Nonparametric Reduced-Rank Regression with Tensor-Structured Response

Guillaume Rabusseau Hachem Kadri François Denis

Journée "Tenseurs et estimation de matrices de covariance"

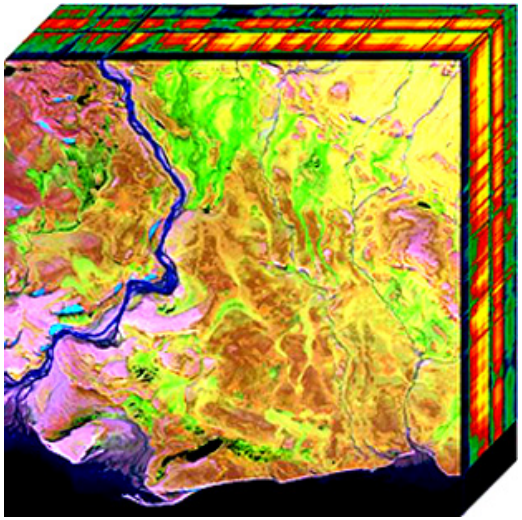
November 27, 2015

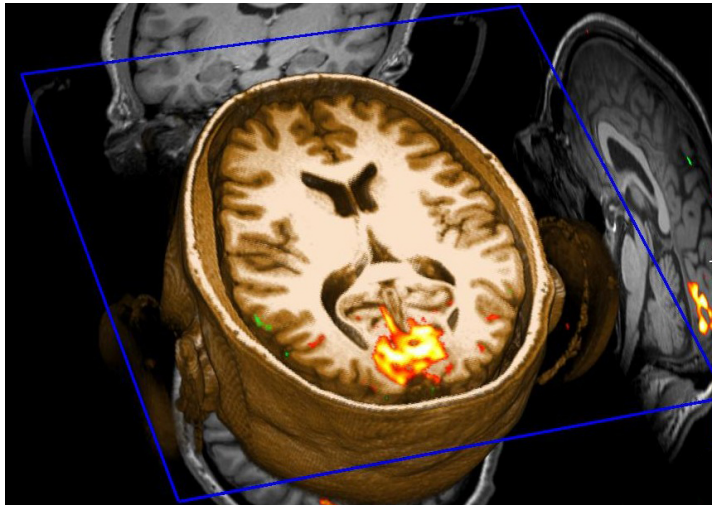
Overview

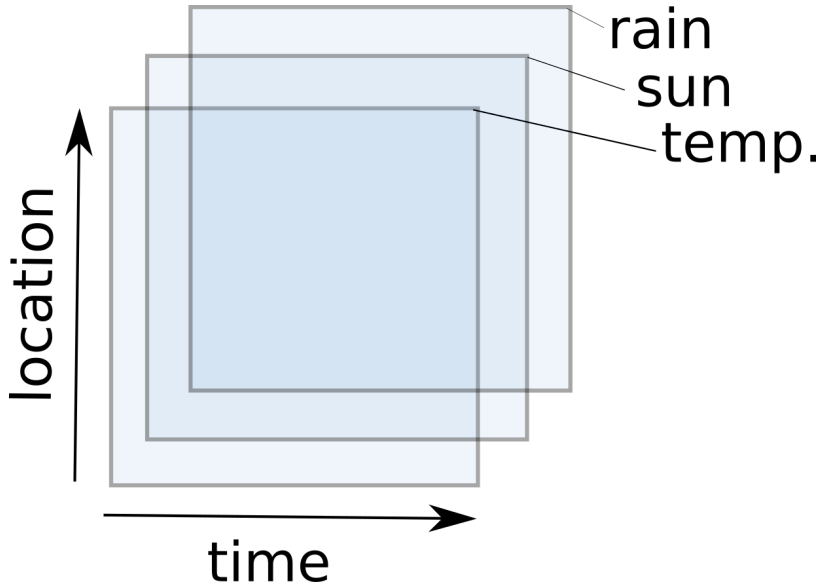
Outline

Introduction

- Data with tensor structure: EEG, hyperspectral images, videos, ...







Introduction

- Data with tensor structure: EEG, hyperspectral images, videos, ...
- Tensor learning
  - Tensor decomposition and latent variable models
  - Source separation
  - Tensor completion, ...

Introduction

- Data with tensor structure: EEG, hyperspectral images, videos, ...
- Tensor learning
  - Tensor decomposition and latent variable models
  - Source separation
  - Tensor completion, ...
  - Regression with tensor input
    - ★ Multilinear model:

$$f(\mathcal{X}) = \langle \mathcal{W}, \mathcal{X} \rangle$$

    - ★ Multi-View/Factorisation Machines:

$$f(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) = \langle \mathcal{W}, \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \mathbf{x}^{(3)} \rangle$$

    - ★ Learning Tensors in RKHS with Multilinear Spectral Penalties (Signoretto et al., 2013)

Introduction

- Data with tensor structure: EEG, hyperspectral images, videos, ...
- Tensor learning
  - Tensor decomposition and latent variable models
  - Source separation
  - Tensor completion, ...
  - Regression with tensor input
    - ★ Multilinear model:

$$f(\mathcal{X}) = \langle \mathcal{W}, \mathcal{X} \rangle$$

    - ★ Multi-View/Factorisation Machines:

$$f(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) = \langle \mathcal{W}, \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \mathbf{x}^{(3)} \rangle$$

    - ★ Learning Tensors in RKHS with Multilinear Spectral Penalties (Signoretto et al., 2013)
  - Regression with tensor output
    - ★ Multilinear models (multitask and spatio-temporal forecasting)
    - ★ Nonlinear / Nonparametric?

Outline

Tensors

- **Tensor** $\mathcal{T} \in \mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_p} \simeq$ Multi-array $(\mathcal{T}_{i_1 \ldots i_k}) \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.
  - $p = 3$: higher order tensor $\mathcal{T}_{i_1 i_2 i_3} \in \mathbb{R}$ for $i_1 \in [d_1], i_2 \in [d_2], i_3 \in [d_3]$.

Tensors

- **Tensor** $\mathcal{T} \in \mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_p} \simeq$ Multi-array $(\mathcal{T}_{i_1 \ldots i_k}) \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.
  - $p = 3$: higher order tensor $\mathcal{T}_{i_1 i_2 i_3} \in \mathbb{R}$ for $i_1 \in [d_1], i_2 \in [d_2], i_3 \in [d_3]$.
- The **CP-rank** of $\mathcal{T}$ is the smallest integer R such that

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{v}_r^1 \otimes \cdots \otimes \mathbf{v}_r^p \qquad \text{with } \mathbf{v}_r^1 \in \mathbb{R}^{d_1}, \cdots, \mathbf{v}_r^p \in \mathbb{R}^{d_p} \ .$$

  - We write $\mathcal{T} = [\![\mathbf{V}^1, \cdots, \mathbf{V}^p]\!]$ where $\mathbf{V}^i = (\mathbf{v}_1^i \cdots \mathbf{v}_R^i) \in \mathbb{R}^{d_i \times R}$.

Tensors

- **Tensor** $\mathcal{T} \in \mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_p} \simeq$ Multi-array $(\mathcal{T}_{i_1 \ldots i_k}) \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.
  - $p = 3$: higher order tensor $\mathcal{T}_{i_1 i_2 i_3} \in \mathbb{R}$ for $i_1 \in [d_1], i_2 \in [d_2], i_3 \in [d_3]$.
- The **CP-rank** of $\mathcal{T}$ is the smallest integer R such that

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{v}_r^1 \otimes \cdots \otimes \mathbf{v}_r^p \qquad \text{with } \mathbf{v}_r^1 \in \mathbb{R}^{d_1}, \cdots, \mathbf{v}_r^p \in \mathbb{R}^{d_p} \ .$$

  - We write $\mathcal{T} = [\![\mathbf{V}^1, \cdots, \mathbf{V}^p]\!]$ where $\mathbf{V}^i = (\mathbf{v}_1^i \cdots \mathbf{v}_R^i) \in \mathbb{R}^{d_i \times R}$.
- **Multiplication** between tensors. Let $\mathcal{A} \in R^{m_1 \times m_2 \times n_1 \times n_2}$ and $\mathcal{B} \in R^{n_1 \times n_2 \times p_1 \times p_2}$, we define $\mathcal{AB} \in \mathbb{R}^{m_1 \times m_2 \times p_1 \times p_2}$ by

$$(\mathcal{AB})_{i_1 i_2 k_1 k_2} = \sum_{j_1 j_2} \mathcal{A}_{i_1 i_2 j_1 j_2} \mathcal{B}_{j_1 j_2 k_1 k_2}$$

$\Rightarrow$ Composition of multilinear maps...

Multivariate Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^p$ from samples $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$ where $\mathbf{y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ $\qquad (\mathbf{W} \in \mathbb{R}^{d \times p})$

Multivariate Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^p$ from samples $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$ where $\mathbf{y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ $\qquad (\mathbf{W} \in \mathbb{R}^{d \times p})$
- Ordinary Least Squares

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d \times p}}{\arg\min} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \qquad (\mathbf{X} \in \mathbb{R}^{N \times d}, \mathbf{Y} \in \mathbb{R}^{N \times p})$$

$\Rightarrow$ Equivalent to perform p independent linear regressions!
How can we capture linear dependencies in the output?

Multivariate Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^p$ from samples $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$ where $\mathbf{y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ $\qquad (\mathbf{W} \in \mathbb{R}^{d \times p})$
- Ordinary Least Squares

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d \times p}}{\arg\min} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \qquad (\mathbf{X} \in \mathbb{R}^{N \times d}, \mathbf{Y} \in \mathbb{R}^{N \times p})$$

  $\Rightarrow$ Equivalent to perform p independent linear regressions!
  How can we capture linear dependencies in the output?

- Reduced Rank Regression (Izenman, 1975)

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d \times p}}{\arg\min} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \quad \text{s.t. } \mathit{rank}(\mathbf{W}) \leq R$$

Tensor-valued Regression

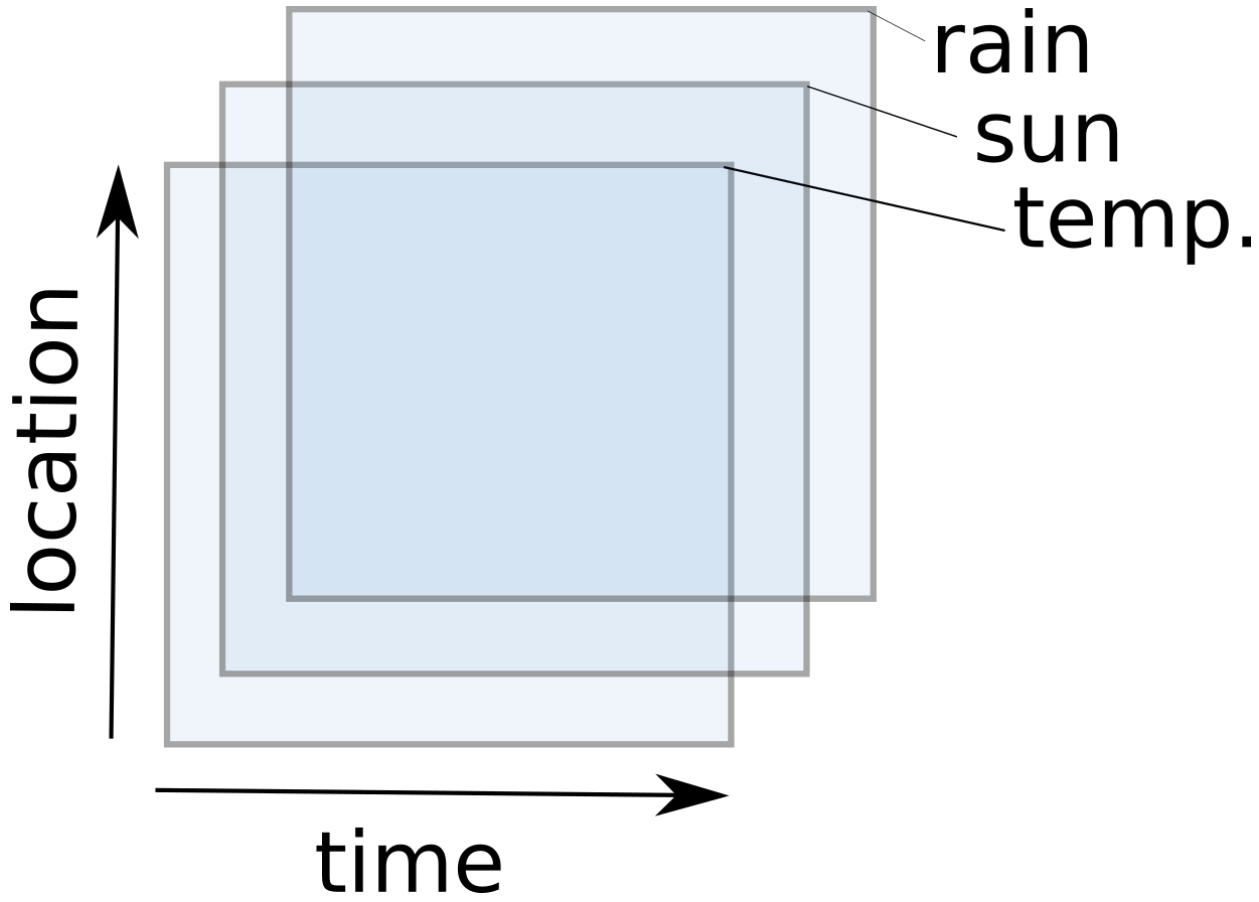
Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{Y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

Tensor-valued Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$ where $\mathbf{y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Spatio-Temporal Forecasting (Bahadori et al., 2014)

$$f(\mathbf{x}) \in \mathbb{R}^{(\text{Times}) \times (\text{Locations}) \times (\text{Variables})}$$



Tensor-valued Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{Y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Multilinear Multitask Learning (Romera-Paredes et al., 2013)

$$f(\mathbf{x}) \in \mathbb{R}^{(\text{Restaurant Critics}) \times (\text{Evaluation Criteria})}$$

Rest. 1	Critic 1	Critic 2	Critic 3
host	5	3	6
food	7	8	6.5
price	5	6.5	4

Rest. 2	Critic 1	Critic 2	Critic 3
host	7	8	6
food	8.5	9	9
price	8	9.5	7

...

Tensor-valued Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{Y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{x}\mathcal{W}$ $\qquad (\mathcal{W} \in \mathbb{R}^{d \times d_1 \times \cdots \times d_p})$

Tensor-valued Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{Y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{x}\mathcal{W}$ $\qquad (\mathcal{W} \in \mathbb{R}^{d \times d_1 \times \cdots \times d_p})$
- Vectorisation of the outputs $\to$ Reduced Rank Regression

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d \times d_1 d_2 \cdots d_p}}{\arg\min} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \quad \text{s.t. } \mathit{rank}(\mathbf{W}) \leq R$$

where $\mathbf{Y}_{n,:} = \mathrm{vec}(\mathcal{Y}^{(n)})^\top$

$\Rightarrow$ Tensor structure of the output is lost!

Tensor-valued Regression

Learn $f : \mathbb{R}^d \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ from $\{(\mathbf{x}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{Y}^{(n)} \simeq f(\mathbf{x}^{(n)})$.

- Linear model: $f(\mathbf{x}) = \mathbf{x}\mathcal{W}$ $\qquad (\mathcal{W} \in \mathbb{R}^{d \times d_1 \times \cdots \times d_p})$
- Vectorisation of the outputs $\to$ Reduced Rank Regression

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d \times d_1 d_2 \cdots d_p}}{\arg\min} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \quad \text{s.t. } \mathit{rank}(\mathbf{W}) \leq R$$

where $\mathbf{Y}_{n,:} = \mathrm{vec}(\mathcal{Y}^{(n)})^\top$

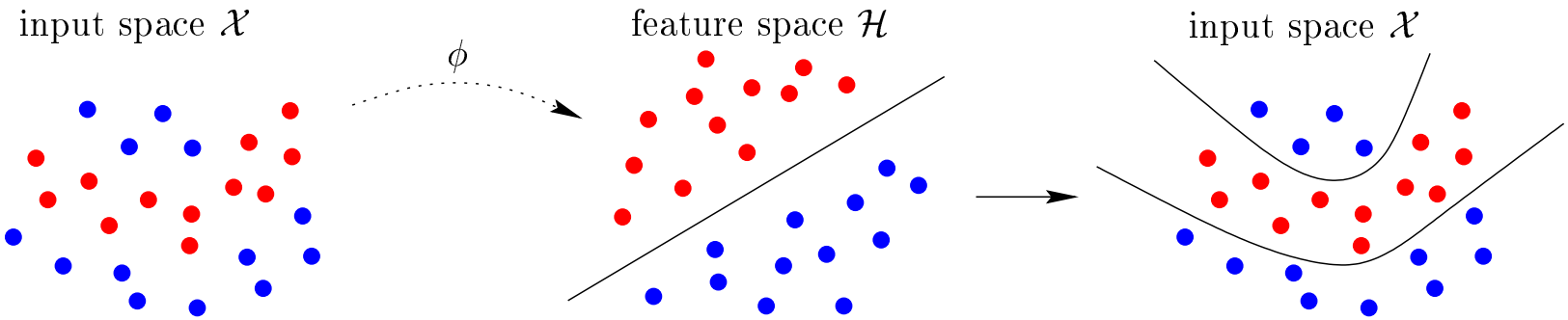
$\Rightarrow$ Tensor structure of the output is lost!

- **Reduced Rank Regression for Tensor Structured Response**

$$\hat{\mathcal{W}} = \underset{\mathcal{W} \in \mathbb{R}^{d \times d_1 \times d_2 \cdots \times d_p}}{\arg\min} \|\mathbf{X}\mathcal{W} - \mathcal{Y}\|_F^2 \quad \text{s.t. } \mathrm{rank}(\mathcal{W}) \leq R$$

Outline

Scalar-valued Kernels

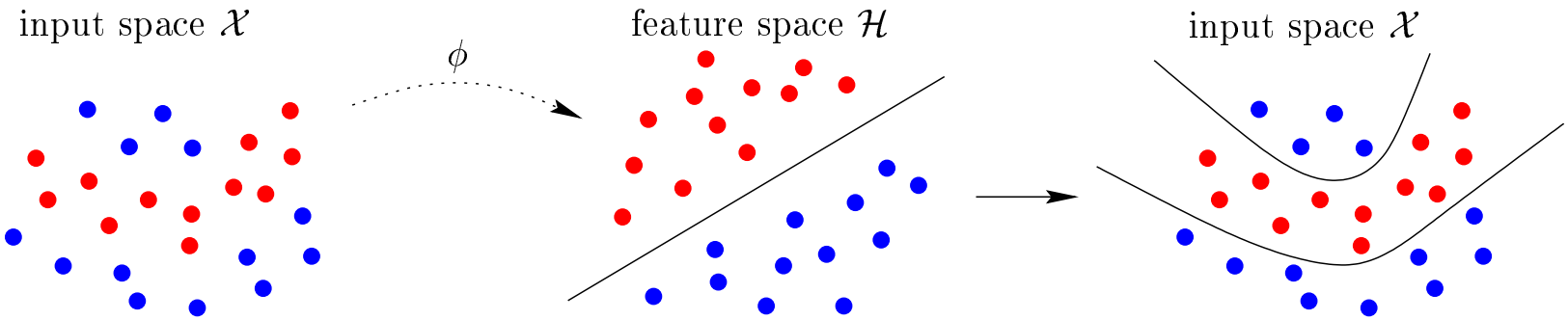


no separating hyperplane one possible separating hyperplane separating surface

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}$.

- **Scalar-valued** kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ positive semidefinite.
- Learn f in the RKHS $\mathcal{H}_k = \overline{span\{k(\mathbf{z}, \cdot) \mid \mathbf{z} \in \mathcal{X}\}}$

Scalar-valued Kernels



no separating hyperplane | one possible separating hyperplane | separating surface

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}$.

- **Scalar-valued** kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ positive semidefinite.
- Learn f in the RKHS $\mathcal{H}_k = \overline{span\{k(\mathbf{z}, \cdot) \mid \mathbf{z} \in \mathcal{X}\}}$
- Representer theorem

$$\hat{f} = \arg\min_{f \in \mathcal{H}_k} \sum_{n=1}^{N} (f(\mathbf{x}^{(n)}) - y^{(n)})^2 + \gamma \|f\|_{\mathcal{H}_k}^2 \Longrightarrow \hat{f} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}^{(n)}, \cdot)$$

for some $\alpha_1, \cdots, \alpha_N \in \mathbb{R}$.

Matrix-valued Kernels (Micchelli and Pontil, 2005)

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^p$.

- **Matrix-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{p \times p}$ positive semidefinite.
- Learn f in the RKHS $\mathcal{H}_K = \overline{span\{K(\mathbf{z}, \cdot)\mathbf{c} \mid \mathbf{z} \in \mathcal{X}, \mathbf{c} \in \mathcal{Y}\}}$

Matrix-valued Kernels (Micchelli and Pontil, 2005)

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^p$.

- **Matrix-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{p \times p}$ positive semidefinite.
- Learn f in the RKHS $\mathcal{H}_K = \overline{span\{K(\mathbf{z}, \cdot)\mathbf{c} \mid \mathbf{z} \in \mathcal{X}, \mathbf{c} \in \mathcal{Y}\}}$
- Representer theorem

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \sum_{n=1}^{N} (f(\mathbf{x}^{(n)}) - \mathbf{y}^{(n)})^2 + \gamma \|f\|^2_{\mathcal{H}_K} \Longrightarrow \hat{f} = \sum_{n=1}^{N} K(\mathbf{x}^{(n)}, \cdot)\mathbf{c}^{(n)}$$

for some $\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(N)} \in \mathbb{R}^p$.

Matrix-valued Kernels (Micchelli and Pontil, 2005)

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^p$.

- **Matrix-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{p \times p}$ positive semidefinite.
- Learn f in the RKHS $\mathcal{H}_K = \overline{span\{K(\mathbf{z}, \cdot)\mathbf{c} \mid \mathbf{z} \in \mathcal{X}, \mathbf{c} \in \mathcal{Y}\}}$
- Representer theorem

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \sum_{n=1}^{N} (f(\mathbf{x}^{(n)}) - \mathbf{y}^{(n)})^2 + \gamma \|f\|_{\mathcal{H}_K}^2 \Longrightarrow \hat{f} = \sum_{n=1}^{N} K(\mathbf{x}^{(n)}, \cdot)\mathbf{c}^{(n)}$$

  for some $\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(N)} \in \mathbb{R}^p$.

- **Separable** kernel $K(\cdot, \cdot) = k(\cdot, \cdot)\mathbf{T}$ for some scalar kernel k and some positive semidefinite $\mathbf{T} \in \mathbb{R}^{p \times p}$

$$\arg\min_{\mathbf{C} \in \mathbb{R}^{N \times p}} \|\mathbf{KCT} - \mathbf{Y}\|_F^2 + \gamma \langle \mathbf{KCT}, \mathbf{C} \rangle$$

Tensor-valued Kernels

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^{d_1 \times \cdots \times d_p}$.

- **Tensor-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{d_1 \times \cdots \times d_p \times d_1 \times \cdots \times d_p}$ positive semidefinite.

Tensor-valued Kernels

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^{d_1 \times \cdots \times d_p}$.

- **Tensor-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{d_1 \times \cdots \times d_p \times d_1 \times \cdots \times d_p}$ positive semidefinite.
- Representer theorem

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \sum_{n=1}^{N} (f(\mathbf{x}^{(n)}) - \mathcal{Y}^{(n)})^2 + \gamma \|f\|^2_{\mathcal{H}_K} \Longrightarrow \hat{f} = \sum_{n=1}^{N} K(\mathbf{x}^{(n)}, \cdot)\mathcal{C}^{(n)}$$

for some $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(N)} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.

Tensor-valued Kernels

Nonparametric learning of $f : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^{d_1 \times \cdots \times d_p}$.

- **Tensor-valued kernel** $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \simeq \mathbb{R}^{d_1 \times \cdots \times d_p \times d_1 \times \cdots \times d_p}$ positive semidefinite.
- Representer theorem

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} \sum_{n=1}^{N} (f(\mathbf{x}^{(n)}) - \mathcal{Y}^{(n)})^2 + \gamma \|f\|_{\mathcal{H}_K}^2 \Longrightarrow \hat{f} = \sum_{n=1}^{N} K(\mathbf{x}^{(n)}, \cdot)\mathcal{C}^{(n)}$$

  for some $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(N)} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.

- **Separable** kernel $K(\cdot, \cdot) = k(\cdot, \cdot)\mathcal{T}$ for some scalar kernel k and some positive semidefinite $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_p \times d_1 \times \cdots \times d_p}$

$$\operatorname*{arg\,min}_{\mathcal{C} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}} \|\mathbf{K}\mathcal{C}\mathcal{T} - \mathcal{Y}\|_F^2 + \gamma \langle \mathbf{K}\mathcal{C}\mathcal{T}, \mathcal{C} \rangle$$

Nonparmateric Tensor-valued Reduced Rank Regression

Separable kernel $K(\cdot,\cdot) = k(\cdot,\cdot)\mathfrak{T}$

$$\underset{\mathcal{C}\in\mathbb{R}^{N\times d_1\times\cdots\times d_p}}{\arg\min} \|\mathbf{K}\mathcal{C}\mathfrak{T} - \mathcal{Y}\|_F^2 + \gamma\langle\mathbf{K}\mathcal{C}\mathfrak{T}, \mathcal{C}\rangle$$

$\Rightarrow$ Does not take the tensor structure of the output into account!

We add a low CP-rank constraint on the tensor $\mathcal{C}\mathfrak{T}$:

$$\underset{\mathcal{C}\in\mathbb{R}^{N\times d_1\times\cdots\times d_p}}{\arg\min} \|\mathbf{K}\mathcal{C}\mathfrak{T} - \mathcal{Y}\|_F^2 + \gamma\langle\mathbf{K}\mathcal{C}\mathfrak{T}, \mathcal{C}\rangle \quad \text{s.t. } \mathit{rank}(\mathcal{C}\mathfrak{T}) \leq R$$

$\Rightarrow$ Back to the linear case when $k(\cdot,\cdot)$ is the linear kernel and $\mathfrak{T} = \mathfrak{I}$

Learning Algorithms

- `TVK-RRR-I`: Special case $\mathcal{T} = \mathcal{I}$.

$$\underset{\mathcal{C} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}}{\arg\min} \quad \|\mathbf{K}\mathcal{C} - \mathcal{Y}\|_F^2 + \gamma \langle \mathbf{K}\mathcal{C}, \mathcal{C} \rangle \qquad \text{s.t. } \mathit{rank}(\mathcal{C}) \leq R$$

$\Rightarrow$ Alternating Least Squares ($\mathcal{C} = [\![\mathbf{U}^0, \mathbf{U}^1, \cdots, \mathbf{U}^p]\!]$)

Learning Algorithms

Algorithm 1 TVK-RRR-I

Require: Gram matrix $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$, output tensor $\mathcal{Y} \in \mathbb{R}^N \otimes \mathcal{Y}$, regularization parameter R.

Ensure: $\mathcal{C} \in \mathbb{R}^N \otimes \mathcal{Y}$

Randomly initialize $\mathbf{U}^i \in R^{d_i \times R}$ for $1 \leq i \leq p+1$

repeat

$\mathbf{U}^1 \leftarrow (\mathbf{K}(\mathbf{K}+\gamma\mathbf{I}))^+ (\mathbf{K}\mathcal{Y})_{(1)} \mathbf{U}_{\odot^1} (\mathbf{U}_{\odot^1}^\top \mathbf{U}_{\odot^1})^+$

for $i = 2, \cdots, p+1$ **do**

$\widetilde{\mathbf{U}} \leftarrow (\mathbf{K}(\mathbf{K}+\gamma\mathbf{I})\mathbf{U}^1, \mathbf{U}^2, \cdots, \mathbf{U}^{p+1})$

$\mathbf{U}^i \leftarrow (\mathbf{K}\mathcal{Y})_{(i)} \mathbf{U}_{\odot^i} (\widetilde{\mathbf{U}}_{\odot^i}^\top \mathbf{U}_{\odot^i})^+$

end for

$\mathcal{C} \leftarrow [\![\mathbf{U}^1, \cdots, \mathbf{U}^{p+1}]\!]$

until convergence of $\mathcal{C}$

Learning Algorithms

- `TVK-RRR-T`: Jointly learn $\mathcal{C}$ and $\mathcal{T}$.

$$\underset{\substack{\mathcal{C}\in\mathbb{R}^{N\times d_1\times\cdots\times d_p},\\ \mathcal{T}\in\mathbb{R}^{d_1\times\cdots\times d_p\times d_1\times\cdots\times d_p},}}{\arg\min} \|\mathbf{K}\mathcal{C}\mathcal{T}-\mathcal{Y}\|_F^2+\gamma\langle\mathbf{K}\mathcal{C}\mathcal{T},\mathcal{C}\rangle \quad \text{s.t.} \quad \begin{array}{c} rank(\mathcal{T})\leq R,\\ \mathcal{T} \text{ is } p.s.d.\end{array}$$

$\Rightarrow$ Alternating minimisation ($\mathcal{C}$ and $\mathcal{T}=[\![\mathbf{V}^1,\cdots,\mathbf{V}^p,\mathbf{V}^1,\cdots,\mathbf{V}^p]\!]$) using gradient descent

Learning Algorithms

Algorithm 2 TVK-RRR-T

Require: Gram matrix $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$, output tensor $\mathcal{Y} \in \mathbb{R}^N \otimes \mathcal{Y}$, regularization parameter R.

Ensure: $\mathcal{C} \in \mathbb{R}^N \otimes \mathcal{Y}$, $\mathcal{T} \in \mathcal{Y} \otimes \mathcal{Y}$ a p.s.d. tensor.

Randomly initialize $\mathbf{V}^i \in R^{d_i \times R}$ for $1 \leq i \leq p$

repeat

$\mathcal{T} \leftarrow [\![\mathbf{V}^1, \cdots, \mathbf{V}^p, \mathbf{V}^1, \cdots, \mathbf{V}^p]\!]$

$\mathcal{C} \leftarrow$ solution of the Sylvester equation $\mathbf{K}\mathcal{C}\mathcal{T} + \gamma\mathcal{C} - \mathcal{Y} = \mathbf{0}$

for $i = 1, \cdots, p$ **do**

$\mathbf{V}^i \leftarrow \arg\min_{\mathbf{V}^i} \frac{1}{2}\|\mathbf{K}\mathcal{C}\mathcal{T} - \mathcal{Y}\|^2 + \frac{\gamma}{2}\langle \mathbf{K}\mathcal{C}\mathcal{T}, \mathcal{C}\rangle$ (solved by gradient descent).

$\mathcal{T} \leftarrow [\![\mathbf{V}^1, \cdots, \mathbf{V}^p, \mathbf{V}^1, \cdots, \mathbf{V}^p]\!]$

end for

until convergence of $\mathcal{C}$ and $\mathcal{T}$

Outline

Synthetic Data

Generate data from the relation

$$\mathcal{Y}^{(n)} = f(\mathbf{x}^{(n)}) + \boldsymbol{\xi}$$

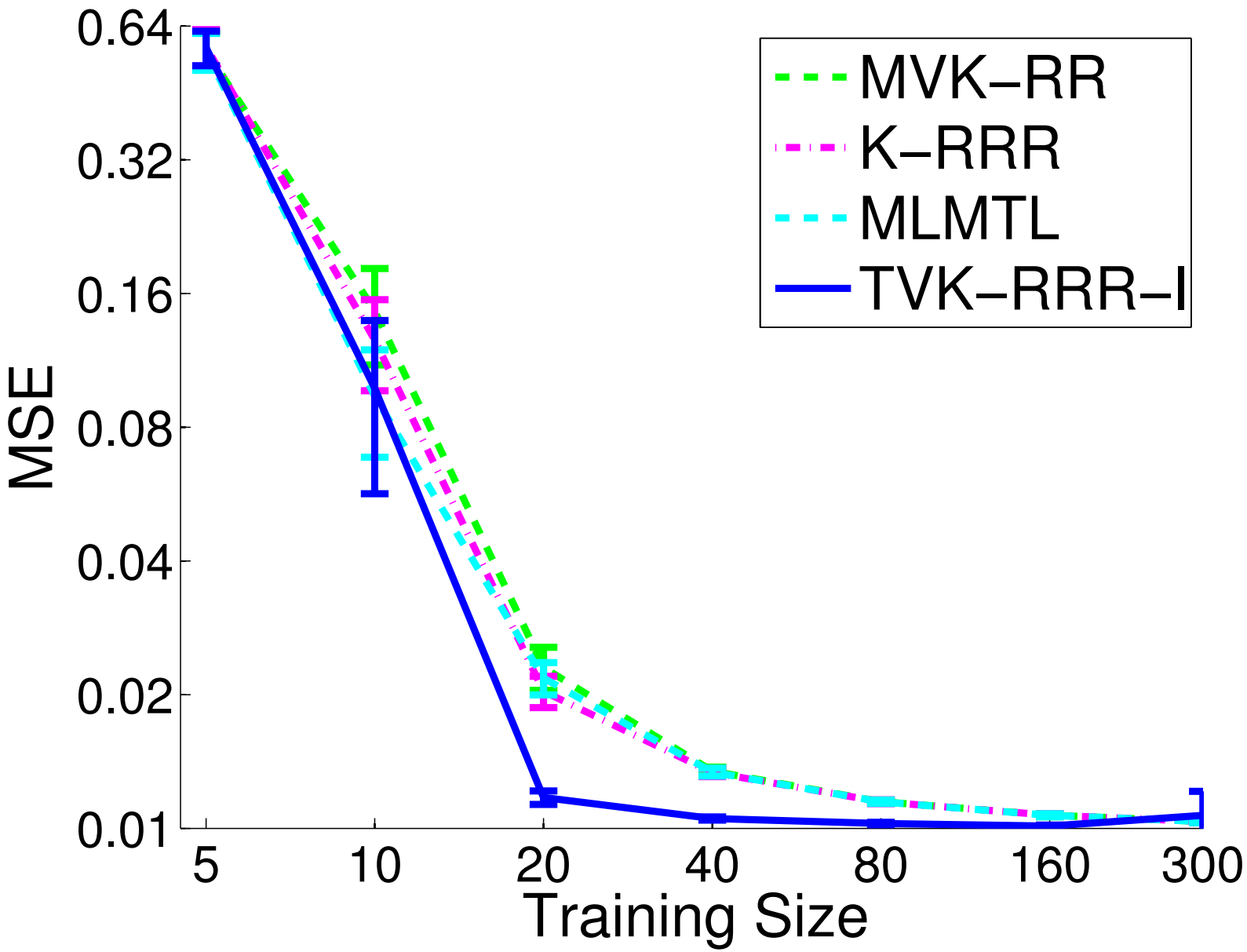
where

$$f : \mathbb{R}^{10} \to \mathbb{R}^{5\times5\times5}$$

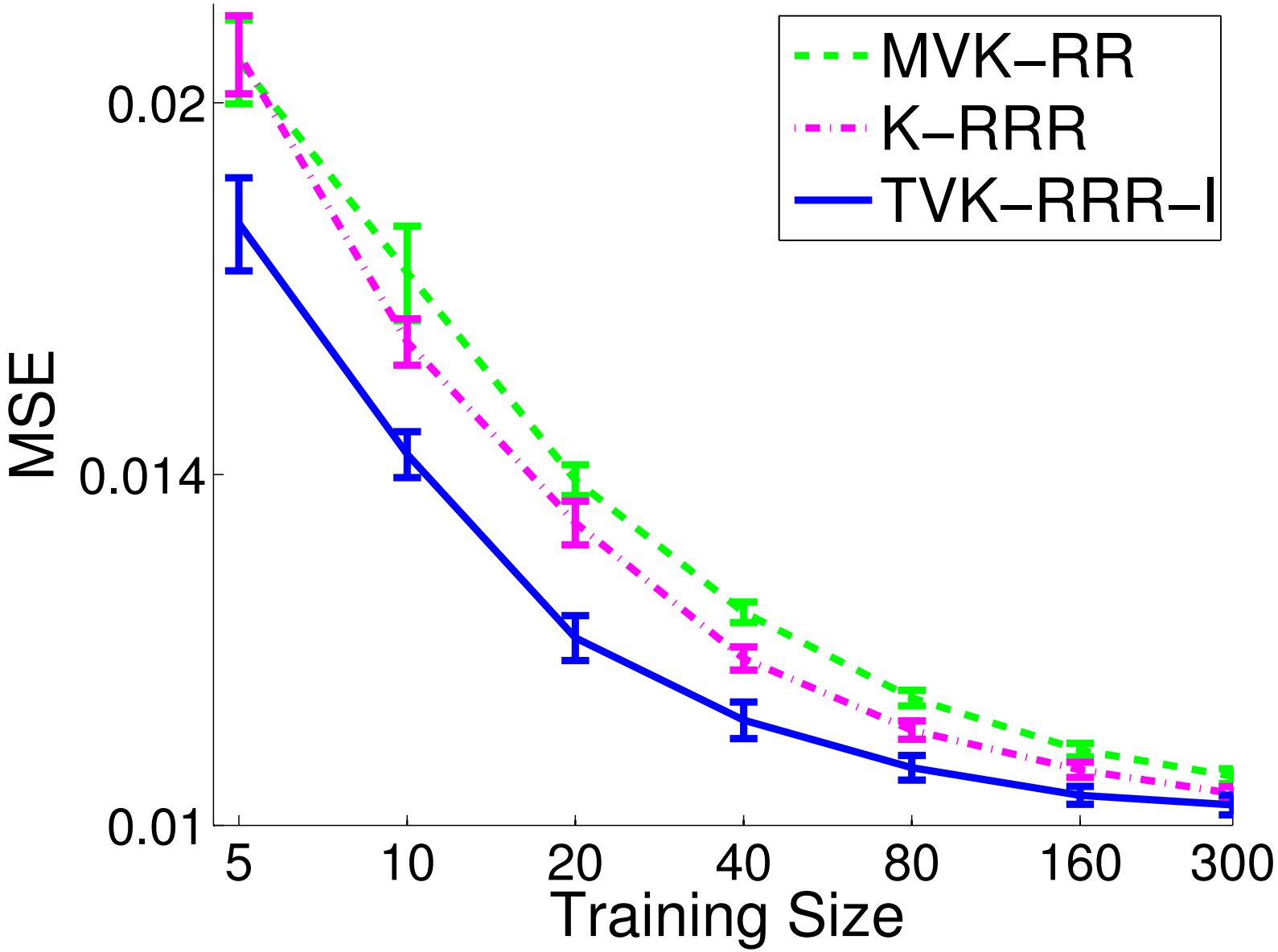
$$\mathbf{x} \mapsto \sum_{i=1}^{100} k(\mathbf{b}_i, \mathbf{x})\mathcal{T}\mathcal{C}^{(i)}$$

- Combination of 100 basis functions $k(\mathbf{b}_i, \cdot)\mathcal{T}\mathcal{C}^{(i)}$ (drawn at random)
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a scalar valued kernel
- $\mathcal{C} \in \mathbb{R}^{100\times5\times5\times5}$ and $\mathcal{T} \in \mathbb{R}^{5\times5\times5\times5\times5\times5}$ are (low-rank) tensors drawn at random.
- $\mathbf{x}^{(n)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- $\boldsymbol{\xi}_{i_1 i_2 i_3} \sim \mathcal{N}(0, 0.1)$

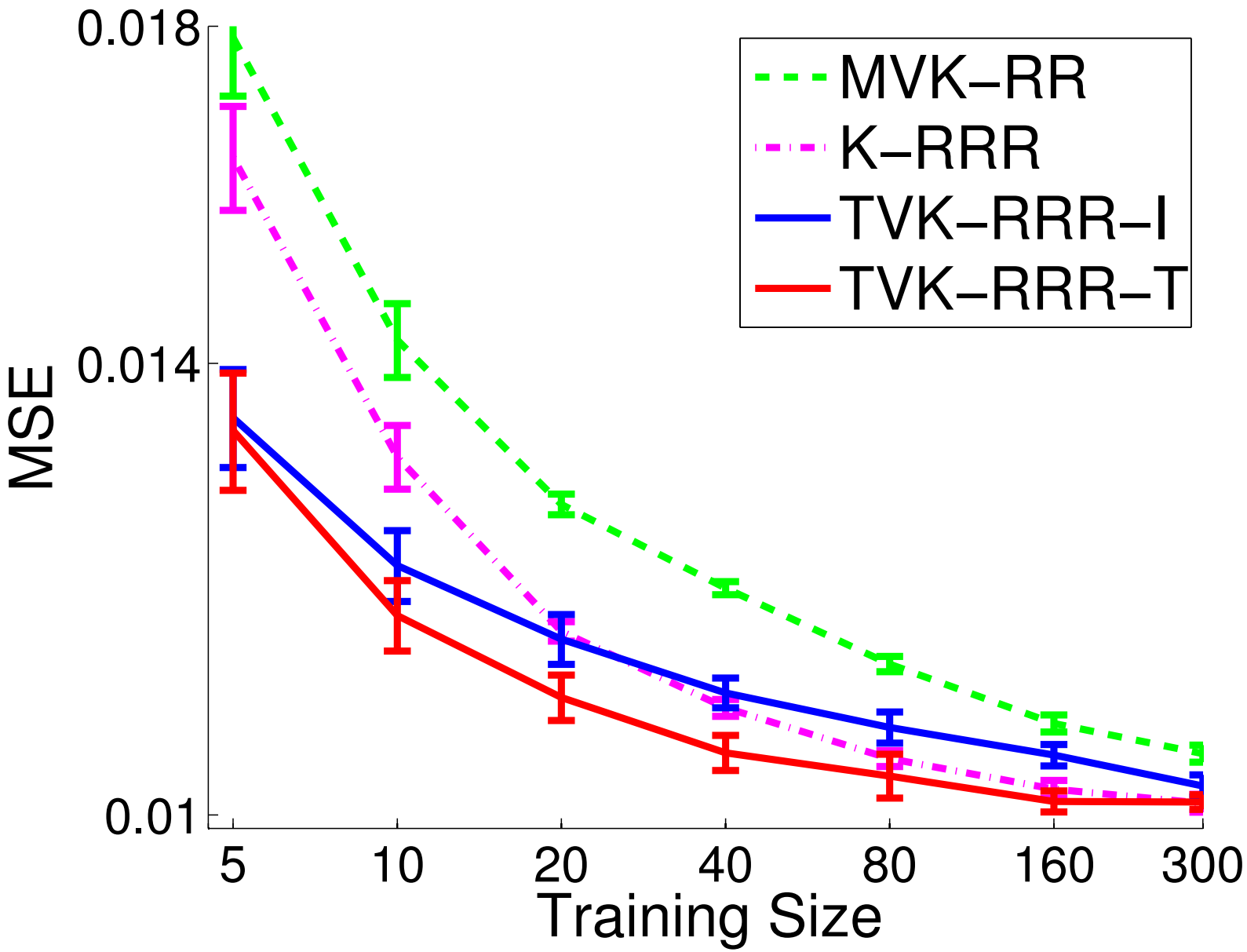
Linear kernel, $\mathcal{T} = \mathcal{I}$, $rank(\mathcal{C}) = 10$



RBF kernel, $\mathcal{T} = \mathcal{I}$, $rank(\mathcal{C}) = 10$



RBF kernel, $rank(\mathcal{T}) = 5$, $rank(\mathcal{C}) = 100$



Real Data: Meteo Forecasting

- Data from the meteorological office of the UK[1]
- Monthly average measurements of 5 variables in 16 stations across the UK from 1960 to 2000
- Predict the 5 variables in the 16 stations from values in the 5 preceding months
- Training set and validation set size: 24 months

[1] `http://www.metoffice.gov.uk/public/weather/climate-historic/`

Real Data: Meteo Forecasting

5-to-1	MLMTL	MVK-RR-I	K-RRR	TVK-RRR-I	TVK-RRR-T
linear	0.3511 ±0.018	0.3701 ±0.024	0.3514 ±0.021	**0.3443** ±0.019	0.3446 ±0.019
RBF	-	0.3501 ±0.012	0.3439 ±0.014	0.3316 ±0.011	**0.3315** ±0.011
poly	-	0.4219 ±0.019	0.4112 ±0.017	**0.4032** ±0.019	0.4033 ±0.018
time (s)	14.470	0.028	0.029	12.682	56.464

5-to-4	MLMTL	MVK-RR-I	K-RRR	TVK-RRR-I	TVK-RRR-T
linear	0.3751 ±0.009	0.3855 ±0.008	0.3641 ±0.007	0.3589 ±0.008	**0.3587** ±0.008
RBF	-	0.3630 ±0.008	0.3595 ±0.007	0.3492 ±0.007	**0.3487** ±0.008
poly	-	0.4121 ±0.017	**0.3970** ±0.015	0.3984 ±0.017	0.3981 ±0.014
time (s)	61.497	0.043	0.038	25.151	262.284

Table : MSE (mean ± std) and average running time for the forecasting task: (top) 1 month prediction, (bottom) 4 months prediction.

Outline

Conclusion and Perspectives

- Nonlinear / Nonparametric extension of low rank tensor learning techniques
- Introduction of Tensor-valued Kernels

- Other form of low-rankness (e.g. Tucker)
- Real-world data which could benefits from nonlinear models

Conclusion and Perspectives

- Nonlinear / Nonparametric extension of low rank tensor learning techniques
- Introduction of Tensor-valued Kernels

- Other form of low-rankness (e.g. Tucker)
- Real-world data which could benefits from nonlinear models

Thank you for your attention.

Bahadori, M. T., Yu, Q. R., and Liu, Y. (2014). Fast multivariate spatio-temporal analysis via low rank tensor learning. In *NIPS*.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264.

Micchelli, C. A. and Pontil, M. (2005). On learning vector-valued functions. *Neural Computation*, 17:177–204.

Romera-Paredes, B., Aung, M. H., Bianchi-Berthouze, N., and Pontil, M. (2013). Multilinear multitask learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1444–1452.

Signoretto, M., De Lathauwer, L., and Suykens, J. K. (2013). Learning tensors in reproducing kernel hilbert spaces with multilinear spectral penalties. *arXiv preprint arXiv:1310.4977*.