UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

FACULTAD DE INGENIERÍA
Vicedecanatura Académica
POSGRADOS

PROPOSAL SUBMISSION

DOCTORAL THESIS:	☒	MASTER THESIS:	□
MASTER FINAL WORK:	□	SPECIALIZATION FINAL WORK:	□

1. **BIDDER:** Robinson Andrés Jaque Pirabán **ID:** 80190790
2. **PROGRAM:** Phylosophy Doctoral in Computer Science and Systems Engineering
3. **ADVISOR:** Fabio Augusto González Osorio
   DEPARTMENT: Computer Science and Industrial Engineering
4. **TITLE: Kernel Tensor Factorization**
5. **AREA:** Computer Science
6. **LINE OF RESEARCH:** Machine Learning
7. COMMENTARY WITH ADVISOR APROVAL
8. BIDDER SIGNATURE
9. SIGNATURE OF ADVISOR

Contents

1 Introduction

Tensors are multidimensional arraies, i.e. an N-way or N-order tensor is an element of tensor product of N vector spaces, each of which has its own coordinate system. A first order tensor is a vector, a second order tensor is a matrix, tensors of higher order are called high-order tensors. The order (ways or modes) of a tensor is the number of dimensions. Figure 1 represents a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$.

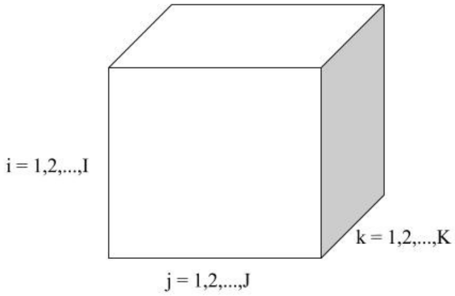
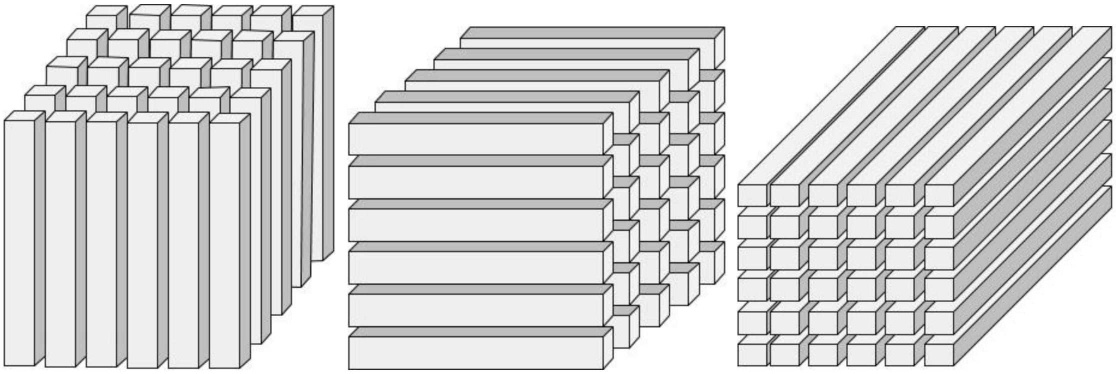


Figure 1: Third-order tensor

Fibers are defined by fixing every index by one. In a third-order tensor a column is a mode-1 fiber, denoted by $x_{:jk}$; a row is a mode-2 fiber, denoted by $x_{i:k}$; while a tube is a mode-3 fiber, denoted by $x_{i:k}$. 2 shows a fibers representation in 3rd-order tensor.



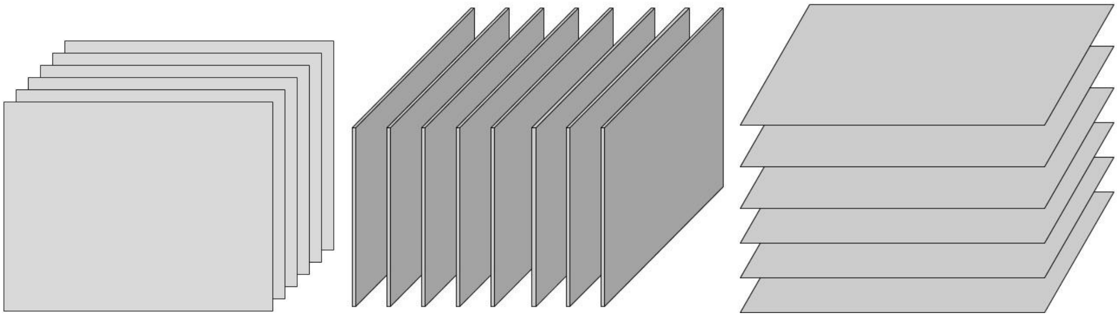
(a) columns (b) rows (c) tubes

Figure 2: 3rd-order tensor fibers

Slices are two-dimensional sections of a tensor defined by fixing two indexes. For instance, slices of 3rd-order tensor $\mathcal{X}$ are denoted by $X_{i::}$ (horizontal), $X_{:j:}$ (lateral) and $X_{::k}$ (frontal).

The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ is analogous to the matrix Frobenius norm, i.e.

$$||\mathcal{X}|| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N}^2} \tag{1}$$



(a) Frontal slices

(b) Horizontal slices

(c) Vertical slices

Figure 3: 3rd-order tensor slices

1.1 Unfolding and Folding Tensors

Unfolding is the process of *matricization* of a tensor. In other words, elements of a tensors are sorted to assemble a matrix. The mode-k unfolding of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ is denoted by $X_{(k)} \in \mathbb{R}^{I_1 \times \prod_{k' \neq k} I_{k'}}$ and arrenges the mode-k tensor fibers as columns of resulting matrix. In addition, Kolda [?] presents a more general procedures of unfolding

Ding and Wei [?] present a fast algorithm for Hankel tensor-vector products.

[?] A method of fast linear transform algorithm synthesis for an arbitrary tensor

2 Tensor Decomposition

Tensor decomposition originated with Hitchcock in 1927 [?], and the the multi-way model is attribuited to Cattell in 1944 [?].

Tensor works had attention in 60s with Tucker ([?], [?], [?]) and Carroll and Chang [?] and Harshman in 1970 [?] with applications in psychometrics. In 1981 Appellof and Davidson [?] used tensor decomposition in chemometrics which have been an popular field of application of tensor decomposition since then.

In last twenty years tensor decomposition applications have expanded to many fields such as signal processing, numberical linear algebra, computer vision, numerical analysis, neuroscience, data mining, graph analysis. Figure 4 attempt to summarize application fields of tensor decomposition.

Fanaee and Gama [?] introduce an interdisciplinary survey about tensor-based anomaly detection.

We suggest to readers to refer to Kolda [?], Acar [?] and [?] for an exhaustive a detailed review of fundamental decomposition methods and applications. Furthermore, [?] presents tensor properties as extension of estructural properties of matrices.

Chemometrics [?] A computationally efficient technique for the solution of multidimensional PBMs of granulation via tensor decomposition

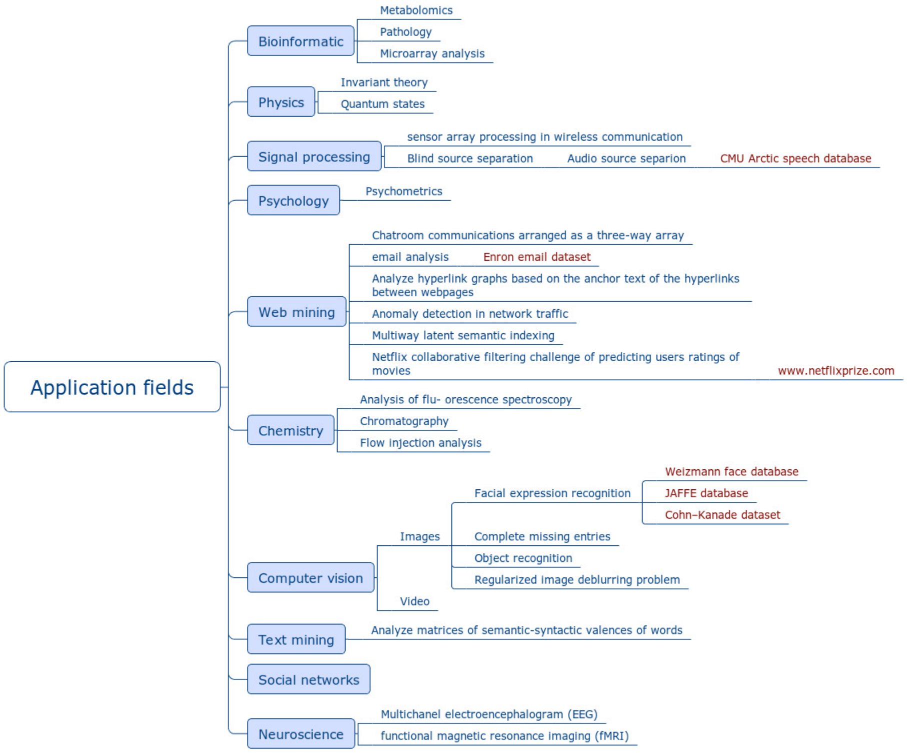

Application fields
Bioinformatic
Metabolomics
Pathology
Microarray analysis
Physics
Invariant theory
Quantum states
Signal processing
sensor array processing in wireless communication
Blind source separation
Audio source separion
CMU Arctic speech database
Psychology
Psychometrics
Web mining
Chatroom communications arranged as a three-way array
email analysis
Enron email dataset
Analyze hyperlink graphs based on the anchor text of the hyperlinks between webpages
Anomaly detection in network traffic
Multiway latent semantic indexing
Netflix collaborative filtering challenge of predicting users ratings of movies
www.netflixprize.com
Chemistry
Analysis of flu- orescence spectroscopy
Chromatography
Flow injection analysis
Computer vision
Images
Facial expression recognition
Weizmann face database
JAFFE database
Cohn–Kanade dataset
Complete missing entries
Object recognition
Regularized image deblurring problem
Video
Text mining
Analyze matrices of semantic-syntactic valences of words
Social networks
Neuroscience
Multichanel electroencephalogram (EEG)
functional magnetic resonance imaging (fMRI)

Figure 4: Application fields of tensor decomposition

Bioinformatics In [?] and [?] a multimodal problem is addressed, the authors formulate data fusion as a coupled matrix and tensor factorization problem and discuss its extension to a structure-revealing data fusion model in metabolomics.

Image processing [?] NNTF for facial expression recognition

Machine Learning [?] Tensor decompositions for learning latent variable models

Text mining [?] This paper describes a method for automatic detection of semantic relations between concept nodes of a networked ontological knowledge base by analyzing matrices of semantic-syntactic valences of words. These matrices are obtained by means of nonnegative factorization of tensors of syntactic compatibility of words.

Numerical analysis [?] use of the sum-factorization for the calculation of the integrals arising in Galerkin isogeometric analysis. While introducing very little change in an isogeometric code based on element-by-element quadrature and assembling, the sum-factorization approach, taking advantage of the tensor-product structure of splines or NURBS shape functions, significantly reduces the quadrature computational cost.

[?] Fast iterative solution of the Bethe-Salpeter eigenvalue problem using low-rank and QTT tensor approximation.

[?] Blind identification of a second order Volterra-Hammerstein series using cumulant cubic tensor analysis.

Neuroscience [?] Decomposition of brain diffusion imaging data uncovers latent schizophrenias with distinct patterns of white matter anisotropy, using NNTF to clustering.

Signal processing Cichocki et.al. [?] sum up tensor decomposition approaches for signal processing problems.

Barker and Virtanen [?] deal with monaural sound source separation problem using NNTF of modulation spectrograms.

[?] Necessity to manually assign the NTF components to audio sources in order to be able to enforce prior information on the sources during the estimation process, Automatic Allocation of NTF Components for User-Guided Audio Source Separation

[?] propose a shifted 2D non-negative tensor factorisation algorithm which extends non-negative matrix factor 2D deconvolution to the multi-channel case. The use of this algorithm for multi-channel sound source separation of pitched instruments is demonstrated.

Other applications [?] Discovering and Characterizing Mobility Patterns in Urban Spaces: A Study of Manhattan Taxi Data. by using non-negative tensor factorization (NTF), we are able to cluster human behavior based on spatio-temporal dimensions. Second, for understanding these clusters, we propose to use HypTrails, a Bayesian approach for expressing and comparing hypotheses about human trails.

[?] NNTF factorization for household electrical seasonal consumption disaggregation

2.1 Canonica Polyadic Decomposition / PARAFAC

Canonica Polyadic (CP) decomposition or PARAFAC decompose a tensor as a finite sum of rank-one tensors.

Domanov [?] shows relaxed uniqueness conditions and algebraic algorithm for Canonical polyadic decomposition, as well as a reduction to generalized eigenvalue decomposition [?] and uniqueness properties [?] of third-order tensors.

2.2 Other methods

[?] The tensor decomposition addressed in this paper may be seen as a generalization of Singular Value Decomposition of matrices. We consider general multilinear and multihomogeneous tensors. We show how to reduce the problem to a truncated moment matrix problem and give a new criterion for flat extension of Quasi-Hankel matrices. We connect this criterion to the commutation characterization of border bases.

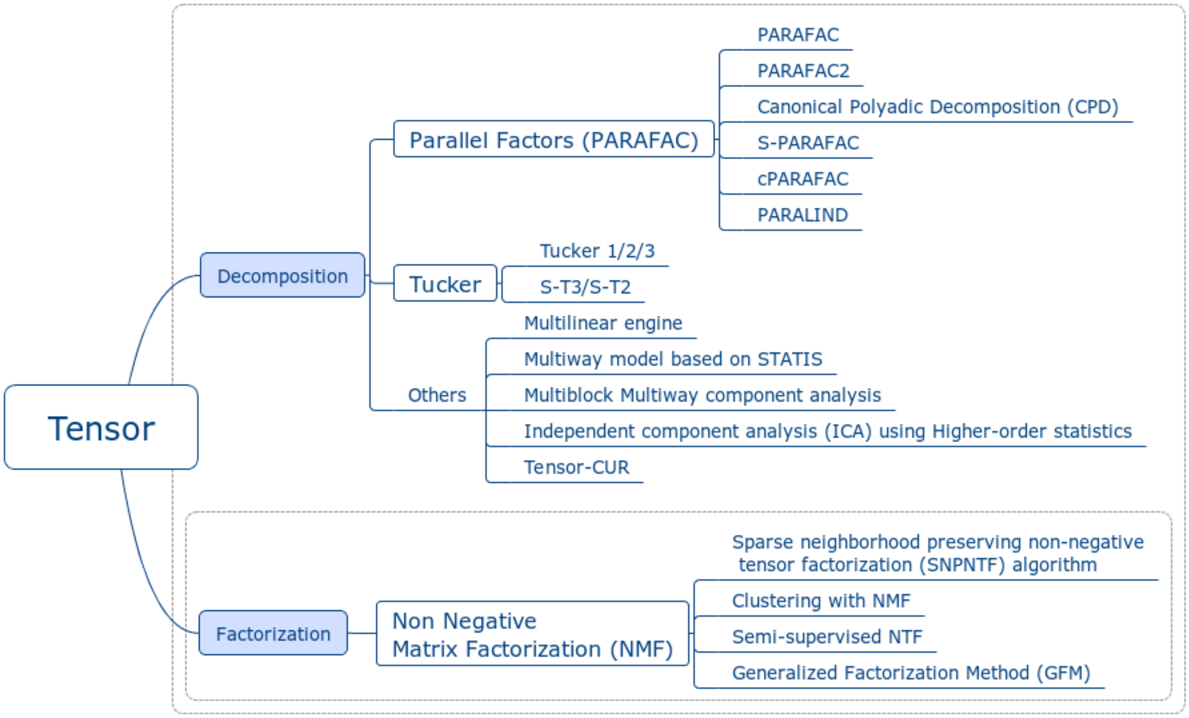

Tensor
Decomposition
Parallel Factors (PARAFAC)
PARAFAC
PARAFAC2
Canonical Polyadic Decomposition (CPD)
S-PARAFAC
cPARAFAC
PARALIND
Tucker
Tucker 1/2/3
S-T3/S-T2
Others
Multilinear engine
Multiway model based on STATIS
Multiblock Multiway component analysis
Independent component analysis (ICA) using Higher-order statistics
Tensor-CUR
Factorization
Non Negative
Matrix Factorization (NMF)
Sparse neighborhood preserving non-negative
tensor factorization (SNPNTF) algorithm
Clustering with NMF
Semi-supervised NTF
Generalized Factorization Method (GFM)

Figure 5: Tensor decomposition methods

2.3 Tensor completion

Acar et.al [?] presents a scalable tensor factorization method to deal with completion problem using PARAFAC method.

[?] The existing studies involving matrix or tensor completion problems are commonly under the nuclear norm penalization framework due to the computational efficiency of the resulting convex optimization problem. Folded-concave penalization methods have demonstrated surprising developments in sparse learning problems due to their nice practical and theoretical properties. To share the same light of folded-concave penalization methods, we propose a new tensor completion model via folded-concave penalty for estimating missing values in tensor data. Two typical folded-concave penalties, the minmax concave plus (MCP) penalty and the smoothly clipped absolute deviation (SCAD) penalty, are employed in the new model. To solve the resulting nonconvex optimization problem, we develop a local linear approximation augmented Lagrange multiplier (LLA-ALM) algorithm which combines a two-step LLA strategy to search a local optimum of the proposed model efficiently. Finally, we provide numerical experiments with phase transitions, synthetic data sets, real image and video data sets to exhibit the superiority of the proposed model over the nuclear norm penalization method in terms of the accuracy and robustness.

[?] The success of research on matrix completion is evident in a variety of real-world applications. Tensor completion, which is a high-order extension of matrix completion, has also generated a great deal of research interest in recent years. Given a tensor with incomplete entries, existing methods use either factorization or completion schemes to recover the missing parts. However, as the number of missing entries increases, factorization schemes may overfit the model because of incorrectly predefined ranks, while completion schemes may fail to interpret the model factors. In this paper, we introduce a novel concept: complete the missing entries and simultaneously capture the underlying model structure. To this end, we propose a method called simultaneous tensor decomposition and completion (STDC) that combines a rank minimization technique with Tucker model decomposition. Moreover, as the model structure is implicitly included in the Tucker model, we use factor priors, which are usually known a priori in real-world tensor objects, to characterize the underlying joint-manifold drawn from the model factors. By exploiting this auxiliary information, our method leverages two classic schemes and accurately estimates the model factors and missing entries. We conducted experiments to empirically verify the convergence of our algorithm on synthetic data and evaluate its effectiveness on various kinds of real-world data. The results demonstrate the efficacy of the proposed method and its potential usage in tensor-based applications. It also outperforms state-of-the-art methods on multilinear model analysis and visual data completion tasks.

Following Ji Lu et.al [?] notation

low rank matrix completion

$$\begin{aligned} &\min_X \ \mathrm{rank}(X) \\ &\text{s.t. } X_\Omega = M_\Omega \end{aligned} \tag{2}$$

where Ω is an index set, then X_Ω is coping entries of X in the indexes Ω and missed entries $\hat{\Omega}$ would be 0

The missing entries in X are determined in order to minimize the matrix X rank. i.e. a non convex optimization problem since rank is nonconvex.

Frequently, trace norm (or nuclear norm) $||\cdot||_*$ is used to approximate the rank of matrices. Trace norm is the tighest convex envelop for the matrices rank.

$$\begin{aligned} &\min_X ||X||_* \\ &\text{s.t. } X_\Omega = M_\Omega \end{aligned} \tag{3}$$

Since tensor is a generalization of the matrix concept, we generalize the optimization problem as

$$\begin{aligned} &\min_{\mathcal{X}} ||\mathcal{X}||_* \\ &\text{s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega \end{aligned} \tag{4}$$

Where $\mathcal{X}$ and $\mathcal{T}$ are n-order tensors with identical size.

2.4 Kernel methods

In contrast with traditional learning techniques, kernel methods do not need a vectorial representation of data. Instead, they use a kernel function. Therefore, kernel methods are naturally applied to unstructured, or complex structured, data such as texts, strings, trees and images [?].

Informally, a kernel function measures the similarity of two objects. Formally, a kernel function, $k : X \times X \to \mathbb{R}$, maps pairs (x, y) of objects in a set X, the problem space, to the reals. A kernel function implicitly generates a map, $\Phi : X \to F$, where F corresponds to a Hilbert space called the feature space. The dot product in F is calculated by k, specifically $k(x, y) =< \Phi(x), \Phi(y) >_F$. Given an appropriate kernel function, complex patterns in the problem space may correspond to simpler patterns in the feature space. For instance, non-linear patterns in the problem space may correspond to linear patterns in the feature space.

Both k-means and SNMF have kernelized versions, which receive as input a kernel matrix instead of a set of sample represented by feature vectors. The kernel version of k-means is called, unsurprisingly, kernel k-means (KKM). In the case of SNMF, the kernelized version works as follows.

SNMF starts with an initial estimation of the matrix factor H and iteratively update it using the updating equation:

$$H_{i,k} = H_{i,k}(1 - \beta + \beta \frac{((X^T X)H)_{i,k}}{(HH^T H)_{i,k}})$$

The kernel version of the algorithm is obtained by using a kernel matrix K instead of the expression $(X^T X)$, where K is an $l \times l$ matrix with $K_{i,j} = k(x_i, x_j)$. There are different types of kernels some of them general and some of them specifically defined for different types of data. The most popular general kernels are the linear kernel

$$k(x, y) =< x, y >, \tag{5}$$

the polynomial kernel

$$k(x, y) = p(< x, y >),$$

where $p(\,)$ is a polynomial with positive coefficients, and the Gaussian (or RBF) kernel

$$k(x, y) = e^{\frac{\|x-y\|^2}{2\sigma^2}}. \tag{6}$$

The cluster centroids estimated by the kernel versions of both algorithms are in the feature space and correspond to the points $C_j = \frac{1}{n} \sum_{x_i \in C_j} \Phi(x_i)$. However, we are interested on the pre-image in the original space of this centroids, i.e., points $\hat{C}_j$ such that $\Phi(\hat{C}_j) = C_j$. However, it is possible that a exact pre-image may not even exist, so we look for the $\hat{C}_j$ that minimizes the following objective function:$\min_{\hat{C}_j} \left\| \hat{C}_j - C_j \right\|^2$. According to Kwok et al. [?], the optimum C_j can be found by iterating the following fixed-point formula:

$$\hat{C}_j^{t+1} = \frac{\sum_{i=1}^{N} \exp(\frac{-||\hat{C}_j^t - x_i||)}{s}) x_i}{\sum_{i=1}^{N} \exp(\frac{-||\hat{C}_j^t - x_i||}{s})} \tag{7}$$

3 Kernel Non-negative Matrix Factorization

Kernel Non-negative Matrix Factorization (KNMF) can be naturally derivated of convex NMF (insert cites 92, 98 and 120 from Cichocki book). Given a kernel function $\phi : x \in X \to \phi(x) \in F$, mapping for N elements $\phi(X) = [\phi(x_1), \dots \phi(x_N)]$. Then, KNMF can be defined as

$$\phi(X) \cong \phi(X) W H^T \tag{8}$$

Therefore, the cost function to minimize is

$$||\phi(X) - \phi(X) W H^T||_F^2 = tr(K) - 2tr(H^T K W) + tr(W^T K W H^T H) \tag{9}$$

Where kernel $K = \phi^T(X) phi(X)$

4 Problem Statement

The general problem addressed by this research proposal is the design of non-supervised learning algorithms, in particular tensor factorization algorithms, applied in the space induced by a kernel function. Matrix and analogous tensor factorization is central to different important tasks in machine learning and information retrieval such as: clustering, latent topic analysis, recommendation, among others. Another important focus of this research is to study robustness from a kernel method perspective.

Kernel methods are ubiquitous in machine learning and there are some connections between some types of kernels (Guassian kernels) and robustness that has not been fully explored yet. The general research question to be addressed by this research is whether the use of some types of kernels may bring robustness to particular factorization methods. A satisfactory solution of this general challenge requires to answer some particular **research question**:

- How to decompose tensors in an space induced by a kernel function?

An answer to the question derivate in a method which factorize a given tensor in the feature space induced by a Kernel function. Naturally, to inquire about the effects of decompose tensors in an space induced by a kernel function open space to evaluate the proposal method performance as well as its capabilities dealing with multimodal data, scalability and robustness to noise and outliers.

References