MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Optimization of Tensor Train Canonical Decomposition in the Support Tensor Machine

Peter Benner (Max Planck Institute DCTS and TU Chemnitz)
Sergey Dolgov (University of Bath)
Kirandeep Kour (Max Planck Institute DCTS)
Martin Stoll (TU Chemnitz)

May 17-21, 2021

SIAM Applied Linear Algebra, 2021

Supported by:

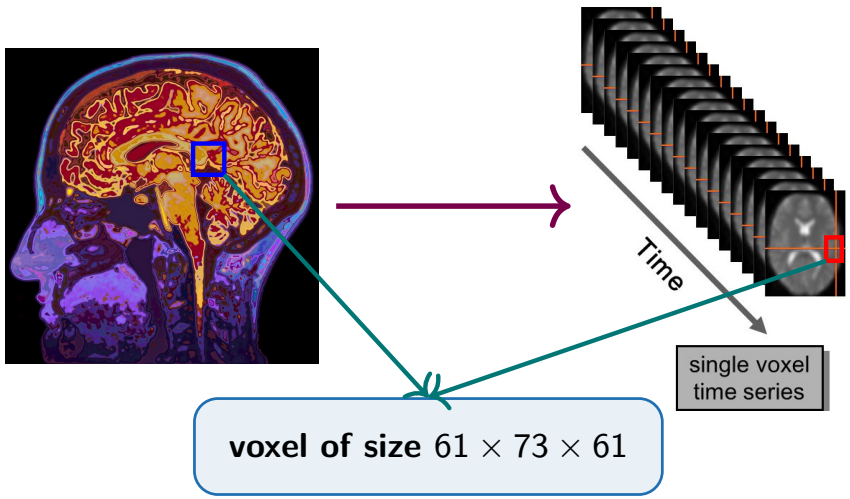
IMPRS ProEng
International Max Planck Research School for Advanced Methods in Process and Systems Engineering
Magdeburg

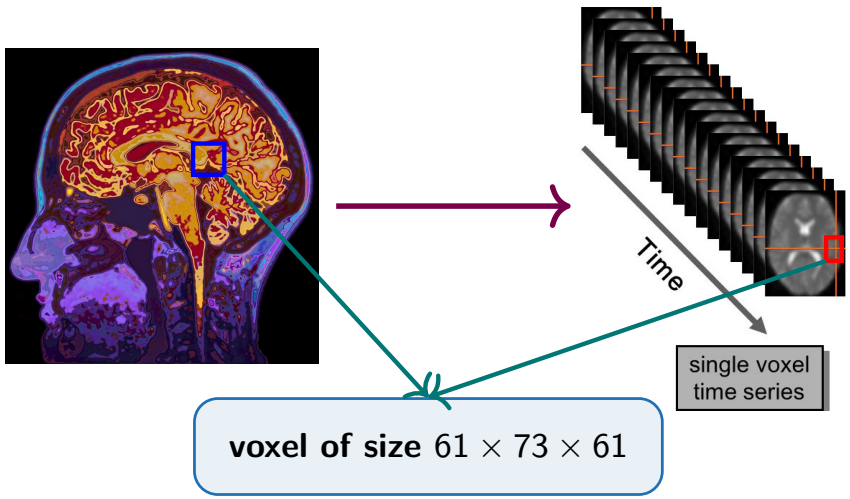
Partners:

TECHNISCHE UNIVERSITÄT CHEMNITZ

UNIVERSITY OF BATH

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG


Time
single voxel
time series
voxel of size $61 \times 73 \times 61$



Dataset
+ Small Sample Size (SSS)
+ Multi Dimensional (MD)

- Binary Classification
- Learning Model
- Nonlinear Boundary
  + SSS + MD

Efficienct Model

- Robustness
- State-of-the-art

for SSS + MD

Model for Small Sample Size

- Binary Classification $\rightarrow$ X - input (all the data points), y - output $\{-1, 1\}$
- Learning Model $\rightarrow$ Support Vector Machine (SVM)
- Nonlinear Boundary $\rightarrow$ projection onto higher-dimensional space

Support Vector Machine

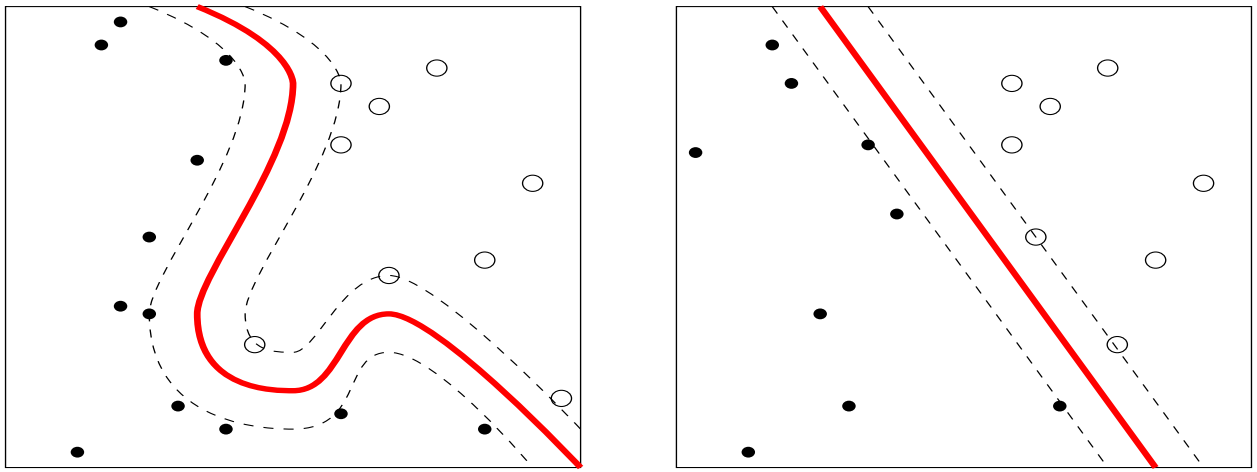


Figure: Non-linear(kernel) and linear SVM.

SVM: It is a discriminative classifier formally defined by a separating hyperplane. Generally, this hyperplane has a margin from both side of the labeled data.

SVM: It is a discriminative classifier formally defined by a separating hyperplane. Generally, this hyperplane has a margin from both side of the labeled data.

- X - input (all the data points),
- y - output $\{-1, 1\}$,
- L - hyperplane,

SVM: It is a discriminative classifier formally defined by a separating hyperplane. Generally, this hyperplane has a margin from both side of the labeled data.

- X - input (all the data points),
- y - output $\{-1, 1\}$,
- L - hyperplane,

we initialize $w \perp L$ and $\text{sign}(\langle x, w \rangle)$ helps in classification. The hypothesis is,

$$h_w(x) = \begin{cases} \langle x, w \rangle + b = 0, & y = 1, \\ \langle x, w \rangle + b \neq 0, & y = -1. \end{cases}$$

Machine Learning Model

Support Vector Machine I [C. Cortes; V. Vapnik, 1995]

SVM: It is a discriminative classifier formally defined by a separating hyperplane. Generally, this hyperplane has a margin from both side of the labeled data.

- X - input (all the data points),
- y - output $\{-1, 1\}$,
- L - hyperplane,

we initialize $w \perp L$ and $\text{sign}(\langle x, w \rangle)$ helps in classification. The hypothesis is,

$$h_w(x) = \begin{cases} \langle x, w \rangle + b = 0, & y = 1, \\ \langle x, w \rangle + b \neq 0, & y = -1. \end{cases}$$

Taking maximum margin to the nearest point for selecting w, b and L.

$$\max_{w} \min_{x_i} \quad |\langle x_i, \frac{w}{||w||} \rangle + b|$$
$$\text{subject to} \quad \text{sign}(\langle x_i, w \rangle + b) = \text{sign}(y_i).$$

Linear Boundary Optimization for SVM

$$\min_{w,b,\xi} J(w,b,\xi) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad y_i \underbrace{(w^T x_i + b)}_{L} \geq 1 - \xi_i, \qquad \xi_i \geq 0, i = 1, \ldots, N.$$

Linear Boundary Optimization for SVM

$$\min_{w,b,\xi} J(w,b,\xi) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad y_i \underbrace{(w^T x_i + b)}_{L} \geq 1 - \xi_i, \qquad \xi_i \geq 0, i = 1,\ldots,N.$$

(Dual) Linear Boundary Optimization for SVM

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,=1}^{N}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad i = 1,\ldots,N.$$

Projection onto higher-dimensional space

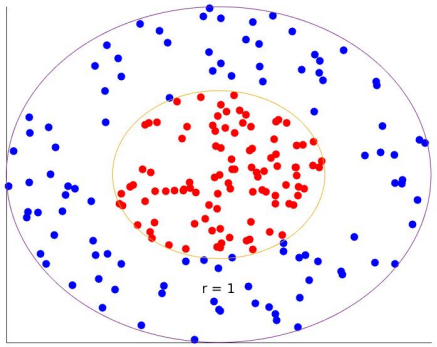


Figure: Nonlinear classification of data in ($\mathbb{R}^2$).

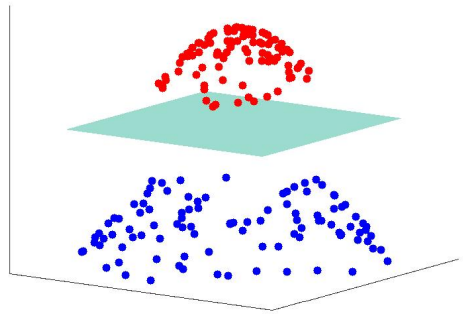


Figure: Linear classification in higher-dimension ($\mathbb{R}^3$).

Projection onto higher-dimensional space

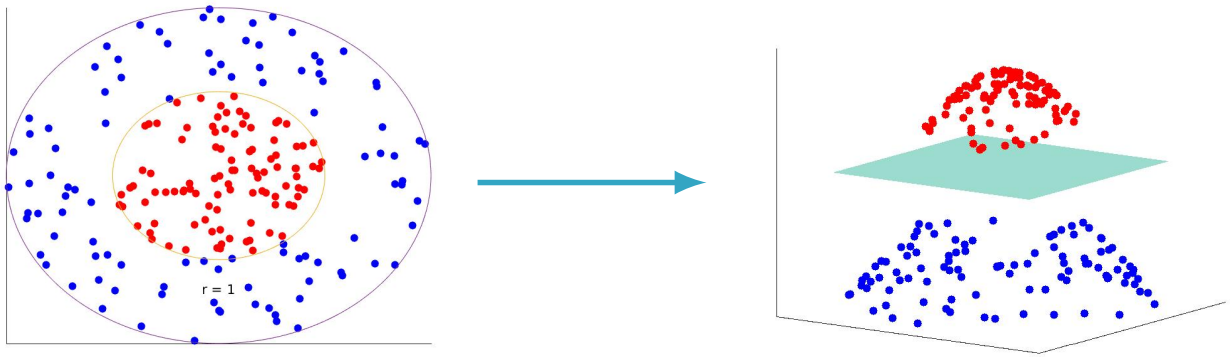


Figure: Nonlinear classification of data in ($\mathbb{R}^2$).

Figure: Linear classification in higher-dimension ($\mathbb{R}^3$).

Non-linear Boundary Optimization for SVM

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad i = 1, \ldots, N.$$

Feature map: $\phi : \mathcal{X}(\text{input}) \longrightarrow \mathcal{V}$ (feature space) $\langle \phi(x), \phi(x') \rangle = k(x, x')$

Feature map: $\phi : \mathcal{X}(\text{input}) \longrightarrow \mathcal{V}$ (feature space) $\langle \phi(x), \phi(x') \rangle = k(x, x')$

Non-linear Boundary Optimization for SVM

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x_i}, \mathbf{x_j})$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad i = 1, \ldots, N.$$

Feature map: $\phi : \mathcal{X}(\text{input}) \longrightarrow \mathcal{V}$ (feature space) $\langle \phi(x), \phi(x') \rangle = k(x, x')$

Non-linear Boundary Optimization for SVM

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x_i}, \mathbf{x_j})$$

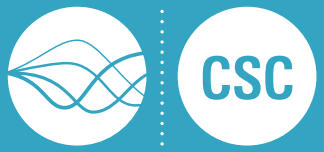
$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad i = 1, \ldots, N.$$

Including Multi-dimensional Constraint

Classification for different datatypes

(SVM) $k(x, x') \rightarrow$ RBF

$k(\mathcal{X}, \mathcal{X}') \rightarrow$ How??



Why Tensor for SVM??

Model for Multi-dimensionality

- SVM has input as a vector,
  - Storing memory for multidimensional data can be explosive,
  - Structural data can loose information during tensor-vector conversion,

Why Tensor for SVM??

Model for Multi-dimensionality

- SVM has input as a vector,
  - Storing memory for multidimensional data can be explosive,
  - Structural data can loose information during tensor-vector conversion,
- Extension of SVM is available and it is called Support Tensor Machine (STM),

Why Tensor for SVM??

Model for Multi-dimensionality

- SVM has input as a vector,
  - Storing memory for multidimensional data can be explosive,
  - Structural data can loose information during tensor-vector conversion,
- Extension of SVM is available and it is called Support Tensor Machine (STM),
- STM works directly with tensor as an input.

Why Tensor for SVM??

Model for Multi-dimensionality

- SVM has input as a vector,
  - Storing memory for multidimensional data can be explosive,
  - Structural data can loose information during tensor-vector conversion,
- Extension of SVM is available and it is called Support Tensor Machine (STM),
- STM works directly with tensor as an input.

Dual Non-linear Boundary Optimization Problem for STM

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathcal{X}_i, \mathcal{X}_j)$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad i = 1, \ldots, N.$$

- Manipulating a tensor is often prone to the curse of dimensionality $\mathcal{O}(\mathcal{I}^M)$.
- A tensor decomposition works efficiently with curse of dimensionality.

- Manipulating a tensor is often prone to the curse of dimensionality $\mathcal{O}(\mathcal{I}^M)$.
- A tensor decomposition works efficiently with curse of dimensionality.
- Specially, Tensor-Train decomposition provides a compact form with storage $\mathcal{O}(M\mathcal{I}R^2)$

Tensor-Train Decomposition [I.V. Oseledets; E. Tyrtyshnikov, 2009]

$$x_{i_1 i_2 \ldots i_M} \cong \sum_{r_0,\ldots,r_M} \mathcal{G}^{(1)}_{r_0,i_1,r_1} \mathcal{G}^{(2)}_{r_1,i_1,r_2} \cdots \mathcal{G}^{(M)}_{r_{M-1},i_M,r_M},$$

$$\mathcal{X} \cong \langle\!\langle \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(M)} \rangle\!\rangle,$$

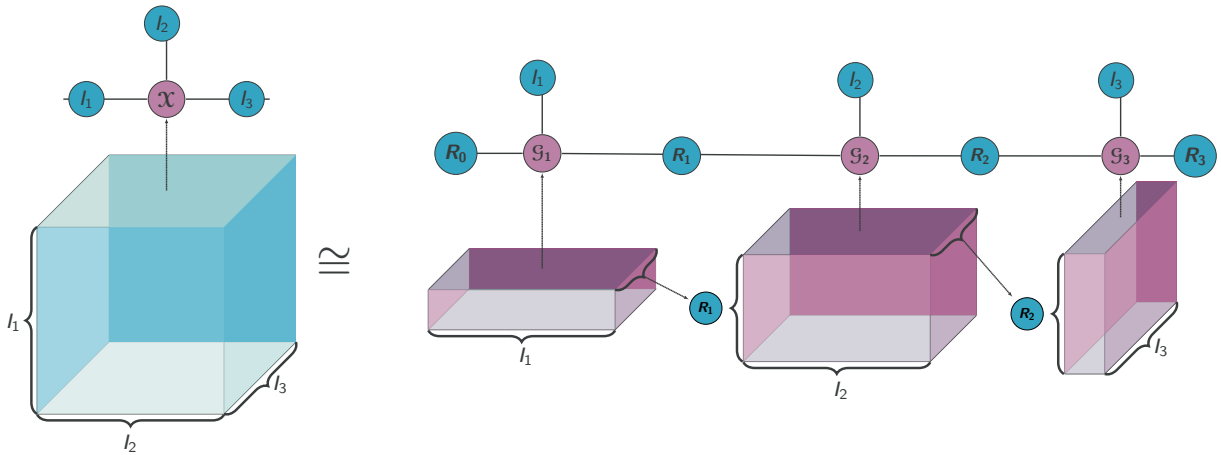


Figure: TT decomposition of a 3-way tensor.

Approximation of Kernel Computation

- Uses Canonical Polyadic (CP) decomposition [F.L. Hitchcock, 1927]

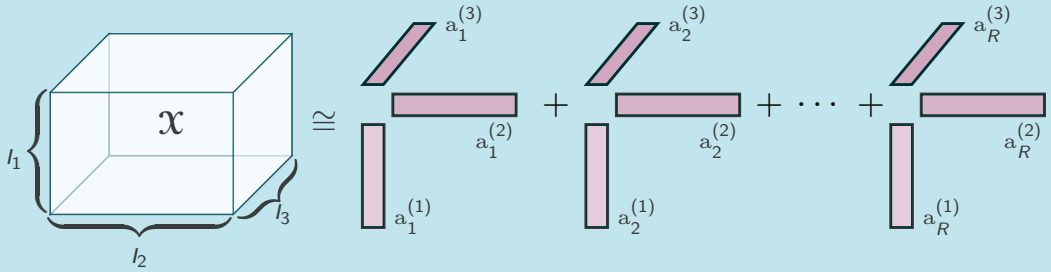


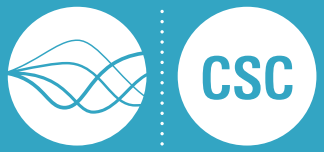
Figure: CP decomposition of a 3-way tensor.

- Dual Structure-preserving Kernel [L. He; X. Kong; P.S. Yu; A.B. Ragin; Z. Hao; X. Yang, 2014]

(Feature map) $\Psi : \mathbf{X} \times \mathbf{Y} \times \mathbf{Z} \mapsto \mathbb{R}^{\mathcal{H}_1 \times \mathcal{H}_2 \times \mathcal{H}_3}$

$$\Psi : \sum_{r=1}^{R} a_r^{(1)} \otimes a_r^{(2)} \otimes a_r^{(3)} \mapsto \sum_{r=1}^{R} \phi(a_r^{(1)}) \otimes \phi(a_r^{(2)}) \otimes \phi(a_r^{(3)})$$

$$\langle \Psi(\mathcal{X}), \Psi(\mathcal{Y}) \rangle = k(\mathcal{X}, \mathcal{Y}) = \sum_{i,j=1}^{R} k(a_i^{(1)}, b_j^{(1)}) k(a_i^{(2)}, b_j^{(2)}) k(a_i^{(3)}, b_j^{(3)})$$



Building Efficient Model

Low-rank Method

- Using TT decomposition
  - Stable algorithm
  - Depends on matrix SVD
  - Leads to over-fitting
  - Blocks might not be unique
- Using CP decomposition
  - Simplicity
  - Finding best rank: NP-hard [J. Håstad, 1989]

TT-CP decomposition: [K. Kour; S. Dolgov; M. Stoll; P. Benner, 2020]

Simplicity: Exact TT-CP Expansion

Given TT decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$

$$x_{i_1 i_2 \ldots i_M} \cong \sum_{r_1, \ldots, r_{M-1}}^{R_1, \ldots, R_{M-1}} \mathcal{G}^{(1)}_{i_1, r_1} \mathcal{G}^{(2)}_{r_1, i_2, r_2} \cdots \mathcal{G}^{(M)}_{r_{M-1}, i_M},$$

CP expansion can be written,

$$x_{i_1 i_2 \ldots i_M} \approx \sum_{r=1}^{R} H^{(1)}_{i_1, r} H^{(2)}_{i_2, r} \cdots H^{(M)}_{i_M, r},$$

The rank r can be merged $r = r_1 + (r_2 - 1)R_1 + \ldots + (r_m - 1)\prod_{\ell=1}^{m-1} R_\ell$,
where the CP factors are defined as

$$H^{(m)}_{i_m, r} = \mathcal{G}^{(m)}_{i_m, r}, \quad m = 1, \ldots, M$$

Algorithm 1: Uniqueness Enforcing TT-SVD

Input: M-dimensional tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$, prescribed accuracy ϵ.
Output: Return tensor $\mathcal{X}'$ in unique TT decomposition with cores $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(M)}$.
The computed approximation satisfies $\left\|\mathcal{X} - \mathcal{X}'\right\|_F \leq \epsilon \left\|\mathcal{X}\right\|_F$.
Temporary tensor: $\mathcal{Z} = \mathcal{X}$, $r_0 = 1$.
for $m = 1$ to $M - 1$ **do**
  $\mathbf{Z} :=$ reshape$(\mathcal{Z}, [R_{m-1} I_m, I_{m+1} \cdots I_M])$
  Compute δ-truncated SVD: $\mathbf{Z} = \mathbf{U}\mathbf{S}\mathbf{V}^T + E$, $\|E\|_F \leq \delta$.
  $\mathbf{U} = [u_1, u_2, \ldots, u_{R_m}], \mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{R_m}), \mathbf{V} = [v_1, v_2, \ldots, v_{R_m}]$
  $\mathbf{Z} = \sigma_1 u_1 v_1^\top + \cdots + \sigma_{I_m} u_{I_m} v_{I_m}^\top$
  for $r_m = 1$ to R_m **do**
    $i^*_{r_m} = \arg\max_{i=1,\ldots,I_m} |u_{r_m}(i)|$,
    $\bar{u}_{r_m} := u_{r_m} / \text{sign}(u_{r_m}(i^*_{r_m}))$, $\bar{v}_{r_m} := v_{r_m} \cdot \text{sign}(u_{r_m}(i^*_{r_m}))$.
    $\mathcal{G}^{(m)}_{r_{m-1}, i_m, r_m} = \bar{u}_{r_m}(i_m)$
  end for
  New core: $\mathcal{G}^{(m)} :=$ reshape$(\mathbf{U}, [R_{m-1}, I_m, R_m])$
  $\mathbf{Z} := \mathbf{S}\mathbf{V}^T$
end for
$\mathcal{G}^{(m)} = \mathbf{Z}$

Stability: Uniqueness Enforcing TT-SVD

- This ensures that "close" tensor produce "close" TT-cores
- Makes model stable w.r.t. different rank truncation

State-of-the-art: Norm Equilibration

Given a rank-r TT-CP expansion $[\![H^{(1)}, H^{(2)}, \cdots, H^{(M)}]\!]$, we compute the total norm of each of the rank-1 tensors

$$\|\mathcal{N}_r\| = \left\|H_r^{(1)}\right\| * \left\|H_r^{(2)}\right\| * \cdots * \left\|H_r^{(M)}\right\|,$$

and distribute this norm equally among the factors,

$$H_r^{(k)} := \frac{H_r^{(k)}}{\left\|H_r^{(k)}\right\|} * \|\mathcal{N}_r\|^{1/M}, \qquad k = 1, 2, \cdots, M.$$

TT Multi-way Multi-level Kernel (TT-MMK)

[K. Kour; S. Dolgov; M. Stoll; P. Benner, 2020]

Algorithm 2: TT-MMK: STM Kernel Approximation via TT-CP

Input: data $\{\mathcal{X}_n\}_{n=1}^{N} \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$, TT-rank R.
Output: Kernel Approximation $k \in \mathbb{R}^{N \times N}$
for $n = 1$ **to** N **do**
  Compute TT decomposition $\mathcal{X}_n \cong \langle\!\langle \mathcal{G}^{(1,n)}, \mathcal{G}^{(2,n)}, \ldots, \mathcal{G}^{(M,n)} \rangle\!\rangle$,
  Compute TT-CP decomposition
  $[\![H^{(1,n)}, H^{(2,n)}, \ldots, H^{(M,n)}]\!] = \langle\!\langle \mathcal{G}^{(1,n)}, \mathcal{G}^{(2,n)}, \ldots, \mathcal{G}^{(M,n)} \rangle\!\rangle$,
end for
for $u, v = 1$ **to** N **do**
  $k\left(\mathcal{X}_u, \mathcal{X}_v\right) \approx \sum_{i,j=1}^{R} k(h_i^{(1,u)}, h_j^{(1,v)}) k(h_i^{(2,u)}, h_j^{(2,v)}) \cdots k(h_i^{(M,u)}, h_j^{(M,v)})$,
end for

Datasets:

1. Resting-state fMRI Brain Images
   - Alzheimer Disease (ADNI[1]) - 33 data points
   - Attention Deficit Hyperactivity Disorder (ADHD[2]) - 200 data points

[1] http://adni.loni.usc.edu/
[2] http://neurobureau.projects.nitrc.org/ADHD200/Data.html
[3] https://aviris.jpl.nasa.gov/

Numerical Experiment
Datasets

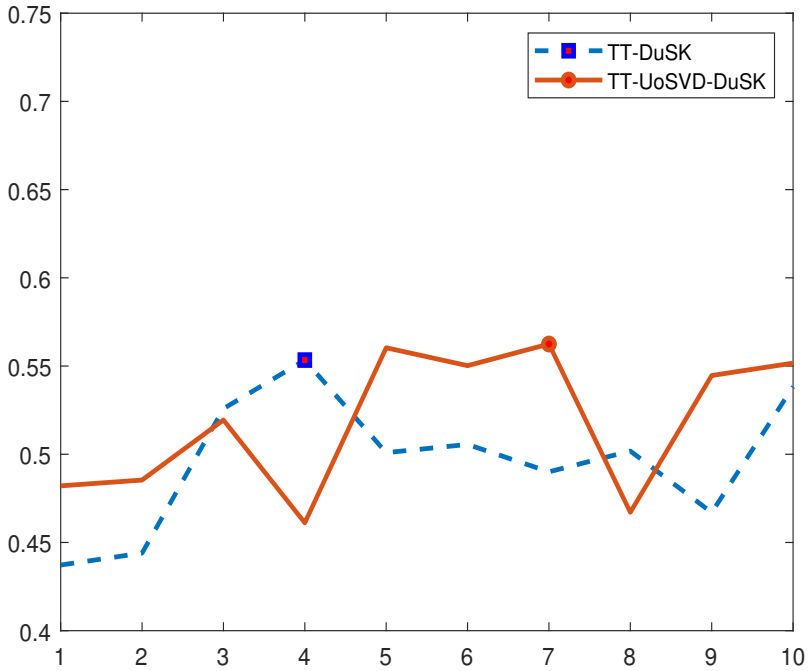
Datasets:

1. Resting-state fMRI Brain Images
   - Alzheimer Disease (ADNI[1]) - 33 data points
   - Attention Deficit Hyperactivity Disorder (ADHD[2]) - 200 data points

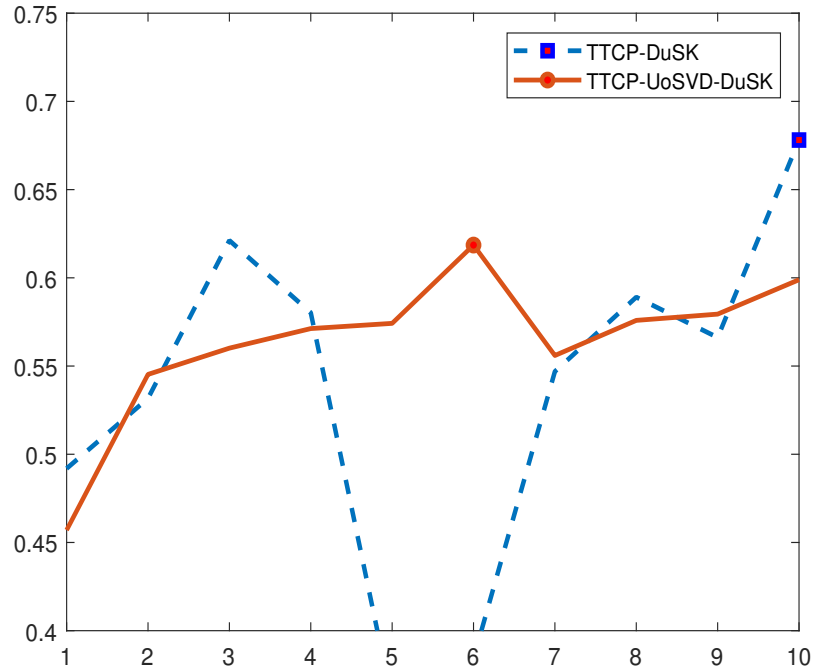
2. Hyperspectral Images
   - Indian Pines: The HSI images was collected via Aviris Sensor[3] over Indian Pines test site - 100 data points
   - Salinas: This HSI images data was collected by 224 band Aviris Sensor over Salinas velly, California - 100 data points

[1] http://adni.loni.usc.edu/
[2] http://neurobureau.projects.nitrc.org/ADHD200/Data.html
[3] https://aviris.jpl.nasa.gov/

Numerical Experiment
Results I

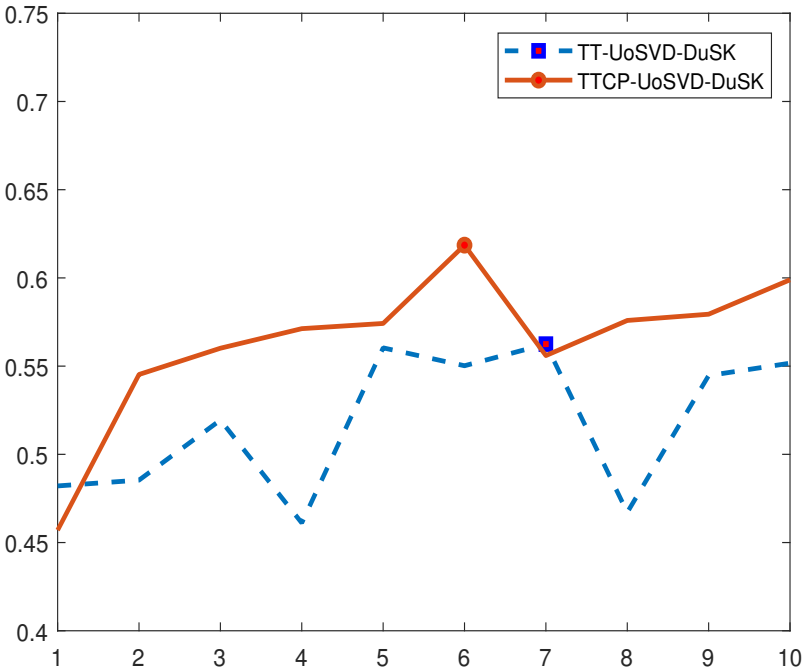


(a) TT with and without enforced uniqueness

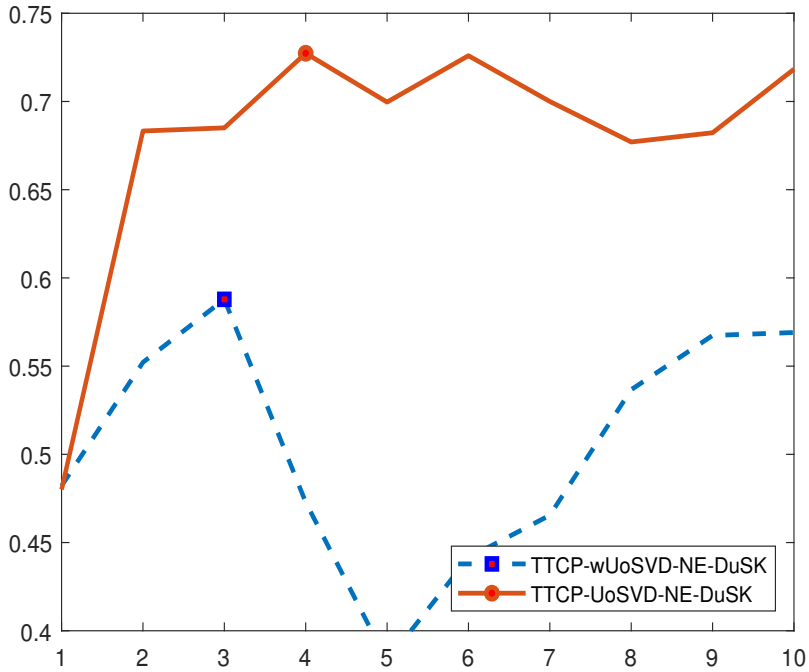


(b) TTCP with and without enforced uniqueness

Figure: Effect of each step of Algorithm for ADNI data set(Rank vs Accuracy).



(a) TT vs TTCP with enforced uniqueness



(b) TTCP norm equilibrium with and without eforced uniqueness

Figure: Effect of each step of Algorithm for ADNI data set (Rank vs Accuracy).

Numerical Experiment
Results II

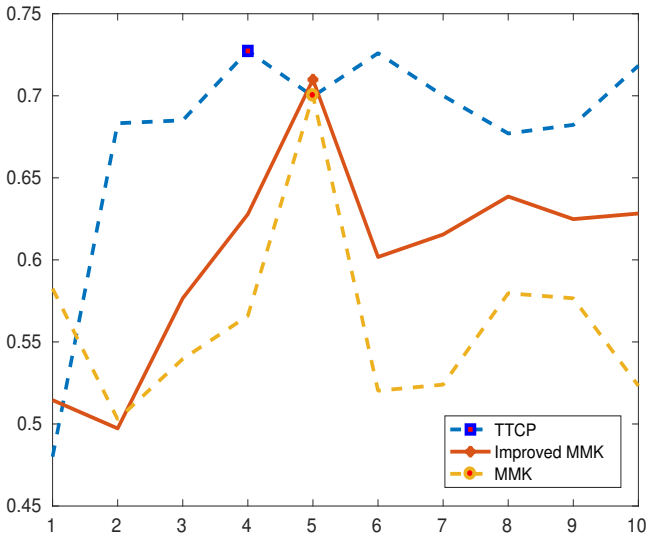
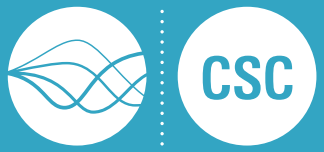


Figure: Accuracy for ADNI dataset w.r.t. truncated TT-CP rank

Table: Best average classification accuracy for different methods and datasets

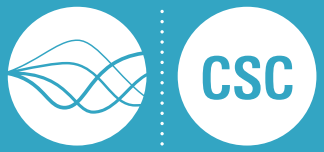
Methods	ADNI	ADHD	I. Pines	Salinas
SVM	49	52	46	47
STuM	51	54	57	74
DuSK	55	58	60	92
MMK	69	60	93	98
TT-MMK	**73**	**64**	**99**	**99**



Conclusions

Summary

- Low rank-method “TT-CP decomposition”
- Binary classification kernel model for less data with tensor input
- Approximating kernel not only reduced cost, also increases accuracy
- Stability and reduced computational cost with higher accuracy



Conclusions

Summary

- Low rank-method "TT-CP decomposition"
- Binary classification kernel model for less data with tensor input
- Approximating kernel not only reduced cost, also increases accuracy
- Stability and reduced computational cost with higher accuracy

https://arxiv.org/abs/2002.05079

Thankful to SIAM Student Travel Award