Multilinear Spectral Regularization for Kernel-based Multitask Learning

Marco Signoretto
ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10
B-3001 Leuven (BELGIUM)
marco.signoretto@esat.kuleuven.be

Johan A.K. Suykens
ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10
B-3001 Leuven (BELGIUM)
johan.suykens@esat.kuleuven.be

Abstract

We outline a novel regularization approach to learn kernel-based tensor product functions. Our main problem formulation admits as special cases a number of different learning frameworks. Here we focus on implications for multilinear multitask learning. We show that the proposed formalism is instrumental to derive kernel-based extensions of existing model classes; beside allowing for nonlinear models, the methodology enables to deal with non-Euclidean features.

1 Introduction

Recent research in machine learning witnessed a renewed interest in tensors. On the one hand tensor-based data representations have been proven effective to preserve useful structures in a number of applications, see [17] and references therein. On the other hand, multilinear algebra has been leveraged to derive structured finite dimensional parametric models [14, 12]. In [13] these ideas have been generalized to reproducing kernel Hilbert spaces. The arising framework comprises existing problem formulations, such as tensor completion [9, 7, 14, 15, 9], as well as novel functional formulations. The approach is based on a class of regularizers for tensor product functions that is related to spectral regularization for operator estimation [1]. In this paper we outline the main ideas and focus on the implications for (multilinear) multitask learning.

The paper is structured as follows: in the next section we present the main functional tools drawing a parallel with their finite-dimensional counterparts. We then present the class of penalty functions of interest and introduce our main problem formulation; we conclude the section by discussing finite dimensional representation and out-of-sample evaluations. Section 3 shows how the approach can be used for kernel-based multilinear multitask learning. Finally, Section 4 presents experiments.

2 Learning Tensor-based Models in RKHSs with Multilinear Spectral Penalties

2.1 Preliminaries

For a positive integer I we denote by $[I]$ the set of integers up to and including I. We write $\times_{m=1}^{M}[I_m]$ to mean $[I_1] \times [I_2] \times \cdots \times [I_M]$, i.e., the cartesian product of such index sets, the elements of which are $M-$tuples $(i_1, i_2, \ldots, i_M)$. Before proceeding, we recall here the notion of *Reproducing kernel Hilbert space*. The reader is referred to [5] for a detailed account. Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be a Hilbert space (HS) of real-valued functions on some set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be the *reproducing kernel* of $\mathcal{H}$ if and only if [3]: *a.)* $k(\cdot, x) \in \mathcal{H}$, $\forall x \in \mathcal{X}$ *b.)* $\langle f, k(\cdot, x) \rangle = f(x)$ $\forall x \in \mathcal{X}$, $\forall f \in \mathcal{H}$ (*reproducing property*). In the following we often denote by k_x the function $k(\cdot, x) : t \mapsto k(t, x)$. A HS of functions $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ that possesses a reproducing kernel k is a Reproducing kernel Hilbert space (RKHS); we denote it by $(\mathcal{H}, \langle \cdot, \cdot \rangle, k)$.

In this paper we are concerned with supervised learning problems. The predictive model will be based upon a tensor product function, the structure of which will be accounted for in the regularization mechanism. More precisely we will learn models in tensor product reproducing kernel Hilbert spaces that we introduce next.

2.2 Tensor Product Reproducing Kernel Hilbert Spaces and Functional Unfoldings

Assume that $\big(\mathcal{H}_m, \langle\cdot,\cdot\rangle_m, k^{(m)}\big)$ is a RKHS of functions defined on a domain $\mathcal{X}_m$, for any $m \in [M]$. Let $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_M$ and for $\beta \subseteq [M]$ denote by $x^{(\beta)}$ the restriction of $x = \big(x^{(m)} \,:\, m \in [M]\big) \in \mathcal{X}$ to $\beta \subseteq [M]$, i.e., $x^{(\beta)} = \big(x^{(m)} \,:\, m \in \beta\big)$. Our interest is in the Hilbert space generated by the linear combinations of *rank-1 functions* $\otimes_{m\in\beta} f^{(m)}$, defined by:

$$\otimes_{m\in\beta} f^{(m)} : x^{(\beta)} \;\;\mapsto\;\; \prod_{m\in\beta} f^{(m)}\big(x^{(m)}\big)\;, \qquad f^{(m)} \in \mathcal{H}_m \text{ for any } m \in \beta\;. \tag{1}$$

Whenever $\beta \subset [M]$ we denote the space generated by rank-1 functions by[1] $(\boldsymbol{\mathcal{H}}_\beta, \langle\cdot,\cdot\rangle_\beta)$. When $\beta = [M]$ we write simply $(\boldsymbol{\mathcal{H}}, \langle\cdot,\cdot\rangle)$, i.e., we omit the β. In the rest of this paragraph we refer to this case, without loss of generality. General elements of $\boldsymbol{\mathcal{H}}$, i.e., linear combinations of rank-1 tensors (1), will be denoted by small bold type letters $(\boldsymbol{f}, \boldsymbol{g}, \ldots)$. The symmetric function:

$$\boldsymbol{k} : \big(x, y\big) \mapsto k^{(1)}\big(x^{(1)}, y^{(1)}\big)\, k^{(2)}\big(x^{(2)}, y^{(2)}\big) \,\cdots\, k^{(M)}\big(x^{(M)}, y^{(M)}\big) \tag{2}$$

is the reproducing kernel of $\boldsymbol{\mathcal{H}}$. Indeed, for any $x \in \mathcal{X}$, $\boldsymbol{k}_x$ is a rank-1 tensor that belongs to $\boldsymbol{\mathcal{H}}$; furthermore $\langle \boldsymbol{f}, \boldsymbol{k}_x\rangle = \boldsymbol{f}(x)$. The space $\boldsymbol{\mathcal{H}}$ is called a *Tensor Product* RKHS (TP-RKHS) and denote it by $(\boldsymbol{\mathcal{H}}, \langle\cdot,\cdot\rangle, \boldsymbol{k})$. The generic partial space will be denoted by $(\boldsymbol{\mathcal{H}}_\beta, \langle\cdot,\cdot\rangle_\beta, \boldsymbol{k}^{(\beta)})$, with obvious meaning of the symbols.

Note that a $I-$dimensional vector f can be regarded as a function $f : [I] \mapsto \mathbb{R}$. In light of this one can see that the definition of tensors given above includes, as a special case, the notion of finite dimensional tensors usually found in multilinear algebra textbooks. In fact, if we take $\beta = [M]$ and $\mathcal{X} = [I_1] \times [I_2] \times \cdots \times [I_M]$, a rank-1 function (1) boils down to the outer product of finite dimensional vectors. Correspondingly, $\boldsymbol{\mathcal{H}}$ is identified with the space of $I_1 \times I_2 \times \cdots \times I_M-$dimensional tensors. Notably if, for rank-1 tensors $\boldsymbol{f} = f^{(1)} \otimes f^{(2)} \otimes \cdots f^{(M)}$ and $\boldsymbol{g} = g^{(1)} \otimes g^{(2)} \otimes \cdots g^{(M)}$, the inner product of $\boldsymbol{\mathcal{H}}$ is taken to be[2]:

$$\langle \boldsymbol{f}, \boldsymbol{g}\rangle = \big(f^{(1)\top} g^{(1)}\big)\big(f^{(2)\top} g^{(2)}\big) \cdots \big(f^{(M)\top} g^{(M)}\big) \tag{3}$$

the corresponding kernel (2) can be seen to be $\boldsymbol{k}\big((i_1, i_2, \ldots, i_M), (i'_1, i'_2, \ldots, i'_M)\big) = \delta\big(i_1 - i'_1\big)\delta\big(i_2 - i'_2\big) \cdots \delta\big(i_M - i'_M\big)$ where $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise.

For arbitrary $\beta \subset [M]$ denote now by β^c its complement, i.e., $\beta^c := [M] \setminus \beta$. Note that for each pair of vectors $(\boldsymbol{f}, \boldsymbol{g}) \in \boldsymbol{\mathcal{H}}_\beta \times \boldsymbol{\mathcal{H}}_{\beta^c}$ there corresponds a rank-1 operator $\boldsymbol{f} \otimes \boldsymbol{g} : \boldsymbol{\mathcal{H}}_{\beta^c} \to \boldsymbol{\mathcal{H}}_\beta$ defined by $(\boldsymbol{f} \otimes \boldsymbol{g})\boldsymbol{z} := \langle \boldsymbol{g}, \boldsymbol{z}\rangle_{\beta^c} \boldsymbol{f}$. Our regularization approach for tensor product functions relies on this fundamental fact. Instrumental to our approach is the *functional unfolding operator* M_β, defined for a rank-1 function by:

$$M_\beta : \;\; \otimes_{m=1}^M f^{(m)} \;\;\mapsto\;\; \boldsymbol{f}^{(\beta)} \otimes \boldsymbol{f}^{(\beta^c)} \quad \text{with} \quad \begin{array}{rcl} \boldsymbol{f}^{(\beta)} & = & \otimes_{m\in\beta} f^{(m)} \in \boldsymbol{\mathcal{H}}_\beta, \\ \boldsymbol{f}^{(\beta^c)} & = & \otimes_{m\in\beta^c} f^{(m)} \in \boldsymbol{\mathcal{H}}_{\beta^c}\;. \end{array} \tag{4}$$

Clearly the definition extends by linearity to generic elements of $\boldsymbol{\mathcal{H}}$, i.e., linear combination of rank-1 functions. The operator M_β represents a generalization of the *matrix unfolding operator* used in the context of multilinear algebra, see [14] and references therein.

2.3 Spectral Regularization for Tensor Product Functions

The facts presented above suggests a notion of model complexity for a tensor product function $\boldsymbol{f}$ related to the image of operators attached to $\boldsymbol{f}$. Let $\mathcal{B}$ be a partition of the set of variables $[M]$; a possible approach is to penalize the $\mathcal{B}-$*multilinear rank* of $\boldsymbol{f}$ [13], defined by:

$$\operatorname{mlrank}_{\mathcal{B}}(\boldsymbol{f}) := \big(\operatorname{rank}_\beta(\boldsymbol{f}) \;:\; \beta \in \mathcal{B}\big) \qquad \text{where} \quad \operatorname{rank}_\beta(\boldsymbol{f}) := \dim\big\{(M_\beta(\boldsymbol{f}))\,\boldsymbol{z} \;:\; \boldsymbol{z} \in \boldsymbol{\mathcal{H}}_{\beta^c}\big\}\;. \tag{5}$$

[1] Throughout the paper we will use bold-face letters to denote tensor product spaces and functions.

[2] As before the definition extends by linearity to arbitrary linear combinations of rank-1 tensors.

More generally one can define *multilinear spectral penalties* (MSPs) based upon the spectral content of $M_\beta(\boldsymbol{f}),\ \beta \in \mathcal{B}$. The reader is referred to [13] for a detailed discussion, formal definitions and interesting special cases. Here we focus on a specific instance of MSP. To this end we recall that, for $p \geq 1$, the operator A is called p-*summable* if it satisfies $\sum_{n\geq 1} \sigma_n(A)^p < \infty$ where $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq 0$ are the singular values of A. For a p-summable operator the *Shatten-p* norm is defined by $\|A\|_p := (\sum_{n\geq 1} \sigma_n(A)^p)^{1/p}$. The MSP of interest is now constituted by Shatten-p norms with an upper-bound on the $\mathcal{B}-$multilinear rank:

$$\Omega_{\mathcal{B}}(\boldsymbol{f}) = \begin{cases} \sum_{q\in[Q]} \|M_{\beta_q}(\boldsymbol{f})\|_p^p, & \text{if } \operatorname{rank}_{\beta_q}(\boldsymbol{f}) \leq R_q \ \forall\, q \in [Q] \\ \infty, & \text{otherwise .} \end{cases} \tag{6}$$

A discussion on the properties of this penalty is postponed to later sections. In the following, unless stated differently, we will always assume that the MSP is given by (6).

2.4 Main Learning Problem, Finite Dimensional Representation and Out-of-sample Evaluations

Next we use the machinery introduced above to specify a class of supervised learning problems. Based upon a dataset $\mathcal{D}_N$ of N input-output training pairs:

$$\mathcal{D}_N := \left\{ \left(x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(M)}, y_n \right) \in \mathcal{X} \times \mathcal{Y} \ :\ n \in [N] \right\} \tag{7}$$

where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_M$ and $\mathcal{Y} \subseteq \mathbb{R}$, we aim at finding a predictive model $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ in a TP-RKHS $(\boldsymbol{\mathcal{H}}, \langle\cdot,\cdot\rangle, \boldsymbol{k})$. With reference to (6), fix p and the partition of $[M]$ into Q sets, $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_Q\}$. Let further $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be some loss function and $\lambda > 0$ be a a trade-off parameter. We are concerned with the following penalized empirical risk minimization problem:

$$\min_{\boldsymbol{f} \in \boldsymbol{H}_{\mathcal{B}}} \left\{ \sum_{(x,y)\in\mathcal{D}_N} l\left(y, \langle \boldsymbol{f}, \boldsymbol{k}_x \rangle\right) + \lambda\, \Omega_{\mathcal{B}}(\boldsymbol{f}) \right\} \tag{8}$$

where $\boldsymbol{H}_{\mathcal{B}} := \left\{ \boldsymbol{f} \in \boldsymbol{\mathcal{H}} \ :\ M_{\beta_q}(\boldsymbol{f}) \text{ is } p\text{-summable for any } q \in [Q] \right\}$ is, informally speaking, the set of well-behaved functions in $\boldsymbol{\mathcal{H}}$. Interestingly, different learning frameworks arise from different specifications of $\mathcal{X}$, loss l and MSP $\Omega_{\mathcal{B}}$. For instance when $\mathcal{X} = [I_1] \times [I_2] \times \cdots \times [I_M]$, $\Omega_{\mathcal{B}}$ is given by a sum of nuclear norms and $l(y, \hat{y}) = \delta(y - \hat{y})$, (8) boils down to tensor completion [9, 7, 14, 15, 9], see [13] for details. Here we focus on applications to Multilinear Multitask Learning. Before delving into this topic, however, we illustrate few properties of (8). To simplify the discussion, we will assume that $\Omega_{\mathcal{B}}$ is the MSP in (6) with $p = 1$ and that $l(y, \hat{y}) = (y - \hat{y})^2$. The approach requires to obtained a factorization of the kernel matrices associated to the sets of variables in $\mathcal{B}$. Specifically, for any $q \in [Q]$ we need to find a *factor matrix* $\boldsymbol{F}^{(q)} \in \mathbb{R}^{N \times I_q}$ such that:

$$\boldsymbol{K}^{(q)} = \boldsymbol{F}^{(q)} \boldsymbol{F}^{(q)\top} \quad \text{where} \quad \boldsymbol{K}^{(q)}_{ij} := k^{(\beta_q)}\left(x_i^{(\beta_q)}, x_j^{(\beta_q)} \right) \text{ for } (x_i, y_i), (x_j, y_j) \in \mathcal{D}_N \ . \tag{9}$$

This can be obtained, for instance, by Cholesky decomposition. It now follows from the representer theorem proved in [13] that computing a solution $\hat{\boldsymbol{f}}$ to (8) practically requires to find a finite dimensional tensor $\hat{\boldsymbol{\alpha}}$ that solves:

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_Q}} \quad & \textstyle\sum_{n\in[N]} \left(y_n - \langle \boldsymbol{z}_n, \boldsymbol{\alpha} \rangle\right)^2 + \lambda \sum_{q\in[Q]} \|M_{\{q\}}(\boldsymbol{\alpha})\|_1 \\ \text{subject to} \quad & \operatorname{rank}\left(M_{\{q\}}(\boldsymbol{\alpha})\right) \leq R_q \ \forall\, q \in [Q] \ . \end{aligned} \tag{10}$$

In this problem, $\langle\cdot,\cdot\rangle$ is the canonical inner product in (3), $M_{\{q\}}$ is the $q-$mode matrix unfolding operator (a special instance of (4), see [14]) and $\boldsymbol{z}_n$ is the rank-1 tensor given by the outer product of the nth row of the factor matrices:

$$\boldsymbol{z}_n := \boldsymbol{F}^{(1)}_{n:} \otimes \boldsymbol{F}^{(2)}_{n:} \otimes \cdots \otimes \boldsymbol{F}^{(Q)}_{n:} \ . \tag{11}$$

One can show that the evaluation of $\hat{\boldsymbol{f}}$ to a test point $x \in \mathcal{X}$ is given by:

$$x \mapsto \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{z} \rangle \ :\ \boldsymbol{z} = \left(\bar{k}^{(1)\top}(x) \boldsymbol{F}^{(1)\ddagger} \right) \otimes \left(\bar{k}^{(2)\top}(x) \boldsymbol{F}^{(2)\ddagger} \right) \otimes \cdots \otimes \left(\bar{k}^{(Q)\top}(x) \boldsymbol{F}^{(Q)\ddagger} \right) \ . \tag{12}$$

in which, for any $q \in [Q]$, $\boldsymbol{F}^{(q)\ddagger}$ is the transpose of the pseudo-inverse of $\boldsymbol{F}^{(q)}$ and $\bar{k}^{(q)}$ is the vector $\bar{k}^{(q)}(x) := \left[k^{(\beta_q)}\left(x_1^{(\beta_q)}, x^{(\beta_q)}\right),\ k^{(\beta_q)}\left(x_2^{(\beta_q)}, x^{(\beta_q)}\right), \ldots, k^{(\beta_q)}\left(x_N^{(\beta_q)}, x^{(\beta_q)}\right) \right]^\top$, see [14] for details.

3 Kernel-based Approach to Multilinear Multitask Learning

Multi-task Learning (MTL) aims at simultaneously finding multiple predictive models each of which corresponds to related learning tasks, see [2, 4, 6] and references therein. Recently [12] has proposed an extension, termed Multilinear Multi-task Learning (MLMTL), to account for multi-modal interactions between the tasks. This is a departure from classical tensor-based methods, such as [16], where the multilinear decomposition is performed directly on the input data. Importantly, the approach allows one to make predictions even in absence of training data for one or more of the tasks. Therefore it is suitable to perform *transfer learning* [11].

It was shown in [13] that the formulation in [12] is equivalent to a special instance of the penalized empirical risk minimization problem in (8). In turn the kernel-based view entailed by (8) enables for useful generalizations. The approach in [12] assumes that there are T linear regression tasks, each of which is represented by a vector $w_t \in \mathbb{R}^{I_1},\ t \in [T]$. In the case of interest $T = \prod_{m=2}^{M} I_m$ and the generic t is represented as a multi-index $(i_2, i_3, \ldots, i_M)$. One concrete example [12] corresponds to the case where learning tasks aim at modelling ratings on I_3 different aspects of I_2 restaurants based upon a vector of I_1 features. For the general case, consider a tensor $\boldsymbol{f} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and let $\kappa : [I_2] \times \cdots \times [I_M] \to [T]$ be a one-to-one mapping. We can associate the linear regression tasks with the set of vectors obtained fixing all but the first index $\{\boldsymbol{f}_{:\kappa^{-1}(t)} \in \mathbb{R}^{I_1} \ : \ t \in [T]\}$ (a.k.a. *mode-1 fibers* of $\boldsymbol{f}$). Simultaneously finding a model for all the learning tasks based upon the task-dependent datasets $\tilde{\mathcal{D}}^{(t)}$ for $t \in [T]$, can now be approached by minimizing the empirical risk :

$$J(\boldsymbol{f}) := \sum_{t \in T} \sum_{(z,y) \in \tilde{\mathcal{D}}^{(t)}} l\left(y, \boldsymbol{f}_{:\kappa^{-1}(t)}^{\top} z\right) \tag{13}$$

while imposing via a suitable penalty that $\boldsymbol{f}$ should have low multilinear rank [12]. It was shown in [13] that if one takes the dataset $\mathcal{D}_N = \{\big(z, i_2, \ldots, i_M, y\big) \in \mathcal{X} \times \mathcal{Y} : \text{there exists } t = \kappa\big(i_2, \ldots, i_M\big) \text{ such that } (z, y) \in \mathcal{D}^{(t)}\}$ then (13) can be equivalently stated as $J(\boldsymbol{f}) = \sum_{(x,y) \in \mathcal{D}_N} l\left(y, \langle \boldsymbol{f}, \boldsymbol{k}_x \rangle\right)$ in which $\boldsymbol{k}$ is the reproducing kernel:

$$\boldsymbol{k}\left(\big(z, i_2, \cdots, i_M\big), \big(z', i_2', \cdots, i_M'\big)\right) = g(z, z') \prod_{m=2}^{M} \delta\big(i_m - i_m'\big) \quad \text{where} \quad g(z, z') = z^{\top} z' \ . \tag{14}$$

This shows that, if the regularization is based on an MSP, then MLMTL can be seen as a special instance of (8). A first implication is that one can readily replace $g = z^{\top} z'$ in (14), for instance with the Gaussian RBF kernel $g(z, z') = \exp\left(-\|z - z'\|^2 / \sigma^2\right)$, to obtain nonlinear models. Moreover the kernel-based view allows one to go beyond Euclidean features: in fact z does not need to belong to the Euclidean space R^{I_1}, as in [12]; input data might consist of, e.g., (probability) distributions, graphs or dynamical systems [8].

4 Experiments

As an illustration of the aforementioned ideas we focus on the shoulder pain dataset [10], which contains video clips of the faces of people who suffer from shoulder pain. Each video is labelled frame by frame according to certain Action Units (AU) which refer to a contraction or relaxation of a determined set of muscles. As in [12] we aim at predicting the AU intensity levels of 5 different AUs for 5 different subjects based upon a vector of features per frame, thereby dealing with a matrix of 5×5 regression tasks. We test the convex algorithm proposed in [12] to solve MLMTL based on linear models (MLMTL-C) against the algorithm proposed in [13]. The latter solves (8) with MSP (6) and uses either the kernel in (14) (LIN-MLRANK-SNN), or the kernel obtained replacing g in (14) with a Gaussian RBF (RBF-MLRANK-SNN). As a baseline we additionally considered Least Squares Support Vector Machines for Regression (LS-SVR); details are found in [13]. The results in terms of Mean Square Error (MSE) are shown in Figure 1.

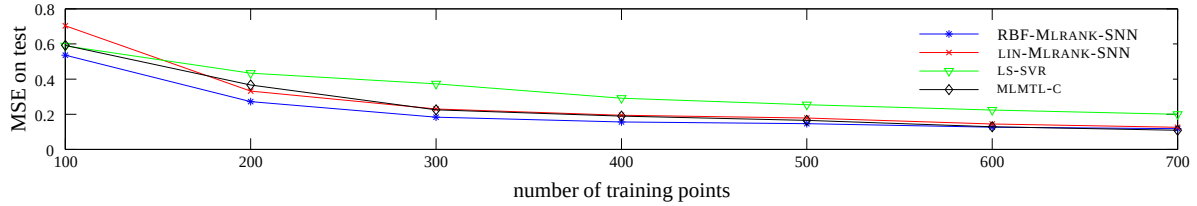

Figure 1: A comparison of different methods on the shoulder pain dataset

Acknowledgments

We thank Bernardino Romera-Paredes for kindly providing the code for MLMTL-C. Research supported by Research Council KUL: GOA/10/09 MaNet, PFV/10/002 (OPTEC); CIF1 STRT1/08/23; Flemish Government: IOF: IOF/KP/SCORES4CHEM, FWO: projects: G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC), G.0377.12 (Structured systems), G.0427.10N (EEG-fMRI), IWT: projects: SBO LeCoPro, SBO Climaqs, SBO POM, EUROSTARS SMART iMinds 2013, Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017), EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B (290923). COST: Action ICO806: IntelliCIS.

References

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[4] J. Baxter. Theoretical models of learning to learn. In *T. Mitchell and S. Thrun (Eds.), Learning*, pages 71–94. Kluwer, 1997.

[5] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[6] R. Caruana. *Learning to learn*, chapter Multitask Learning, pages 41–75. Springer, 1998.

[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):19pp, 2011.

[8] T. Gärtner. *Kernels for structured data*, volume 72 of *Machine Perception and Artificial Intelligence*. World Scientific Publishing, 2008.

[9] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208– 220, 2013.

[10] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, and I. Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. *IEEE Facial and Gesture (FG)*, pages 57–64, 2011.

[11] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

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

[13] M. Signoretto, L. De Lathauwer, and J. A. K. Suykens. Learning tensors in reproducing kernel Hilbert spaces with multilinear spectral penalties. *Internal Report 13-189, ESAT-STADIUS, K.U.Leuven (Leuven, Belgium)*, 2013.

[14] M. Signoretto, Q. Tran Dinh, L. De Lathauwer, and J. A. K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning, DOI 10.1007/s10994-013-5366-3*, 2013.

[15] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Precessing Systems 24*, 2011.

[16] M. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensor faces. In *7th European Conference on Computer Vision (ECCV), in Computer Vision – ECCV 2002, Lecture Notes in Computer Science*, volume 2350, pages 447–460, 2002, 2002.

[17] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki. Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data. *IEEE Signal Processing Magazine*, 30:137–148, 2013.