Fast High-order Tensor Learning Based on Grassmann Manifold.

O.KARMOUDA, R.BOYER and J.BOULANGER

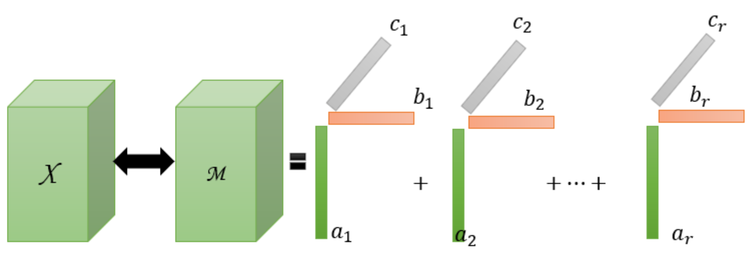
Laboratoire CristAL, CNRS, Université de Lille, France.

1. Introduction

Support Vector Machines (SVMs) are a powerful tool for different classification tasks. In order to use SVMs when input data are multidimensional, recently, several works aim to represent data with Canonical Polyadic Decomposition (CPD) and express the kernel between tensors in terms of kernels between their CP factors. However, the CPD ambiguities affects severely the performance of classification. Moreover, the factor estimation algorithm associated to the CPD suffers from the curse of dimensionality. In fact, the part of time spent to compute the CPD with the well known algorithm Alternated Least Squares (ALS) is approximatively 80% of the of the total running time of the method in the state of the art. In this work, we show that the use of kernel between tensors on Grassmann manifold is able to cirumvent the scaling ambiguity of the CPD decomposition. Next, the complexity problem is addressed by using the equivalent algebraic representation of the CP model into a Tensor Train (TT) model.

2. Tensor Factorizations

CP Decomposition (3rd tensor)



$$x(i,j,k) \approx m(i,j,k) = \sum_{l=1}^{r} a(i,l)b(j,l)c(k,l).$$

Tensor Notation : [Kolda et Bader, SIAM 2009]

$$\mathcal{X} \approx \mathcal{M} = \sum_{l=1}^{r} a_l \circ b_l \circ c_l = [\![A, B, C]\!] \ .$$

- Fitting CP: Finding $\mathcal{M}$ that best approximates $\mathcal{X}$ by solving the optimization problem:

$$\min_{\mathcal{M}} ||\mathcal{X} - \mathcal{M}|| \ \ with \ \ \mathcal{M} = \sum_{r=1}^{R} a_r \circ b_r \circ c_r.$$

  The ALS algorithm is the well known algorithm for fitting CP.

- CP ambiguities:
  - Permutation ambiguity :

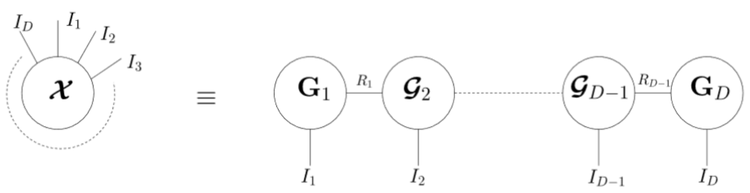
$$\mathcal{X} = [\![A, B, C]\!] = [\![A\Pi, B\Pi, C\Pi]\!]$$

    for any $R \times R$ permutation matrix Π.
  - Scaling ambiguity :

$$\mathcal{X} = \sum_{r=1}^{R} (\alpha_r a_r) \circ (\beta_r b_r) \circ (\gamma_r c_r)$$

    as long as $\alpha_r \beta_r \gamma_r = 1$ for $r = 1, ..., R$

Tensor Train Model [Oseledets, SIAM 2011]



$$\mathcal{X}(i_1, \ldots, i_Q) = \sum_{r_1, \ldots, r_{Q-1}} G_1(i_1, r_1)\mathcal{G}_2(r_1, i_2, r_2)$$

$$\ldots \mathcal{G}_{Q-1}(r_{Q-2}, i_{Q-1}, r_{Q-1})G_Q(r_{Q-1}, i_Q),$$

JIRAFE method is an alternative method of the ALS algorithm, faster and don't suffer from the curse of dimensionality. If $[\![P_1, P_2..., P_{Q-1}, P_Q]\!]$ is the CPD of $\mathcal{X}$, JIRAFE method consists on optimizing the following criterion :[Zniyed, Boyer et al. LAA 2019]

$$\min_{M,P}\{||G_1 - P_1 M_1^{-1}||_F + ||G_Q - M_{Q-1}P_Q^T||_F + \sum_{q=2}^{Q-2} ||\mathcal{G}_q - [\![M_{q-1}, P_q, M_q^{-T}||_F]\!]\}.$$

3. Kernel-based classification of high-order tensors

- Given a binary classification problem where data is composed of M tensors $\mathcal{X}_i \in \mathbb{R}^{N_1 \times \ldots \times N_Q}$ of rank R labeled with $y_i \in \{-1, 1\}$, we look for a hyperplane to discrimate the classes. Applying SVM for vectorized tensors leads to a prohibtive computational cost and destroys the multidimensional structure of data.

- In order to compte the kernel between a couple of data $(\mathcal{X}_i, \mathcal{X}_j)$, the idea of [Dusk, SIAM 2014] is to compute their CPD and then compute kernels between the CP factors using :

$$k_{\text{dusk}}(\mathcal{X}_i, \mathcal{X}_j) := \sum_{r=1}^{R}\sum_{r'=1}^{R}\prod_{q=1}^{Q} k_{gauss}(\mathbf{x}_r^{(q)}, \mathbf{x'}_{r'}^{(q)}),$$

  where $\mathbf{x}_r^{(q)}, \mathbf{x'}_{r'}^{(q)}$ are respectively the CP factors of $\mathcal{X}_i$ and $\mathcal{X}_j$.

- The decision function for a new point $\mathcal{X}$ to classify is given by:

$$f(\mathcal{X}) = sgn(\sum_{i=1}^{M} \alpha_i y_i k(\mathcal{X}_i, \mathcal{X}) + b),$$

  where b and $(\alpha_i)_i$ are parameters of SVM.

Due to the scaling ambiguity, we can prove the two following results:

- Two identical tensors are viewed as two distinct objects for the classification
- The decision function may become data-invariant.

4.Tensor Learning on a Grassmann manifold

For integers $n \geq k > 0$, a Grassmann Manifold is defined as the set of subspaces of dimension k embeded on a space of dimension n. Mathematically, $\mathbb{G}(n, k)$ is given by:

$$\mathbb{G}(n, k) = \{span(N) : N \in \mathbb{R}^{n \times k} N^T N = I_k\}.$$

Figure 1: $\mathbb{G}(3, 1)$ with two classes.

A suitable distance between $X, Y \in \mathbb{G}(n, k)$ that gives rise to a positive definite gaussian kernel on $\mathbb{G}_{n,k}$ is the projection Frobenius norm:

$$d_c(X, Y) := ||\Pi_X - \Pi_Y||_2 = \sqrt{2}||\sin(\theta)||,$$

where $\theta = \{(\theta_i)\}_{i=1}^{k}$ is a vector of principle angles between X and Y.
The kernel that we propose to use is then:

$$k_{\text{grass}}(\mathcal{X}_i, \mathcal{X}_j) := \sum_{r=1}^{R}\sum_{r'=1}^{R}\prod_{q=1}^{Q} \exp\left(-\frac{d_c(span(\mathbf{x}_r^{(q)}), span(\mathbf{x'}_{r'}^{(q)}))^2}{2\sigma^2}\right),$$

5. Numerical Experiments

In the figure at the top left: Accuracy score for different methods to discriminate 3 classess in the Extended Yale dataset. [A. Georghiades, P. Belhumeur, and D. Kriegman, IEEE 2001]
In the figure at the top right:Accuracy score for different methods to discriminate 3 classess in the UCF11 dataset [J. Liu, Jiebo Luo, and M. Shah, SIAM 2009].
In the figure at the bottom in the middle : Gain in time when computing CPD with ALS vs JIRAFE.

R	3	4
DUSK+ALS	0.833	0.9166
ALS+Grassmann	1	1
JIRAFE+Grassmann	1	1

R	2	3
DUSK+ALS	0.56+/- 0.06	0.63+/- 0.09
ALS+Grassmann	0.83+/-0.06	0.72+/-0.04
JIRAFE+Grassmann	0.86+/-0.03	0.81+/-0.04

R	2	3	4
Extended Yale	87	92	60
UCF11	196	444	-