High-dimensional Regression and Dictionary Learning: Some Recent Advances for Tensor Data

Waheed U. Bajwa

Department of Electrical and Computer Engineering
Rutgers University–New Brunswick, NJ USA
www.inspirelab.us

One World Signal Processing Seminar
October 29, 2020

CCF-1453073
CCF-1910110

W911NF-17-1-0546

Students and Collaborators

Dr. Talal Ahmed

Dr. Zahra Shakeri

Dr. Haroon Raja

Mohsen Ghassemi

Prof. Anand Sarwate

Outline

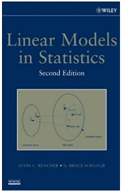
1. Motivation: High-dimensional Data and Its Implications
2. High-dimensional Tensor Regression
3. Dictionary Learning for High-dimensional Tensor Data
4. Summary

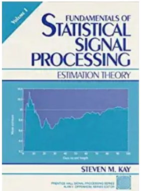
Outline

Classical data-driven inference problems

Data in classical signal processing, machine learning, and statistics problems tended to be *extrinsically* low-dimensional

- Number of data samples exceeds the number of features in each sample
- Examples: Social sciences, medical sciences, paleontology, etc., in yesteryears





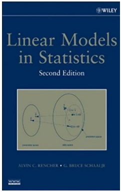
Classical data-driven inference problems

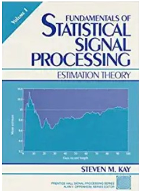
Data in classical signal processing, machine learning, and statistics problems tended to be *extrinsically* low-dimensional

- Number of data samples exceeds the number of features in each sample
- Examples: Social sciences, medical sciences, paleontology, etc., in yesteryears

Classical linear regression: Regress a response variable y over p covariates (predictors) using $n \geq p$ observations (data samples)

- Mathematically, recover regression parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ from n observations $\mathbf{y} \in \mathbb{R}^n$ modeled as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}$ for the case of $n \geq p$ observations





Classical data-driven inference problems

Data in classical signal processing, machine learning, and statistics problems tended to be *extrinsically* low-dimensional

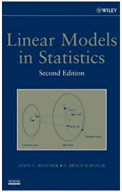
- Number of data samples exceeds the number of features in each sample
- Examples: Social sciences, medical sciences, paleontology, etc., in yesteryears

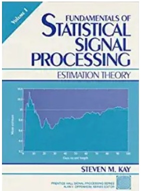
Classical linear regression: Regress a response variable y over p covariates (predictors) using $n \geq p$ observations (data samples)

- Mathematically, recover regression parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ from n observations $\mathbf{y} \in \mathbb{R}^n$ modeled as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}$ for the case of $n \geq p$ observations

Advantages of 'low-dimensional' data settings

- There is less fear of overfitting
- Memory requirements can be low
- Computations can be easier



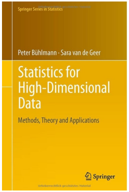


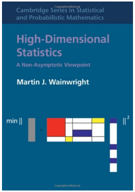
Modern-day inference problems

Confluence of cheap sensors, abundant storage, and digitization of the world has led a shift to 'high-dimensional' inference problems

High-dimensional data setting: Data dimension (number of features, independent variables, predictors, etc.) far exceeds number of samples (observations)

- Examples: Social sciences, medical sciences, paleontology, etc.





Modern-day inference problems

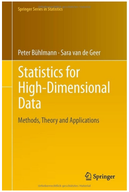
Confluence of cheap sensors, abundant storage, and digitization of the world has led a shift to 'high-dimensional' inference problems

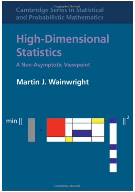
High-dimensional data setting: Data dimension (number of features, independent variables, predictors, etc.) far exceeds number of samples (observations)

- Examples: Social sciences, medical sciences, paleontology, etc.

Challenges of high-dimensional data settings

- Overfitting is a real concern
  - More unknowns than the number of observations
- Potentially large computational and memory overhead





Modern-day inference problems

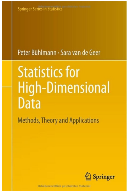
Confluence of cheap sensors, abundant storage, and digitization of the world has led a shift to 'high-dimensional' inference problems

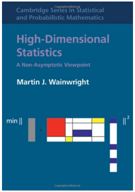
High-dimensional data setting: Data dimension (number of features, independent variables, predictors, etc.) far exceeds number of samples (observations)

- Examples: Social sciences, medical sciences, paleontology, etc.

Challenges of high-dimensional data settings

- Overfitting is a real concern
  - More unknowns than the number of observations
- Potentially large computational and memory overhead



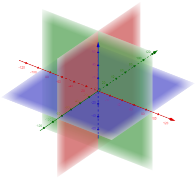


Solution: Exploit *intrinsic* low-dimensional geometry of high-dimensional data through the use of an appropriate regularizer

Popular regularizers for high-dimensional problems

Sparsity-based regularizers

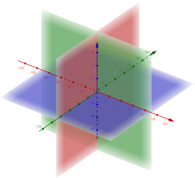
- Sparse regression
- Sparse approximation
- Compressed sensing
- Dictionary learning



Popular regularizers for high-dimensional problems

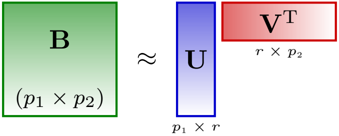
Sparsity-based regularizers

- Sparse regression
- Sparse approximation
- Compressed sensing
- Dictionary learning



Low-rankness based regularizers

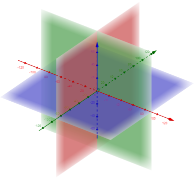
- Matrix regression
- Matrix completion
- Background subtraction
- Principal component analysis



Popular regularizers for high-dimensional problems

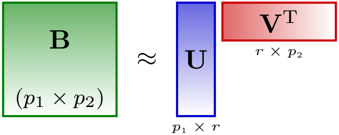
Sparsity-based regularizers

- Sparse regression
- Sparse approximation
- Compressed sensing
- Dictionary learning



Low-rankness based regularizers

- Matrix regression
- Matrix completion
- Background subtraction
- Principal component analysis

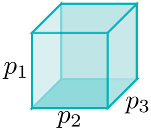


Sample Complexity ➠ n can be on the order of intrinsic dimensionality

- Sparse regression (sparsity $s \ll p$): $n = O(s \log(p))$ for p-dimensional data
- Matrix regression (rank $r \ll \min(p_1, p_2)$): $n = O((p_1 + p_2) r \log(\cdot))$ for $p := p_1 p_2$-dimensional data

Tensor data and the 'old' regularizers

Many of today's problems give rise to multidimensional data samples, also referred to as multiway data or tensor data

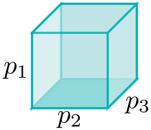


3rd-order tensor

- Examples: Colored / depth / multispectral images, grayscale / colored videos, MIMO channels, lidar data, (f)MRI data, etc.

Tensor data and the 'old' regularizers

Many of today's problems give rise to multidimensional data samples, also referred to as multiway data or tensor data



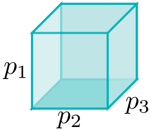
3rd-order tensor

- Examples: Colored / depth / multispectral images, grayscale / colored videos, MIMO channels, lidar data, (f)MRI data, etc.

Tensor data can be massively high-dimensional, rendering the old (tensor-agnostic) regularizers highly suboptimal

Tensor data and the 'old' regularizers

Many of today's problems give rise to multidimensional data samples, also referred to as multiway data or tensor data

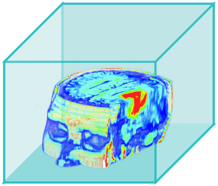


3rd-order tensor

- Examples: Colored / depth / multispectral images, grayscale / colored videos, MIMO channels, lidar data, (f)MRI data, etc.

Tensor data can be massively high-dimensional, rendering the old (tensor-agnostic) regularizers highly suboptimal

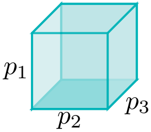
Tensor $\underline{\mathbf{B}} \in \mathbb{R}^{256 \times 256 \times 64}$



Tensor data and the ‘old’ regularizers

Many of today’s problems give rise to multidimensional data samples, also referred to as multiway data or tensor data

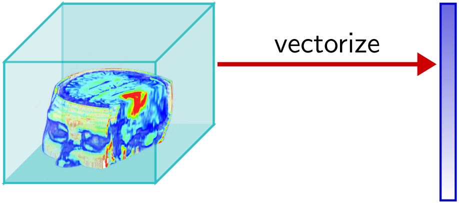
- Examples: Colored / depth / multispectral images, grayscale / colored videos, MIMO channels, lidar data, (f)MRI data, etc.



3rd-order tensor

Tensor data can be massively high-dimensional, rendering the old (tensor-agnostic) regularizers highly suboptimal

Tensor $\underline{\mathbf{B}} \in \mathbb{R}^{256 \times 256 \times 64}$



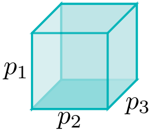
Vector dimensions: $p = 4,194,304$

10% sparsity $\Rightarrow n \geq 419,430$

Tensor data and the ‘old’ regularizers

Many of today’s problems give rise to multidimensional data samples, also referred to as multiway data or tensor data

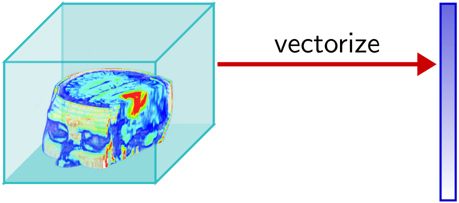
- Examples: Colored / depth / multispectral images, grayscale / colored videos, MIMO channels, lidar data, (f)MRI data, etc.



3rd-order tensor

Tensor data can be massively high-dimensional, rendering the old (tensor-agnostic) regularizers highly suboptimal

Tensor $\underline{\mathbf{B}} \in \mathbb{R}^{256 \times 256 \times 64}$



Vector dimensions: $p = 4,194,304$

10% sparsity $\Rightarrow n \geq 419,430$

High-dimensional inference from tensor data necessitates newer regularizers

High-dimensional inference from tensor data

Goal: Use regularizers that exploit tensor geometry based on tensor decompositions [KoldaBader'09]

Review-style references summarizing related works

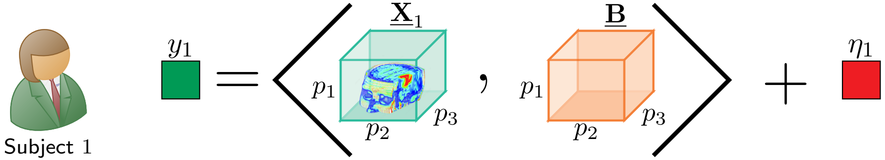
- [Cichocki et al.'09], [Sidiropoulos et al.'17], [Rabanser et al.'17], [Fu et al.'20]

This talk

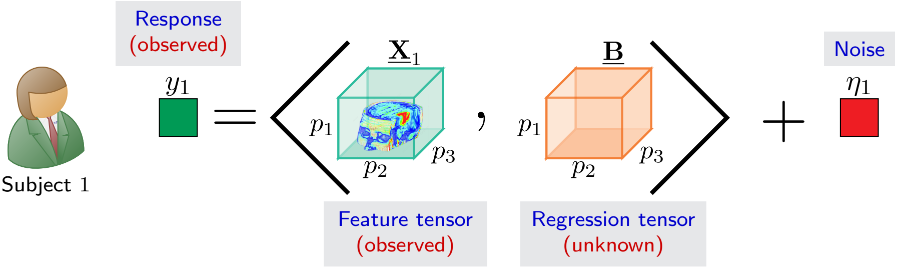
- High-dimensional tensor regression
  - Ahmed, Raja, **B.**, "Tensor regression using low-rank and sparse Tucker decompositions," SIAM J. Math. Data Science, 2020 (in press)
- High-dimensional tensor dictionary learning
  - Ghassemi, Shakeri, Sarwate, **B.**, "Learning mixtures of separable dictionaries for tensor data: Analysis and algorithms," IEEE Trans. Signal Processing, 2020
  - Shakeri, Sarwate, **B.**, "Identifiability of Kronecker-structured dictionaries for tensor data," IEEE J. Sel. Topics Signal Processing, 2018
  - Shakeri, **B.**, Sarwate, "Minimax lower bounds on dictionary learning for tensor data," IEEE Trans. Inform. Theory, 2018

Outline

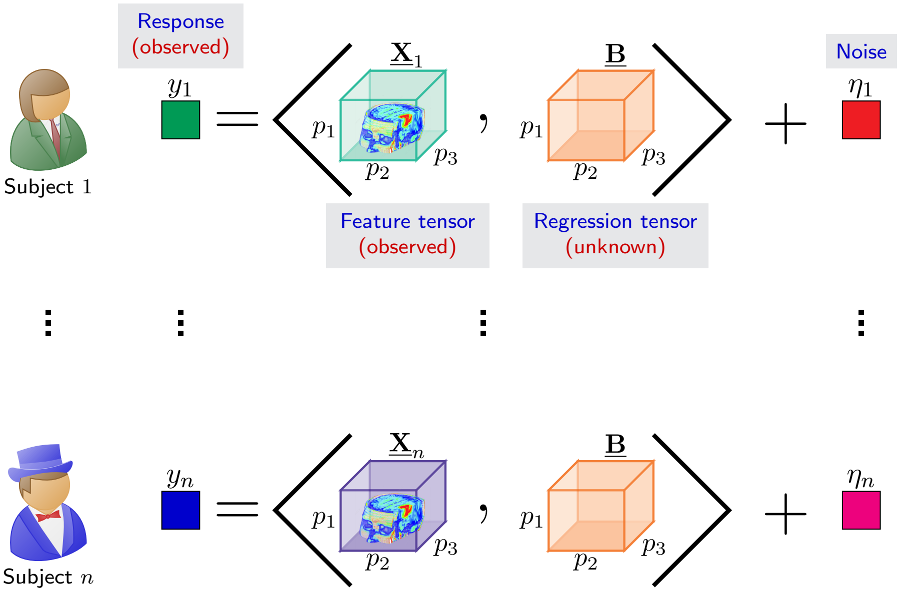
Tensor regression model for 3rd-order tensors



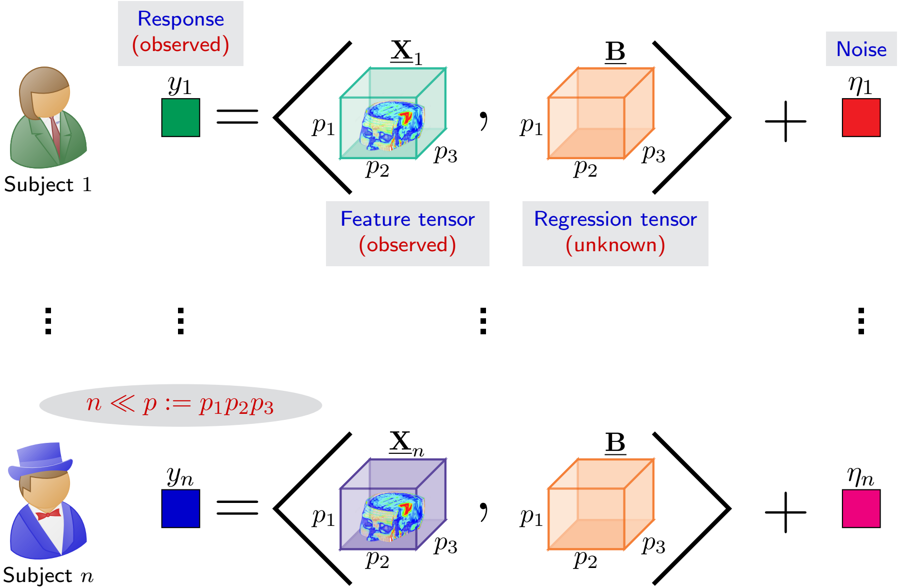
Tensor regression model for 3rd-order tensors



Tensor regression model for 3rd-order tensors



Tensor regression model for 3rd-order tensors



Mathematical model for general tensor regression

Observations: $y_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle + \eta_i, \ i = 1, \ldots, n$

- Tensor of predictors: $\underline{\mathbf{X}}_i \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
- Tensor of regression parameters: $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
  - Number of *extrinsic* degrees of freedom: $p := \Pi_{k=1}^{K} p_k$
- Scalar-valued response variable: $y_i \in \mathbb{R}$
- Modeling error / additive noise: $\eta_i \in \mathbb{R}$

Goal: Obtain an estimate of $\underline{\mathbf{B}}$ using data $\{(\underline{\mathbf{X}}_i, y_i)\}_{i=1}^{n}$

Challenge: Ill-posed $(n \ll p)$ for even modest values of $p_1, \ldots, p_K$

Mathematical model for general tensor regression

Observations: $y_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle + \eta_i,\ i = 1, \ldots, n$

- Tensor of predictors: $\underline{\mathbf{X}}_i \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
- Tensor of regression parameters: $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
  - Number of *extrinsic* degrees of freedom: $p := \Pi_{k=1}^{K} p_k$
- Scalar-valued response variable: $y_i \in \mathbb{R}$
- Modeling error / additive noise: $\eta_i \in \mathbb{R}$

Goal: Obtain an estimate of $\underline{\mathbf{B}}$ using data $\{(\underline{\mathbf{X}}_i, y_i)\}_{i=1}^{n}$

Challenge: Ill-posed $(n \ll p)$ for even modest values of $p_1, \ldots, p_K$

Impose additional structure on $\underline{\mathbf{B}}$ to reduce its *intrinsic* degrees of freedom

Structured tensor as a regularizer

Related prior works

- [Gandy et al.'11], [Tomioka et al.'11], [Liu et al.'12], [Mu et al.'14], [YuLiu'16], [Rauhut et al.'17], [He et al.'18], [Chen et al.'19], [Raskutti et al.'19], ...

Structured tensor as a regularizer

Related prior works

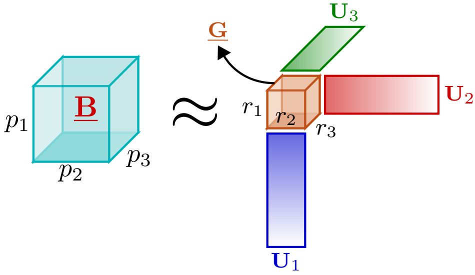
- [Gandy et al.'11], [Tomioka et al.'11], [Liu et al.'12], [Mu et al.'14], [YuLiu'16], [Rauhut et al.'17], [He et al.'18], [Chen et al.'19], [Raskutti et al.'19], ...

Typical imposed structure: $\underline{\mathbf{B}}$ admits a *low-rank* Tucker decomposition

Related prior works

- [Gandy et al.'11], [Tomioka et al.'11], [Liu et al.'12], [Mu et al.'14], [YuLiu'16], [Rauhut et al.'17], [He et al.'18], [Chen et al.'19], [Raskutti et al.'19], ...

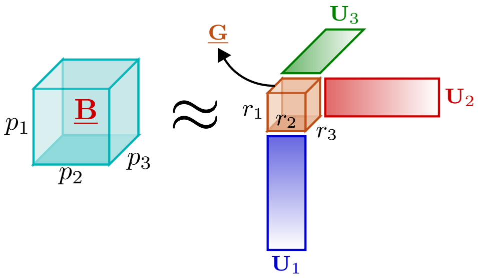
Typical imposed structure: $\underline{\mathbf{B}}$ admits a *low-rank* Tucker decomposition



Related prior works

- [Gandy et al.'11], [Tomioka et al.'11], [Liu et al.'12], [Mu et al.'14], [YuLiu'16], [Rauhut et al.'17], [He et al.'18], [Chen et al.'19], [Raskutti et al.'19], ...

Typical imposed structure: $\underline{\mathbf{B}}$ admits a *low-rank* Tucker decomposition



$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: Mode-1 factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: Mode-2 factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: Mode-3 factor matrix $(p_3 \times r_3)$

Low rank: $r_1 \ll p_1$, $r_2 \ll p_2$, $r_3 \ll p_3$

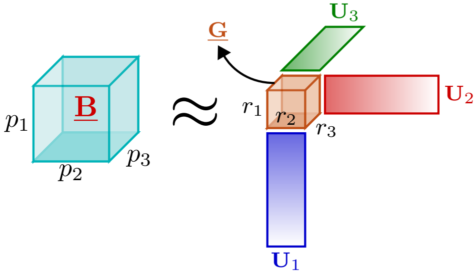
Mathematically: $\underline{\mathbf{B}} \approx \sum_{i,j,k} \underline{g}_{i,j,k} \mathbf{u}_{1,i} \circ \mathbf{u}_{2,j} \circ \mathbf{u}_{3,k}$

Structured tensor as a regularizer

Related prior works

- [Gandy et al.'11], [Tomioka et al.'11], [Liu et al.'12], [Mu et al.'14], [YuLiu'16], [Rauhut et al.'17], [He et al.'18], [Chen et al.'19], [Raskutti et al.'19], ...

Typical imposed structure: $\underline{\mathbf{B}}$ admits a *low-rank* Tucker decomposition

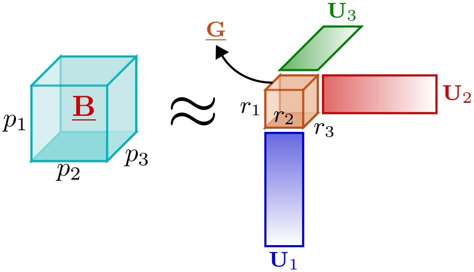


$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: Mode-1 factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: Mode-2 factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: Mode-3 factor matrix $(p_3 \times r_3)$

Low rank: $r_1 \ll p_1,\ r_2 \ll p_2,\ r_3 \ll p_3$

Mathematically: $\underline{\mathbf{B}} \approx \sum_{i,j,k} \underline{g}_{i,j,k} \mathbf{u}_{1,i} \circ \mathbf{u}_{2,j} \circ \mathbf{u}_{3,k} = \boxed{\underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3}$

Tensor regression and the low-rank Tucker model



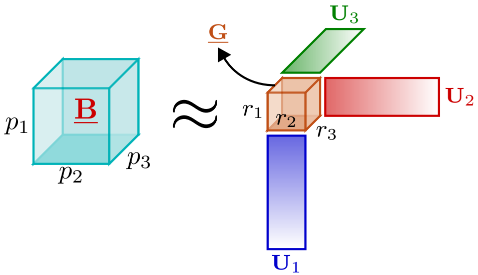
$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: Mode-1 factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: Mode-2 factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: Mode-3 factor matrix $(p_3 \times r_3)$

Low rank: $r_1 \ll p_1$, $r_2 \ll p_2$, $r_3 \ll p_3$

Sample complexity under the low-rank Tucker model [Rauhut et al.'17]

$$n = O\left(\left(p_{\max} r_{\max} K + r_{\max}^K\right) \log(K)\right)$$



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

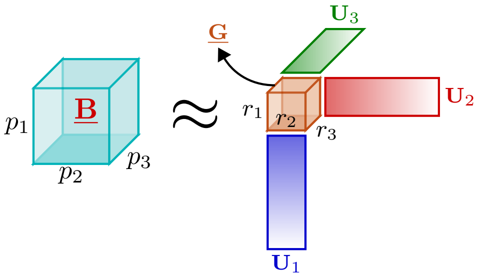
$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: Mode-1 factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: Mode-2 factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: Mode-3 factor matrix $(p_3 \times r_3)$

Low rank: $r_1 \ll p_1$, $r_2 \ll p_2$, $r_3 \ll p_3$

Sample complexity under the low-rank Tucker model [Rauhut et al.'17]

$$n = O\left(\left(p_{\max} r_{\max} K + r_{\max}^K\right) \log(K)\right)$$

- The sample complexity can still be infeasible for large values of $p_{\max}$



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: Mode-1 factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: Mode-2 factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: Mode-3 factor matrix $(p_3 \times r_3)$

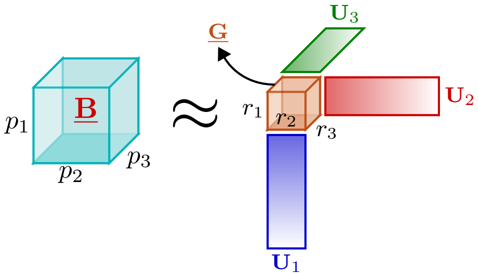
Low rank: $r_1 \ll p_1$, $r_2 \ll p_2$, $r_3 \ll p_3$

Sample complexity under the low-rank Tucker model [Rauhut et al.'17]

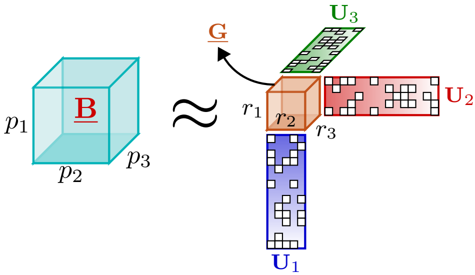
$$n = O\left(\left(p_{\max} r_{\max} K + r_{\max}^K\right) \log(K)\right)$$

- The sample complexity can still be infeasible for large values of $p_{\max}$
- Identification of a parsimonious set of significant predictors remains a challenge

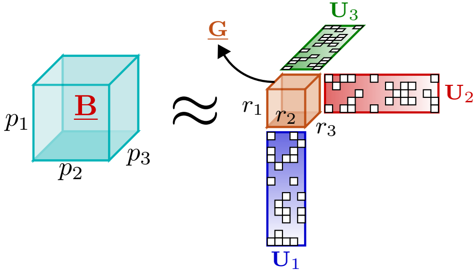
Low-rank and sparse Tucker decomposition



Low-rank and sparse Tucker decomposition



Low-rank and sparse Tucker decomposition

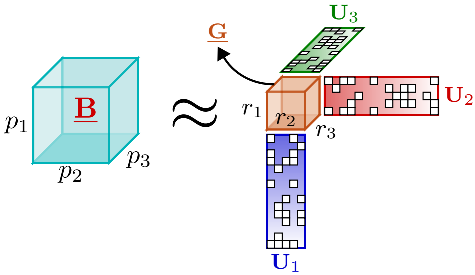


$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Low-rank and sparse Tucker decomposition



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

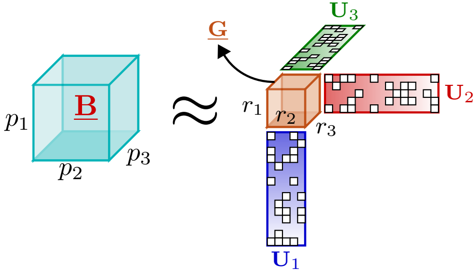
$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Definition ($(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition)

A K-th order tensor $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$ admits an $(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition if $\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_K \mathbf{U}_K$ and

- $\dim(\underline{\mathbf{G}}) = \mathbf{r} := (r_1, \ldots, r_K) \ll (p_1, \ldots, p_K)$
- $(\|\mathbf{U}_1\|_{0,\infty}, \ldots, \|\mathbf{U}_K\|_{0,\infty}) \leq \mathbf{s} := (s_1, \ldots, s_K) \ll (p_1, \ldots, p_K)$



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

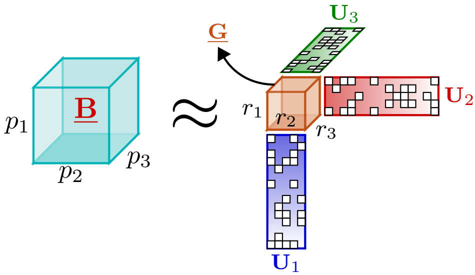
Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Definition ($(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition)

A K-th order tensor $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$ admits an $(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition if $\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_K \mathbf{U}_K$ and

- $\dim(\underline{\mathbf{G}}) = \mathbf{r} := (r_1, \ldots, r_K) \ll (p_1, \ldots, p_K)$
- $(\|\mathbf{U}_1\|_{0,\infty}, \ldots, \|\mathbf{U}_K\|_{0,\infty}) \leq \mathbf{s} := (s_1, \ldots, s_K) \ll (p_1, \ldots, p_K)$

Low-rank and sparse Tucker decomposition



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

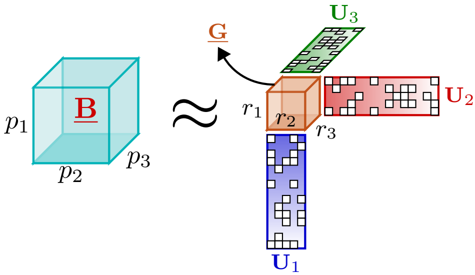
$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Definition ($(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition)

A K-th order tensor $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$ admits an $(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition if $\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_K \mathbf{U}_K$ and

- $\dim(\underline{\mathbf{G}}) = \mathbf{r} := (r_1, \ldots, r_K) \ll (p_1, \ldots, p_K)$
- $(\|\mathbf{U}_1\|_{0,\infty}, \ldots, \|\mathbf{U}_K\|_{0,\infty}) \leq \mathbf{s} := (s_1, \ldots, s_K) \ll (p_1, \ldots, p_K)$



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

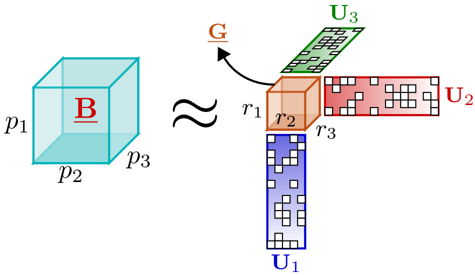
Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Definition ($(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition)

A K-th order tensor $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$ admits an $(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition if $\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_K \mathbf{U}_K$ and

- $\dim(\underline{\mathbf{G}}) = \mathbf{r} := (r_1, \ldots, r_K) \ll (p_1, \ldots, p_K)$
- $(\|\mathbf{U}_1\|_{0,\infty}, \ldots, \|\mathbf{U}_K\|_{0,\infty}) \leq \mathbf{s} := (s_1, \ldots, s_K) \ll (p_1, \ldots, p_K)$

Low-rank and sparse Tucker decomposition



$\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\underline{\mathbf{G}}$: Core tensor $(r_1 \times r_2 \times r_3)$
$\mathbf{U}_1$: s_1-sparse factor matrix $(p_1 \times r_1)$
$\mathbf{U}_2$: s_2-sparse factor matrix $(p_2 \times r_2)$
$\mathbf{U}_3$: s_3-sparse factor matrix $(p_3 \times r_3)$

Low rank: $\mathbf{r} := (r_1, r_2, r_3) \ll (p_1, p_2, p_3)$
Sparsity: $\mathbf{s} := (s_1, s_2, s_3) \ll (p_1, p_2, p_3)$

Definition ($(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition)

A K-th order tensor $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$ admits an $(\mathbf{r}, \mathbf{s})$–low-rank and sparse Tucker decomposition if $\underline{\mathbf{B}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_K \mathbf{U}_K$ and

- $\dim(\underline{\mathbf{G}}) = \mathbf{r} := (r_1, \ldots, r_K) \ll (p_1, \ldots, p_K)$
- $(\|\mathbf{U}_1\|_{0,\infty}, \ldots, \|\mathbf{U}_K\|_{0,\infty}) \leq \mathbf{s} := (s_1, \ldots, s_K) \ll (p_1, \ldots, p_K)$

Why? Reduces the number of degrees of freedom and can impart sparsity on $\underline{\mathbf{B}}$

Model for low-rank and sparse tensor regression

Observations: $y_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle + \eta_i,\ i = 1, \ldots, n$

- Tensor of predictors: $\underline{\mathbf{X}}_i \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
- Tensor of regression parameters: $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
  - Tensor $\underline{\mathbf{B}}$ is $(\mathbf{r}, \mathbf{s})$–Tucker decomposable
- Scalar-valued response variable: $y_i \in \mathbb{R}$
- Modeling error / additive noise: $\eta_i \in \mathbb{R}$

Compact Notation

- $\mathbf{y} = \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}) + \boldsymbol{\eta}$, with $\boldsymbol{\mathcal{X}} : \mathbb{R}^{p_1 \times \cdots \times p_K} \to \mathbb{R}^n$ s.t. $[\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}})]_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle$

Model for low-rank and sparse tensor regression

Observations: $y_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle + \eta_i, \ i = 1, \ldots, n$

- Tensor of predictors: $\underline{\mathbf{X}}_i \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
- Tensor of regression parameters: $\underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K}$
  - Tensor $\underline{\mathbf{B}}$ is $(\mathbf{r}, \mathbf{s})$–Tucker decomposable
- Scalar-valued response variable: $y_i \in \mathbb{R}$
- Modeling error / additive noise: $\eta_i \in \mathbb{R}$

Compact Notation

- $\mathbf{y} = \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}) + \boldsymbol{\eta}$, with $\boldsymbol{\mathcal{X}} : \mathbb{R}^{p_1 \times \cdots \times p_K} \to \mathbb{R}^n$ s.t. $[\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}})]_i = \langle \underline{\mathbf{X}}_i, \underline{\mathbf{B}} \rangle$

Goals

- A provably convergent algorithm for estimating $\underline{\mathbf{B}}$ using data $\{(\underline{\mathbf{X}}_i, y_i)\}_{i=1}^n$
- A characterization of sample complexity of the developed algorithm

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r},\mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Optimization formulation: $\widehat{\underline{\mathbf{B}}} = \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2$

TPGD Algorithm

1: **Initialize:** Tensor $\underline{\mathbf{B}}^{(0)}$ and $t \leftarrow 0$

2: **while** Stopping criterion **do**

3: $\quad \widetilde{\underline{\mathbf{B}}}^{(t)} \leftarrow \underline{\mathbf{B}}^{(t)} - \mu \boldsymbol{\mathcal{X}}^*(\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(t)}) - \mathbf{y})$

4: $\quad \underline{\mathbf{B}}^{(t+1)} \leftarrow \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \|\widetilde{\underline{\mathbf{B}}}^{(t)} - \underline{\mathbf{Z}}\|_F^2$

5: $\quad t \leftarrow t + 1$

6: **end while**

7: **return** Tensor $\widehat{\underline{\mathbf{B}}} = \underline{\mathbf{B}}^{(t)}$

Algorithm: Tensor Projected Gradient Descent (TPGD)

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r},\mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Optimization formulation: $\widehat{\underline{\mathbf{B}}} = \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2$

TPGD Algorithm

1: **Initialize:** Tensor $\underline{\mathbf{B}}^{(0)}$ and $t \leftarrow 0$

2: **while** Stopping criterion **do**

3: $\quad \widetilde{\underline{\mathbf{B}}}^{(t)} \leftarrow \underline{\mathbf{B}}^{(t)} - \mu \boldsymbol{\mathcal{X}}^*(\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(t)}) - \mathbf{y})$ → Gradient descent step

4: $\quad \underline{\mathbf{B}}^{(t+1)} \leftarrow \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \|\widetilde{\underline{\mathbf{B}}}^{(t)} - \underline{\mathbf{Z}}\|_F^2$

5: $\quad t \leftarrow t + 1$

6: **end while**

7: **return** Tensor $\widehat{\underline{\mathbf{B}}} = \underline{\mathbf{B}}^{(t)}$

Algorithm: Tensor Projected Gradient Descent (TPGD)

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r},\mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Optimization formulation: $\widehat{\underline{\mathbf{B}}} = \arg\min_{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2$

TPGD Algorithm

Initialize: Tensor $\underline{\mathbf{B}}^{(0)}$ and $t \leftarrow 0$

while Stopping criterion **do**

$\widetilde{\underline{\mathbf{B}}}^{(t)} \leftarrow \underline{\mathbf{B}}^{(t)} - \mu \boldsymbol{\mathcal{X}}^*(\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(t)}) - \mathbf{y})$ → Gradient descent step

$\underline{\mathbf{B}}^{(t+1)} \leftarrow \arg\min_{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}} \|\widetilde{\underline{\mathbf{B}}}^{(t)} - \underline{\mathbf{Z}}\|_F^2$ → Projection step

$t \leftarrow t + 1$

end while

return Tensor $\widehat{\underline{\mathbf{B}}} = \underline{\mathbf{B}}^{(t)}$

Algorithm: Tensor Projected Gradient Descent (TPGD)

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r}, \mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Optimization formulation: $\widehat{\underline{\mathbf{B}}} = \arg\min_{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2$

TPGD Algorithm

1: **Initialize:** Tensor $\underline{\mathbf{B}}^{(0)}$ and $t \leftarrow 0$

2: **while** Stopping criterion **do**

3: $\widetilde{\underline{\mathbf{B}}}^{(t)} \leftarrow \underline{\mathbf{B}}^{(t)} - \mu \boldsymbol{\mathcal{X}}^*(\boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(t)}) - \mathbf{y})$ → Gradient descent step

4: $\underline{\mathbf{B}}^{(t+1)} \leftarrow \arg\min_{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}} \|\widetilde{\underline{\mathbf{B}}}^{(t)} - \underline{\mathbf{Z}}\|_F^2$ → Projection step

5: $t \leftarrow t + 1$

6: **end while**

7: **return** Tensor $\widehat{\underline{\mathbf{B}}} = \underline{\mathbf{B}}^{(t)}$

Exact tensor projection can be NP-hard, so we have to work with a “good” approximation

TPGD: Approximate Projection Step

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r}, \mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Projection step: $\widehat{\underline{\mathbf{W}}} = \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \|\underline{\mathbf{W}} - \underline{\mathbf{Z}}\|_F^2$

TPGD: Approximate Projection Step

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r},\mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Projection step: $\widehat{\underline{\mathbf{W}}} = \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \|\underline{\mathbf{W}} - \underline{\mathbf{Z}}\|_F^2$

Sparse Higher-order SVD [Allen'12]

1: **Input:** Tensor $\underline{\mathbf{W}}$, rank tuple $\mathbf{r}$, and sparsity tuple $\mathbf{s}$
2: **for** $k = 1, \ldots, K$ **do**
3: $\quad \mathbf{U}_k \leftarrow$ First r_k, s_k-*sparse* principal components of $\mathbf{W}_{(k)}$
4: **end for**
5: $\underline{\mathbf{G}} \leftarrow \underline{\mathbf{W}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_K \mathbf{U}_K$
6: **return** Tensor $\widehat{\underline{\mathbf{W}}} \leftarrow \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_K \mathbf{U}_K$

TPGD: Approximate Projection Step

Define $\mathcal{B}_{\mathbf{r},\mathbf{s},\tau} := \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_K} | \underline{\mathbf{B}} \text{ is } (\mathbf{r}, \mathbf{s})\text{–Tucker decomposable and } \|\underline{\mathbf{G}}\|_1 \leq \tau \right\}$

Projection step: $\widehat{\underline{\mathbf{W}}} = \underset{\underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}}{\arg\min} \|\underline{\mathbf{W}} - \underline{\mathbf{Z}}\|_F^2$

Sparse Higher-order SVD [Allen'12]

Mode-k matricization

1: **Input:** Tensor $\underline{\mathbf{W}}$, rank tuple $\mathbf{r}$, and sparsity tuple $\mathbf{s}$
2: **for** $k = 1, \ldots, K$ **do**
3: $\mathbf{U}_k \leftarrow$ First r_k, s_k-*sparse* principal components of $\mathbf{W}_{(k)}$
4: **end for**
5: $\underline{\mathbf{G}} \leftarrow \underline{\mathbf{W}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_K \mathbf{U}_K$
6: **return** Tensor $\widehat{\underline{\mathbf{W}}} \leftarrow \underline{\mathbf{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_K \mathbf{U}_K$

Mode-k matricization/unfolding $\mathbf{W}_{(k)}$ of $\underline{\mathbf{W}}$

- Stacking of mode-k 'fibers' of $\underline{\mathbf{W}}$ into columns of $\mathbf{W}_{(k)} \in \mathbb{R}^{p_k \times \Pi_{j \neq k} p_j}$

Convergence of TPGD for tensor regression

$(\mathbf{r}, \mathbf{s}, \tau, \delta_{\mathbf{r},\mathbf{s},\tau})$-Restricted Isometry Property (RIP)

A linear map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{p_1 \times \cdots \times p_K} \to \mathbb{R}^n$ acting on tensors of order K satisfies the RIP with constant $\delta_{\mathbf{r},\mathbf{s},\tau}$ if the following holds:

$$\forall \underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}, \quad (1 - \delta_{\mathbf{r},\mathbf{s},\tau})\|\underline{\mathbf{Z}}\|_F^2 \leq \|\boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2 \leq (1 + \delta_{\mathbf{r},\mathbf{s},\tau})\|\underline{\mathbf{Z}}\|_F^2.$$

Convergence of TPGD for tensor regression

$(\mathbf{r}, \mathbf{s}, \tau, \delta_{\mathbf{r},\mathbf{s},\tau})$-Restricted Isometry Property (RIP)

A linear map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{p_1 \times \cdots \times p_K} \to \mathbb{R}^n$ acting on tensors of order K satisfies the RIP with constant $\delta_{\mathbf{r},\mathbf{s},\tau}$ if the following holds:

$$\forall \underline{\mathbf{Z}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}, \quad (1 - \delta_{\mathbf{r},\mathbf{s},\tau})\|\underline{\mathbf{Z}}\|_F^2 \leq \|\boldsymbol{\mathcal{X}}(\underline{\mathbf{Z}})\|_2^2 \leq (1 + \delta_{\mathbf{r},\mathbf{s},\tau})\|\underline{\mathbf{Z}}\|_F^2.$$

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ and the map $\boldsymbol{\mathcal{X}}$ satisfies RIP with constant $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ for $\gamma \in (0,1)$. Then, fixing step size $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ and defining $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$, the estimation error of TPGD after t iterations satisfies

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1 - \delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1 - \delta_{2\mathbf{r},\mathbf{s},2\tau}}\Big(1 + \frac{b}{1-\gamma}\Big).$$

Implications of convergence guarantees for TPGD

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$. *Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$, *the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}\Big(1 + \frac{b}{1-\gamma}\Big).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

Implications of convergence guarantees for TPGD

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$*. Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$*, the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}\left(1 + \frac{b}{1-\gamma}\right).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

Implications of convergence guarantees for TPGD

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$*. Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$*, the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}\left(1 + \frac{b}{1-\gamma}\right).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$*. Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$*, the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}\left(1 + \frac{b}{1-\gamma}\right).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

Implications of convergence guarantees for TPGD

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$*. Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$*, the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\|\mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)})\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}\left(1 + \frac{b}{1-\gamma}\right).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

Theorem (Convergence of TPGD [AhmedRajaB.**'20])**

Suppose the regression tensor $\underline{\mathbf{B}} \in \mathcal{B}_{\mathbf{r},\mathbf{s},\tau}$ *and the map* $\boldsymbol{\mathcal{X}}$ *satisfies RIP with constant* $\delta_{2\mathbf{r},\mathbf{s},2\tau} < \frac{\gamma}{4+\gamma}$ *for* $\gamma \in (0,1)$*. Then, fixing step size* $\mu = \frac{1}{1+\delta_{2\mathbf{r},\mathbf{s},2\tau}}$ *and defining* $b := \frac{1+3\delta_{2\mathbf{r},\mathbf{s},2\tau}}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}}$*, the estimation error of TPGD after* t *iterations satisfies*

$$\|\underline{\mathbf{B}}^{(t)} - \underline{\mathbf{B}}\|_F^2 \leq \frac{2\gamma^t}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left\| \mathbf{y} - \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}^{(0)}) \right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\mathbf{r},\mathbf{s},2\tau}} \left(1 + \frac{b}{1-\gamma}\right).$$

- Convergence guarantees for constant stepsize
- Geometric / linear rate of convergence for the algorithm
- Estimation error scales linearly with (deterministic) noise power

But are there linear maps operating on tensor spaces that satisfy the RIP?

Sample complexity of TPGD for sub-Gaussian maps

Sub-Gaussian random variable with parameter α

- Moment generating function is dominated by that of a Gaussian random variable with variance α^2
  - Tail of the distribution is dominated by that of a Gaussian distribution
- Examples: Gaussian, bounded, uniform, and binary random variables

Theorem (Sample Complexity of Sub-Gaussian Maps [AhmedRajaB.**'20])**

Let the entries of $\{\underline{\mathbf{X}}_i\}_{i=1}^n$ be independently drawn from zero-mean, $\frac{1}{n}$-variance sub-Gaussian distributions, and define $p_{\max} := \max_k p_k$. Then, $\forall \delta, \varepsilon \in (0,1)$, the map $\boldsymbol{\mathcal{X}}$ satisfies $\delta_{\mathbf{r},\mathbf{s},\tau} \leq \delta$ with probability at least $1-\varepsilon$ as long as

$$n \geq \delta^{-2} \max\left\{ C_1\, \tau^2 \left(\sum_{k=1}^{K} s_k r_k + \prod_{k=1}^{K} r_k \right) \log^2(3 p_{\max} K),\ C_2 \log(\varepsilon^{-1}) \right\},$$

where the constants C_1, $C_2 > 0$ depend on τ and the sub-Gaussian parameter α.

Sample complexity of TPGD for sub-Gaussian maps

Sub-Gaussian random variable with parameter α

- Moment generating function is dominated by that of a Gaussian random variable with variance α^2
  - Tail of the distribution is dominated by that of a Gaussian distribution
- Examples: Gaussian, bounded, uniform, and binary random variables

Theorem (Sample Complexity of Sub-Gaussian Maps [AhmedRajaB.**'20])**

Let the entries of $\{\underline{\mathbf{X}}_i\}_{i=1}^n$ be independently drawn from zero-mean, $\frac{1}{n}$-variance sub-Gaussian distributions, and define $p_{\max} := \max_k p_k$. Then, $\forall \delta, \varepsilon \in (0,1)$, the map $\boldsymbol{\mathcal{X}}$ satisfies $\delta_{\mathbf{r},\mathbf{s},\tau} \leq \delta$ with probability at least $1-\varepsilon$ as long as

$$n \geq \delta^{-2} \max\left\{C_1\, \tau^2 \left(\sum_{k=1}^{K} s_k r_k + \prod_{k=1}^{K} r_k\right) \log^2(3 p_{\max} K),\ C_2 \log(\varepsilon^{-1})\right\},$$

where the constants C_1, $C_2 > 0$ depend on τ and the sub-Gaussian parameter α.

Sample complexity of TPGD for sub-Gaussian maps

Sub-Gaussian random variable with parameter α

- Moment generating function is dominated by that of a Gaussian random variable with variance α^2
  - Tail of the distribution is dominated by that of a Gaussian distribution
- Examples: Gaussian, bounded, uniform, and binary random variables

Theorem (Sample Complexity of Sub-Gaussian Maps [AhmedRajaB.**'20])**

Let the entries of $\{\underline{\mathbf{X}}_i\}_{i=1}^n$ be independently drawn from zero-mean, $\frac{1}{n}$-variance sub-Gaussian distributions, and define $p_{\max} := \max_k p_k$. Then, $\forall \delta, \varepsilon \in (0,1)$, the map $\boldsymbol{\mathcal{X}}$ satisfies $\delta_{\mathbf{r},\mathbf{s},\tau} \leq \delta$ with probability at least $1-\varepsilon$ as long as

$$n \geq \delta^{-2} \max\left\{ C_1\, \tau^2 \left(\sum_{k=1}^{K} s_k r_k + \prod_{k=1}^{K} r_k \right) \log^2(3 p_{\max} K),\ C_2 \log(\varepsilon^{-1}) \right\},$$

where the constants C_1, $C_2 > 0$ depend on τ and the sub-Gaussian parameter α.

Sample complexity comparison with prior works

Assume $p_1 = \cdots = p_K \equiv \bar{p}$, $r_1 = \ldots r_K \equiv \bar{r}$, and $s_1 = \cdots = s_K \equiv \bar{s}$

Reference	Regression Tensor $(\underline{\mathbf{B}})$	Sample Complexity
Tomioka et al.'11	Low-rank Tucker	$\bar{r}\bar{p}^{K-1}$
Mu et al.'13	Low-rank Tucker	$\bar{r}^{\lfloor K/2 \rfloor}\bar{p}^{\lfloor K/2 \rfloor}$
Rauhut et al.'17	Low-rank Tucker	$(\bar{r}\bar{p}K + \bar{r}^K)\log(K)$
This Talk	Low-rank and sparse Tucker	$\tau^2(\bar{s}\bar{r}K + \bar{r}^K)\log^2(\bar{p}K)$

Sample complexity comparison with prior works

Assume $p_1 = \cdots = p_K \equiv \bar{p}$, $r_1 = \ldots r_K \equiv \bar{r}$, and $s_1 = \cdots = s_K \equiv \bar{s}$

Reference	**Regression Tensor** ($\underline{\mathbf{B}}$)	**Sample Complexity**
Tomioka et al.'11	Low-rank Tucker	$\bar{r}\bar{p}^{K-1}$
Mu et al.'13	Low-rank Tucker	$\bar{r}^{\lfloor K/2 \rfloor}\bar{p}^{\lfloor K/2 \rfloor}$
Rauhut et al.'17	Low-rank Tucker	$(\bar{r}\bar{p}K + \bar{r}^K)\log(K)$
This Talk	Low-rank and sparse Tucker	$\tau^2(\bar{s}\bar{r}K + \bar{r}^K)\log^2(\bar{p}K)$

Typical values obtained from neuroimaging datasets

- $K = 3, \bar{p} = 128, \bar{r} = 3$, and $\bar{s} = 10$
    - An order of magnitude difference in sample complexity!!!

Sample complexity comparison with prior works

Assume $p_1 = \cdots = p_K \equiv \bar{p}$, $r_1 = \ldots r_K \equiv \bar{r}$, and $s_1 = \cdots = s_K \equiv \bar{s}$

Reference	Regression Tensor ($\underline{\mathbf{B}}$)	Sample Complexity
Tomioka et al.'11	Low-rank Tucker	$\bar{r}\bar{p}^{K-1}$
Mu et al.'13	Low-rank Tucker	$\bar{r}^{\lfloor K/2 \rfloor}\bar{p}^{\lfloor K/2 \rfloor}$
Rauhut et al.'17	Low-rank Tucker	$(\bar{r}\bar{p}K + \bar{r}^K)\log(K)$
This Talk	Low-rank and sparse Tucker	$\tau^2(\bar{s}\bar{r}K + \bar{r}^K)\log^2(\bar{p}K)$

Typical values obtained from neuroimaging datasets

- $K = 3, \bar{p} = 128, \bar{r} = 3$, and $\bar{s} = 10$
  - An order of magnitude difference in sample complexity!!!

Synthetic data experiments: Setup

- Regression tensor $\underline{\mathbf{B}} \in \mathbb{R}^{50\times 50\times 30}$
  - $r_1 = r_2 = r_3 = 3$
  - $s_1 = 6, s_2 = 6, s_3 = 4$
  - Randomly generated core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{3\times 3\times 3}$
  - Uniformly-at-random locations of nonzero (random) entries

Synthetic data experiments: Setup

- Regression tensor $\underline{\mathbf{B}} \in \mathbb{R}^{50\times 50\times 30}$
    - $r_1 = r_2 = r_3 = 3$
    - $s_1 = 6, s_2 = 6, s_3 = 4$
    - Randomly generated core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{3\times 3\times 3}$
    - Uniformly-at-random locations of nonzero (random) entries
- Gaussian regression map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{50\times 50\times 30} \to \mathbb{R}^{n}$

Synthetic data experiments: Setup

- Regression tensor $\underline{\mathbf{B}} \in \mathbb{R}^{50\times 50\times 30}$
  - $r_1 = r_2 = r_3 = 3$
  - $s_1 = 6, s_2 = 6, s_3 = 4$
  - Randomly generated core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{3\times 3\times 3}$
  - Uniformly-at-random locations of nonzero (random) entries
- Gaussian regression map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{50\times 50\times 30} \to \mathbb{R}^n$
- Gaussian additive noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$
  - $\sigma^2 = 0.1$ and $\sigma^2 = 0.4$

Synthetic data experiments: Setup

- Regression tensor $\underline{\mathbf{B}} \in \mathbb{R}^{50\times 50\times 30}$
  - $r_1 = r_2 = r_3 = 3$
  - $s_1 = 6, s_2 = 6, s_3 = 4$
  - Randomly generated core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{3\times 3\times 3}$
  - Uniformly-at-random locations of nonzero (random) entries
- Gaussian regression map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{50\times 50\times 30} \to \mathbb{R}^n$
- Gaussian additive noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$
  - $\sigma^2 = 0.1$ and $\sigma^2 = 0.4$
- Vector of n observations of the response variable $\mathbf{y} = \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}) + \boldsymbol{\eta}$

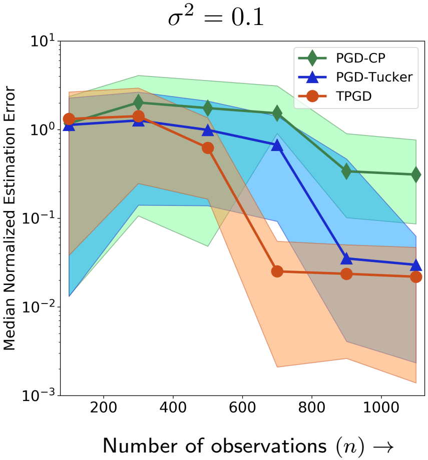
Synthetic data experiments: Setup

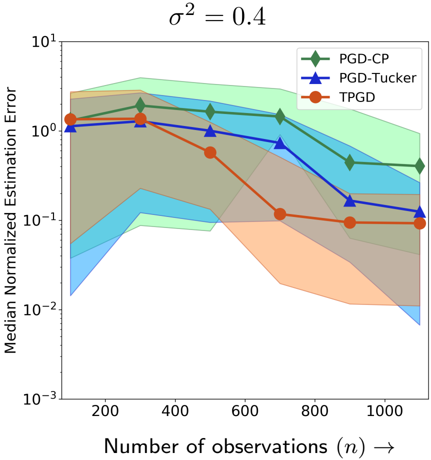
- Regression tensor $\underline{\mathbf{B}} \in \mathbb{R}^{50\times 50\times 30}$
  - $r_1 = r_2 = r_3 = 3$
  - $s_1 = 6, s_2 = 6, s_3 = 4$
  - Randomly generated core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{3\times 3\times 3}$
  - Uniformly-at-random locations of nonzero (random) entries
- Gaussian regression map $\boldsymbol{\mathcal{X}} : \mathbb{R}^{50\times 50\times 30} \to \mathbb{R}^n$
- Gaussian additive noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$
  - $\sigma^2 = 0.1$ and $\sigma^2 = 0.4$
- Vector of n observations of the response variable $\mathbf{y} = \boldsymbol{\mathcal{X}}(\underline{\mathbf{B}}) + \boldsymbol{\eta}$

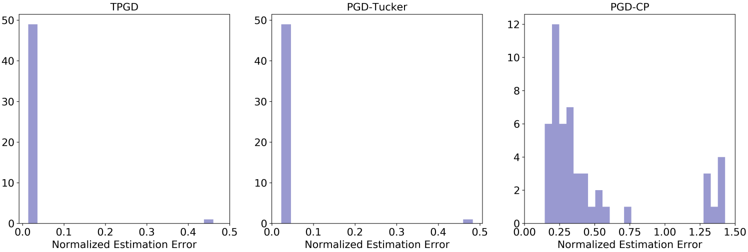
Comparison of the performance of TPGD with

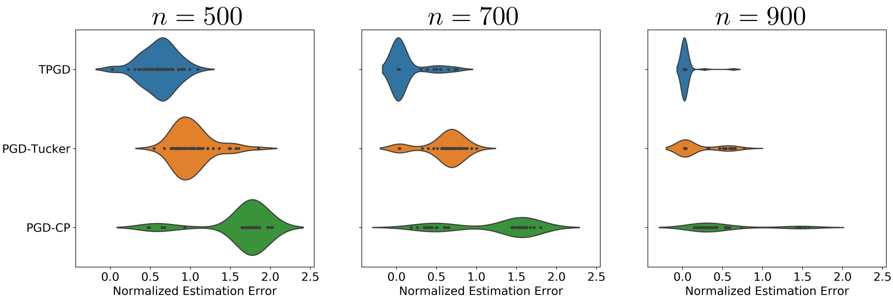
- Sparse regression (e.g., lasso)
- Low-rank CP regression (PGD-CP) [Zhou et al.'13]
- Low-rank Tucker regression (PGD-Tucker) [Rauhut et al.'17]

Synthetic data experiments: Results






TPGD
PGD-Tucker
PGD-CP
Normalized Estimation Error
Normalized Estimation Error
Normalized Estimation Error


$n = 500$
$n = 700$
$n = 900$
TPGD
PGD-Tucker
PGD-CP
Normalized Estimation Error
Normalized Estimation Error
Normalized Estimation Error

Real-world neuroimaging data experiments: Setup

ADHD-200 Sample: A collaboration of 8 international imaging sites studying *attention deficit/hyperactivity disorder* (ADHD) in children and adolescents

Dataset description

- Task: Predict ADHD diagnosis of subjects who participated in ADHD studies at the NYU (New York University Child Study Center), the NeuroImage (The Donders Institute) and the KKI (Kennedy Krieger Institute) imaging sites
- Raw data: *Functional magnetic resonance imaging* (fMRI) data

Real-world neuroimaging data experiments: Setup

ADHD-200 Sample: A collaboration of 8 international imaging sites studying *attention deficit/hyperactivity disorder* (ADHD) in children and adolescents

Dataset description

- Task: Predict ADHD diagnosis of subjects who participated in ADHD studies at the NYU (New York University Child Study Center), the NeuroImage (The Donders Institute) and the KKI (Kennedy Krieger Institute) imaging sites
- Raw data: *Functional magnetic resonance imaging* (fMRI) data
- Preprocessed data: Brain maps of *fractional amplitude of low-frequency fluctuations* (fALFF) obtained from fMRI data
    - $p_1 = 49, p_2 = 58, p_3 = 47 \Rightarrow p := p_1 p_2 p_3 = 133,574$

Real-world neuroimaging data experiments: Setup

ADHD-200 Sample: A collaboration of 8 international imaging sites studying *attention deficit/hyperactivity disorder* (ADHD) in children and adolescents

Dataset description

- Task: Predict ADHD diagnosis of subjects who participated in ADHD studies at the NYU (New York University Child Study Center), the NeuroImage (The Donders Institute) and the KKI (Kennedy Krieger Institute) imaging sites
- Raw data: *Functional magnetic resonance imaging* (fMRI) data
- Preprocessed data: Brain maps of *fractional amplitude of low-frequency fluctuations* (fALFF) obtained from fMRI data
  - $p_1 = 49, p_2 = 58, p_3 = 47 \Rightarrow p := p_1 p_2 p_3 = 133,574$
- Collective training data: 305 subjects (134 w/ ADHD, 171 controls)
- Collective test data: 77 subjects, divided into w/ ADHD and controls

Real-world neuroimaging data experiments: Results

NeuroImage Dataset $(n = 39;$ ADHD $= 17)$

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Specificity (TNR)	0.68	0.57	0.57	1	0.89
Sensitivity (TPR)	0.73	0.45	0.64	0.18	0.36
Harmonic mean	**0.70**	0.50	0.60	0.31	0.51

KKI Dataset $(n = 78;$ ADHD $= 20)$

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Specificity (TNR)	0.63	0.50	0.50	1	1
Sensitivity (TPR)	0.67	0.33	0.33	0	0
Harmonic mean	**0.65**	0.40	0.40	0	0

NYU Dataset $(n = 188;$ ADHD $= 97)$

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Harmonic mean	0.55	**0.59**	0.56	0.48	0.26

Outline

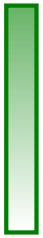
Review of dictionary learning for vector data

Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods

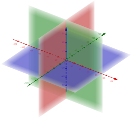
Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

Review of dictionary learning for vector data

Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods



$\mathbf{y} \in \mathbb{R}^m$



$p \geq m$

$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\|_0 \leq s\}$

Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

Review of dictionary learning for vector data

Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods

$$\mathcal{M}_{\mathbf{D}} : \mathbf{y} \mapsto \mathbf{x} = \arg\min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2$$

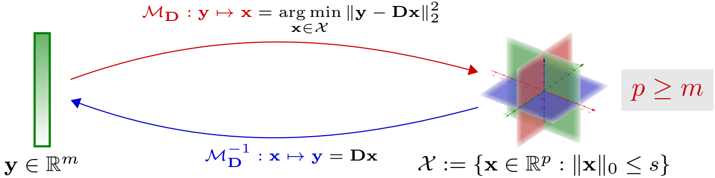
$p \geq m$

$\mathbf{y} \in \mathbb{R}^m$

$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\|_0 \leq s\}$

Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

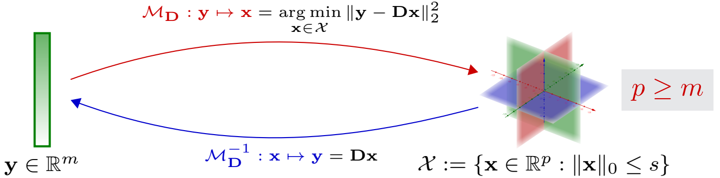
Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods

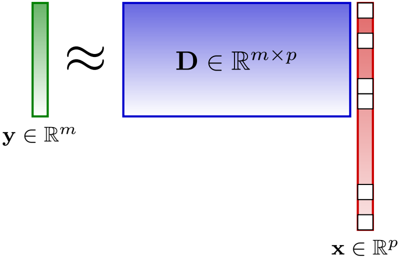


Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

Review of dictionary learning for vector data

Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods

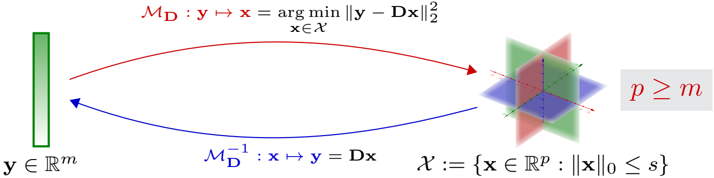


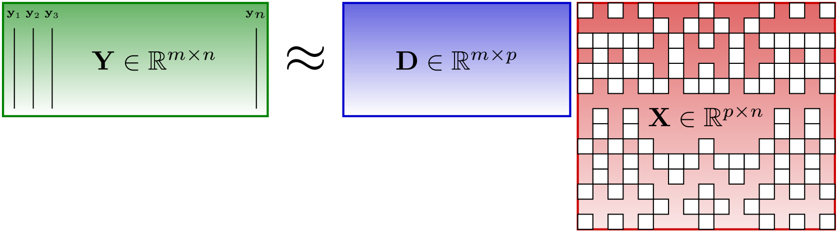


Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

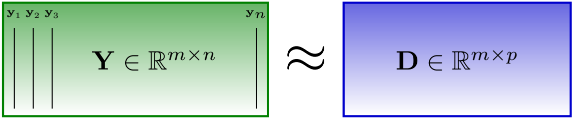
Review of dictionary learning for vector data

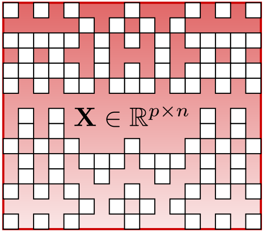
Dictionary learning: A nonlinear feature learning approach that sits between (linear) principal component analysis and (nonlinear) kernel-based methods





Review article: Tosic, Frossard, "Dictionary learning," IEEE Signal Processing Magazine, 2011

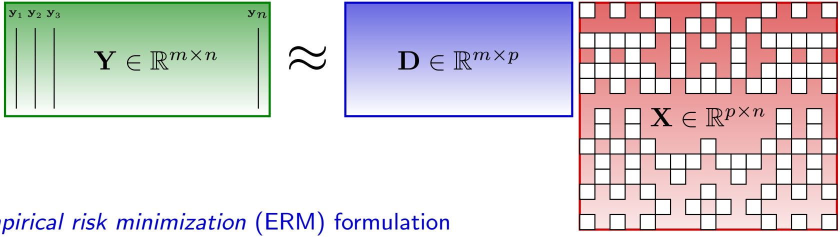




Empirical risk minimization (ERM) formulation

$$\widehat{\mathbf{D}} \in \underset{\mathbf{D} \in \mathcal{D}}{\arg\min} \left[\mathbf{F}_{\mathbf{Y}}(\mathbf{D}) := \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \|\mathbf{y}_j - \mathbf{D}\mathbf{x}_j\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\} \right]$$

Review chapter: Shakeri, Sarwate, **B.**, "Sample complexity bounds for dictionary learning from vector- and tensor-valued data," in Information-Theoretic Methods in Data Science, Cambridge University Press, 2020

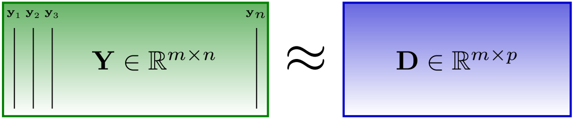


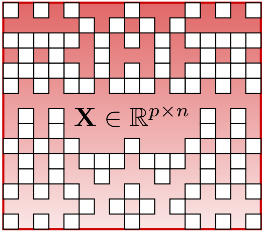
Empirical risk minimization (ERM) formulation

$$\widehat{\mathbf{D}} \in \arg\min_{\mathbf{D} \in \mathcal{D}} \left[\mathbf{F}_{\mathbf{Y}}(\mathbf{D}) := \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \|\mathbf{y}_j - \mathbf{D}\mathbf{x}_j\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\} \right]$$

- Methods: [Engan et al.'99], [Aharon et al.'06], [Mairal et al.'10], [ZhangLi'10], ...
- Sample complexity results: [Schnass'14], [Arora et al.'14], [GengWright'14], [Gribonval et al.'15], [Jung et al.'16], ...

Review chapter: Shakeri, Sarwate, **B.**, "Sample complexity bounds for dictionary learning from vector- and tensor-valued data," in Information-Theoretic Methods in Data Science, Cambridge University Press, 2020





Empirical risk minimization (ERM) formulation

$$\widehat{\mathbf{D}} \in \underset{\mathbf{D} \in \mathcal{D}}{\arg\min} \left[\mathbf{F}_{\mathbf{Y}}(\mathbf{D}) := \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \|\mathbf{y}_j - \mathbf{D}\mathbf{x}_j\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\} \right]$$

- Methods: [Engan et al.'99], [Aharon et al.'06], [Mairal et al.'10], [ZhangLi'10], ...
- Sample complexity results: [Schnass'14], [Arora et al.'14], [GengWright'14], [Gribonval et al.'15], [Jung et al.'16], ...

Bounds for $\|\cdot\|_F$ error: $mp^2\varepsilon^{-2} \preceq n \preceq mp^3\varepsilon^{-2}$ → Impractical for most tensor data!!!

Review chapter: Shakeri, Sarwate, **B.**, "Sample complexity bounds for dictionary learning from vector- and tensor-valued data," in Information-Theoretic Methods in Data Science, Cambridge University Press, 2020

Dictionary learning for tensor data

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K},\ j = 1, \ldots, n$

Dictionary learning for tensor data

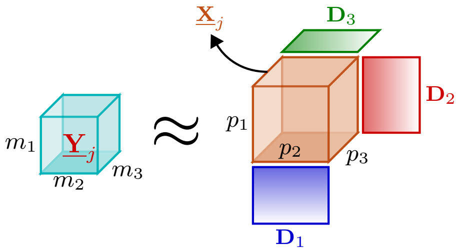
Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K},\ j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition

Dictionary learning for tensor data

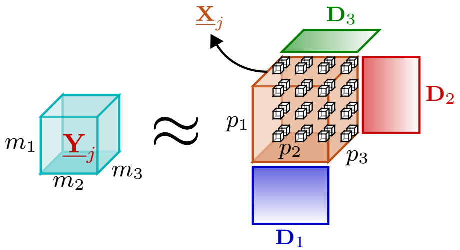
Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K},\ j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \; j = 1, \ldots, n$

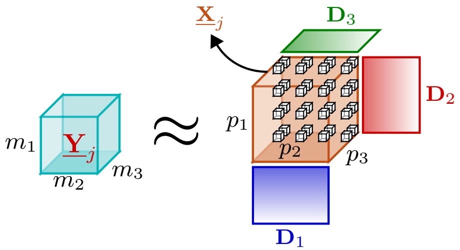
Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



Dictionary learning for tensor data

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \; j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



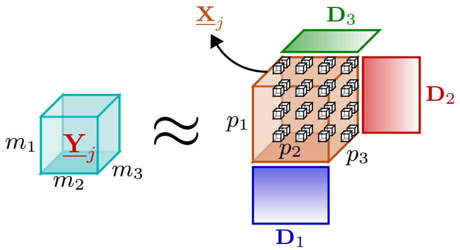
$\underline{\mathbf{X}}_j$: Sparse core tensor $(p_1 \times p_2 \times p_3)$
$\mathbf{D}_1$: Mode-1 subdictionary $(m_1 \times p_1)$
$\mathbf{D}_2$: Mode-2 subdictionary $(m_2 \times p_2)$
$\mathbf{D}_3$: Mode-3 subdictionary $(m_3 \times p_3)$

Sparsity: $s := \|\underline{\mathbf{X}}\|_0 \ll p := p_1 p_2 p_3$
Overcomplete: $(m_1, m_2, m_3) \leq (p_1, p_2, p_3)$

- $\mathbf{y}_j :=$ vec$(\underline{\mathbf{Y}}_j) \Rightarrow$ length $m := \prod_k m_k$
- $\mathbf{x}_j :=$ vec$(\underline{\mathbf{X}}_j) \Rightarrow$ length $p := \prod_k p_k$

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \ j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



$\underline{\mathbf{X}}_j$: Sparse core tensor $(p_1 \times p_2 \times p_3)$
$\mathbf{D}_1$: Mode-1 subdictionary $(m_1 \times p_1)$
$\mathbf{D}_2$: Mode-2 subdictionary $(m_2 \times p_2)$
$\mathbf{D}_3$: Mode-3 subdictionary $(m_3 \times p_3)$

Sparsity: $s := \|\underline{\mathbf{X}}\|_0 \ll p := p_1 p_2 p_3$
Overcomplete: $(m_1, m_2, m_3) \leq (p_1, p_2, p_3)$

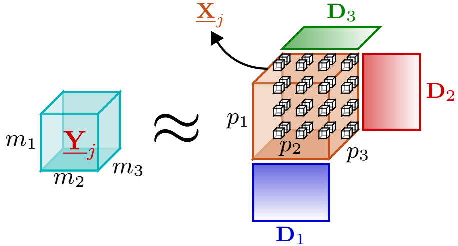
- $\mathbf{y}_j := \text{vec}(\underline{\mathbf{Y}}_j) \Rightarrow$ length $m := \prod_k m_k$
- $\mathbf{x}_j := \text{vec}(\underline{\mathbf{X}}_j) \Rightarrow$ length $p := \prod_k p_k$

- $\underline{\mathbf{Y}}_j \approx \sum_{i_1, i_2, i_3} x_{j,(i_1,i_2,i_3)} \mathbf{d}_{1,i_1} \circ \mathbf{d}_{2,i_2} \circ \mathbf{d}_{3,i_3} = \underline{\mathbf{X}}_j \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \mathbf{D}_3$

Dictionary learning for tensor data

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}$, $j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



$\underline{\mathbf{X}}_j$: Sparse core tensor $(p_1 \times p_2 \times p_3)$
$\mathbf{D}_1$: Mode-1 subdictionary $(m_1 \times p_1)$
$\mathbf{D}_2$: Mode-2 subdictionary $(m_2 \times p_2)$
$\mathbf{D}_3$: Mode-3 subdictionary $(m_3 \times p_3)$

Sparsity: $s := \|\underline{\mathbf{X}}\|_0 \ll p := p_1 p_2 p_3$
Overcomplete: $(m_1, m_2, m_3) \leq (p_1, p_2, p_3)$

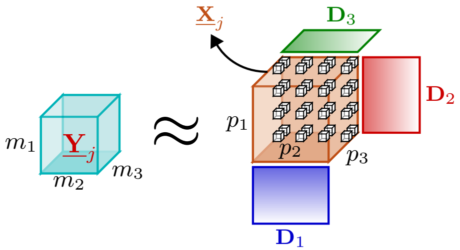
- $\mathbf{y}_j :=$ vec$(\underline{\mathbf{Y}}_j) \Rightarrow$ length $m := \prod_k m_k$
- $\mathbf{x}_j :=$ vec$(\underline{\mathbf{X}}_j) \Rightarrow$ length $p := \prod_k p_k$

- $\underline{\mathbf{Y}}_j \approx \sum_{i_1, i_2, i_3} x_{j,(i_1,i_2,i_3)} \mathbf{d}_{1,i_1} \circ \mathbf{d}_{2,i_2} \circ \mathbf{d}_{3,i_3} = \underline{\mathbf{X}}_j \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \mathbf{D}_3$
- Vectorize tensor sample ➠ $\mathbf{y}_j \approx (\mathbf{D}_3 \otimes \mathbf{D}_2 \otimes \mathbf{D}_1)\mathbf{x}_j$

Dictionary learning for tensor data

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \; j = 1, \ldots, n$

Sparse representation in a dictionary ⇔ Overcomplete, sparse Tucker decomposition



$\underline{\mathbf{X}}_j$: Sparse core tensor $(p_1 \times p_2 \times p_3)$
$\mathbf{D}_1$: Mode-1 subdictionary $(m_1 \times p_1)$
$\mathbf{D}_2$: Mode-2 subdictionary $(m_2 \times p_2)$
$\mathbf{D}_3$: Mode-3 subdictionary $(m_3 \times p_3)$

Sparsity: $s := \|\underline{\mathbf{X}}\|_0 \ll p := p_1 p_2 p_3$
Overcomplete: $(m_1, m_2, m_3) \le (p_1, p_2, p_3)$

- $\mathbf{y}_j :=$ vec$(\underline{\mathbf{Y}}_j) \Rightarrow$ length $m := \prod_k m_k$
- $\mathbf{x}_j :=$ vec$(\underline{\mathbf{X}}_j) \Rightarrow$ length $p := \prod_k p_k$

- $\underline{\mathbf{Y}}_j \approx \sum_{i_1, i_2, i_3} x_{j,(i_1,i_2,i_3)} \mathbf{d}_{1,i_1} \circ \mathbf{d}_{2,i_2} \circ \mathbf{d}_{3,i_3} = \underline{\mathbf{X}}_j \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \mathbf{D}_3$
- Vectorize tensor sample ➠ $\mathbf{y}_j \approx (\mathbf{D}_3 \otimes \mathbf{D}_2 \otimes \mathbf{D}_1)\mathbf{x}_j$
- **General case:** $\mathbf{y}_j \approx \mathbf{D}\mathbf{x}_j, \|\mathbf{x}_j\|_0 \le s$ such that $\boxed{\mathbf{D} := \mathbf{D}_K \otimes \mathbf{D}_{K-1} \otimes \cdots \otimes \mathbf{D}_1}$

Degrees of freedom in a Kronecker-structured dictionary

Unstructured Dictionary

$m_1 = 512$

$m_2 = 512$

m | $\mathbf{D} \in \mathbb{R}^{m \times p}$ | p

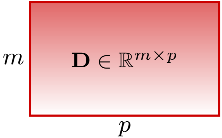
$$m = m_1 m_2 = 2^{18}$$
$$p = m \Rightarrow mp = 2^{36}$$

6,871 billion parameters

$m_1 = 512$

$m_2 = 512$

Unstructured Dictionary



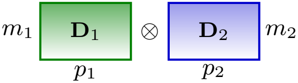
$m = m_1 m_2 = 2^{18}$

$p = m \Rightarrow mp = 2^{36}$

6,871 billion parameters

Kronecker-structured Dictionary

$$\mathbf{D} = \mathbf{D}_1 \otimes \mathbf{D}_2$$



$p_1 = p_2 = 2^9$

$\Rightarrow (m_1 p_1 + m_2 p_2) = 2^{19}$

524,288 parameters

$m_1 = 512$

$m_2 = 512$

Unstructured Dictionary

m $\mathbf{D} \in \mathbb{R}^{m \times p}$ p

$m = m_1 m_2 = 2^{18}$

$p = m \Rightarrow mp = 2^{36}$

6,871 billion parameters

Kronecker-structured Dictionary

$\mathbf{D} = \mathbf{D}_1 \otimes \mathbf{D}_2$

m_1 $\mathbf{D}_1$ p_1 $\otimes$ $\mathbf{D}_2$ m_2 p_2

$p_1 = p_2 = 2^9$

$\Rightarrow (m_1 p_1 + m_2 p_2) = 2^{19}$

524,288 parameters

Related prior works

- [Hawe et al.'13], [Zubair et al.'13], [CaiafaCichocki'13], [Roemer et al.'14], [Dantas et al.'17], ...

Tensor dictionary learning: Sample complexity bounds

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \; j = 1, \dots, n$

Empirical risk minimization (ERM) formulation

$$(\widehat{\mathbf{D}}_1, \dots, \widehat{\mathbf{D}}_K) \in \underset{(\mathbf{D}_1, \dots, \mathbf{D}_K)}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \mathsf{vec}(\underline{\mathbf{Y}}_j) - \left(\bigotimes_k \mathbf{D}_k \right) \mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$$

Error metrics: $\varepsilon := \left\| \bigotimes_k \widehat{\mathbf{D}}_k - \bigotimes_k \mathbf{D}_k \right\|_F$ and $\varepsilon_k := \left\| \widehat{\mathbf{D}}_k - \mathbf{D}_k \right\|_F$

Tensor dictionary learning: Sample complexity bounds

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \ j = 1, \ldots, n$

Empirical risk minimization (ERM) formulation

$$(\widehat{\mathbf{D}}_1, \ldots, \widehat{\mathbf{D}}_K) \in \underset{(\mathbf{D}_1, \ldots, \mathbf{D}_K)}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \left(\bigotimes_k \mathbf{D}_k \right) \mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$$

Error metrics: $\varepsilon := \left\| \bigotimes_k \widehat{\mathbf{D}}_k - \bigotimes_k \mathbf{D}_k \right\|_F$ and $\varepsilon_k := \left\| \widehat{\mathbf{D}}_k - \mathbf{D}_k \right\|_F$

Theorem (Informal Bounds [ShakeriB.Sarwate'18, ShakeriSarwateB.'18])

Assuming independent and identically distributed samples $\underline{\mathbf{Y}}_j$, possibly corrupted by additive noise, under the overcomplete, sparse Tucker decomposition model, the following sample complexity bounds hold for Kronecker-structured dictionary learning:

- *Minimax lower bound:* $n \succeq p \left(\sum_k m_k p_k \right) \varepsilon^{-2} / K$
- *Achievability upper bound:* $n \preceq \max_k \left(m_k p_k^3 \varepsilon_k^{-2} \right)$

Tensor dictionary learning: Sample complexity bounds

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}$, $j = 1, \ldots, n$

Empirical risk minimization (ERM) formulation

$$(\widehat{\mathbf{D}}_1, \ldots, \widehat{\mathbf{D}}_K) \in \underset{(\mathbf{D}_1,\ldots,\mathbf{D}_K)}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \left(\bigotimes_k \mathbf{D}_k \right) \mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$$

Error metrics: $\varepsilon := \left\| \bigotimes_k \widehat{\mathbf{D}}_k - \bigotimes_k \mathbf{D}_k \right\|_F$ and $\varepsilon_k := \left\| \widehat{\mathbf{D}}_k - \mathbf{D}_k \right\|_F$

Theorem (Informal Bounds [ShakeriB.Sarwate'18, ShakeriSarwateB.'18])

Assuming independent and identically distributed samples $\underline{\mathbf{Y}}_j$, possibly corrupted by additive noise, under the overcomplete, sparse Tucker decomposition model, the following sample complexity bounds hold for Kronecker-structured dictionary learning:

- *Minimax lower bound:* $n \succeq p \left(\sum_k m_k p_k \right) \varepsilon^{-2} / K$
- *Achievability upper bound:* $n \preceq \max_k \left(m_k p_k^3 \varepsilon_k^{-2} \right)$

- Vectorization-based lower bound: $n \succeq p \, (mp) \, \varepsilon^{-2}$ [Jung et al.'16]

Tensor dictionary learning: Sample complexity bounds

Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K},\ j = 1, \ldots, n$

Empirical risk minimization (ERM) formulation

$$(\widehat{\mathbf{D}}_1, \ldots, \widehat{\mathbf{D}}_K) \in \underset{(\mathbf{D}_1, \ldots, \mathbf{D}_K)}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \left(\bigotimes_k \mathbf{D}_k \right) \mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$$

Error metrics: $\varepsilon := \left\| \bigotimes_k \widehat{\mathbf{D}}_k - \bigotimes_k \mathbf{D}_k \right\|_F$ and $\varepsilon_k := \left\| \widehat{\mathbf{D}}_k - \mathbf{D}_k \right\|_F$

Theorem (Informal Bounds [ShakeriB.Sarwate'18, ShakeriSarwateB.'18])

Assuming independent and identically distributed samples $\underline{\mathbf{Y}}_j$, possibly corrupted by additive noise, under the overcomplete, sparse Tucker decomposition model, the following sample complexity bounds hold for Kronecker-structured dictionary learning:

- *Minimax lower bound:* $n \succeq p \left(\sum_k m_k p_k \right) \varepsilon^{-2} / K$
- *Achievability upper bound:* $n \preceq \max_k \left(m_k p_k^3 \varepsilon_k^{-2} \right)$

- Vectorization-based lower bound: $n \succeq p\,(mp)\,\varepsilon^{-2}$ [Jung et al.'16]
- Vectorization-based upper bound: $n \preceq mp^3 \varepsilon^{-2}$ [Gribonval et al.'15]

The road to algorithms for tensor dictionary learning

Existing algorithms for Kronecker-structured dictionary learning

- SeDiL [Hawe et al.'13], GradTensor [Zubair et al.'13], Kronecker DL [CaiafaCichocki'13], K-HOSVD [Roemer et al.'14], SuKro [Dantas et al.'17], ...

The road to algorithms for tensor dictionary learning

Existing algorithms for Kronecker-structured dictionary learning

- SeDiL [Hawe et al.'13], GradTensor [Zubair et al.'13], Kronecker DL [CaiafaCichocki'13], K-HOSVD [Roemer et al.'14], SuKro [Dantas et al.'17], ...

Tucker decomposition / Kronecker structure enforces strict separability in modes

- Can a model provide a tradeoff between representation power and sample complexity?

The road to algorithms for tensor dictionary learning

Existing algorithms for Kronecker-structured dictionary learning

- SeDiL [Hawe et al.'13], GradTensor [Zubair et al.'13], Kronecker DL [CaiafaCichocki'13], K-HOSVD [Roemer et al.'14], SuKro [Dantas et al.'17], . . .

Tucker decomposition / Kronecker structure enforces strict separability in modes

- Can a model provide a tradeoff between representation power and sample complexity?

Model: Low-separation-rank, overcomplete, sparse Tucker decomposition

The road to algorithms for tensor dictionary learning

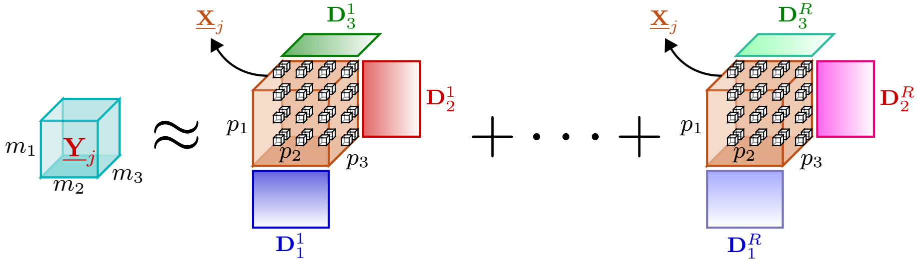
Existing algorithms for Kronecker-structured dictionary learning

- SeDiL [Hawe et al.'13], GradTensor [Zubair et al.'13], Kronecker DL [CaiafaCichocki'13], K-HOSVD [Roemer et al.'14], SuKro [Dantas et al.'17], ...

Tucker decomposition / Kronecker structure enforces strict separability in modes

- Can a model provide a tradeoff between representation power and sample complexity?

Model: Low-separation-rank, overcomplete, sparse Tucker decomposition



- $\underline{\mathbf{Y}}_j \approx \sum_{i_1,i_2,i_3} \underline{x}_{j,(i_1,i_2,i_3)} \left(\sum_{r=1}^{R} \mathbf{d}_{1,i_1}^r \circ \mathbf{d}_{2,i_2}^r \circ \mathbf{d}_{3,i_3}^r \right)$

The road to algorithms for tensor dictionary learning

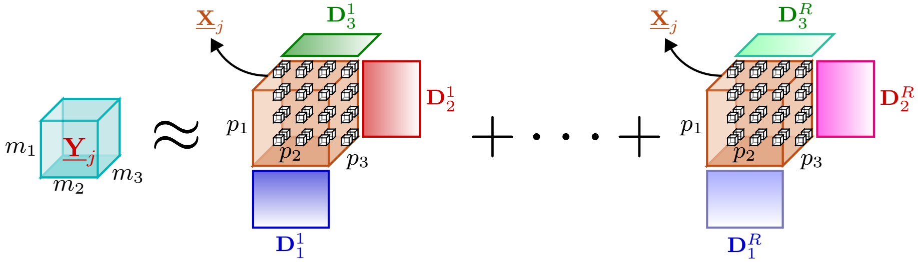
Existing algorithms for Kronecker-structured dictionary learning

- SeDiL [Hawe et al.'13], GradTensor [Zubair et al.'13], Kronecker DL [CaiafaCichocki'13], K-HOSVD [Roemer et al.'14], SuKro [Dantas et al.'17], ...

Tucker decomposition / Kronecker structure enforces strict separability in modes

- Can a model provide a tradeoff between representation power and sample complexity?

Model: Low-separation-rank, overcomplete, sparse Tucker decomposition



- $\underline{\mathbf{Y}}_j \approx \sum_{i_1,i_2,i_3} \underline{x}_{j,(i_1,i_2,i_3)} \left(\sum_{r=1}^{R} \mathbf{d}^r_{1,i_1} \circ \mathbf{d}^r_{2,i_2} \circ \mathbf{d}^r_{3,i_3} \right)$ ➠ $\mathbf{y}_j \approx \left(\sum_{r=1}^{R} \mathbf{D}_3^r \otimes \mathbf{D}_2^r \otimes \mathbf{D}_1^r \right) \mathbf{x}_j$

Tensor dictionary learning and low separation rank

- Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \ j = 1, \ldots, n$
- Dictionary model: $\text{vec}(\underline{\mathbf{Y}}_j) \approx \mathbf{D}\mathbf{x}_j, \|\mathbf{x}_j\|_0 \leq s$ s.t. $\boxed{\mathbf{D} := \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r}$
  - The case of $R = 1$ corresponds to a Kronecker-structured dictionary
  - The parameter R is termed separation rank of the dictionary [BeylkinMohlenkamp'02] [TsiligkaridisHero'13]

Tensor dictionary learning and low separation rank

- Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \ j = 1, \ldots, n$
- Dictionary model: $\text{vec}(\underline{\mathbf{Y}}_j) \approx \mathbf{D}\mathbf{x}_j, \|\mathbf{x}_j\|_0 \leq s$ s.t. $\boxed{\mathbf{D} := \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r}$
  - The case of $R = 1$ corresponds to a Kronecker-structured dictionary
  - The parameter R is termed separation rank of the dictionary [BeylkinMohlenkamp'02] [TsiligkaridisHero'13]
- ERM formulation: $\mathbf{D} \in \arg\min_{\mathbf{D} \in \mathcal{D}_K^R} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \mathbf{D}\mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$
  - $\mathcal{D}_K^R := \left\{ \mathbf{D} \in \mathbb{R}^{m \times p} : \mathbf{D} = \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r, \ \mathbf{D}_k^r \in \mathbb{R}^{m_k \times p_k}, \ \|\mathbf{d}_{k,i}^r\|_2 = 1 \right\}$

Tensor dictionary learning and low separation rank

- Tensor data samples: $\underline{\mathbf{Y}}_j \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}, \ j = 1, \ldots, n$
- Dictionary model: $\text{vec}(\underline{\mathbf{Y}}_j) \approx \mathbf{D}\mathbf{x}_j, \|\mathbf{x}_j\|_0 \leq s$ s.t. $\boxed{\mathbf{D} := \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r}$
  - The case of $R = 1$ corresponds to a Kronecker-structured dictionary
  - The parameter R is termed separation rank of the dictionary [BeylkinMohlenkamp'02] [TsiligkaridisHero'13]
- ERM formulation: $\mathbf{D} \in \arg\min_{\mathbf{D} \in \mathcal{D}_K^R} \sum_{j=1}^{n} \inf_{\mathbf{x}_j \in \mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \mathbf{D}\mathbf{x}_j \right\|_2^2 + \mathcal{R}(\mathbf{x}_j) \right\}$
  - $\mathcal{D}_K^R := \left\{ \mathbf{D} \in \mathbb{R}^{m \times p} : \mathbf{D} = \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r, \ \mathbf{D}_k^r \in \mathbb{R}^{m_k \times p_k}, \ \|\mathbf{d}_{k,i}^r\|_2 = 1 \right\}$

Lemma (The Rearrangement Lemma [GhassemiShakeriSarwateB.**'20])**

Every low-separation rank matrix $\mathbf{D} := \sum_{r=1}^{R} \mathbf{D}_K^r \otimes \cdots \otimes \mathbf{D}_1^r$ *can be rearranged into a* K*-th order tensor* $\underline{\mathbf{D}}^\pi$ *of rank* R *as follows:*

$$\underline{\mathbf{D}}^\pi = \sum_{r=1}^{R} \mathbf{d}_1^r \circ \mathbf{d}_1^r \circ \cdots \circ \mathbf{d}_K^r, \quad \mathbf{d}_k^r := \text{vec}(\mathbf{D}_k^r).$$

Algorithms for tensor dictionary learning

STARK: A regularization-based algorithm [GhassemiShakeriSarwate**B.**'20]

- Uses a convex regularizer for implicit enforcement of the separation rank

$$\widehat{\mathbf{D}} = \operatorname*{arg\,min}_{\mathbf{D}\in\mathcal{D}} \sum_{j=1}^{n} \inf_{\mathbf{x}_j\in\mathcal{X}} \left\{ \frac{1}{2} \left\| \mathsf{vec}(\underline{\mathbf{Y}}_j) - \mathbf{D}\mathbf{x}_j \right\|_2^2 + \lambda \left\| \mathbf{x}_j \right\|_1 \right\} + \lambda_1 \sum_{k=1}^{K} \left\| \mathbf{D}^{\pi}_{(k)} \right\|_{\mathrm{tr}}$$

- Makes use of ADMM to solve the resulting dictionary learning problem

Algorithms for tensor dictionary learning

STARK: A regularization-based algorithm [GhassemiShakeriSarwate**B.**'20]

- Uses a convex regularizer for implicit enforcement of the separation rank

$$\widehat{\mathbf{D}} = \underset{\mathbf{D}\in\mathcal{D}}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j\in\mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \mathbf{D}\mathbf{x}_j \right\|_2^2 + \lambda \left\| \mathbf{x}_j \right\|_1 \right\} + \lambda_1 \sum_{k=1}^{K} \left\| \mathbf{D}^{\pi}_{(k)} \right\|_{\text{tr}}$$

- Makes use of ADMM to solve the resulting dictionary learning problem

TeFDiL: A factorization-based algorithm [GhassemiShakeriSarwate**B.**'20]

- Uses the factored formulation for explicit enforcement of the separation rank

$$\widehat{\mathbf{D}} = \underset{\mathbf{D}:\mathbf{D}=\sum_{r=1}^{R}\otimes_k \mathbf{D}_k^r}{\arg\min} \sum_{j=1}^{n} \inf_{\mathbf{x}_j\in\mathcal{X}} \left\{ \frac{1}{2} \left\| \text{vec}(\underline{\mathbf{Y}}_j) - \mathbf{D}\mathbf{x}_j \right\|_2^2 + \lambda \left\| \mathbf{x}_j \right\|_1 \right\}$$

- Makes use of the rearrangement lemma along with rank-R CP decompositions

Real-world data experiments: Setup

Dataset description

- Task: Denoising of four images (House, Castle, Mushroom, and Lena)
  - All images corrupted with AWGN of standard deviation $\sigma \in \{10, 50\}$
- Training data: Overlapping $8 \times 8 \times 3$ patches
  - $(m_1, m_2, m_3) = (8, 8, 3)$




Real-world data experiments: Setup

Dataset description

- Task: Denoising of four images (House, Castle, Mushroom, and Lena)
  - All images corrupted with AWGN of standard deviation $\sigma \in \{10, 50\}$
- Training data: Overlapping $8 \times 8 \times 3$ patches
  - $(m_1, m_2, m_3) = (8, 8, 3)$

Dictionary dimensions: $(p_1, p_2, p_3) = (16, 16, 3)$

Real-world data experiments: Setup

Dataset description

- Task: Denoising of four images (House, Castle, Mushroom, and Lena)
  - All images corrupted with AWGN of standard deviation $\sigma \in \{10, 50\}$
- Training data: Overlapping $8 \times 8 \times 3$ patches
  - $(m_1, m_2, m_3) = (8, 8, 3)$

Dictionary dimensions: $(p_1, p_2, p_3) = (16, 16, 3)$

Performance metric: *Peak Signal-to-Noise Ratio*

$$\text{PSNR} := 20 \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right)$$

Real-world data experiments: Results

		Unstructured	Kronecker-structured Dictionary			Low-separation-rank Dictionary		
	Noise	K-SVD	SeDiL	BCD	TeFDiL	BCD	STARK	TeFDiL
House	$\sigma = 10$	35.670	23.189	31.609	36.295	32.295	33.400	**37.127**
	$\sigma = 50$	25.468	23.692	24.830	**27.541**	21.613	27.394	26.590
Castle	$\sigma = 10$	33.091	23.695	32.759	34.503	30.356	**37.043**	35.100
	$\sigma = 50$	22.418	23.266	22.306	**24.667**	20.441	24.496	23.337
Mushroom	$\sigma = 10$	34.496	25.814	33.280	36.538	32.210	36.944	**37.703**
	$\sigma = 50$	22.549	22.946	22.855	22.928	21.779	**25.108**	22.837
Lena	$\sigma = 10$	33.269	23.660	30.957	34.885	31.131	33.881	**35.301**
	$\sigma = 50$	22.507	23.421	21.698	23.499	19.599	**24.821**	23.166

Real-world data experiments: Results

		Unstructured	Kronecker-structured Dictionary			Low-separation-rank Dictionary		
	Noise	K-SVD	SeDiL	BCD	TeFDiL	BCD	STARK	TeFDiL
House	$\sigma = 10$	35.670	23.189	31.609	36.295	32.295	33.400	**37.127**
	$\sigma = 50$	25.468	23.692	24.830	**27.541**	21.613	27.394	26.590
Castle	$\sigma = 10$	33.091	23.695	32.759	34.503	30.356	**37.043**	35.100
	$\sigma = 50$	22.418	23.266	22.306	**24.667**	20.441	24.496	23.337
Mushroom	$\sigma = 10$	34.496	25.814	33.280	36.538	32.210	36.944	**37.703**
	$\sigma = 50$	22.549	22.946	22.855	22.928	21.779	**25.108**	22.837
Lena	$\sigma = 10$	33.269	23.660	30.957	34.885	31.131	33.881	**35.301**
	$\sigma = 50$	22.507	23.421	21.698	23.499	19.599	**24.821**	23.166

	Noise	$R = 1$	$R = 4$	$R = 8$	$R = 16$	$R = 32$	K-SVD
Mushroom	$\sigma = 10$	36.538	36.754	37.417	37.491	**37.702**	34.496
	$\sigma = 50$	**22.928**	22.835	22.838	22.842	22.837	22.549
Number of parameters		265	1060	2120	4240	8480	147456

Real-world data experiments: Results

	Noise	Unstructured K-SVD	Kronecker-structured Dictionary SeDiL	BCD	TeFDiL	Low-separation-rank Dictionary BCD	STARK	TeFDiL
House	$\sigma = 10$	35.670	23.189	31.609	36.295	32.295	33.400	**37.127**
	$\sigma = 50$	25.468	23.692	24.830	**27.541**	21.613	27.394	26.590
Castle	$\sigma = 10$	33.091	23.695	32.759	34.503	30.356	**37.043**	35.100
	$\sigma = 50$	22.418	23.266	22.306	**24.667**	20.441	24.496	23.337
Mushroom	$\sigma = 10$	34.496	25.814	33.280	36.538	32.210	36.944	**37.703**
	$\sigma = 50$	22.549	22.946	22.855	22.928	21.779	**25.108**	22.837
Lena	$\sigma = 10$	33.269	23.660	30.957	34.885	31.131	33.881	**35.301**
	$\sigma = 50$	22.507	23.421	21.698	23.499	19.599	**24.821**	23.166

	Noise	$R = 1$	$R = 4$	$R = 8$	$R = 16$	$R = 32$	K-SVD
Mushroom	$\sigma = 10$	36.538	36.754	37.417	37.491	**37.702**	34.496
	$\sigma = 50$	**22.928**	22.835	22.838	22.842	22.837	22.549
Number of parameters		265	1060	2120	4240	8480	147456

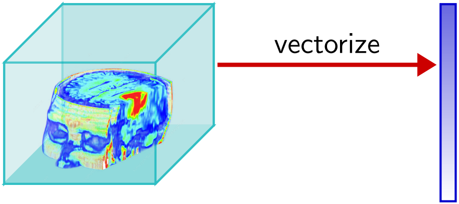
0.18% of K-SVD

Outline

Summary of the talk

Tensor data can be massively high-dimensional, rendering the old (tensor-agnostic) regularizers highly suboptimal

Tensor $\underline{\mathbf{B}} \in \mathbb{R}^{256 \times 256 \times 64}$



Vector dimensions: $p = 4,194,304$

10% sparsity $\Rightarrow n \geq 419,430$

- High-dimensional tensor regression
  - Contributions: Low-rank and sparse Tucker model for regression parameters; provable recovery using a linearly convergent algorithm; sample complexity analysis
- High-dimensional tensor dictionary learning
  - Contributions: Tucker-based models for dictionary learning; lower and upper bounds on sample complexity; algorithms along with characterization of their sample complexities

Complete list of relevant publications and code: www.inspirelab.us/publications