of Physics*, vol. 326, no. 1, pp. 96 – 192, 2011, january 2011 Special Issue.
- [11] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (coil-20),” *Technical Report CUCS-005-96*, Feb 1996.
- [12] —, “Columbia object image library (coil-100),” *Technical Report CUCS-005-96*, Feb 1996.
- [13] M. Pontil and A. Verri, “Support vector machines for 3d object recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 6, pp. 637–646, Jun 1998.
- [14] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: illumination cone models for face recognition under variable lighting and pose,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 643–660, Jun 2001.
- [15] M.-H. Yang, D. Roth, and N. Ahuja, “Learning to recognize 3d objects with snow,” in *Computer Vision - ECCV 2000*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, vol. 1842, pp. 439–454.
- [16] R. Maree, P. Geurts, J. Piater, and L. Wehenkel, “Random subwindows for robust image classification,” *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 34–40 vol. 1, June 2005.

- [17] J. Matas and S. Obdržálek, "Object recognition methods based on transformation covariant features," *Proc. 12th European Signal Processing Conference (EUSIPCO 2004)*, 2004.