Convex Graph Laplacian Multi-Task Learning SVM

Carlos Ruiz[1](✉), Carlos M. Alaíz[1], and José R. Dorronsoro[1,2]

[1] Dept. Computer Engineering, Universidad Autónoma de Madrid, Madrid, Spain
carlos.ruizp@estudiante.uam.es, {carlos.alaiz,jose.dorronsoro}@uam.es
[2] Inst. Ing. Conocimiento, Universidad Autónoma de Madrid, Madrid, Spain

Abstract. Multi-Task Learning (MTL) goal is to achieve a better generalization by using data from different sources. MTL Support Vector Machines (SVMs) embrace this idea in two main ways: by using a combination of common and task-specific parts, or by fitting individual models adding a graph Laplacian regularization that defines different degrees of task relationships. The first approach is too rigid since it imposes the same relationship among all tasks. The second one does not have a clear way of sharing information among the different tasks. In this paper, we propose a model that combines both approaches. It uses a convex combination of a common model and of task specific models, where the relationships between these specific models are determined through a graph Laplacian regularization. We write the primal problem of this formulation and derive its dual problem, which is shown to be equivalent to a standard SVM dual using a particular kernel choice. Empirical results over different regression and classification problems support the usefulness of our proposal.

1 Introduction

Standard Machine Learning (ML) often seeks to minimize a fixed overall loss. This is the optimal goal when the training dataset is associated to a single homogeneous task, but less so when there might be somehow different subtasks underlying the common objective. If this is the case, it is natural to share the common task learning while allowing for specific learning procedures for the individual tasks. Among other advantages, this approach, known as Multi-Task Learning (MTL), complements data augmentation with task focusing, introduces inductive bias in the learning process and even performs implicit regularization. Starting with the work of R. Caruana [1], MTL has been applied to a large number of problems and under different underlying ML techniques.

Support Vector Machines (SVMs) are a natural choice for MTL. Although SVM models were originally formulated as linear models, the kernel trick allows to find the optimal hyperplane in a high-dimensional, even theoretically infinite space. Additionally, the ϵ-insensitive loss makes these models robust to noise in regression problems. Among the first approaches to SVM-based MTL is the Regularized Multi-Task Learning proposal in [2], where the primal problem for

I. Farkaš et al. (Eds.): ICANN 2020, LNCS 12397, pp. 142–154, 2020.
https://doi.org/10.1007/978-3-030-61616-8_12

linear SVM is expressed in a multi-task framework by introducing common and specific parts of each task model and penalizing these independently in the regularizer. This work is extended in [3], where a variety of kernel methods for multi-task problems with quadratic regularizers are reduced to solving a single-task problem using a multi-task kernel. This result is used in [4,5] for multi-task feature and structure learning. Also, multi-task regularizers with different goals have been proposed: for instance, in [6] the tasks are clustered, the intra-cluster distance is minimized while the inter-cluster distance is maximized. The ideas of Evgeniou *et al.* presented in [2] are extended in [7] to the use of multiple kernels for different tasks in regression, and they are generalized in [8] for classification and regression, addressing the use of task specific biases.

The initial approach, used in [2,8], is to consider the task models to be a sum of a common model and a task specific one, where a penalty μ controls the regularization balance between these common and specific parts. This is then transformed into a dual problem where μ is incorporated into the kernel matrix. With this formulation, the relationship between all tasks is assumed to be the same. The differences from the common model are all equally penalized, forcing the tasks to be equidistant to the common model. Other interesting approach shown in [3] and extended in [9] is to use a Graph Laplacian regularization, where the tasks are represented as nodes in a graph, and the distance between two task parameters is penalized according to the edge weight between those tasks. In this multi-task approximation, one can define different relations between the task pairs.

In this work we propose a new formulation, which we name Convex Graph Laplacian SVM-MTL, where the MTL models are a convex combination of common and specific components. A graph defines the relationships between the task-specific models, while the common model ensures the sharing of information across tasks. By using this formulation we can obtain the flexibility of using both different task relationships and the explicit shared information, represented in the common model. More precisely, our contributions in this work are:

- We introduce linear Convex Graph Laplacian MTL-SVMs.
- We extend this initial linear set-up to a multi-kernel setting where each component of the multi-task model can have its own kernel.
- We show numerically that our proposal gives competitive and often better results that either a single SVM model for all tasks, a combination of independent models, or Graph Laplacian MTL-SVMs.

The rest of the paper is organized as follows. In Sect. 2 we will briefly review previous formulations of the MTL and Graph MTL primal and dual problems and we present our approach in Sect. 3. We show our experimental results in Sect. 4, and the paper ends in Sect. 5, where we briefly discuss our results, offer some conclusions on them and present lines of further work.

2 Multi-Task Learning Support Vector Machine

We briefly review first standard SVMs. In order to show a more general result, we introduce a notation that allows to write Support Vector Classification (SVC) and Support Vector Regression (SVR) problems in a unified way. Following [10], consider a sample $S = \{(x_i, y_i, p_i),\ 1 \leq i \leq N\}$, where $y_i = \pm 1$, and a primal problem of the form

$$\begin{aligned} \underset{w,b,\xi}{\arg\min} \quad & J(w,b,\xi) = C\sum_{n=1}^{N} \xi_i + \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq p_i - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, N. \end{aligned} \tag{1}$$

It is easy to check [10] that for a classification sample $\{(x_i, y_i),\ 1 \leq i \leq M\}$, Problem (1) is equivalent to the SVC primal problem when choosing $N = M$ and $p_i = 1$ for all i. In a similar way, for a regression sample $\{(x_i, t_i),\ 1 \leq i \leq M\}$, Problem (1) is equivalent to the ϵ-insensitive SVR primal problem when we set $N = 2M$ and $y_i = 1,\ p_i = t_i - \epsilon,\ y_{M+i} = -1,\ p_{M+i} = -t_i - \epsilon$ for $i = 1, \ldots, M$. With this notation any result obtained for (1) will be valid for both SVC and SVR. The dual problem for this general formulation can be written as follows:

$$\begin{aligned} \underset{\alpha}{\arg\min} \quad & \Theta(\alpha) = \alpha^\intercal Q \alpha - p^\intercal \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, N,\ \sum_{i=1}^{N} y_i \alpha_i = 0, \end{aligned} \tag{2}$$

where we use the vectors $\alpha^\intercal = (\alpha_1, \ldots, \alpha_N)$, $p^\intercal = (p_1, \ldots, p_N)$ and Q is the kernel matrix. To present our results in a compact way we will use this unified formulation in the rest of this work.

Turning our attention to Convex Multi-Task SVM, their formulation in [11] has the following primal problem:

$$\begin{aligned} \underset{w,v_r,b_r,\xi}{\arg\min} \quad & J(w, v_r, b_r, \xi) = C\sum_{r=1}^{T}\sum_{i=1}^{N} \xi_i^r + \frac{1}{2}\|w\|^2 + \frac{1}{2}\sum_{r=1}^{T}\|v_r\|^2 \\ \text{s.t.} \quad & y_i^r\left(\lambda w \cdot x_i^r + (1-\lambda) v_r \cdot x_i^r + b_r\right) \geq p_i^r - \xi_i^r, \\ & \xi_i^r \geq 0,\ i = 1, \ldots, n_r,\ r = 1, \ldots, T. \end{aligned} \tag{3}$$

It can be shown that the dual problem of (3) is the following:

$$\begin{aligned} \underset{\alpha}{\arg\min} \quad & \Theta(\alpha) = \alpha^\intercal \widehat{Q} \alpha - p^\intercal \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i^r \leq C,\ i = 1, \ldots, n_r,\ r = 1, \ldots, T, \\ & \sum_{i=1}^{n_r} y_i \alpha_i^r = 0\ ,\ r = 1, \ldots, T, \end{aligned} \tag{4}$$

where $\widehat{Q}$ is the multi-task kernel matrix defined by the multi-task kernel $\widehat{k}$:

$$\widehat{k}(x_i^r, x_j^s) = \lambda^2 k(x_i^r, x_j^s) + (1-\lambda)^2 \delta_{rs} k_r(x_i^r, x_j^s).$$

Here k and k_r are the common and task-specific kernels, and δ denotes the Kronecker delta function. Also, multiple equality constraints are included in (4), which are not compatible with the SMO algorithm used to solve the SVM dual. We will discuss below how to deal with this issue. One drawback of this approach is that every task-independent part is equally penalized. This implicitly assumes all models f_r to be equidistant to the common model f. This could be detrimental in those cases where not all the tasks are related in the same way.

Finally, another approach that can introduce different relations between tasks is the Graph Laplacian Multi-Task SVM introduced in [3]. Here the tasks are seen as nodes in a complete graph $\mathcal{G}$ and the edge weights A_{rs} control the relationship between the task nodes that they connect. The primal problem is defined as

$$\begin{aligned} \underset{v_r, b_r, \xi}{\arg\min} \quad & J(v_r, b_r, \xi) = C \sum_{r=1}^{T} \sum_{i=1}^{N} \xi_i^r + \frac{\mu}{4} \sum_{r=1}^{T} \sum_{s=1}^{T} A_{rs} \left\| v_r - v_s \right\|^2 \\ \text{s.t.} \quad & y_i^r \left(v_r \cdot x_i^r + b_r \right) \geq p_i^r - \xi_i^r,\ \xi_i^r \geq 0,\ i = 1, \ldots, n_r,\ r = 1, \ldots, T; \end{aligned} \tag{5}$$

note that in this formulation no common part is shared across tasks. Moreover, consider the following extended vector $v \in \mathbb{R}^{T \times d}$ with $v^\intercal = (v_1^\intercal, \ldots, v_T^\intercal)$ and the graph Laplacian $L = D - A$, where A is the graph weight matrix and D is the corresponding degree matrix, i.e., $D_{rs} = \delta_{rs} \sum_{q=1}^{T} A_{rq}$. Denoting by $\otimes$ the Kronecker product, it can be proved that

$$v^\intercal (L \otimes I_d) v = \frac{1}{2} \sum_{r=1}^{T} \sum_{s=1}^{T} A_{rs} \left\| v_r - v_s \right\|^2 \ . \tag{6}$$

Given this, and as shown in [3], the corresponding dual problem is

$$\begin{aligned} \underset{\alpha}{\arg\min} \quad & \Theta(\alpha) = \alpha^\intercal \tilde{Q} \alpha - p^\intercal \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i^r \leq C,\ i = 1, \ldots, n_r,\ r = 1, \ldots, T, \\ & \sum_{i=1}^{n_r} y_i \alpha_i = 0,\ r = 1, \ldots, T, \end{aligned} \tag{7}$$

where $\widetilde{Q}$ is the kernel matrix corresponding to the multi-task kernel $\widetilde{k}(x_i^r, x_j^s) = L_{rs}^{+} k(x_i^r, x_j^s)$, and L^{+} is the pseudo-inverse of the graph Laplacian matrix L. Notice that problem (7) is formally identical to (4), although using a different multi-task kernel.

We point out that in (5) only the distance between vectors is penalized, but the weight vector norms v_r are not regularized. This can lead to overfitting when the tasks are highly related. Also, the sharing of information is only made through the Graph Laplacian regularization term. To improve on this, we propose the Convex Graph Laplacian Multi-Task SVM described next.

3 Convex Graph Laplacian Multi-Task SVM

Convex Graph Laplacian Multi-Task SVM combines the two approaches above, working with a convex combination of a common component w and of specific models v_r. We also use both their individual regularizers and a Graph Laplacian regularization term. The multi-task models f_r are defined as $f_r = \lambda f + (1 - \lambda) g_r + b_r$ where f is the common model, g_r are the individual models, b_r are the bias terms and $\lambda \in [0, 1]$. Hence, this reduces to the common model by setting $\lambda = 1$, and to the individual models when $\lambda = 0$; in this last case we would have a Graph Laplacian model with additional individual weight regularization.

3.1 Convex Graph Laplacian Linear Multi-Task SVM

We consider first the case of the common and specific models being linear. More precisely, the primal problem is defined now as:

$$\begin{aligned}
\underset{w, v_r, b_r, \xi}{\arg\min} \quad & J(w, v_r, b_r, \xi) = C \sum_{r=1}^{T} \sum_{i=1}^{N} \xi_i^r + \frac{1}{2} \|w\|^2 \\
& \qquad + \frac{1}{2} \sum_{r=1}^{T} \|v_r\|^2 + \frac{\mu}{4} \sum_{r=1}^{T} \sum_{s=1}^{T} A_{rs} \|v_r - v_s\|^2 \\
\text{s.t.} \quad & y_i^r \left(\lambda w \cdot x_i^r + (1 - \lambda) v_r \cdot x_i^r + b_r \right) \geq p_i^r - \xi_i^r, \\
& \xi_i^r \geq 0, \ i = 1, \dots, n_r, \ r = 1, \dots, T.
\end{aligned} \tag{8}$$

We can write this primal problem in a more compact way as

$$\begin{aligned}
\underset{w, v_r, b_r, \xi}{\arg\min} \quad & J(w, v, b_r, \xi) = C \sum_{r=1}^{T} \sum_{i=1}^{N} \xi_i^r + \frac{1}{2} \|w\|^2 + \frac{1}{2} v^\intercal (B \otimes I_d) v \\
\text{s.t.} \quad & y_i^r \left(\lambda w \cdot x_i^r + (1 - \lambda) v_r \cdot x_i^r + b_r \right) \geq p_i^r - \xi_i^r, \\
& \xi_i^r \geq 0, \ i = 1, \dots, n_r, \ r = 1, \dots, T.
\end{aligned} \tag{9}$$

Here we have $v^\intercal = (v_1^\intercal, \dots, v_T^\intercal)$ and $B = (I_T + \mu L)$; also, $\otimes$ denotes again the Kronecker product, L is the Laplacian matrix of the task graph and I_d is the identity matrix of dimension d. To prove the equivalence between (8) and (9), we simply observe that

$$v^\intercal (I_T \otimes I_d) v = \sum_{r=1}^{T} \|v_r\|^2, \ v^\intercal (L \otimes I_d) v = \frac{1}{2} \sum_{r=1}^{T} \sum_{s=1}^{T} A_{rs} \|v_r - v_s\|^2,$$

and that the Kronecker product is bilinear. The second equality also uses (6). We derive next the dual problem corresponding to (9). Its Lagrangian is

$$\begin{aligned}
\mathcal{L}(w, v, b_r, \xi, \alpha, \beta) = & \, C \sum_{r=1}^{T} \sum_{i=1}^{N} \xi_i^r + \frac{1}{2} \|w\|^2 + \frac{1}{2} v^\intercal (B \otimes I_d) v \\
& + \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r p_i^r - \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r \xi_i^r - \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r b_r \\
& - \lambda \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r w \cdot x_i^r - (1 - \lambda) \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r v_r \cdot x_i^r - \sum_{r=1}^{T} \sum_{i=1}^{n_r} \beta_i^r \xi_i^r.
\end{aligned} \tag{10}$$

Taking derivatives with respect to the primal variables and equating them to zero, we obtain the stationary conditions, of which the one involving v becomes

$$v = (1 - \lambda)(B \otimes I_d)^{-1} \Psi \alpha, \tag{11}$$

where $\alpha^{\intercal} = (\alpha_1^1, \dots, \alpha_{n_T}^T)$, and where the matrix Ψ of extended patterns is defined as:

$$\Psi = \underbrace{\begin{pmatrix} \Psi_1 & 0 & \dots & 0 \\ 0 & \Psi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Psi_T \end{pmatrix}}_{(Td) \times N}, \ \Psi_r^{\intercal} = \underbrace{\begin{pmatrix} y_1^r (x_1^r)^{\intercal} \\ y_2^r (x_2^r)^{\intercal} \\ \vdots \\ y_{n_r}^r (x_{n_r}^r)^{\intercal} \end{pmatrix}}_{n_r \times d}.$$

Note that the inverse in (11) is well defined since $(B \otimes I_d)^{-1} = (B^{-1} \otimes I_d)$, and $B = I_T + \mu L$ is an invertible matrix. Using the stationary conditions, the Lagrangian becomes the function of α:

$$\begin{aligned} \mathcal{L}(\alpha) = &-\frac{\lambda^2}{2} \sum_{r,s=1}^{\intercal} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} \alpha_i^r \alpha_j^s y_i^r y_j^s x_i^r x_j^s \\ &- \frac{(1-\lambda)^2}{2} \sum_{r,s=1}^{\intercal} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} \alpha_i^r \alpha_j^s y_i^r y_j^s B_{rs}^{-1} x_i^r x_j^s + \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r p_i^r, \end{aligned}$$

and, therefore, we arrive at the dual problem:

$$\begin{aligned} \arg\min_{\alpha} \quad & \Theta(\alpha) = \frac{1}{2} \sum_{r,s=1}^{\intercal} \sum_{i,j=1}^{n_r, n_s} \alpha_i^r \alpha_j^s y_i^r y_j^s \left[\lambda^2 + (1-\lambda)^2 B_{rs}^{-1} \right] x_i^r x_j^s - \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r p_i^r \\ \text{s.t.} \quad & 0 \le \alpha_i^r \le C, \ i = 1, \dots, n_r, \ r = 1, \dots, T, \\ & \sum_{i=1}^{n_r} \alpha_i^r y_i^r = 0, \ r = 1, \dots, T. \end{aligned} \tag{12}$$

Note that the quadratic part of the objective function has two different terms. The first one, corresponding to the common part, involves the dot products of all the points in the training set independently of their task, while the second term, which corresponds to the specific part, takes into account the task relationships via B_{rs}^{-1}. Once the dual problem is solved, the prediction of this multi-task model for a new point z from task $t \in \{1, \dots, T\}$ can also be written as $f_t(z^t) = \lambda f(z^t) + (1 - \lambda) g_t(z^t) + b_t$, where the f and g_t models are defined as:

$$f(z^t) = \lambda \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r x_i^r \cdot z^t, \ g_t(z^t) = (1 - \lambda) \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r B_{rt}^{-1} x_i^r \cdot z^t.$$

3.2 Convex Graph Laplacian Kernel Multi-Task SVM

The above division in two differentiated parts is the starting point to extend the preceding linear discussion to a kernel setting, where we will work in different

Hilbert spaces for the common f and specific g_r model functions. We can observe this by extending (12) to the kernel case, which can be expressed as a standard SVM dual problem with an MTL kernel, namely

$$\begin{aligned} \underset{\alpha}{\arg\min} \quad & \Theta(\alpha) = \frac{1}{2}\alpha^\intercal \widetilde{Q}\alpha - p\alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i^r \leq C,\ i = 1, \ldots, n_r,\ r = 1, \ldots, T, \\ & \sum_{i=1}^{n_r} \alpha_i^r y_i^r = 0,\ r = 1, \ldots, T, \end{aligned} \tag{13}$$

where the kernel matrix $\widetilde{Q}$ is computed using the kernel function $\widetilde{k}$ defined as:

$$\widetilde{k}(x_i^r, x_j^s) = \lambda^2 k(x_i^r, x_j^s) + (1-\lambda)^2 (I_T + \mu_2 L)_{rs}^{-1} k_{\mathrm{g}}(x_i^r, x_j^s);$$

here, k and k_{g} are the kernels corresponding to the common and specific parts respectively. When comparing (13) with the standard SVM dual (2), the differences are in the definition of the kernel matrix and the multiple equality constraints in (13), which have their origin at the multiple biases in (8). However, if we impose a single bias in all models, we have a dual problem that can be solved using the standard SMO algorithm.

Finally, we can write the kernel multi-task model prediction over a new pattern z^t from task t as $f_t(z^t) = \lambda f(z^t) + (1-\lambda) g_t(z^t) + b_t$, where

$$f(z^t) = \lambda \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r k(x_i^r, z^t),\ g_t(z^t) = (1-\lambda) \sum_{r=1}^{T} \sum_{i=1}^{n_r} \alpha_i^r y_i^r B_{rt}^{-1} k_{\mathrm{g}}(x_i^r, z^t).$$

4 Numerical Experiments

4.1 Datasets and Models

We test our method over eight different problems: `majorca`, `tenerife`, `california`, `boston`, `abalone` and `crime` for regression and `landmine` and `binding` for classification. In `majorca` and `tenerife` each task goal is to predict the photovoltaic production in these islands at different hours. In `california` and `boston` datasets the target is the price of houses and the tasks are defined using different location categories of these houses. In `abalone` we define three tasks: the prediction for male, female and infant specimens. The target in `crime` is to predict the number of crimes per 100 000 people in different cities of the U.S.; the prediction in each state is considered a task. For classification, in `binding`, the goal is to predict whether peptides will bind to a certain MHC molecule and each molecule represents a different task. In `landmine` the goal is the detection of landmines; each type of landmine defines a task. In Table 1 we can see the characteristics of the different datasets. We will compare the performance of our multi-task approach against four alternative models, described next. All of them are built using Gaussian kernels.

Table 1. Sample sizes, dimensions and number of tasks of the datasets used.

Dataset	Size	No. features	No. tasks	Avg. task size	Min. task size	Max. task size
majorca	15330	765	14	1095	1095	1095
tenerife	15330	765	14	1095	1095	1095
binding	32302	184	47	687	59	3089
landmine	14820	10	28	511	445	690
california	19269	9	5	3853	5	8468
boston	506	12	2	253	35	471
abalone	4177	8	3	1392	1307	1527
crime	1195	127	9	132	60	278

Common Task Learning SVM (CTL). A single SVM model is fitted on all the data, ignoring task information.

Independent Task learning SVM (ITL). Specific models are fitted for each task using only the tasks data; no cross-model learning takes place.

Convex Multi-Task learning SVM (cvxMTL). Here a convex combination between the common and the independent models is used. This multi-task approach uses both common and task-specific kernels.

Graph Laplacian MTL-SVM (GLMTL). This is the multi-task approach defined in [3]. It only uses specific models with a Graph Laplacian regularization term. In this approach, a single kernel is used for all tasks and there is no common part to be shared among the specific models.

Convex Graph Laplacian MTL-SVM (cvxGLMTL). This is our proposal, in which we use a convex combination of the common model and the specific models with their own regularizers to which we add a Graph Laplacian regularization.

4.2 Experimental Setup

Since each model taken has a different set of hyperparameters, their selection has been done in various ways. Model hyperparameters are basically chosen by cross-validation (CV) with some simplifications that we detail next. The three parameters of CTL, i.e., (C, ϵ, γ_c), are all chosen via CV and we do the same for the parameters $(C^r, \epsilon^r, \gamma_s^r)$ of each specific model in the ITL approach. For cvxMTL we will use the width γ_c selected for CTL and the specific widths γ_s^r obtained for ITL, whereas C, λ and ϵ are selected by CV. We use the γ_c selected for CTL in the GLMTL kernel and we select (C, ϵ, μ) by CV. For cvxGLMTL we use the γ_c from CTL in both the common and graph Laplacian kernels, the μ selected for GLMTL, and apply a CV procedure to select C, λ and, for regression, ϵ. In Table 2 we show the grids where the optimal values are searched and the procedure used to select each model's hyperparameters. Notice that only three hyperparameters per model are chosen by CV, to alleviate computational costs.

Table 2. Hyper-parameters, grids used to select them (when appropriate) and hyper-parameter selection method for each model.

	Grid	CTL	ITL	cvxMTL	GLMTL	cvxGLMTL
C	$\{4^k : -2 \leq k \leq 2\}$	CV	CV	CV	CV	CV
ϵ	$\{\frac{\sigma}{4^k} : 1 \leq k \leq 6\}$	CV	CV	CV	CV	CV
γ_c	$\left\{\frac{4^k}{d} : -2 \leq k \leq 3\right\}$	CV	-	CTL	CTL	CTL
γ_s	$\left\{\frac{4^k}{d} : -2 \leq k \leq 3\right\}$	-	CV	ITL	-	CTL
λ	$\{0.2k : 0 \leq k \leq 5\}$	-	-	CV	-	CV
μ	$\{4^k : -1 \leq k \leq 3\}$	-	-	-	CV	GLMTL

Cross-validation has been done in the following way. In majorca and tenerife, with time-dependent data, we have data for the years 2013, 2014 and 2015, which have been used for train, validation and test respectively. For the rest of the problems we have used a nested cross-validation scheme, using the inner CV to select the optimal hyperparameters and the outer folds to measure the fitness of our models. We work with 3 outer folds, using cyclically two thirds of the data for train and validation and keeping one third for test. We also use 3 inner folds, with 2 folds used for training and the remaining one for validation. These folds are selected randomly using the StratifiedKFold class of *Scikit-learn*; the stratification is made using the task labeling, so every fold has a similar task distribution. The regression CV score is the Mean Absolute Error (MAE), the natural measure for SVR fitness. The classification CV score is the F1 score, more informative than accuracy when we deal with unbalanced datasets. For all problems, we scale the data feature-wise into the $[0, 1]$ interval and normalize the regression targets to have zero-mean and one-standard deviation. As mentioned before, the multiple biases of the multi-task approaches cvxMTL and cvxGLMTL imply the existence of multiple dual equality constraints. To avoid this and be able to apply the standard SMO algorithm, we use a simplified version of the MTL models in which a common bias is shared among all tasks.

Finally, for cvxGLMTL it is necessary to define a graph over the tasks. The weights of the edges connecting two tasks define the degree of relationship wanted or expected between them. This predefined graph information is included in the model through the Laplacian matrix regularization. Choosing a useful graph is not a trivial task and it may also be harmful when the prior information used does not match the characteristics of the data. In our experiments no prior information is given to the model, and we use a graph in which every task (node) is connected to all the others with the same constant weight. To normalize the Graph Laplacian regularization term we will use $A_{rs} = \frac{1}{T(T-1)}$.

4.3 Experimental Results

Table 3 shows the scores obtained in every problem considered. In the case of the regression tasks, we give both the MAE and R2 scores. Moreover, in Table 5 we

Table 3. Test MAE (top), and test R2 scores (bottom) in the regression problems.

	maj.	ten.	boston	california	abalone	crime
MAE						
CTL	5.265	5.786	2.254±0.035	41870.820 ± 76.723	1.483±0.039	0.078±0.001
ITL	5.119	5.341	2.779±0.134	37043.664 ± 371.549	1.488±0.038	0.082±0.006
cvxMTL	5.077	5.351	**2.228±0.006**	36848.971 ± 242.052	**1.466±0.028**	**0.074±0.003**
GLMTL	5.291	5.840	3.070±0.391	37123.515 ± 404.205	1.690±0.017	0.094±0.006
cvxGLMTL	**4.917**	**5.335**	2.230±0.038	**36720.854 ± 225.335**	1.467±0.026	**0.074±0.003**
R2						
CTL	0.831	0.902	0.843±0.044	0.638 ± 0.005	0.560±0.017	0.743± 0.022
ITL	0.843	0.904	0.776±0.017	0.696 ± 0.005	0.550±0.024	0.711±0.006
cvxMTL	0.845	**0.907**	0.850±0.045	0.700 ± 0.003	**0.566±0.013**	**0.755±0.016**
GLMTL	0.832	0.894	0.490±0.264	0.695 ± 0.007	0.366±0.027	0.596±0.033
cvxGLMTL	**0.849**	0.905	**0.852±0.046**	**0.702± 0.003**	**0.566±0.013**	0.752±0.016

Table 4. Test F1 score (left), and accuracy (right) in the classification problems.

	F1		Accuracy	
	landmine	binding	landmine	binding
CTL	0.106 ± 0.016	0.868 ± 0.002	0.942 ± 0.004	0.791 ± 0.003
ITL	0.183 ± 0.034	0.901 ± 0.000	0.942 ± 0.004	0.850 ± 0.000
cvxMTL	0.150 ± 0.023	0.906 ± 0.001	0.943 ± 0.004	0.858 ± 0.002
GLMTL	**0.227 ± 0.042**	0.896 ± 0.003	0.935 ± 0.002	0.844 ± 0.005
cvxGLMTL	0.163 ± 0.031	**0.908 ± 0.001**	**0.944 ± 0.004**	**0.862 ± 0.002**

show the p-values of the paired signed rank Wilcoxon tests we will perform. With these tests we can reject the null hypothesis, which states that the distribution of the differences of two related samples is symmetrical around zero. Given that there are several models to be compared, we proceed in the following manner: we first rank the models by their MAE score and, then, the absolute and quadratic error distributions of each model are compared using the Wilcoxon test with the immediately following model. With this, we can determine whether the error distributions of two consecutive models are significantly different. The rankings given in the Table show ties for those model pairs where the null hypothesis is rejected at the 5% significance level. It can be observed that, in terms of MAE, the proposed cvxGLMTL model obtains the best results in most regression problems and, even when cvxGLMTL does not achieve the smaller error, Table 5 shows that it is not significantly worse than the best model. Only for abalone the cvxMTL model obtains the significantly best result in terms of R2 scores.

In the case of classification, we show in Table 4 both accuracy and F1 score. We notice that in the landmine problem the accuracies obtained are high whereas the F1 scores are low, due to the unbalanced nature of the problem. In contrast, in binding, a balanced problem, both F1 score and accuracy have similar values.

Table 5. Top: Wilcoxon p-values of absolute errors of a regression model and the one following it in the MAE ranking and similar accuracy p values. Bottom: with the same scheme, p values of quadratic errors and the R2 score ranking and F1 scores.

	majorca		tenerife		boston		california		abalone		crime		classif.	
CTL	0.0000	(3)	0.0000	(4)	**0.2554**	**(1)**	0.0000	(4)	0.0002	(2)	0.0000	(2)	0.0277	(3)
ITL	0.8131	(2)	0.0035	(2)	0.0001	(2)	0.0318	(3)	0.2546	(2)	0.3995	(2)	**0.3454**	**(1)**
cvxMTL	0.0000	(2)	0.0000	(3)	-	**(1)**	0.0000	(2)	-	**(1)**	-	**(1)**	0.0277	(2)
GLMTL	0.4183	(3)	0.5962	(4)	0.0621	(2)	0.5658	(3)	0.0000	(3)	0.0000	(3)	-	**(1)**
cvxGLMTL	-	**(1)**	-	**(1)**	**0.4113**	**(1)**	-	**(1)**	**0.0771**	**(1)**	**0.6093**	**(1)**	**0.3454**	**(1)**
CTL	0.0032	(3)	0.0000	(2)	**0.1791**	**(1)**	0.0000	(4)	0.0016	(3)	0.0001	(2)	0.3454	(4)
ITL	0.6340	(2)	**0.5999**	**(1)**	0.0001	(2)	0.0035	(3)	0.3096	(3)	0.3972	(2)	0.0277	(3)
cvxMTL	0.0000	(2)	**0.0815**	**(1)**	-	**(1)**	0.0000	(2)	-	**(1)**	-	**(1)**	0.0431	(2
GLMTL	0.2040	(3)	0.7790	(2)	0.0384	(3)	0.6759	(3)	0.0000	(4)	0.0000	(3)	0.0277	(4)
cvxGLMTL	-	**(1)**	-	**(1)**	**0.2606**	**(1)**	-	**(1)**	0.0181	(2)	**0.7262**	**(1)**	-	**(1)**

Table 6. Train MAE in the regression problems (smallest values in bold face).

	maj.	ten.	boston	california	abalone	crime
MAE						
CTL	3.440	4.183	1.557 ± 0.198	40502.686 ± 222.209	1.434 ± 0.019	0.055 ± 0.006
ITL	3.590	3.914	1.883 ± 0.224	34403.940 ± 83.583	1.399 ± 0.025	0.050 ± 0.004
cvxMTL	3.649	3.921	1.522 ± 0.248	35061.556 ± 118.259	1.399 ± 0.027	0.055 ± 0.007
GLMTL	**2.630**	**3.728**	2.077 ± 0.447	**33984.568 ± 151.998**	1.594 ± 0.023	**0.038 ± 0.002**
cvxGLMTL	3.344	4.141	**1.516 ± 0.270**	34409.942 ± 101.472	**1.406 ± 0.023**	0.057 ± 0.007

Given the small number of accuracy or F1 values, the validity of applying a Wilcoxon test is not guaranteed. In any case and for illustration purposes, we have combined the score (either F1 or accuracy) obtained by the models in each one of the three CV outer folds of both landmine and binding problems. We thus obtain six paired samples which we use as inputs for the Wilcoxon test; we show the resulting p values and rankings in the last column of Table 5.

Finally, when comparing the two graph based MTL approaches, GLMTL performs quite well in the classification problems but less so in the regression ones. As a possible explanation we point out to Table 6, which shows the train MAEs of each regression model. Recall that GLMTL does not have an explicit weight regularization term and, thus, may be more susceptible of overfitting the training sample. This may be the case here since, as it can be seen, GLMTL has the smallest MAE in majorca, tenerife, california and crime. In these problems, where the tasks we consider may be more informative, it seems that GLMTL overfits on them and, hence, has worse test MAE values than cvxGLMTL.

5 Discussion and Conclusions

The Multi-Task learning paradigm incorporates data from multiple sources with the goal of achieving a better generalization than that of a common model or independent models per task. The idea is to make use of all the information, but at the same time refining each model for its particular task. Multi-Task Support Vector Machines are adapted into this framework usually in two ways: either a common part shared by all tasks and a task-specific part are combined, or a graph is defined over the tasks and an independent model is fitted for each task, while trying to be similar to the models of the most related tasks. The first approach imposes the same relationship between all the tasks, while the second one allows for different degrees of task relationships but loses the common part where the information is shared across tasks. In this work we have proposed a hybrid model that combines both approaches in a convex manner by incorporating both the common part and a graph which defines the task relationships. The numerical results over eight different problems show that our proposal performs better than both previous MTL approaches, and also better than either a global model or task-independent models, while the computational cost is similar. To finish, we mention two possible venues of further research that we are pursuing. The first one would be to learn the task relationship graph by exploring the data, instead of using predefined task relation values as we have done here. The second one would be to improve on using just a single convex combination parameter for all tasks by learning task specific λ values.

Acknowledgments. With partial support from Spain's grants TIN2016-76406-P and PID2019-106827GB-I00/AEI/10.13039/501100011033. Work supported also by the UAM–ADIC Chair for Data Science and Machine Learning. We thank Red Eléctrica de España for making available solar energy data and AEMET and ECMWF for access to the MARS repository. We also gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at UAM.

References

1. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
2. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM (2004)
3. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. J. Mach. Learn. Res. **6**, 615–637 (2005)
4. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Advances in Neural Information Processing Systems, pp. 41–48 (2007)
5. Argyriou, A., Pontil, M., Ying, Y., Micchelli, C.A.: A spectral regularization framework for multi-task structure learning. In: Advances in Neural Information Processing Systems, pp. 25–32 (2008)
6. Jacob, L., Vert, J.-P., Bach, F.R.: Clustered multi-task learning: a convex formulation. In: Advances in Neural Information Processing Systems, pp. 745–752 (2009)
7. Cai, F., Cherkassky, V.: SVM+ regression and multi-task learning. In: Proceedings of the 2009 International Joint Conference on Neural Networks, IJCNN 2009, pp. 503–509. IEEE Press, Piscataway (2009)
8. Cai, F., Cherkassky, V.: Generalized SMO algorithm for SVM-based multitask learning. IEEE Trans. Neural Netw. Learn. Syst. **23**(6), 997–1003 (2012)
9. Zhang, Y., Yeung, D.-Y.: A convex formulation for learning task relationships in multi-task learning. arXiv preprint arXiv:1203.3536 (2012)

10. Lin, C.-J.: On the convergence of the decomposition method for support vector machines. IEEE Trans. Neural Networks **12**(6), 1288–1298 (2001)
11. Ruiz, C., Alaíz, C.M., Dorronsoro, J.R.: A convex formulation of SVM-based multi-task learning. In: Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E. (eds.) HAIS 2019. LNCS (LNAI), vol. 11734, pp. 404–415. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29859-3_35