Graph Based Regularization for Multilinear Multitask Learning

M. Signoretto*[1], R. Langone†[1], M. Pontil‡[2], and J. Suykens§[1]

[1]ESAT-STADIUS, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10
B-3001 Leuven, BELGIUM
[2]Department of Computer Science, University College London, Malet Place
London WC1E 6BT, UK

August 28, 2014

Abstract

In Multi-Task Learning (MTL) several related tasks are considered simultaneously, often improving over the case where tasks are learned in isolation. We focus on MultiLinear MTL (MLMTL) problems where, in view of the inherent multi-modal structure, the tasks are naturally associated with a multi-index. We consider the cases where the relationship between a pair of tasks is encoded via a weight, and propose classes of weighted graphs suitable for MLMTL. Specifically, we model the relationship between the tasks via sparse graphs defined upon the Hamming distance between the multi-indices. The idea implies a significant reduction in the number of parameters that fully determine the graph. This is important in the common situation where task relationships are unknown and need to be learned from data. To deal with such cases we give a simple and scalable procedure based on a regularized Gauss-Seidel method. Experiments show significant improvement over alternative techniques found in the literature.

1 Introduction

Multi-Task Learning (MTL) has emerged as an important avenue of machine learning research. The aim of MTL is to jointly learn several predictive models, hereby improving the generalization error achieved by models learned in isolation [4, 6]. The approach is particularly effective when there is a small number of available observations per task, provided that the correct task relationships are leveraged [10]. In practice the task relationships are often unknown and need to be learned from data. To this end, several techniques have been proposed in the literature. An important class of such methods, in particular, relies on a graph based formalism [10, 17]. In this setting the relationship between a pair of tasks is encoded via a weight; a graph based regularization scheme is then used to enforce that the parameter vectors of related pairs should be close to each other.

In this paper we are interested in the setting where *1*) the number of tasks is very high relative to the number of observations available and *2*) in view of the inherent multi-modal structure, the tasks are naturally identified with a multi-index. Due to point 1 learning the task relationships from data is particularly problematic. In view of point 2, however, we can make sensible assumptions on the graph encoding task relationships. These assumptions imply a substantial reduction in the number of parameters to learn. The arising data-driven regularization approach entails a significant improvement in the generalization error with respect to alternative approaches, as we demonstrate in experiments.

*marco.signoretto@esat.kuleuven.be
†rocco.langone@esat.kuleuven.be
‡m.pontil@cs.ucl.ac.uk
§johan.suykens@esat.kuleuven.be

The setting of point 2 has already been considered in [21], under the name of MultiLinear Multi-Task Learning (MLMTL). The approach there relies on arranging the weight vectors in a parameter tensor; solutions with low rank matricizations [19] are then encouraged via regularization. The method proposed in this paper is substantially different. We argue that, in view of the inherent multi-modal structure, we can effectively learn from few observations by encoding the task relationships via certain classes of sparse graphs. The latter are defined based upon the Hamming distance between the multi-indices associated to pairs of tasks. An interesting special case arises from *multi-way Cartesian product graphs*, that we describe in detail and further consider in experiments. Whereas we mainly focus on task similarities we also discuss an extension to account for task dissimilarities via *mixed graphs* [14].

We propose an optimization problem that models the tasks and their relationships jointly. The approach relies on a graph-induced penalty and finds a model in a restricted hypothesis space contained within a suitably defined norm ball. This restriction is imposed to conveniently counteract the fact that sparse graphs might leave unpenalized a large class of models. In the special case of trace norm balls the procedure is in line with the conventional MTL assumption that the parameter vectors associated to the different tasks lie near a common low dimensional subspace.

The remainder of the paper is organized as follows. Section 2 details the notation used for tensors and recalls MLMTL. Section 3 deals with graph formalisms for modeling task relationships within MLMTL problems. In Section 4 we introduce our problem formulation to learn models and task relationships jointly; we then discuss a solution approach and the related algorithmical aspects. Section 5 compares our techniques with a number of alternative approaches found in the literature. We draw concluding remarks in Section 6.

2 Tensors and Multilinear Multitask Learning

For a positive integer I we write $[I]$ to mean the set $\{1, 2, \ldots, I\}$. Our approach relies on modeling learning tasks jointly based upon a multidimensional array, i.e., a *tensor* [19]. We recall that the order of a tensor is its number of dimensions, also known as modes. The m-mode fibers of the Mth order tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m \times \cdots \times I_M}$ are column vectors each of which is obtained from $\mathcal{W}_{i_1 i_2 \cdots i_m \cdots i_M}$ by fixing the values of all the indices but i_m. For $J_m = I_1 I_2 \cdots I_{m-1} I_{m+1} \cdots I_M$ the m-mode unfolding of $\mathcal{W}$, denoted by $\mathcal{W}^{(m)}$, is then the (I_m, J_m)-matrix whose columns are the m-mode fibers, arranged according to a given ordering[1]. For two compatible tensors $\mathcal{W}, \mathcal{Z}$ of order $M \geq 1$ we define the inner product by $\langle \mathcal{W}, \mathcal{Z} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_M} \mathcal{W}_{i_1 i_2 \cdots i_M} \mathcal{Z}_{i_1 i_2 \cdots i_M}$; the corresponding norm is $\|\mathcal{W}\|_2 = \sqrt{\langle \mathcal{W}, \mathcal{W} \rangle}$. We conclude by recalling the definition of *m-mode product* [8]. For a generic Mth order tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m \times \cdots \times I_M}$ and a (F_m, I_m)-matrix A we write $\mathcal{W} \times_m A$ to mean:

$$\mathcal{M} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times F_m \times \cdots \times I_M} \;\; : \;\; \mathcal{M} = \mathcal{W} \times_m A \Leftrightarrow \mathcal{M}^{(m)} = A \mathcal{S}^{(m)} \; . \tag{1}$$

One has $(\mathcal{W} \times_1 A) \times_2 B = (\mathcal{W} \times_2 B) \times_1 A$, which justify the notation $\mathcal{W} \times_1 A \times_2 B$.

We consider T related learning tasks; for each $t \in [T]$ we aim at finding the parameter vector corresponding to a linear function over $\mathcal{X} \subseteq \mathbb{R}^D$, from a task-specific dataset of input-output pairs

$$\mathscr{D}^{(t)} = \{(x_n, y_n) \;\; : \;\; n \in [N_t]\} \subset \mathcal{X} \times \mathcal{Y} \; , \tag{2}$$

where $\mathcal{Y} \subseteq \mathbb{R}$ denotes the target space. As in [21] we assume that each task is associated with M modes; that is, we consider the case where $T = I_1 I_2 \cdots I_M$ and we model the tasks jointly based upon a $(I_1, I_2, \ldots, I_M, D)$-tensor $\mathcal{W}$. See below and [21] for a selection of practical problems where this setting arises in. In the following we use $w_i^{(m)}$ to denote the column vector corresponding to the ith column of $\mathcal{W}^{(m)}$, and $\overline{w}_j^{(m)}$ to denote the column vector corresponding to the jth row of $\mathcal{W}^{(m)}$. With this notation the $(M+1)$-mode unfolding of $\mathcal{W}$ is the matrix:

$$\mathcal{W}^{(M+1)} = \left[w_1^{(M+1)}, w_2^{(M+1)}, \cdots, w_T^{(M+1)} \right] = \left[\overline{w}_1^{(M+1)}, \overline{w}_2^{(M+1)}, \cdots, \overline{w}_D^{(M+1)} \right]^\top \; , \tag{3}$$

[1] In what follow we assume the same ordering rule given in [19, Section 2.4].

whose columns correspond to the parameter vectors representing the linear task functions.

The tensor formalism highlighted above was used in [21] to extend MTL to account for multi-modal structure among the tasks. In particular the authors proposed a convex approach based on a penalty that encourages common latent features. The penalty was defined as the average of the trace norm of each mode unfolding [13, 22, 23, 24]:

$$\Omega_{\mathrm{tr}}(\mathcal{W}) = \sum_{m\in[M+1]} \left\|\mathcal{W}^{(m)}\right\|_1 \tag{4}$$

where, for a matrix A, the trace norm is given by $\|A\|_1 = \mathrm{trace}\big((A^\top A)^{1/2}\big)$. On a different track, it has become a prevailing trend in MTL to model the common structure via a graph. In this case learning the tasks relationship amounts to find the adjacency matrix or, is some cases, the corresponding Laplacian [1, 10, 12, 26].

3 Multimodal Task Relationships, Hamming Distance and Mode Graphs

Our next goal is to introduce a standard graph formalism and the induced regularization scheme. We will highlight an important drawback of this methodology and propose a novel approach suitable for MTL problems with multi-modal structure. We conclude by discussing an extension that allows to incorporate also task dissimilarities.

3.1 Leveraging Multimodal Task Relationships via Weighted Undirected Graphs

A convenient way to model task relationships is via a weighted undirected graph $G = ([T], A)$ where $[T]$ is identified with the set of tasks and A is the adjacency matrix, a symmetric (T, T)-matrix with non-negative entries. In the following we always assume that, to be admissible, G must have no self loops, i.e., $A_{st} = 0$ if $s = t$; we denote by $\mathscr{A}_+^T$ the corresponding set of admissible adjacency matrices. The (unnormalized) graph Laplacian is $L = D - A$, where D is the diagonal matrix defined by $D_{tt} = \sum_{s\in[T]} A_{st}$ for all $t \in [T]$. When learning the T tasks jointly one can rely on the following structure-inducing penalty to enforce relevant similarities [10]:

$$\Omega(\mathcal{W}, A) = \sum_{s<t} A_{st} \left\| w_s^{(M+1)} - w_t^{(M+1)} \right\|_2^2 . \tag{5}$$

Note that the number of entries of A scales quadratically with T; this represents a major obstacle when A needs to be learned from few input-output pairs.

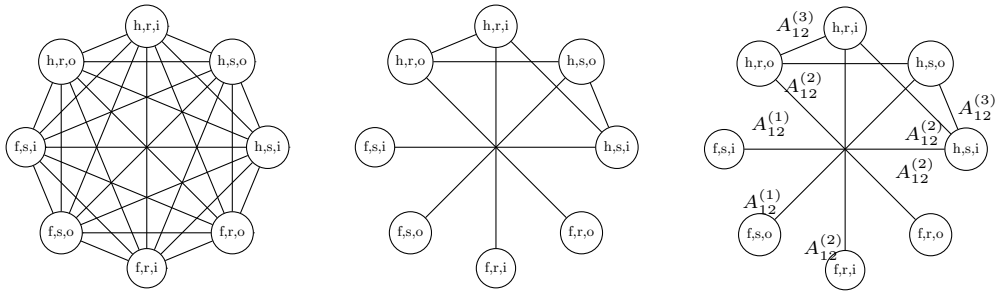
To overcome the above problem we exploit the multi-modal structure among the learning tasks. Specifically, we consider the setting in which each $t \in [T]$ is naturally identified with a multi-index $i = (i_1, i_2, \ldots, i_M) \in [I_1] \times [I_2] \times \cdots \times [I_M]$ via a one-to-one *index mapping* $\kappa : [T] \to [I_1] \times [I_2] \times \cdots \times [I_M]$. Denote by $\alpha(i, j)$ the *Hamming distance* between the multi-indices i, j, i.e.,

$$\alpha(i, j) = |\{m \ : \ i_m \neq j_m\}| \ , \tag{6}$$

where $|\cdot|$ is used to denote the cardinality of a set. We can leverage the inherent multi-modal structure by assuming that tasks that are distant according to α are disconnected, which leads to sparse graphs. This is formalized by the following definition.

Definition 1. *We call $G = ([T], A)$ (α, q)-sparse if $A \in \mathscr{A}_+^T$ and for all $s, t \in [T]$ it holds that $A_{st} = 0$ if $\alpha(\kappa(s), \kappa(t)) > q$. We denote by $\mathscr{G}_q^\alpha$ the set of (α, q)-sparse graphs.*

The definition implies that the sparsity decreases with q; one has $\mathscr{G}_1^\alpha \subset \mathscr{G}_2^\alpha \subset \cdots \subset \mathscr{G}_M^\alpha$ where $\mathscr{G}_M^\alpha$ is the set of dense graphs. Correspondingly, learning task relationships within $\mathscr{G}_q^\alpha$ is expected to be increasingly more difficult with q, both computationally and statistically, due to the explosion in the number of parameters to learn. The following definition formalizes a class of graphs playing a key role in our context.



(a) dense (b) $(\alpha, 1)-$sparse (c) CP

Figure 1: Type of graphs for MTL problems with an inherent multi-modal structure. For the sake of clarity we refer to an hypothetical subject-activity-condition preference model. Here $T = 8$ and tasks refer to $I_1 = 2$ subjects (**h**enry/**f**elicia), $I_2 = 2$ activity (**r**unning/**s**tudying) and $I_3 = 2$ conditions (**i**ndoor/**o**utdoor). In order to fully determine a dense graph we need to specify 28 weights (Fig 1a); only 8 if the graph is $(\alpha, 1)-$sparse (Fig 1b). For a CP graph (Fig 1c) it is enough to specify 3 parameters, i.e., $A_{12}^{(1)}$, $A_{12}^{(2)}$ and $A_{12}^{(3)}$. In this case the relationships between two tasks depends on the entry that differs in the corresponding multi-indices. For instance the task relationships between the models (**h**,**r**,**i**) and (**h**,**s**,**i**) is solely determined by the weight in the 2nd mode graph corresponding to the pair $(\mathbf{r}, \mathbf{s})$.

Definition 2. *We call $G = ([T], A)$ a Cartesian Product (CP) graph if $A \in \mathscr{A}_+^T$ and for all $m \in [M]$ there exists a (I_m, I_m)-matrix $A^{(m)}$ such that:*

$$A_{st} = \begin{cases} A^{(m)}_{i_m, j_m}, & \text{if} \quad i_\ell = j_\ell \quad \forall\, \ell \neq m \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

for all $(s, t) \in [T] \times [T]$ and $\kappa(s) = (i_1, i_2, \ldots, i_M)$, $\kappa(t) = (j_1, j_2, \ldots, j_M)$.

The latter can be seen as a multi-way generalization of the notion of (two-way) Cartesian product graph found in the literature see, e.g., [16]. A CP graph is $(\alpha, 1)$-sparse; its adjacency matrix is further characterized by being completely determined by M small matrices $A^{(1)}, \ldots, A^{(M)}$. In particular, the definition implies that the mth matrix corresponds to a factor graph $G^{(m)} = \left([I_m], A^{(m)}\right)$, that we call the mth *mode graph*. Denote by $A \otimes B$ the Kronecker product between the matrices A and B. When the index mapping κ in Definition 2 matches the ordering rule of [19, Section 2.4], one can check that equation (7) can be restated as:

$$A = \left(\mathrm{Id} \otimes \cdots \otimes \mathrm{Id} \otimes A^{(1)}\right) + \left(\mathrm{Id} \otimes \cdots \otimes A^{(2)} \otimes \mathrm{Id}\right) + \cdots + \left(A^{(M)} \otimes \cdots \otimes \mathrm{Id} \otimes \mathrm{Id}\right) , \tag{8}$$

where we have denoted by Id the identity matrix of the appropriate dimensions. Figure 1 illustrates the different type of graphs for MTL problems with an inherent multi-modal structure. We have given emphasis on the number of parameters required to fully determine graphs in each class.

It should be clear from above that CP graphs constitute a natural way to exploit multi-modality when representing task relationships in MLMTL problems. The next result shows the impact on the graph based regularization scheme.

Proposition 1. *Suppose that $G = ([T], A)$ is a CP graph; then the penalty function* (5) *can be restated as:*

$$\Omega(\mathcal{W}, A) = \sum_{m=1}^{M} \sum_{i_m < j_m} A^{(m)}_{i_m j_m} \left\| \overline{w}^{(m)}_{i_m} - \overline{w}^{(m)}_{j_m} \right\|_2^2 = \sum_{m=1}^{M} \left\langle \mathcal{W}, \mathcal{W} \times_m L^{(m)} \right\rangle \tag{9}$$

where $A^{(1)}, A^{(2)}, \ldots, A^{(M)}$ are the adjacency matrices of the mode graphs corresponding to G and $L^{(1)}, L^{(2)}, \ldots, L^{(M)}$ are the associated graph Laplacians.

Proof. See Appendix. □

3.2 Accounting for Dissimilarities Via Mixed Graphs

Before concluding this section we show how the above ideas extend to *mixed graphs* [14]. Mixed graphs are a generalization of weighted undirected graphs; besides possessing an adjacency matrix A and a corresponding Laplacian L, specified as above, they are characterised by a symmetric matrix $E \in \{-1,1\}^{T\times T}$. The latter expresses the type of edge: $E_{st} = 1$ (respectively -1) if (s,t) is a similarity (respectively dissimilarity) edge. The analogue of the quadratic penalty (5) is here:

$$\tilde{\Omega}(\mathcal{W}, A) = \sum_{s<t} A_{st} \left\| w_s^{(M+1)} - E_{st} w_t^{(M+1)} \right\|_2^2 \tag{10}$$

which simultaneusly pull together similar tasks while pushing away those that are dissimilar.

It is not difficut to generalize Definition 2 to the notion of *mixed CP graph*. Besides (7), in this setting we require for all $m \in [M]$ the existence of a symmetric matrix $E^{(m)} \in \{-1,1\}^{I_m \times I_m}$ such that:

$$E_{st} = \begin{cases} E^{(m)}_{i_m, j_m}, & \text{if} \quad i_\ell = j_\ell \ \ \forall\, \ell \neq m \\ 1, & \text{otherwise} \end{cases} \tag{11}$$

for all $(s,t) \in [T] \times [T]$ and $\kappa(s) = (i_1, i_2, \ldots, i_M)$, $\kappa(t) = (j_1, j_2, \ldots, j_M)$. Note that, correspondingly, E factorizes as A does in (8). Consider now the positive semidefinite matrix $\tilde{L} = L + (1 - E) \bullet A$, where 1 is the matrix of ones and we denoted by $\bullet$ the Hadamard product between compatible matrices. We can state the following generalization of Proposition 1.

Proposition 2. *Suppose that $G = ([T], A)$ is a mixed CP graph; then the we have:*

$$\tilde{\Omega}(\mathcal{W}, A) = \sum_{m=1}^{M} \sum_{i_m < j_m} A^{(m)}_{i_m j_m} \left\| \overline{w}^{(m)}_{i_m} - E^{(m)}_{i_m j_m} \overline{w}^{(m)}_{j_m} \right\|_2^2 = \sum_{m=1}^{M} \left\langle \mathcal{W}, \mathcal{W} \times_m \tilde{L}^{(m)} \right\rangle \tag{12}$$

where for all $m \in [M]$, $A^{(m)}$, $E^{(m)}$, $\tilde{L}^{(m)}$ are associated to the m-th mode mixed graph.

Proof. See Appendix. □

The penalty (4) used in [21] is defined upon the different mode matricizations of the parameter tensor. Propositions 1 and 2 show that, whenever the underlying graph is CP (respectively, mixed CP), the quadratic penalty (5) (respectively, (10)) can be similarly restated in terms of the first M mode matricizations. The computational implication of this fact will be discussed in the next section.

4 Learning Models and Task Relationships Jointly

In real-life applications task relationships are usually unknown and need to be learned from data. Our next goal is to give a simple and scalable approach to learn the T predictive models and their relationships simultaneously. The method outlined below can be applied to weighted undirected graphs, in which case only task similarities can be accounted for. The approach can also work with dissimilarities via mixed graphs, provided that the similarity matrix E (or the similarity matrices of the mode graphs) are given (fixed). In the following we focus on the first case; it should be clear that for mixed graph it is enough to replace Ω and L by the analogue entities $\tilde{\Omega}$ and $\tilde{L}$, given above.

4.1 Working Assumptions and Hypothesis Spaces

For weighted undirected graphs the method is based upon the main working assumptions that *1)* the parameter vectors of related tasks should be close in ℓ_2 norm and *2)* tasks are inherently multi-modal with the unknown graph of relationships $G = ([T], A)$ being either CP or (α, q)-sparse, with small q. Point 1 motivates the use of the graph based penalty (5) whereas point 2 implies restrictions on the support of the unknown adjacency matrix A. In the special case of CP graphs, only the upper part of M small matrices $A^{(1)}$, $A^{(2)}, \ldots, A^{(M)}$ needs to be learned. In the experiments we

focus on this case; the outcome shows that the proposed data-driven regularization approach leads to significant improvements over the generalization error of alternative MTL methods.

In some cases only few pairs of tasks in the pool might be close in ℓ_2 norm. The matching regularization graph would then leave unpenalized a large set of tensors. The set of parameter tensors $\mathscr{S}^*(A)$ that achieve zero penalty are associated to the null space of the correspnding graph Laplacian L since

$$\mathscr{S}^*(A) = \big\{\mathcal{W} \;:\; \Omega(\mathcal{W}, A) = 0\big\} = \big\{\mathcal{W} \;:\; \mathcal{W}^{(M+1)} L = 0\big\} \;. \tag{13}$$

This represents the limiting case where related tasks have identical parameter vectors; in fact, if $\mathcal{W} \in \mathscr{S}^*(A)$, then $w_s^{(M+1)} = w_t^{(M+1)}$ whenever $A_{st} > 0$. If the graph is very sparse the null space of L might have large dimension resulting into an ill-posed problem. This occurs when the graph consists of many connected components, possibly given by isolated nodes. To counteract this fact we further restrict the hypothesis space by searching for models in the norm ball:

$$\mathscr{W}_\gamma^p = \left\{\mathcal{W} \in \mathbb{R}^{I_1 \times \cdots \times I_M \times D} \;:\; \left\|\mathcal{W}^{(M+1)}\right\|_p^p \leq \gamma\right\} \tag{14}$$

where $p \in \{1, 2\}$ and $\gamma > 0$ is a used-defined parameter. The case $p = 2$ is amenable to simple computations and leads to Tikhonov regularization. The case $p = 1$ attracted a lot of attention in the MTL literature, see [20] and references therein. The trace-norm is the convex envelope of the rank function within the spectral norm unit ball [11]. This motivates its use under the typical MTL assumption that parameter vectors lie near a common low dimensional subspace.

4.2 Main Problem Formulation

We define the empirical risk and the regularized objective function, respectively, by:

$$E(\mathcal{W}; \mathscr{D}) = \sum_{t \in [T]} \sum_{(x,y) \in \mathscr{D}^{(t)}} \left\|y - x^\top w_t^{(M+1)}\right\|^2 \quad \text{and} \quad h(\mathcal{W}, A) = E(\mathcal{W}; \mathscr{D}) + \Omega(\mathcal{W}, A) \;. \tag{15}$$

We further denote by $\mathscr{A}_\lambda$ the set of admissible adjacency matrices defined upon λ, a used-defined hyper-parameter vector with positive entries. The nature of $\mathscr{A}_\lambda$ is dictated by the chosen family of graphs, as well as by additional user-defined constraints. In the following we assume that $\mathscr{A}_\lambda$ involves a set of linear equalities and inequalities constraints. This allows one to encode a large class of prior assumptions over the task relationships, while still resulting into a manageable convex set. A special case that we consider in the following is the set of CP graphs with prescribed sum of weights and upper-bounds on each weight, namely:

$$\mathscr{A}_\lambda = \left\{A : (8) \text{ holds with } A^{(m)} \in \mathscr{A}_+^{I_m}, A_{ij}^{(m)} \leq \lambda_m \;\forall\; (i,j) \in [I_m] \times [I_m], \sum_{j<i} A_{ij}^{(m)} = \lambda_m,\; m \in [M]\right\}. \tag{16}$$

We then learn models and task relationships jointly by solving the following non-convex problem:

$$\min\big\{h(\mathcal{W}, A) : \mathcal{W} \in \mathscr{W}_\gamma^p,\; A \in \mathscr{A}_\lambda\big\} \;. \tag{17}$$

4.3 Template Algorithm

To solve problem (17) we consider the proximal regularization of a two block Gauss-Seidel method. This scheme has been proposed and analyzed in [2] and was considered in [3] for dictionary learning. When applied to our setting it consists of the following main iteration[2]:

$$\mathcal{W}_{k+1} = \arg\min \left\{h\big(\mathcal{W}, A_k\big) + \frac{1}{2\eta_k}\big\|\mathcal{W} - \mathcal{W}_k\big\|_2^2 \;:\; \mathcal{W} \in \mathscr{W}_\gamma^p\right\} \tag{18a}$$

$$A_{k+1} = \arg\min \left\{\Omega\big(\mathcal{W}_{k+1}, A\big) + \frac{1}{2\xi_k}\big\|A - A_k\big\|_2^2 :\; A \in \mathscr{A}_\lambda\right\} \;. \tag{18b}$$

[2]See also the additional material.

Suppose that $\eta_k, \xi_k \in [a, b]$ for any $k \in \mathbb{N}$ and $0 < a \leq b$. It is possible to show [2] that the sequence $(h(\mathcal{W}_k, A_k))_{k\in\mathbb{N}}$ is decreasing and that $(\mathcal{W}_k, A_k)_{k\in\mathbb{N}}$ converges to a critical point for problem (17). At $k = 0$ the approach gives the unique solution of a strictly convex learning problem, the nature of which depends on p. The procedure then produces a data-drive iterative refinement of the hypothesis space based upon the chosen family of graphs.

Whenever in (17) the set $\mathscr{A}_\lambda$ is defined upon a set of linear equalities and inequalities constraints, (18b) results into a convex Quadratic Program (QP). Importantly, further simplifications occur for CP graphs as soon as $\mathscr{A}_\lambda$ prescribes only conditions on the entries of the mode adjacency matrices. One such case, in particular, is given by (16). In light of (9) problem (18b) in this case decomposes into M small convex QPs aimed at finding the matrices $A^{(1)}, \cdots, A^{(M)}$; these can be effectively approached by standard QP solvers.

The computational bottleneck is represented by (18a). For the practical problems of interest, in fact, the high dimensionality of $\mathcal{W}$ prevents the use of second order information in the solution strategy. A suitable approach is obtained by specializing the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) given in [5]. Specifically, $\mathcal{W}_{k+1}$ in (18a) is approached for $i \to \infty$ by $\mathcal{X}_i$ given as:

$$\mathcal{X}_i = \Pi_p \left(\mathcal{Z}_{i-1} - t_i \left(\partial_{\mathcal{Z}} h\big(\mathcal{Z}_{i-1}, A_k\big) + 1/\eta_k \big(\mathcal{Z}_{i-1} - \mathcal{W}_k\big) \right) \right) \tag{19a}$$

$$\mathcal{Z}_i = \mathcal{X}_i + \frac{i-1}{i+2} \left(\mathcal{X}_i - \mathcal{X}_{i-1} \right) \tag{19b}$$

where Π_p denotes the Euclidean projection onto the ball (14). For $p = 2$ it is not difficult to see that the projection requires simply a rescaling step. For $p = 1$ the projection requires to compute the Singular Value Decomposition (SVD) of its matrix argument first, and then to project the obtained vector of singular values onto the ℓ_1-ball. This latter step can be performed by fast algorithms, see e.g. [9]. An alternative approach suitable for very large scale problems is given by a generalized Frank-Wolfe procedure, see, e.g., [18]. In this case, the full SVD in Π_1 is replaced by computing only the leading singular vectors, which can be approached for instance via the power method or via Lanczos' algorithm. In this case a lower iteration complexity is traded for a worse convergence rate of $O\big(\frac{1}{i}\big)$ instead of the optimal rate $O\big(\frac{1}{i^2}\big)$ achieved by the iteration (19).

5 Case Studies

Here we consider synthetic and real-life case studies. We begin by listing the compared techniques.

G-MLMTL1(respectively 2) refers to the graph based approach in (17) with $p = 1$ (respectively $p = 2$) and $\mathscr{A}_\lambda$ specified as in (16). A solution is found via the main iteration (18); the subproblem (18a) is solved via (19). In the synthetic experiments **G-MLMTLoracle** refers to the case where the graph is fixed and corresponding to the true task relationships.

TN-MLMTL refers to the convex approach in [21]; the method uses the average of trace norm of mode matricizations (4) as a regularization mechanism[3].

TN-MTL refers to the standard MTL approach based on constraining the trace-norm of $\mathcal{W}^{(M+1)}$ see, e.g., [20] and references therein. A solution can be obtained for $\eta_1 \to \infty$ as a by-product of the first step in the main iteration (18), using the initialization $(\mathcal{W}_0, A_0) = 0$.

C-MTL refers to the Clustered MTL approach[4] proposed in [17]; we use the convex relaxation based on the cluster norm.

[3] We thank Bernardino Romera Paredes for providing a MATLAB implementation.

[4] We considered the MATLAB implementation found at `http://cbio.ensmp.fr/~ljacob/documents/cmtl-code.tgz` .

STL refers to single task learning. Each model is learned independently via ridge regression, based on the corresponding task-specific dataset[5].

In all the cases optimal hyper-parameters within a wide range are found via 5-fold cross-validation based on training data. The Mean Squared Error (MSE) obtained on an independent test dataset and averaged across tasks is reported in Table 1 as a measure of performance. Next we give a description of the different case studies that we have considered.

Table 1: MSE results on test data

	dataset			
	synthetic	restaurant (×1e2)	load (×1e2)	temperature (×1e2)
STL	35.30(2.8)	40.11(1.9)	6.58(1.8)	26.12(2.3)
C-MTL	24.62(2.9)	35.29(1.7)	10.87(4.8)	23.85(2.1)
TN-MTL	17.95(2.7)	35.17(1.7)	6.24(1.7)	22.92(1.8)
TN-MLMTL	23.73(2.7)	34.69(2.2)	5.58(1.6)	21.15(2.1)
G-MLMTL1	**10.86(2.6)**	**33.52(2.0)**	**4.38(0.9)**	22.41(2.0)
G-MLMTL2	22.41(2.4)	**33.61(2.0)**	4.51(1.0)	**19.38(1.9)**

Synthetic Problems We begin by MTL problems with a multi-modal structure generated as follows. With reference to the symbols introduced in Section 2, we took $D = 30$ and $I_1 = I_2 = I_3 = 5$ so that the number of tasks was $T = 125$. We then generated a random CP-graph $G = ([T], A)$. Specifically, A was generated via (8); the weights of $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$ were taken to follow a Bernoulli distribution with success probability 0.2. The tensor $\mathcal{W}^*$ was generated at random so that $\mathcal{W}^{*(M+1)}$ was a 5-rank matrix with rows in the null-space of the Laplacian matrix L corresponding to A. In this way, the matrix of tasks $\mathcal{W}^{*(M+1)}$ was low rank *and* such that some pairs of columns were exactly the same[6]. Finally we generated N input-output pairs per task. Specifically, denoted by (x, y) a generic pair, we let $y = x^\top w_t^{*(M+1)} + \sigma\,\epsilon$ where x was drawn from a normal multivariate Gaussian distribution, t was the task index in $[T]$, $\sigma = 0.1$ and ϵ was again normally distributed and independent from x. Of these input-output pairs per task, $N_t \ll N$ were used for training and model selection; testing results measured on the left-out sample are reported in Figure 2. Notably G-MLMTL1 is close to the performance of the oracle, even for small N_t. Results in Table 1 refer to $N_t = 5$.

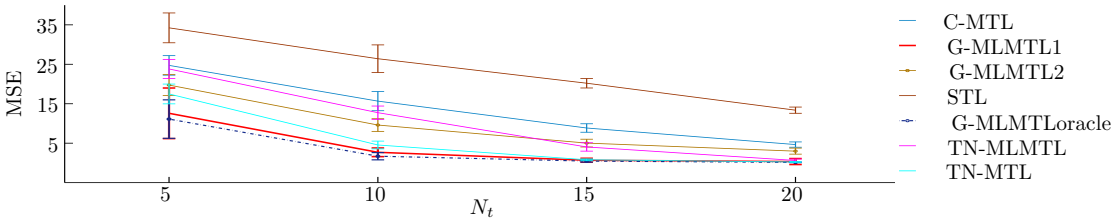


Figure 2: Test results for the synthetic problem

Restaurant & Consumer Dataset As a first real-life test case we considered the same Restaurant & Consumer Dataset [25] that was also analyzed in [21]. The goal is to predict different types of ratings given by consumers to different restaurants. We focused on 20 consumers and tried to predict their ratings on 3 different aspects of each restaurant, based upon a vector of 44 given restaurant features. The MLMTL methods, which then learn a $(20, 3, 44)$-tensor $\mathcal{W}$, are compared agains the MTL and STL techniques listed above. Of the 819 observations available across the tasks, 491 were used for training and model selection and the rest were used for testing.

[5] We considered the MATLAB implementation found at `http://www.esat.kuleuven.be/sista/lssvmlab/` .

[6] Note that, by construction, $\mathcal{W}^*$ belongs to the set of unpenalized parameter tensors in (13).

Short-term Electricity Load Forecasting This is the dataset released for the Global Energy Forecasting Competition (GEFCom2012, [15]). The goal is to perform 1-hour-ahead electricity load forecasting. The data consists of hourly electricity loads for 20 zones from January 1st, 2004 to December 31st, 2008. To predict the load in the next hour we considered nonlinear autoregressive models based on a delay vector of 8 lagged hourly loads. Specifically, we followed [7] and accounted for nonlinearity by a kernel-based explicit feature map constructed from the training delay vectors[7]. As the consumer behaviors is expected to change along the year[8] we model each quarter separately. Correspondingly the MLMTL methods learn a $(20, 4, 11)$-tensor $\mathcal{W}$ where 11 was the dimension of the explicit feature map. We used 1519 training patterns uniformly sampled across the tasks; testing was carried on the left-out sample.

One-day-ahead Temperature Prediction We consider the problem of predicting the next day average temperature at the weather stations of 22 different cities scattered across Europe[9]. To this end daily data between January 1st, 2012 and October 16th, 2013 were collected from www.wunderground.com. The predictors consisted of three lagged values of relevant continuous variables including humidity, wind direction and speed, pressure and min/max/average temperature measured at each location for a total of 1386 variables. The vector of such variables was compressed via linear principal component analysis to obtain 16 uncorrelated features. These features, representative of the global weather state, were then used for each local prediction. As in the previous case we acknowledged that models might vary across time and dealt with each month independently. This results into a $(22, 12, 16)$-tensor in the MLMTL approaches; training patterns were uniformly distributed across the cities and corresponded to measurements collected in the year 2012 only.

6 Concluding Remarks

We have focused on MTL problems where task relationships are encoded via a weighted graph. This has the disadvantage that, in general, the number of weights grows quadratically with the number of tasks. In case of a multi-modal structure we have shown that a natural approach to circumvent this disadvantage is via (α, q)-sparse and CP graphs. Furthermore, to deal with the common situation where task relationships are unknown we have proposed a scalable and easily implementable data-driven algorithm. Learning relationships within the class of CP graphs is particularly simple in view of the decomposability of problem (18b). Although problem (17) is non-convex, a natural initialization for the algorithm (18) is given by the unique solution to a convex MTL problem. The procedure then produces a data-driven iterative refinement of the hypothesis space based upon the chosen family of graphs. The proposed approach compares favorably with alternative MTL techniques. In particular, in our experiments it outperformed the approach in [21] based on the sum of trace norm penalties. Finally we remark that, although we have not dealt with this point in the paper, the proposed approach is amenable to a kernel-based extension, along the same lines of [10].

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. IWT: projects: SBO POM (100031); iMinds Medical Information Technologies SBO 2014 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017).

[7] We used the RBF Gaussian kernel and fixed the kernel parameter based upon the mean distance of the training patterns.

[8] Due in particular to changing work-loads patterns in business and industrial users and the effect of the temperature variation.

[9] The cities were, in alphabetical order: Amsterdam, Antwerp, Athens, Berlin, Brussels, Dortmund, Dublin, Eindhoven, Frankfurt, Groningen, Hamburg, Liege, Lisbon, London, Madrid, Milan, Nantes, Paris, Prague, Rome, Toulouse, Vienna.

Appendix: Proof of Propositions 1 & 2

It is sufficient to prove Propositions 2 as Proposition 1 arises from the special case obtained setting all the entries of the similarity matrix E equal to 1. For simplicity of exposition we prove the result for $M = 2$, in which case the parameter tensor $\mathcal{W}$ is a third order tensor. It follows by the definition of mixed CP graph that the matrix A factorizes: $A = \mathrm{Id} \otimes A^{(1)} + A^{(2)} \otimes \mathrm{Id}$; a similar factorization holds for E. Correspondingly, it can be checked that we can restate $\tilde{L}$ as follows:

$$\tilde{L} = L + (1 - E) \bullet A = \mathrm{Id} \otimes \tilde{L}^{(1)} + \tilde{L}^{(2)} \otimes \mathrm{Id} \tag{20}$$

where $\tilde{L}^{(m)} = L^{(m)} + (1 - E^{(m)}) \bullet A^{(m)}$ refers to the m-th mixed mode graph. Now we have:

$$\sum_{m=1}^{2} \sum_{i_m < j_m} A^{(m)}_{i_m j_m} \left\| \overline{w}^{(m)}_{i_m} - E^{(m)}_{i_m j_m} \overline{w}^{(m)}_{j_m} \right\|_2^2 = \sum_{m=1}^{2} \left\langle \mathcal{W}^{(m)}, \tilde{L}^{(m)} \mathcal{W}^{(m)} \right\rangle = \sum_{m=1}^{2} \left\langle \mathcal{W}, \mathcal{W} \times_m \tilde{L}^{(m)} \right\rangle \tag{21}$$

where the first equality follows from [14, Proposition 1] and in the second equality we relied on the definition of m-th mode product and the fact that matricizations preserve inner product. Now using (20) and that $\mathcal{M} = \mathcal{W} \times_1 A \times_2 B \Leftrightarrow \mathcal{M}^{(3)} = \mathcal{W}^{(3)} (B \otimes A)^\top$, the right hand side of (21) can be restated as follows:

$$\sum_{m=1}^{2} \left\langle \mathcal{W}, \mathcal{W} \times_m \tilde{L}^{(m)} \right\rangle = \left\langle \mathcal{W}, \mathcal{W} \times_1 \tilde{L}^{(1)} \times_2 \mathrm{Id} + \mathcal{W} \times_1 \mathrm{Id} \times_2 \tilde{L}^{(2)} \right\rangle = \left\langle \mathcal{W}^{(3)}, \mathcal{W}^{(3)} \tilde{L} \right\rangle . \tag{22}$$

From this we finally obtain:

$$\sum_{m=1}^{2} \sum_{i_m < j_m} A^{(m)}_{i_m j_m} \left\| \overline{w}^{(m)}_{i_m} - E^{(m)}_{i_m j_m} \overline{w}^{(m)}_{j_m} \right\|_2^2 = \left\langle \mathcal{W}^{(3)}, \mathcal{W}^{(3)} \tilde{L} \right\rangle = \sum_{s<t} A_{st} \left\| w^{(3)}_s - E_{st} w^{(3)}_t \right\|_2^2 . \tag{23}$$

□

References

[1] A. Argyriou, S. Clémençon, and R. Zang. Learning the graph of relations among multiple tasks. Technical Report hal-00940321, GALEN - INRIA Saclay - Ile de France, Laboratoire Traitement et Communication de l'Information, Paris, 2013.

[2] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[3] A. Barla, S. Salzo, and A. Verri. Regularized dictionary learning. In J. A. K. Suykens, M. Signoretto, and A. Argyriou, editors, *Regularization, Optimization, Kernels, and Support Vector Machines*, Machine Learning and Pattern Recognition Series. Chapman Hall/CRC, to appear.

[4] J. Baxter. Theoretical models of learning to learn. In S. Thrun and L. Pratt, editors, *Learning to learn*, pages 71–94. Kluwer Academic Publishers, 1998.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] R. Caruana. Multitask learning. In S. Thrun and L. Pratt, editors, *Learning to learn*. Kluwer Academic Publishers, 1998.

[7] K. De Brabanter, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Optimized fixed-size kernel models for large data sets. *Computational Statistics & Data Analysis*, 54(6):1484–1504, 2010.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[9] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[10] T. Evgeniou, C. Micchelli, M. Pontil, and J. Shawe-Taylor. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(4):615–637, 2005.

[11] M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum ordersystem approximation. In *Proceedings of the American Control Conference, 2001*, volume 6, pages 4734–4739, 2001.

[12] R. Flamary, A. Rakotomamonjy, and G. Gasso. Learning constrained task similarities in graph-regularized multi-task learning. In J. A. K. Suykens, M. Signoretto, and A. Argyriou, editors, *Regularization, Optimization, Kernels, and Support Vector Machines.*, Machine Learning and Pattern Recognition Series. Chapman Hall/CRC, to appear.

[13] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):25010–25028, 2011.

[14] A.B. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. *Artificial Intelligence and Statistics (AISTATS)*, 2007.

[15] T. Hong, P. Pinson, and S. Fan. Global energy forecasting competition 2012. *International Journal of Forecasting*, 30(2):357–363, 2013.

[16] W. Imrich, S. Klavzar, and D. F. Rall. *Topics in graph theory: Graphs and their Cartesian product*. AK Peters Ltd, 2008.

[17] L. Jacob, F. Bach, and J.P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems 21*, volume 21, pages 745–752, 2008.

[18] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[19] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[20] A. Maurer and M. Pontil. Excess risk bounds for multitask learning with trace norm regularization. In *Conference on Learning Theory (COLT)*, volume 30, pages 55–76, 2013.

[21] 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.

[22] 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*, 94(3):303–351, 2014.

[23] M. Signoretto, R. Van de Plas, B. De Moor, and J. A. K. Suykens. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters*, 18(7):403–406, 2011.

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

[25] B. Vargas-Govea, G. González-Serna, and R. Ponce-Medellın. Effects of relevant contextual features in the performance of a restaurant recommender system. In *Workshop on Context Aware Recommender RecSys 11: Workshop on Context Aware Recommender Systems (CARS-2011),*, 2011.

[26] W. Zhong and J. Kwok. Convex multitask learning with flexible task clusters. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012.