DuSK: A Dual Structure-preserving Kernel for Supervised Tensor Learning with Applications to Neuroimages

Lifang He*   Xiangnan Kong†   Philip S. Yu ‡   Ann B. Ragin §   Zhifeng Hao ¶

Xiaowei Yang ‖

Abstract

With advances in data collection technologies, tensor data is assuming increasing prominence in many applications and the problem of supervised tensor learning has emerged as a topic of critical significance in the data mining and machine learning community. Conventional methods for supervised tensor learning mainly focus on learning kernels by flattening the tensor into vectors or matrices, however structural information within the tensors will be lost. In this paper, we introduce a new scheme to design structure-preserving kernels for supervised tensor learning. Specifically, we demonstrate how to leverage the naturally available structure within the tensorial representation to encode prior knowledge in the kernel. We proposed a tensor kernel that can preserve tensor structures based upon dual-tensorial mapping. The dual-tensorial mapping function can map each tensor instance in the input space to another tensor in the feature space while preserving the tensorial structure. Theoretically, our approach is an extension of the conventional kernels in the vector space to tensor space. We applied our novel kernel in conjunction with SVM to real-world tensor classification problems including brain fMRI classification for three different diseases (*i.e.*, Alzheimer's disease, ADHD and brain damage by HIV). Extensive empirical studies demonstrate that our proposed approach can effectively boost tensor classification performances, particularly with small sample sizes.

---

*Computer Science and Engineering, South China University of Technology, China. lifanghescut@gmail.com

†Computer Science Department, University of Illinois at Chicago, USA. xkong4@uic.edu

‡Computer Science Department, University of Illinois at Chicago, USA. psyu@cs.uic.edu

§Department of Radiology, Northwestern University, USA. ann-ragin@northwestern.edu

¶Faculty of Computer, Guangdong University of Technology, China. mazfhao@scut.edu.cn

‖School of Sciences, South China University of Technology, China. xwyang@scut.edu.cn

1 Introduction

Supervised learning is one of the most fundamental data mining tasks. Conventional approaches on supervised learning usually assume, explicitly or implicitly, that data instances are represented as feature vectors. However, in many real-world applications, data instances are more naturally represented as second-order (matrices) or higher-order tensors, where the order of a tensor corresponds to the number of modes or ways. For example, in computer vision, a grey-level image is inherently a 2-D object, which can be represented as a second-order tensor with the column and row modes [21]; in medical neuroimaging, an MRI (Magnetic Resonance Imaging) image is naturally a third-order tensor consisting of 3-D voxels [3]. Supervised learning on this type of data is called supervised tensor learning, where each instance in the input space is represented as a tensor. With the rapid proliferation of tensor data, supervised tensor learning has drawn significant attention in recent years in the machine learning and data mining communities.

A straightforward solution to supervised tensor learning is to convert the input tensors into feature vectors, and feed the feature vectors to a conventional supervised learning algorithm. However, tensor objects are commonly specified in high-dimensional space. For example, a typical MRI image of size $256 \times 256 \times 256$ voxels contains $16,777,216$ features [23]. This makes traditional methods prone to overfitting, especially for small sample size problems [4]. On the other hand, tensorial representations retain the information about the structure of the high-dimensional space the data lie in, such as about the spatial arrangement of the voxel-based features in a 3-D image. When converting tensors into vectors, such important structural information will be lost. In particular, the entries of a tensor object are often highly correlated with surrounding entries. For example, in MRI image data, adjacent voxels usually exhibit similar patterns, which means that the source images contain redundant information at this voxel. It is believed by many researchers that potentially more compact and useful representations can be extracted

from the original tensor data and thus result in more accurate and interpretable models. Therefore, supervised learning algorithms operating directly on tensors rather than their vectorized versions are much desired.

Formally, a major difficulty in supervised tensor learning is how to build predictive models that can leverage the naturally available structure of tensor data to facilitate the learning process. In the literature, several solutions have been proposed. Previous work on supervised tensor learning mainly focuses on linear models [1, 5, 6, 19, 23], which assume, explicitly or implicitly, that data are linearly separable in the input space. However, in practice this assumption is often violated and the linear decision boundaries do not adequately separate the classes. Recently, several approaches try to exploit the tensor structure with nonlinear kernel models [16, 17, 22], which first unfold the tensor along each of its modes, and then use these unfolded matrices to construct nonlinear kernels for supervised tensor learning as shown in Figure 1(b). However, these methods can only capture the relationships within each single mode of the tensor data, because the structural information about inter-mode relationships of tensor data is lost in the unfolding procedures.

In this paper, we study the problem of supervised tensor learning with nonlinear kernels which can adequately preserve and utilize the structure of the tensor data. The major research challenges of supervised tensor learning with structure-preserving kernels can be summarized as follows:

• **High-dimensional tensors**: One fundamental problem in supervised tensor learning lies in the intrinsic high dimensionality of tensor objects. Traditional supervised learning algorithms assume that the instances are represented as vectors. However, in the context of tensors, each data object is usually not represented as a vector but a high-dimensional multi-mode (also known as multi-way) array. If we reshape the tensor into a vector, the number of features is extremely high. Both computability and theoretical guarantee of the traditional models are compromised by this ultra-high dimensionality.

• **Complex tensor structure**: Another fundamental problem in supervised tensor learning lies in complex structure of tensors. Conventional tensor-based kernel approaches focus on unfolding tensor data into matrices [16, 17, 22] which can only preserve the one-way relationships within the tensor data. However, in many real-world applications, the tensor data have multi-way structures. Such prior knowledge about multi-way relationships among features should be incorporated to build more accurate and interpretable models, especially in the case of high dimensional tensor data with

(a) Vector-based kernels

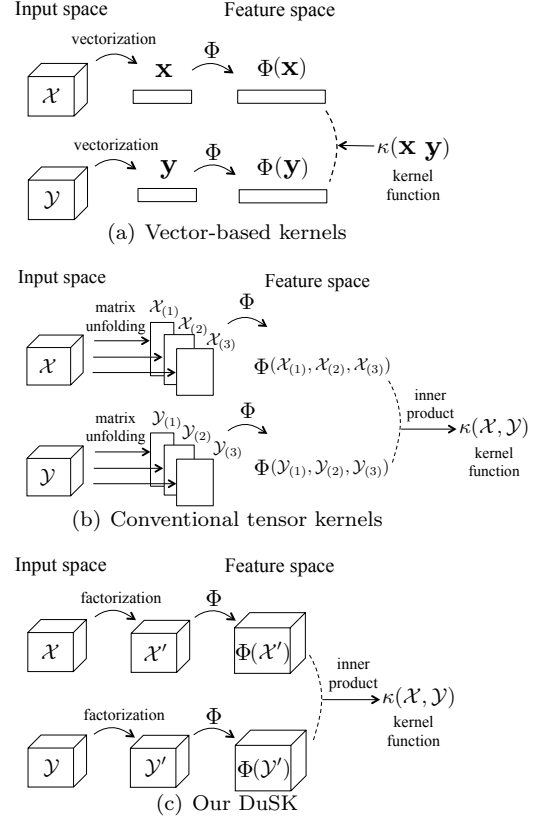(b) Conventional tensor kernels

(c) Our DuSK

Figure 1: Schematic view of the key difference among three kernel learning schemes. Standard kernel (a) works on the vectorized representation and conventional tensor-based kernel (b) applies tensor-to-matrix alignment first, which may lead to loss of structural information. Our method (c) works on the tensor representation directly.

small sample size.

• **Nonlinear separability**: In real-world applications, the data is usually not linearly separable in the input space. Conventional supervised tensor learning methods which can preserve tensor structures are often based upon linear models. Thus these methods cannot efficiently solve nonlinear learning problems on tensor data.

In this paper, we propose a novel approach to supervised tensor learning, called DuSK (Dual Structure-preserving Kernels). Our framework is illustrated in Figure 1(c). Different from conventional methods, our approach is based upon kernel methods and tensor factorization techniques that can fully capture the multiway structures of tensor data. We first extract a more compact and informative representation from the original data using a tensor factorization method, *i.e.*, CANDECOMP/PARAFAC (CP) [10]. Then we define a structure-preserving feature mapping to derive the DuSK kernels in the tensor product feature space,

used in conjunction with kernel machines to solve the supervised tensor learning problems. Empirical studies on real-world tasks (classifying fMRI images of different brain diseases, *i.e.*, Alzheimer's disease, ADHD and HIV) demonstrate that the proposed approach can significantly boost the classification performances on tensor datasets.

2 PRELIMINARIES

Before presenting our approach, we introduce some related concepts and notation of tensors. Table 1 lists some basic symbols defined in this study. We first give a formal mathematical definition of the tensor, which provides an intuitive understanding of the algebraic structure of the tensor that tensor object has the tensor product structure.

DEFINITION 1. (TENSOR) *An Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system.*

We use $\mathcal{A} = (a_{i_1,i_2,\dots,i_N}) \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ to denote a tensor $\mathcal{A}$ of N order. For $n = 1, 2, \cdots, N$, I_n is the dimension of $\mathcal{A}$ along the n-th mode. Based on the above definition, we define inner product, tensor norm, tensor product, and rank of a tensor and give CP model as follows:

DEFINITION 2. (INNER PRODUCT) *The inner product of two same-sized tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as the sum of the products of their entries:*

$$(2.1) \quad \langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} a_{i_1,i_2,\dots,i_N} b_{i_1,i_2,\dots,i_N}.$$

DEFINITION 3. (TENSOR NORM) *The norm of a tensor $\mathcal{A}$ is defined to be the square root of the sum of all entries of the tensor squared, i.e.,*
(2.2)

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} a_{i_1,i_2,\dots,i_N}^2}.$$

As we see the norm of a tensor is a straightforward generalization of the usual Frobenius norm for matrices and of the l_2 norm for vectors.

DEFINITION 4. (TENSOR PRODUCT) *The tensor product $\mathcal{A} \otimes \mathcal{B}$ of tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{I'_1 \times I'_2 \times \dots \times I'_M}$ is defined by*
(2.3)

$$(\mathcal{A} \otimes \mathcal{B})_{i_1,i_2,\dots,i_N,i'_1,i'_2,\dots,i'_M} = a_{i_1,i_2,\cdots,i_N} b_{i'_1,i'_2,\cdots,i'_M}$$

for all values of the indices.

Table 1: List of symbols

Symbol	Definition and Description
s	each lower-case represents a scale
$\mathbf{v}$	each boldface lowercase letter represents a vector
$\mathbf{M}$	each boldface capital letter represents a matrix
$\mathcal{T}$	each calligraphic letter represents a tensor
$\mathfrak{G}$	each gothic letter represent a general set or space
$\otimes$	denotes tensor product
$\langle .,. \rangle$	denotes the inner product in some feature space
$R = Rank(\mathcal{A})$	is the rank of tensor $\mathcal{A}$
$\phi(.)$	denotes the feature mapping
$\kappa(.,.)$	represents a kernel function

It is worth mentioning that a rank-one tensor, is still analogously to the matrix case, a tensor that is a tensor product of vectors (Nth-order tensor requires N vectors). Additionally, notice that for rank-one tensors $\mathcal{A} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \cdots \otimes \mathbf{a}^{(N)}$ and $\mathcal{B} = \mathbf{b}^{(1)} \otimes \mathbf{b}^{(2)} \otimes \cdots \otimes \mathbf{b}^{(N)}$, it holds that
(2.4)

$$\langle \mathcal{A}, \mathcal{B} \rangle = \left\langle \mathbf{a}^{(1)}, \mathbf{b}^{(1)} \right\rangle \left\langle \mathbf{a}^{(2)}, \mathbf{b}^{(2)} \right\rangle \cdots \left\langle \mathbf{a}^{(N)}, \mathbf{b}^{(N)} \right\rangle.$$

DEFINITION 5. (TENSOR RANK) *The rank of a tensor $\mathcal{A}$ is the minimum number of rank-one tensor to fit $\mathcal{A}$ exactly.*

DEFINITION 6. (CP FACTORIZATION) *Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and an integer R, if it can be expressed as*

$$(2.5) \qquad \mathcal{A} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)},$$

we call it CP factorization (see Figure 2 for graphical representations). For convenience, in the following we write $\prod_{n=1}^{N} \otimes \mathbf{a}^{(n)}$ for $\mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \cdots \otimes \mathbf{a}^{(N)}$.

3 APPROACH

In this section, we first formulate the problem of tensor-based kernel learning and then elaborate on our DuSK. For the sake of brevity, hereafter we restrict our discussion to classification problems.

3.1 Problem statement Considering a training set of M pairs of samples $\{\mathcal{X}_i, y_i\}_{i=1}^{M}$ for binary tensor classification problem, where $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ are the input of the sample and $y_i \in \{-1, +1\}$ are the corresponding class labels of $\mathcal{X}_i$. In [6], it was noted that the problem of tensor classification can be stated as a convex quadratic optimization problem in the framework of the standard linear SVM. Based on this result, we show how it can be modeled as a kernel learning problem.

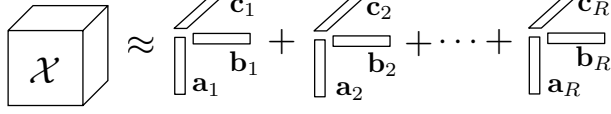Suppose we are given the optimization problem of

Figure 2: CP factorization of a third-order tensor

linear tensor classification as

$$(3.6) \quad \min_{\mathcal{W},b,\xi} \frac{1}{2} \|\mathcal{W}\|_F^2 + C \sum_{i=1}^{M} \xi_i,$$

$$(3.7) \quad \text{s.t. } y_i \left(\langle \mathcal{W}, \mathcal{X}_i \rangle + b \right) \geq 1 - \xi_i,$$

$$(3.8) \quad \xi_i \geq 0, \forall i = 1, \cdots, M.$$

Where $\mathcal{W}$ is the weight tensor of the separating hyperplane, b is the bias, ξ_i is the error of the ith training sample, and C is the trade-off between the classification margin and misclassification error.

Obviously, the optimization problem in (3.6)-(3.8) is the generalization of the problem of the standard linear SVM to tensor patterns in tensor space. When the input samples $\mathcal{X}_i$ are vectors, it degenerates into the standard linear SVM. As such, based on the kernel method for the extension of linear SVM to the nonlinear case−by introducing a nonlinear feature mapping ϕ : $\mathbf{x} \rightarrow \phi(\mathbf{x}) \in \mathfrak{H} \subset \mathbb{R}^H$, we develop a nonlinear extension of (3.6)-(3.8) in the following, which is critical for the derivation of the model for tensor-based kernel learning.

Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we assume it is mapped into the Hilbert space $\mathfrak{H}$ by

$$(3.9) \quad \phi : \mathcal{X} \rightarrow \phi(\mathcal{X}) \in \mathbb{R}^{H_1 \times H_2 \times \cdots \times H_P}.$$

Note that the project tensor $\phi(\mathcal{X})$ in space $\mathfrak{H}$ may have different order with $\mathcal{X}$, and each mode dimension is higher even an infinite dimension depending on the feature mapping function $\phi(.)$. Such a Hilbert space is called the high-dimensional tensor feature space or simply a tensor feature space. According to the same principle as the construction of linear classification model in the original tensor space, we construct the following model in this space:

$$(3.10) \quad \min_{\mathcal{W},b,\xi} \frac{1}{2} \|\mathcal{W}\|_F^2 + C \sum_{i=1}^{M} \xi_i,$$

$$(3.11) \quad \text{s.t. } y_i \left(\langle \mathcal{W}, \phi(\mathcal{X}_i) \rangle + b \right) \geq 1 - \xi_i,$$

$$(3.12) \quad \xi_i \geq 0, \forall i = 1, \cdots, M.$$

From the viewpoint of high-dimensional tensor feature space, this model is a linear model. However, from the viewpoint of the original tensor space, it is a nonlinear model. When the input samples $\mathcal{X}_i$ are vectors, it degenerates into the standard nonlinear SVM. When the feature mapping function $\phi(.)$ is an identical function,

i.e., $\phi(\mathcal{X}) = \mathcal{X}$, it is the same as that in (3.6)-(3.8). Thus, we say that the optimization model (3.10)-(3.12) is the nonlinear counterpart of (3.6)-(3.8).

Let us now show how this model can be exploited to obtain tensor-based kernel optimization model. Using Lagrangian relaxation method [2], it is easy to check that the dual problem of (3.10)-(3.12) is

$$(3.13)$$

$$\max_{\alpha_1, \alpha_2, \cdots, \alpha_M} \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \langle \phi(\mathcal{X}_i), \phi(\mathcal{X}_j) \rangle$$

$$(3.14) \quad \text{s.t.} \sum_{i=1}^{M} \alpha_i y_i = 0,$$

$$(3.15) \quad 0 \leq \alpha_i \leq C, \forall i = 1, \cdots, M.$$

Where α_i are the Lagrangian multipliers and $\langle \phi(\mathcal{X}_i), \phi(\mathcal{X}_j) \rangle$ are the inner product between the mapped tensors of $\mathcal{X}_i$ and $\mathcal{X}_j$ in the tensor feature space.

The advantage of formulation (3.13)-(3.15) over (3.10)-(3.12) is that the training data only appear in the form of inner products. Based on the fundamental principle of kernel method, by substituting the inner product $\langle \phi(\mathcal{X}_i), \phi(\mathcal{X}_j) \rangle$ with a suitable tensor kernel function $\kappa(\mathcal{X}_i, \mathcal{X}_j)$, we thus get the tensor-based kernel model. The resulting decision function is

$$(3.16) \quad f(\mathcal{X}) = sign \left(\sum_{i=1}^{M} \alpha_i y_i \kappa(\mathcal{X}_i, \mathcal{X}) + b \right).$$

3.2 DuSK From the above statement, we can see that tensor-based kernel learning degenerates into the study of kernel function, and the success of kernel methods depends strongly on the data representation encoded into the kernel function. Now we propose the DuSK. Our target is to leverage the naturally available structure of the tensor to facilitate kernel learning.

Tensors provide a natural and efficient representation for multi-way data, but there is no guarantee that such representation will be good for kernel learning. Since learning will only be successful if the regularities that underlie the data can be discerned by the kernel. As with the previous analysis for the characteristics of tensor object, we know that the essential information in the tensor is embedded in its multi-way structure. Thus, one important aspect of kernel learning for such complex objects is to represent them by sets of key structural features easier to manipulate, and design kernels on such sets.

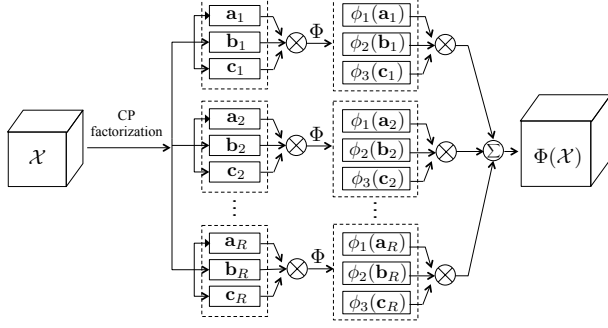According to the mathematical definition of tensor, we can gain a further understanding of the structure

Figure 3: Dual-tensorial mapping

of the tensor that tensor object has the tensor product structure. In previous work, it was found that CP factorization is particularly effective for extracting this structure. Motivated by these observations, we investigate how to exploit the benefits of CP factorization to learn a structure-preserving kernel in the tensor product feature space. More specifically, we will represent each tensor object as a sum of rank-one tensors in the original space and map them into the tensor product feature space for our kernel learning. In the following, we illustrate how to design the feature mapping.

We start by defining the following mapping on a rank-one tensor $\prod_{n=1}^{N} \otimes \mathbf{x}^{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

(3.17)
$$\phi : \prod_{n=1}^{N} \otimes \mathbf{x}^{(n)} \rightarrow \prod_{n=1}^{N} \otimes \phi\left(\mathbf{x}^{(n)}\right) \in \mathbb{R}^{H_1 \times H_2 \times \cdots \times H_N}.$$

Let the CP factorization of $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be $\mathcal{X} = \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{x}_r^{(n)}$ and $\mathcal{Y} = \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{y}_r^{(n)}$ respectively. By using the concept of the kernel function, we see that the kernel can be defined directly with inner product in the feature space. Thus, when $R = 1$, based on the above mapping and Eq. 2.4, we can directly derive the naive tensor product kernels, *i.e.*,

(3.18)
$$\kappa\left(\mathcal{X}, \mathcal{Y}\right) = \prod_{n=1}^{N} \kappa\left(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\right).$$

Despite this, many authors has demonstrated that a simple rank-one tensor cannot provide compact and informative presentation for original data [24]. The key point is how to design feature mapping when the value of R is more than one.

Based on the definition of the kernel function, it is easy to find that the feature space is a high-dimensional space of the original space, equipped with the same operations. Thus, we can factorize tensor data directly in the feature space the same as original space. This is formally equivalent to performing the following mapping:

(3.19)
$$\phi : \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{x}_r^{(n)} \rightarrow \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \phi\left(\mathbf{x}_r^{(n)}\right).$$

In this sense, it corresponds to mapping tensors into high-dimensional tensors that retain the original structure. More precisely, it can be regarded as mapping the original data into tensor feature space and then conducting the CP factorization in the feature space. We call it the dual-tensorial mapping function (see Figure 3).

After mapping the CP factorization of the data into the tensor product feature space, the kernel itself is just the standard inner product of tensors in that feature space. Thus, we derive our DuSK:

(3.20)
$$\kappa\left(\sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{x}_r^{(n)}, \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{y}_r^{(n)}\right)$$
$$= \sum_{i=1}^{R} \sum_{j=1}^{R} \prod_{n=1}^{N} \kappa\left(\mathbf{x}_i^{(n)}, \mathbf{y}_j^{(n)}\right)$$

From its derivation, we know such a kernel can take the multi-way structure flexibility into account. In general, the DuSK is an extension of the conventional kernels in the vector space to tensor space, and each vector kernel can be used in this framework for supervised tensor learning in conjunction with kernel machines.

3.3 Efficiency We consider the case of Gaussian RBF kernel in our framework, which is one of the most popular kernels that have been proven successful in many different contexts. Assume that a set of tensor data $\{(\mathcal{X}_i, y_i)\}_{i=1}^{M}$ is given, where $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The time complexity of computing a Gaussian RBF kernel matrix is $O\left(M^2 \prod_{n=1}^{N} I_n\right)$ and our method DuSK is thus $O\left(M^2 R^2 \sum_{n=1}^{N} I_n\right)$. A typical characteristic associated with tensor data is very high dimensional while R is often very small, which indicates our proposed method is significantly more efficient than its vector counterpart. It is also worth mentioning that our method depends on CP factorization technique, but it is backed with rapid implementation [13]. The storage complexity is reduced from $O\left(M \prod_{n=1}^{N} I_n\right)$ to $O\left(M \sum_{n=1}^{N} I_n\right)$, where the data is compressed without quality loss and can be recovered quickly. Furthermore, since the constituent kernels are Gaussian RBF kernels,
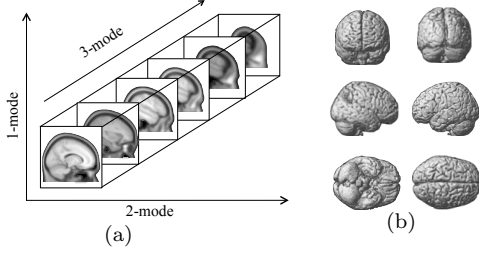
Figure 4: (a) An illustration of a three-order tensor (fMRI image), (b) An visualization of fMRI image.

we can thus reformulate Eq. 3.20 to

$$
\begin{aligned}
(3.21) \quad \kappa\left(\mathcal{X}, \mathcal{Y}\right) &= \sum_{i=1}^{R} \sum_{j=1}^{R} \prod_{n=1}^{N} \kappa\left(\mathbf{x}_i^{(n)}, \mathbf{y}_j^{(n)}\right) \\
&= \sum_{i=1}^{R} \sum_{j=1}^{R} \exp\left(-\sigma \sum_{n=1}^{N} \|\mathbf{x}_i^{(n)} - \mathbf{y}_j^{(n)}\|^2\right)
\end{aligned}
$$

where σ is used to set an appropriate bandwidth. We denote this kernel as DuSK$_{\text{RBF}}$.

4 Experiment Evaluation

In this study, we validate the effectiveness of the DuSK$_{\text{RBF}}$ kernel within standard SVM framework for tensor classification, which we refer to as DuSK$_{\text{RBF}}$. As an application we consider an example of neuroimaging mining.

4.1 Data collection
We use three real-world fMRI datasets in our experimental evaluation as follows.

• *Alzheimer's Disease (ADNI)*: The first dataset is collected from the Alzheimer's Disease Neuroimaging Initiative[1]. The dataset consists of records of patients with Alzheimer's Disease (AD) and Mild Cognitive Impairment (MCI). We downloaded all records of resting-state fMRI images and apply SPM8 toolbox[2] to preprocess the data. We deleted the first ten volumes for each individual, and functional images were realigned to the first volume, slice timing corrected, and normalized to the MNI template and spatially smoothed with an 8-mm FWHM Gaussian kernel. Resting-State fMRI Data Analysis Toolkit (REST[3]) was then used to remove the linear trend of time series and temporally band-pass filtering ($0.01 - 0.08$ Hz). The average value of each subject over time series was calculated within each of those boxes, thereby resulting in 33 samples and a sum total of $61 \times 73 \times 61 = 271633$ voxels (or features). We treat the normal brains as negative class, and AD+MCI as

the positive class. Each individual is linearly rescaled to $[0, 1]$. Feature normalization is an important procedure, since the brain of every individual is different.

• *Attention Deficit Hyperactivity Disorder (ADHD)*: The second dataset is collected from ADHD-200 global competition dataset[4]. The dataset contains records of resting-state fMRI images for 776 subjects with $58 \times 49 \times 47 = 133574$ voxels, which are labeled as real patients (positive) and normal controls (negative). The original dataset is unbalanced, we randomly sampled 100 ADHD patients and 100 normal controls from the dataset for performance evaluation and the average over time series is conducted. Such dataset are quite special, all algorithms perform bad with normalization, we use non-normalized dataset.

• *Human Immunodeficiency Virus Infection (HIV)*: The third dataset is collected from the Department of Radiology in Northwestern University [20]. The dataset contains fMRI brain images of patients with early HIV infection (positive) as well as normal controls (negative). The same preprocessing steps as in ADNI dataset were given. This contains 83 samples with $61 \times 73 \times 61 = 271633$ voxels.

4.2 Baselines and Metrics
In order to establish a comparative study, we use seven state-of-the-art methods as baselines, each representing a different strategy. We here focus on SVM classifier, since it has been proven successful in many applications.

• Gaussian-RBF: a Gaussian-RBF kernel-based SVM, which is now the most widely used vector-based method for classification. In the following methods, if not stated explicitly, we use SVM with Gaussian RBF kernel as the classifier.

• Factor kernel: a matrix unfolding based tensor kernel, which is recently proposed in [16] and the constituent kernels belong to a class of Gaussian RBF kernels.

• K$_{3rd}$ kernel: a class of vector-based tensor kernels, aiming at representing the tensor in each vector space to capture structural information and have been applied to analyze fMRI images in conjunction with Gaussian RBF kernel [14].

• Linear SHTM: a linear support higher-order tensor machine [6], which is one of the most effective methods for tensor classification that generalizes linear SVM to tensor pattern using CP factorization and can be regarded as a special case of DuSK, namely the constituent kernels are linear kernels. This baseline is used to test the ability of our proposed method to cope with complex (possibly nonlinear) structured data.

• Linear kernel: linear SVM has also been increasingly

Table 2: Average classification accuracy comparison: mean (standard deviation).

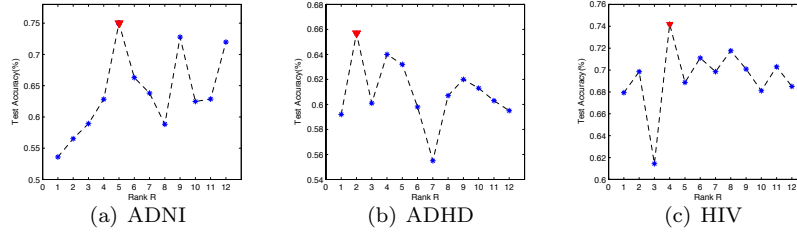| Dataset	DuSK$_{RBF}$	Gaussian RBF	Factor kernel	K$_{3rd}$ kernel	linear SHTM	linear SVM	PCA+SVM	MPCA+SVM
ADNI	**0.75 (0.18)**	0.49 (0.23)	0.51 (0.21)	0.55 (0.14)	0.52 (0.31)	0.42 (0.27)	0.50 (0.02)	0.51 (0.02)
ADHD	**0.65 (0.01)**	0.58 (0.00)	0.50 (0.00)	0.55 (0.00)	0.51 (0.03)	0.51 (0.01)	0.63 (0.01)	0.64 (0.01)
HIV	**0.74 (0.00)**	0.70 (0.00)	0.70 (0.01)	**0.75 (0.02)**	0.70 (0.01)	0.74 (0.01)	0.73 (0.25)	0.72 (0.02)

(a) ADNI   (b) ADHD   (c) HIV

Figure 5: Test accuracy vs. R on (a) ADNI, (b) ADHD, and (c) HIV, where the red triangles indicate the peak positions.

used to handle fMRI data. In some cases, it outperforms SVM using nonlinear kernels.

• PCA+SVM: Principal component analysis (PCA) is a vector-based subspace learning algorithms, which are commonly used for dealing with high-dimensional data, in particular fMRI data.

• MPCA+SVM: Multilinear principal component analysis (MPCA) [12] is a natural extension of PCA to tensors, which are used to handle high-dimensional tensor data.

The first three baselines are used to show the improvement of our proposed method over current kernel approaches to tensor classification. The last two baselines are used to test the effectiveness of our proposed method compared to unsupervised methods for tensor classification.

The effectiveness of an algorithm is always evaluated by test accuracy, we utilize it as metrics in the experiments. For our proposed method and linear SHTM, we choose the most popular and widely used enhanced linear search method [13] as its CP factorization strategy. All of the related methods select the optimal trade-off parameter from $C \in \{2^{-5}, 2^{-4}, \cdots, 2^9\}$ and kernel width parameter from $\sigma \in \{2^{-4}, 2^{-3}, \cdots, 2^9\}$. Considering the fact that there is no known closed-form solution to determine the rank R of a tensor a priori [9], and rank determination of a tensor is still an open problem [18], in our method and linear SHTM, we use grid search to determine the optimal rank and the optimal trade-off parameter together, where the rank $R \in \{1, 2, \cdots, 12\}$. The influence of different rank parameters on the classification performance of our method is also given.

All the experiments are conducted on a computer with Intel Core2^{TM}1.8GHz processor and 3.5GB RAM memory running Microsoft Windows XP.

4.3 Classification Performance In our experiments, we first randomly sample 80% of the whole data

as the training set, and the remaining samples as the test set. This random sampling experiment was repeated 50 times for all methods. The average performances of each method are reported. Table 2 shows the average classification accuracy and standard deviation of seven algorithms on three datasets, where the best result is highlighted in bold type.

From the experimental results in Table 2, we can observe that the classification accuracy of each method on different dataset can be quite different. However, the best method that outperforms other methods in all datasets is DuSK$_{RBF}$, especially for ADNI dataset. It is worth noting that in neuroimaging task it is very hard for classification algorithms to achieve even moderate classification accuracy on ADNI dataset since this data is extremely high dimensional but with small sample size. While we can observe an 20% gain over comparison methods. Based on this result, we can conclude that operation on tensors is much more effective than on matrices and vectors for high-dimensional tensor data analysis.

So far we have demonstrated that our proposed method is effective for tensor classification. However, it is still interesting to show how the data structure for tensor is actually used in our method. We focus on ADNI dataset to conduct an analysis. Figure 6 shows the visualization of original ADNI object and reconstruction result from our chosen CP factorization. As illustrated, CP factorization can fully capture the multi-way structure of the data, thus our method take it into account in the learning process.

4.4 Parameter Sensitivity Although the optimal rank parameter R , the optimal trade-off parameter C and kernel width parameter σ are found by a grid search in DuSK$_{RBF}$, it is still important to see the sensitivity of DuSK$_{RBF}$ to the rank parameter R. For this purpose, we demonstrate a sensitivity study over
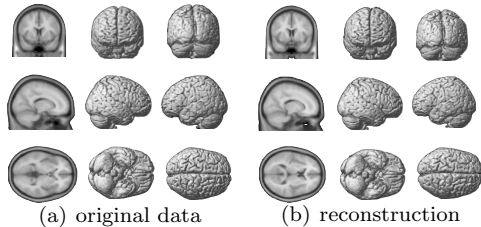
(a) original data (b) reconstruction

Figure 6: (a) is visualization of original ADNI object (a cross section is shown on the left and a 3D plot on the right) and (b) is reconstruction result from our chosen CP factorization.

different $R \in \{1, 2, \cdots, 12\}$ in this section, where the optimal trade-off parameter and kernel width parameter are still selected from $C \in \{2^{-5}, 2^{-4}, \cdots, 2^9\}$ and $\sigma \in \{2^{-4}, 2^{-3}, \cdots, 2^9\}$ respectively. According to the aforementioned analysis, we know that the efficiency of DuSK_{RBF} is reduced when R is increased because a higher value of R implies that more items are included into kernel computations. Thus, we only demonstrate the variation in test accuracy over different R on three datasets. As shown in Figure 5, we can observe that the rank parameter R has a significant effect on the test accuracy and the optimal value of R depends on the data, while the optimal value of R lies in the range $2 \leq R \leq 5$, which may provide a good guidance for selection of the R in advance.

In summary, the parameter sensitivity study indicates that the classification performance of DuSK_{RBF}+SVM relies on parameter R and it is difficult to specify an optimal value for R in advance. However, in most cases the optimal value of R lies in a small range of values as demonstrated in [6] and it is not time-consuming to find it using the grid search strategy in practical applications.

5   Related Work

From the conceptual perspective, two topics can be seen as closely related to our DuSK approach: supervised tensor learning and tensor factorization. This section gives a short overview of these areas and distinguishes DuSK from other existing solutions.

Tensor factorizations: Tensor factorizations are higher-order extensions of matrix factorization that elicit intrinsic multi-way structures and capture the underlying patterns in tensor data. These techniques have been widely used in diverse disciplines to analyze and process tensor data. A thorough survey of these techniques and applications can be found in [10]. The two most commonly factorizations are CP and Tucker. CP is a special case of Tucker decomposition which forces the core array to a (super)diagonal form. It is thus more condensed than that of Tucker. In the supervised ten-

sor learning setting, CP is more frequently applied to explore tensor data because of its properties of uniqueness and simplicity [6, 8, 19, 23]. However, in these applications, CP factorization is used either for exploratory analysis or to deal with linear tensor-based models. In this study, we employ the CP factorization to foster the use of kernel methods for supervised tensor learning.

Supervised tensor learning: Supervised tensor learning has been extensively studied in recent years [1, 5, 11, 19, 23]. Most of previous work has concentrated on learning linear tensor-based models, whereas the problem of how to build nonlinear models directly on tensor data has not been well studied. A first attempt in this direction focused on second-order tensors and led to a non-convex optimization problem [15]. Subsequently, the authors claimed that it can be extended to deal with higher-order tensors at the cost of a higher computational complexity, and proposed a factor kernel for tensors of arbitrary order except for square matrices based upon matrix unfoldings [16]. In the context of this proposal, Signorette et al. [17] introduced a cumulant-based kernel approach for classification of multichannel signals. Zhao et al. [22] presented a kernel tensor partial least squares for regression of lamb movements. A drawback of the approaches in [16, 17, 22] is that they can only capture the one-way relationships within the tensor data, because the tensors are unfolded into matrices. The multi-way structures within tensor data are already lost before the kernel construction process. Different from these methods, we aim to directly exploit the algebraic structure of the tensor to study structure-preserving kernels.

Another recent work by Hardoon et al. [7], although not directly performs supervised tensor learning, is worth mentioning in this context. They introduced the so-called tensor kernels to analyze neuroimaging data from multiple sources, which demonstrated that the tensor product feature space is useful for modeling interactions between feature sets in different domains. In this study, we make use of the tensor product feature space to derive our kernels in vivo the incorporation of CP model. The tensor kernels can be cast as a special case of our framework.

6   Conclusion and Future work

In this paper we have introduced a new tensor-based kernel methodology and first operate directly on tensors. We have applied our method on the problem of fMRI classification. The results indicate that the prior structural information can indeed improve the classification performance, particularly with small-sample size. As previous work limited on learning with matrices and vectors, this paper provides a new insight into the un-

derstanding of the principles and ideas underlying the concept of tensor.

In the future, we will investigate the reconstruction techniques of tensor data, so that our method can handle high-dimensional vector data more effectively. Another interesting topic would be to design some special method to address the parameter problem. Further study on this topic will also include many applications of DuSK kernels in real-world unsupervised learning with tensor representations.

Acknowledgements

References

[1] D. Cai, X. He, and J. Han. Learning with tensor representation. Computer Science Technical Report UIUCDCS-R-2006-2716, University of Illinois at Urbana-Champaign, 2006.

[2] E. Chong and S. Zak. *An introduction to optimization.* Wiley interscience, 2001.

[3] A. Cichocki. Tensors decompositions: New concepts for brain data analysis? *Journal of Control Measurement, and System Integration*, 7:507–517, 2013.

[4] J. Davis and I. Dhillon. Structured metric learning for high dimensional problems. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–203, Las Vegas, Nevada, USA, 2008.

[5] W. Guo, I. Kotsia, and I. Patras. Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827, 2012.

[6] Z. Hao, L. He, B. Chen, and X. Yang. A linear support higher-order tensor machine for classification. *IEEE Transactions on Image Processing*, 22(7):2911–2920, 2013.

[7] D. Hardoon and J. Shawe-Taylor. Decomposing the tensor kernel support vector machine for neuroscience data with structured labels. *Machine Learning*, 79(1):1–18, 2010.

[8] A. Jukić, I. Kopriva, and A. Cichocki. Canonical polyadic decomposition for unsupervised linear feature extraction from protein profiles. In *Proceeding of the European Signal Processing Conference*, Marrakech, Morocco, 2013.

[9] M. Kilmer and C. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[10] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[11] I. Kotsia and I. Patras. Support tucker machines. In *Proceedings of Computer Vision and Pattern Recognition*, pages 633–640, Providence, RI, 2011.

[12] H. Lu, K. Plataniotis, and A. Venetsanopoulos. Mpca: multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks and Learning Systems*, 19(1):18–39, 2008.

[13] D. Nion and L. D. Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. *Signal Processing*, 21:749–755, 2008.

[14] S. Park. Multifactor analysis for fmri brain image classification by subject and motor task. Electrical and computer engineering technical report, Carnegie Mellon University, 2011.

[15] M. Signoretto, L. D. Lathauwer, and J. Suykens. Kernel-based learning from infinite dimensional 2-way tensors. In *Proceedings of the 20th International Conference on Artificial Neural Networks*, pages 59–69, Thessaloniki, Greece, 2010.

[16] M. Signoretto, L. D. Lathauwer, and J. Suykens. a kernel based framework to tensorial data analysis. *Neural Networks*, 24(8):861–874, 2011.

[17] M. Signoretto, E. Olivetti, L. D. Lathauwer, and J. Suykens. Classification of multichannel signals with cumulant-based kernels. *IEEE Transactions on Signal Processing*, 45(12):2304–2314, 2012.

[18] V. Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[19] D. Tao, X. Li, X. Wu, W. Hu, and S. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007.

[20] X. Wang, P. Foryt, R. Ochs, J. Chung, Y. Wu, T. Parrish, and A. Ragin. Abnormalities in resting-state functional connectivity in early human immunodeficiency virus infection. *Brain Connectivity*, 1(3):207, 2011.

[21] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*, 16(1):212–220, 2007.

[22] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki. kernel-based tensor partial least squares for reconstruction of limb movements. In *Proceedings of Acoustics, Speech, and Signal Processing*, pages 633–640, Providence, RI, 2013.

[23] H. Zhou, L. . Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *American Statistical Association*, 2012.

[24] Y. Zhu, J. He, and R. Lawrence. Hierarchical modeling with tensor inputs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1233–1239, Toronto, Canada, 2012.