Mathematics in Engineering, 5(1): 1–16.
DOI:10.3934/mine.2023020
Received: 13 February 2021
Revised: 12 January 2022
Accepted: 14 February 2022
Published: 24 March 2022

http://www.aimspress.com/journal/mine

*Research article*

# Mixed-integer quadratic programming reformulations of multi-task learning models†

**Matteo Lapucci*** **and Davide Pucci**

Department of Information Engineering, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Florence, Italy

† **This contribution is part of the Special Issue:** Mathematical aspects of machine learning
Guest Editors: Ernesto De Vito; Lorenzo Rosasco; Silvia Villa
Link: www.aimspress.com/mine/article/6026/special-articles

* **Correspondence:** Email: matteo.lapucci@unifi.it.

**Abstract:** In this manuscript, we consider well-known multi-task learning (MTL) models from the literature for linear regression problems, such as clustered MTL or weakly constrained MTL. We propose novel reformulations of the training problem for these models, based on mixed-integer quadratic programming (MIQP) techniques. We show that our approach allows to drive the optimization process up to certified global optimality, exploiting popular off-the-shelf software solvers. By computational experiments on both synthetic and real-world datasets, we show that this strategy generally leads to improvements in terms of the predictive performance of the models, if compared to the classical local optimization techniques, based on alternating minimization strategies, that are usually employed. We also suggest a number of possible extensions of our model that should further improve the quality of the obtained regressors, introducing, for example, sparsity and features selection elements.

**Keywords:** multitask learning; clustered MTL; weakly constrained MTL; MIQP; global optimization

## 1. Introduction

In a seminal work of 1997, Rich Caruana introduced the idea of the *multi-task learning* (MTL) paradigm in machine learning [12]. Multi-task learning is based on the intuitive idea that, like humans, machines may jointly learn distinct tasks that are yet somehow related one with the other. In this way, the knowledge acquired from learning one task can be exploited to improve performance with the other tasks, and vice versa. Sharing information between related tasks is a particularly useful idea

in applications where only small amounts of samples are available for each single task, but it is also effective in more general situations.

In fact, multi-task learning strategies have been successfully employed in several settings: with supervised, unsupervised or semi-supervised tasks, with reinforcement learning problems, with graphical models. Also, the range of applications is wide: computer vision [29, 32, 38], bioinformatics [25, 33, 37], natural language processing [22, 36], web applications [2, 5], ubiquitous computing [26, 41]. For the supervised learning setting alone, several different multi-task approaches have been proposed in the literature. Namely, we can list Feature-based approaches [4, 12], Low-rank approaches [1, 3], Task Clustering approaches [6, 16, 21, 24, 42, 43], Task Relation Learning approaches [35, 40] and Decomposition approaches [13, 23]. The latter four classes of methodologies are collectively referred to as *parameter-based approaches*. We refer the reader to [39] for a thorough review of multi-task learning models.

In this work, we are interested in parameter-based multi-task approaches to *regression problems* in a *homogeneous* setting, i.e., where the input space is the same for all tasks [39]. Many strategies have been proposed in the literature to tackle regression problems in a multi-task environment by linear models, employing the common squared error loss function for training.

The main contribution of this work consists in showing that, in some of the aforementioned cases, the underlying optimization problem can in fact be equivalently reformulated as a *Mixed-Integer Quadratic Programming* (MIQP) problem. MIQP solvers, like Gurobi, CPLEX, CBC or GLPK, are nowadays able to efficiently manage problems with a large number of integer variables, finding the certified global optimum. This is in contrast with the local optimization procedures, typically employed to tackle the original continuous formulations, that only attain local minima. Furthermore we argue, and also numerically show, that solving to global optimality the training problems provides benefits in terms of generalization capabilities and predictive performance of the trained models.

The manuscript is organized as follows. In Section 2, we briefly review some basic multi-task learning models from the literature. Then, we show in Section 3 how such approaches can be reformulated by employing mixed-integer programming techniques. In Section 4, we present computational experiments aiming to assess the practical advantages of solving our reformulations, compared to using classical algorithmic schemes. Finally, we draw some conclusions in Section 5. In Appendix A, we list some possible additional elements that can be taken into account within our models, whereas in Appendix B we show the results of a computational study aimed at evaluating the scalability of the proposed approaches.

## 2. Preliminaries

In this section, after introducing the notation employed in this work, we briefly review some of the most basic approaches to multi-task learning in regression problems.

### *2.1. Notation*

We are interested in multi-task linear regression problems. We are given a set of $m$ tasks $\mathcal{T}_1, \dots, \mathcal{T}_m$, each one associated with a dataset $\mathcal{D}_i = (X^i, y^i)$, $i = 1, \dots, m$. We consider the homogeneous setting, therefore $X^i = (x^i_1, \dots, x^i_{N_i})$ with $x^i_j \in \mathbb{R}^n$ for all $i = 1, \dots, m$ and all $j = 1, \dots, N_i$. We also have $y^i_j \in \mathbb{R}$, as we are considering regression tasks. For each task $\mathcal{T}_i$, we want to construct a linear regression model

$w_i \in \mathbb{R}^n$. The loss function $\mathcal{L}^i(\cdot)$ associated to model $w_i$ on the dataset $\mathcal{D}_i$ is the mean squared error:

$$\mathcal{L}^i(w_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} (w_i^T x_j^i - y_j^i)^2.$$

### 2.2. Basic continuous MTL modes

Many works in parameter-based homogeneous MTL have focused on the idea that similarity among tasks shall be transferred to the learned models by enforcing that corresponding feature weights across tasks are close to each other. With linear regression models, one of the simplest ways to enforce this requirement is by introducing an additional regularization term into the ridge regression setting, as first shown by [16, 17]:

$$\min_{w_1,\dots,w_m,\bar{w}} \sum_{i=1}^{m} \left( \mathcal{L}^i(w_i) + \lambda \|w_i\|^2 + \nu \|w_i - \bar{w}\|^2 \right), \tag{2.1}$$

where $\bar{w}$ acts as a connection term. Problem (2.1) is quadratic, convex and unconstrained, hence it is easily solvable to global optimality. However, the model has evident limitations. First, not all tasks may be related to each other, and hence enforcing proximity may in fact deteriorate predictive performance. Moreover, the assumption that weights of related models are similar is often too strong with most real-world data.

For this reason, many variants and alternatives to model (2.1) have been proposed. A first extension is the important class of *Clustered Multi-Task Learning* (CMTL) models. Evegeniou et al. [16] have shown that by a simple extension of (2.1) it is possible to retrieve the Task-clustering setting. Task-clustering models have been reformulated in many different fashions [6, 21, 24, 42, 43]. If the hard-clustering setting is considered, in which any task is associated to one and only one cluster, the following basic formulation can be considered:

$$\min_{\substack{w_1,\dots,w_m \\ z_1,\dots,z_K,\delta}} \sum_{i=1}^{m} \left( \mathcal{L}^i(w_i) + \lambda \|w_i\|^2 + \nu \sum_{k=1}^{K} \delta_{ik} \|w_i - z_k\|^2 \right), \tag{2.2}$$

where $K$ is the number of clusters, $z_k \in \mathbb{R}^n$ for all $k$ and $\delta_{ik}$, for $i = 1, \dots, m$ and $k = 1, \dots, K$, is a binary indicator variable which is set to 1 if task $i$ belongs to cluster $k$ and is 0 otherwise. In the soft-clustering setting, the problem can be formulated similarly, with variables $\delta_{ik}$ representing probability values: $\delta_{ik} \in [0, 1]$, $\sum_{k=1}^{K} \delta_{ik} = 1$.

Problem (2.2) and similar formulations such those in [24,42,43] are typically solved by Alternating Minimization, where three steps are iteratively repeated:

1) minimize the objective function with respect to model weights, having fixed clusters composition and representatives:

$$w_i^{t+1} = \arg\min_{w_i} \ \mathcal{L}^i(w_i) + \lambda \|w_i\|^2 + \nu \sum_{k=1}^{K} \delta_{ik}^t \|w_i - z_k^t\|^2 \quad \forall i = 1, \dots, m;$$

2) assign each task model to the cluster with the nearest representative (for the hard-clustering setting):

$$\delta_{ik}^{t+1} = \begin{cases} 1 & \text{if } z_k^t = \arg\min_{h=1,\dots,K} \|w_i^{t+1} - z_h^t\|^2, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i = 1,\dots,m,\ \forall k = 1,\dots,K;$$

3) set the representative of each cluster as the mean of the models belonging to that cluster:

$$z_k^{t+1} = \frac{1}{\sum_{i=1}^m \delta_{ik}^{t+1}} \sum_{i=1}^m \delta_{ik}^{t+1} w_i^{t+1} \quad \forall k = 1,\dots,K.$$

This approach can be shown to converge, but global optimality of the retrieved solution cannot be guaranteed; in fact, global optimality is rarely attained.

As an alternative to regularization-based approaches, with weaker assumptions on data and task relatedness, a *polarity-constrained multi-task learning* (or *weakly constrained MTL*, wcMTL) model has been recently proposed [30, 34]; the idea is that corresponding weights in related tasks are not necessarily close in magnitude, but reasonably share the polarity, i.e., if a feature is positively relevant to the output of a task, then its weight will also be positive for the other ones. This is modeled by the following optimization problem:

$$\begin{aligned} &\min_{w_1,\dots,w_m} \sum_{i=1}^m \left( \mathcal{L}^i(w_i) + \lambda \|w_i\|^2 \right) \\ &\text{s.t. } w_{i,j} w_{i+1,j} \geq 0 \quad \text{for all } i = 1,\dots,m-1,\ j = 1,\dots,n, \end{aligned} \tag{2.3}$$

which, again, can be solved by alternating minimization approaches such as Block Coordinate Descent (BCD) [7] or the Alternating Direction Method of Multipliers (ADMM) [10] up to stationarity. Specifically, the main steps in the ADMM loop are carried out as follows for the considered problem [34]:

1) Sequentially update models $w_i$, $i = 1,\dots,m$:

$$\begin{aligned} w_i^{t+1} = \arg\min_{w_i} \ & \mathcal{L}(w_i) + \lambda \|w_i\|^2 + \tau \|w_i - z_i^t + y_i^t/\tau\|^2 \\ & \text{s.t. } w_{i,j} w_{i+1,j}^t \geq 0 \ \forall j = 1,\dots,n; \end{aligned}$$

2) Update auxiliary variables:

$$z_i^{t+1} = w_i^{t+1} + y^t/\tau \quad \forall i = 1,\dots,m;$$

3) Compute primal and dual residuals:

$$s_i^{t+1} = z_i^{t+1} - z_i^t, \qquad r_i^{t+1} = w_i^{t+1} - z_i^{t+1} \quad \forall i = 1,\dots,m;$$

4) Update dual variables:

$$y_i^{t+1} = y^t + \tau r^{t+1} \quad \forall i = 1,\dots,m.$$

## 3. Mixed-integer reformulations

In this section, we show how problems (2.2) and (2.3) can be reformulated as MIQP problems. In particular, Task-clustering multi-task learning (2.2) can be formulated, in the hard-clustering setting, as:

$$\min_{\substack{w_1,\dots,w_m\in\mathbb{R}^n\\ z_1,\dots,z_K\in\mathbb{R}^n\\ \delta\in\{0,1\}^{m\times K},s\in\mathbb{R}^{m\times K\times n}}} \quad \sum_{i=1}^{m}\left(\mathcal{L}^i(w_i)+\lambda\|w_i\|^2+\nu\sum_{k=1}^{K}\|s_{ik}\|^2\right) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\delta_{ik}=1 \quad \forall i=1,\dots,m, \tag{3.1b}$$

$$s_{ik}\geq -M(1-\delta_{ik})e+(w_i-z_k) \quad \text{for all } i=1,\dots,m,\ k=1,\dots,K, \tag{3.1c}$$

$$s_{i_k}\leq M(1-\delta_{ik})e+(w_i-z_k) \quad \text{for all } i=1,\dots,m,\ k=1,\dots,K, \tag{3.1d}$$

where $s_{ik}\in\mathbb{R}^n$ and $M$ is a sufficiently large constant to be used in big-M type constraints and $e\in\mathbb{R}^n$ is the vector of all ones.

Constraints (3.1c)–(3.1d) are used to model the implication $\delta_{ik}=1 \implies s_{ik}=w_i-z_k$: if $\delta_{ik}=1$, i.e., the $i$-th task belongs to the $k$-th cluster, $s_{ik}$ is equal to the distance between the model and the cluster representative; this quantity will be quadratically penalized in the objective function; otherwise, $s_{ik}$ is free and, since we are minimizing its squared norm, it will be set to zero, i.e., the distance between the model and the representative is not penalized. Therefore, equivalently as in model (2.2), the squared distance between a model and its cluster representative is penalized in the optimization process.

On the other hand, problem (2.3) can be reformulated as:

$$\begin{aligned}\min_{\substack{w_1,\dots,w_m\in\mathbb{R}^n\\ y\in\{0,1\}^n}} \quad & \sum_{i=1}^{m}\left(\mathcal{L}^i(w_i)+\lambda\|w_i\|^2\right)\\ \text{s.t.} \quad & -M(1-y_j)\leq w_{i,j}\leq My_j \quad \text{for all } j=1,\dots,n,\ i=1,\dots,m.\end{aligned} \tag{3.2}$$

This time, the big-M constraint models the following implication: if $y_j=0$, then the $j$-th weight will be non positive in all models; if $y_j=1$, it will be non negative for all tasks. Basically, $y_j$ denotes the polarity sign of the $j$-th feature, which is shared among all tasks.

Since the number of introduced binary variables is limited - $K\times m$ for problem (3.1), with both $K$ and $m$ usually small, and $n$ for problem (3.2) - such formulations can be solved, in a reasonable amount of time, to certified global optimality by employing off-the-shelf mixed-integer programming solvers such as Gurobi [20]. In the following section, we will show how this is advantageous, in terms of predictive performance, w.r.t. using local optimization techniques such as Alternate Minimization or the Alternating Direction Method of Multipliers.

A remark shall be pointed out at this point; exact optimization approaches for mixed-integer problems are well-known to be computationally hard. Local algorithms are certainly cheaper and possess better scalability properties. However, the intent when multi-task approaches are employed in linear regression tasks is to obtain the best possible improvement for models that have poor performance because of the lack of training data. We are hence in a context where it is worth

employing greater computational resources in order to obtain even slight performance boosts. Nonetheless, we present in Appendix B a computational study suggesting the applicability limits of the proposed approach.

Note that models (3.1) and (3.2) are quite flexible and it is possible to introduce additional modeling elements aimed at further improving the predictive performance. For example, the two considered models can be combined in different ways; moreover, mixed-integer formulations allow to introduce sparsity and feature selection with ease. We formalize these aspects more in detail in Appendix A. However, in the computational analysis we will focus on the basic models, leaving a robust experimentation of these variants to future work.

## 4. Computational experiments

In this section we evaluate the benefits, in terms of generalization performance, of solving the basic CMTL (2.2) and wcMTL (2.3) models to global optimality using reformulations (3.1) and (3.2).

To this aim, we implemented and solved with Gurobi [20] models (3.1) and (3.2). As also suggested in Gurobi documentation, we found that directly implementing big-$M$ constraints, with the reasonable value of $M = 10000$, is computationally more convenient than employing the "indicator constraint" construct provided by the library. We then implemented in Python3 the AM and ADMM procedures to solve respectively problems (2.2) and (2.3). We employed the `numpy` library for all basic operations and the L-BFGS-B solver [11] to solve the bound-constrained subproblems in ADMM. As for the parameter setting, we set to 100 the number of iterations for the Alternate Minimization of problem (2.2), while for ADMM we set $\tau = 1000$ and the tolerance for the stopping criterion based on residual convergence to $\|s^{t+1}\| \leq 0.01$. As for the models hyperparameters, we will detail our choices case by case in the following. As starting points of the local optimization procedures, we initialized each model with the optimal solution of the least squares regression considering each task independently. Single task least-squares regression can easily be obtained by solving the normal equations.

Concerning the starting cluster assignment in CMTL, this is randomly extracted from a uniform distribution. The approach is thus nondeterministic, therefore we took into account 10 independent runs with different random cluster initializations every time we employed the Alternate Minimization method.

In the experiments we employed both synthetic and real-world benchmarks, that we describe in the following section.

### *4.1. Datasets*

#### 4.1.1. Synthetic datasets

We generated 10 datasets as follows. Each dataset contains 16 tasks, all concerning regression problems with data in $\mathbb{R}^{12}$. The size of the training set for each task is equal to $N$, where $N = 3, \ldots, 12$ varies for every dataset, while the size of the test set is 5 for all tasks and all datasets. The samples are generated from the normal distribution $\mathcal{N}_{12}(0, 1)$, while the output for a sample $x$ of the $i$-th task is given by $y = w_i^T x$, where $w_i$ is generated as follows:

$$w_i = \begin{pmatrix} w_{iA} \\ w_{iB} \\ w_{iC} \end{pmatrix} = \begin{pmatrix} \bar{w}_A + 0.3\bar{w}_A^k + 0.1\mathcal{N}_5(0,1) \\ \bar{w}_B \odot \mathcal{U}_6([0,4]) \\ \mathcal{N}_1(0,1), \end{pmatrix}$$

where $A = \{1,2,3,4,5\}$, $B = \{6,7,8,9,10,11\}$, $C = \{12\}$, $\bar{w}_A^k \sim \mathcal{N}_5(0,1)$ for $k = 1,\dots,3$, with the $i$-th task belonging to one of the $K = 3$ clusters, $\bar{w}_B \sim \mathcal{N}_6(0,1)$, $\odot$ denotes the element-wise product and $\mathcal{U}$ the uniform distribution.

### *4.2. Real-world datasets*

We considered a total of 4 real-world datasets for multi-task linear regression problems:

- `school`* [18]: the dataset concerns the estimation problem of examination scores of 15,362 students from 139 British secondary schools in the period 1985–87. Each school is treated as a task, the inputs consist of four school-specific and three student-specific attributes.
- `parkinson`† [31]: the dataset contains 5,875 data points for 42 patients suffering from the Parkinson's disease, each one being a separate task. Given 19 bio-medical features, the aim is to predict the disease progress status at different times. The target can be measured by two different technical scores: motor UPDRS and total UPDRS. We can hence obtain two different datasets: `parkinson_motor` and `parkinson_total`.
- `insurance`‡: the dataset concerns the prediction of the individual medical costs billed by health insurance. There are 1,338 data samples, with 5 features each. We used the sixth feature, i.e., the residential area in the US, to identify 4 different tasks.

### *4.3. Results*

We performed three groups of experiments. In the first one, we considered the synthetic datasets described in Section 4.1.1. We compare the test mean squared error (MSE) attained by the CMTL and the wcMTL models trained by solving the optimization problem both via mixed-integer and continuous (local) optimization procedures. We also report the result of training each task independently (single task learning, ST).

For this experiment we set to 3 the number of clusters in the task-clustering model. We set $\nu = 1$ and $\lambda = 0.01$ for all models. We recall that the results of Alternate Minimization for CMTL are the mean of 10 independent runs with different random clusters initializations. The results of the experiment are reported in Table 1. We can observe that, with the only exception of the CMTL model with the problems of size $N = 3$ and $N = 4$, solving the optimization problem to global optimality has clear benefits in terms of the predictive performance of the models. We can also observe that the CMTL model appears to be superior on this class of problems than the wcMTL. Also, we can note that both models are indeed an upgrade if compared to the single task models.

Next, we turn to the real-world datasets. We begin by evaluating the performance of the MIQP approach, compared to the local minimization approaches, on the `school` dataset for different

*`http://www.bristol.ac.uk/cmm/learning/support/datasets/`
†`https://archive.ics.uci.edu/ml/datasets/parkinsons+telemonitoring`
‡`https://www.kaggle.com/mirichoi0218/insurance`

hyperparameters settings. For this experiment, we have kept fixed the train-test split (80/20) of all the tasks, which is obtained randomly. We show the results of the experiment in Tables 2 and 3. Again, we can see that finding a better solution in terms of the training optimization problems consistently translates into benefits when it comes to generalization performance.

**Table 1.** Test MSE values obtained on the synthetic datasets by the weakly-constrained and the clustered multi-task models, trained by different algorithms. For the wcMTL we set $\lambda = 0.01$, for the CMTL we set $\lambda = 0.01$, $\nu = 1$ and $K = 3$. We also report the score obtained training the models independently (ST).

| $N$ | CMTL-MIQP | CMTL-AM | wcMTL-MIQP | wcMTL-ADMM | ST |
|---|---|---|---|---|---|
| 3 | 10.0681 | **9.5189±1.7874** | 14.0276 | 17.0070 | 16.6861 |
| 4 | 16.8705 | **13.1457±2.6602** | 25.4401 | 26.7594 | 35.3628 |
| 5 | **11.8681** | 20.7552±7.1580 | 22.6021 | 35.1270 | 39.6169 |
| 6 | **6.32260** | 10.4790±2.7821 | 15.9449 | 27.0943 | 24.8687 |
| 7 | **1.6975** | 3.6934±0.5992 | 6.1491 | 9.0188 | 10.4652 |
| 8 | **2.5645** | 2.7944±0.3428 | 4.2290 | 5.9489 | 5.2729 |
| 9 | **1.2142** | 3.5931±1.2425 | 1.7224 | 5.0305 | 5.3522 |
| 10 | **2.8012** | 5.2928±0.7232 | 5.0192 | 7.1570 | 7.9376 |
| 11 | **1.2442** | 1.9041±0.4506 | 1.8354 | 3.7030 | 3.0068 |
| 12 | **0.9910** | 1.7944±0.2132 | 2.1547 | 2.5624 | 2.1963 |

**Table 2.** Test MSE values obtained on a fixed train-test split of the `School` dataset by the weakly-constrained multi-task model, trained by MIQP and ADMM approaches for different values of $\lambda$.

| $\lambda$ | MIQP | ADMM |
|---|---|---|
| 0.005 | **112.5894** | 113.1634 |
| 0.01 | **111.4319** | 112.9807 |
| 0.05 | **108.3934** | 112.0678 |
| 0.1 | **110.4160** | 111.6596 |

**Table 3.** Test MSE values obtained on a fixed train-test split of the `School` dataset by the clustered multi-task model, trained by MIQP and AM approaches for different values of $\lambda$, $\nu$ and $K$. Note that AM is not deterministc (it depends on the starting cluster representatives), so mean and standard deviation are reported for 10 runs with different random initializations.

| ($\lambda$, $\nu$,$K$) | MIQP | AM |
|---|---|---|
| (0.05, 0.2, 2) | **106.0100** | 106.1563 ± 0.1585 |
| (0.05, 0.2, 3) | **106.2453** | 106.4122 ± 0.3213 |
| (0.05, 1, 2) | **105.3637** | 106.3416 ± 0.1816 |
| (0.05, 1, 3) | **105.3434** | 106.2400 ± 0.1337 |
| (0.05, 3, 2) | **106.4584** | 107.5607 ± 0.1299 |
| (0.05, 3, 3) | **106.2833** | 107.4739 ± 0.2285 |

Finally, we carried out a wider experiment on all 4 real-world datasets. This time we repeated the training and testing process on 10 different train-test splits for each dataset. For each run, we selected the hyperparameters of each model by a 5-fold cross-validation step on the training set. The results of the experiment are reported in Tables 4 and 5.

From Table 4 we can observe that the MIQP approach always outperforms the corresponding local minimization one, with the only exception of the polarity-constrained model on the `insurance` dataset. On the other hand, the wcMTL is competitive with the CMTL on the two `parkinson` datasets only if the mixed-integer approach is employed. We also note that Task-clustering solved as a MIQP is the overall best approach among those considered.

From Table 5, we can further see that solving the weakly-constrained model as an MIQP one is consistently advantageous as the train-test splits vary, except for the case of the `insurance` dataset. As for task-clustering, because of the nondeterministic nature of the Alternate Minimization procedure, we consider both the mean and the best MSE obtained among 10 different initializations for each split of each dataset. If the average is taken into account, the mixed-integer approach confirms to be certainly preferable than the AM approach. Even if we consider the best run of Alternate Minimization for each split, the MIQP method continues to be slightly superior.

In the end, we can conclude that basic models for multi-task linear regression should definitely be reformulated as MIQP problems, so that the global optimum of the training problem can be found with benefits in the prediction phase.

**Table 4.** Mean and standard deviation of test MSE values obtained on 10 different random train-test splits of the `school`, `parkinson-total`, `parkinson-motor` and `insurance` datasets by the CMTL and weakly-constrained MTL models, trained by different approaches. For each method, hyperparameters were selected by a 5-fold cross-validation. Note that CMTL-AM is not deterministic, so for each test split we consider the mean test MSE value obtained by 10 independent runs.

| **dataset** | **CMTL-AM** | **CMTL-MIQP** | **wcMTL-ADMM** | **wcMTL-MIQP** |
|---|---|---|---|---|
| `school` | 102.1546 ± 1.9639 | **102.0610 ± 2.4465** | 109.5408 ± 2.2877 | 108.2532 ± 2.5660 |
| `parkinson-total` | 2.0644 ± 0.0765 | **2.0544 ± 0.0832** | 2.9461 ± 0.1755 | 2.0778 ± 0.0791 |
| `parkinson-motor` | 1.6268 ± 0.0621 | **1.6266 ± 0.0609** | 1.9612 ± 0.0753 | 1.6353 ± 0.0665 |
| `insurance` | 0.2653 ± 0.029 | **0.2637 ± 0.0281** | 0.2641 ± 0.0274 | 0.2642 ± 0.0280 |

**Table 5.** Direct comparison of MIQP and local optimization approaches on real datasets in terms of predictive performance. The results take into account 10 different train-test splits for each dataset. For the nondeterministic CMTL-AM approach we consider both the best and the mean results over 10 runs.

| **dataset** | `school` | `parkinson-total` | `parkinson-motor` | `insurance` |
|---|---|---|---|---|
| **CMTL - MIQP/AM_mean wins** | 7/3 | 7/3 | 7/3 | 8/2 |
| **CMTL - MIQP/AM_best wins** | 5/5 | 7/3 | 3/7 | 7/3 |
| **wcMTL - MIQP/ADMM wins** | 8/2 | 10/0 | 10/0 | 5/5 |

## 5. Conclusions

In this work, we showed that basic multi-task learning models for linear regression can be equivalently reformulated by means of mixed-integer quadratic programming techniques. This is useful, as MIQP problems can be solved to global optimality by using off-the-shelf solvers like Gurobi, in contrast with the local optimization strategies usually employed. By a set of computational experiments, we also showed that this strategy indeed leads to a general improvement of the performance of the models at predicting out-of-sample values. In conclusion, the proposed approaches should allow practitioners to obtain stronger performance from classical MTL models with a very limited implementation effort. We shall highlight, however, that the computational cost of the proposed strategy is not just as cheap, especially as the number of employed integer variables grows (see Appendix B).

Future research should be focused on evaluating the performance of the extensions and combinations of the proposed models, such as those discussed in Appendix A. Moreover, the nontrivial extension of the proposed approach to classification problems might be considered. This could be achieved, for example, taking inspiration from works such as [8, 14, 28], where the problem of best subset selection in logistic regression is tackled by mixed-integer programming formulations.

## Acknowledgments

We would like to thank Prof. Marco Sciandrone for giving us the opportunity to work on the topic and for giving us numerous useful suggestions.

## Conflict of interest

The authors declare no conflict of interest.

## References

1. A. Agarwal, S. Gerber, H. Daume, Learning multiple tasks using manifold regularization, In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, 2010, 46–54.
2. A. Ahmed, A. Das, A. J. Smola, Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising, In: *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, 153–162.
3. R. K. Ando, T. Zhang, P. Bartlett, A framework for learning predictive structures from multiple tasks and unlabeled data, *J. Mach. Learn. Res.*, **6** (2005), 1817–1853.
4. A. Argyriou, T. Evgeniou, M. Pontil, Convex multi-task feature learning, *Mach. Learn.*, **73** (2008), 243–272. http://dx.doi.org/10.1007/s10994-007-5040-8
5. J. Bai, K. Zhou, G. Xue, H. Zha, G. Sun, B. Tseng, et al., Multi-task learning for learning to rank in web search, In: *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, 1549–1552. http://dx.doi.org/10.1145/1645953.1646169

6. A. Barzilai, K. Crammer, Convex multi-task learning by clustering, In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, 65–73.
7. D. P. Bertsekas, J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, NJ: Prentice hall Englewood Cliffs, 1989.
8. D. Bertsimas, A. King, Logistic regression: From art to science, *Statist. Sci.*, **32** (2017), 367–384. http://dx.doi.org/10.1214/16-STS602
9. D. Bertsimas, A. King, R. Mazumder, Best subset selection via a modern optimization lens, *Ann. Statist.*, **44** (2016), 813–852. http://dx.doi.org/10.1214/15-AOS1388
10. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Now Publishers Inc, 2011. http://dx.doi.org/10.1561/2200000016
11. R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, *SIAM J. Sci. Comput.*, **16** (1995), 1190–1208. http://dx.doi.org/10.1137/0916069
12. R. Caruana, Multitask learning, *Mach. Learn.*, **28** (1997), 41–75. http://dx.doi.org/10.1023/A:1007379606734
13. J. Chen, J. Zhou, J. Ye, Integrating low-rank and group-sparse structures for robust multi-task learning, In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, 42–50.
14. E. Civitelli, M. Lapucci, F. Schoen, A. Sortino, An effective procedure for feature subset selection in logistic regression based on information criteria, *Comput. Optim. Appl.*, **80** (2001), 1–32. http://dx.doi.org/10.1007/s10589-021-00288-1
15. L. Di Gangi, M. Lapucci, F. Schoen, A. Sortino, An efficient optimization approach for best subset selection in linear regression, with application to model selection and fitting in autoregressive time-series, *Comput. Optim. Appl.*, **74** (2019), 919–948. http://dx.doi.org/10.1007/s10589-019-00134-5
16. T. Evgeniou, C. A. Micchelli, M. Pontil, Learning multiple tasks with kernel methods, *J. Mach. Learn. Res.*, **6** (2005), 615–637.
17. T. Evgeniou, M. Pontil, Regularized multi–task learning, In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, 109–117. http://dx.doi.org/10.1145/1014052.1014067
18. H. Goldstein, Multilevel modelling of survey data, *Journal of the Royal Statistical Society. Series D (The Statistician)*, **40** (1991), 235–244. http://dx.doi.org/10.2307/2348496
19. A. Gómez, O. A. Prokopyev, A mixed-integer fractional optimization approach to best subset selection, *INFORMS J. Comput.*, **33** (2021), 551–565. http://dx.doi.org/10.1287/ijoc.2020.1031
20. Gurobi Optimization, LLC, Gurobi optimizer reference manual. Available from: https://www.gurobi.com/documentation/9.5/refman/index.html.
21. L. Han, Y. Zhang, Learning multi-level task groups in multi-task learning, In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, 2638–2644.
22. Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, R. Maia, Fusion of multiple parameterisations for dnn-based sinusoidal speech synthesis with multi-task learning, In: *Sixteenth annual conference of the international speech communication association*, 2015.

23. A. Jalali, S. Sanghavi, C. Ruan, P. Ravikumar, A dirty model for multi-task learning, In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, 2010, 964–972.

24. Z. Kang, K. Grauman, F. Sha, Learning with whom to share in multi-task feature learning, In: *Proceedings of the 28 th International Conference on Machine Learning*, 2011, 1–8.

25. Q. Liu, Q. Xu, V. W. Zheng, H. Xue, Z. Cao, Q. Yang, Multi-task learning for cross-platform sirna efficacy prediction: an in-silico study, *BMC Bioinformatics*, **11** (2010), 181. http://dx.doi.org/10.1186/1471-2105-11-181

26. X. Lu, Y. Wang, X. Zhou, Z. Zhang, Z. Ling, Traffic sign recognition via multi-modal tree-structure embedded multi-task learning, *IEEE Trans. Intell. Transp. Syst.*, **18** (2016), 960–972. http://dx.doi.org/10.1109/TITS.2016.2598356

27. R. Miyashiro, Y. Takano, Mixed integer second-order cone programming formulations for variable selection in linear regression, *Eur. J. Oper. Res.*, **247** (2015), 721–731. http://dx.doi.org/10.1016/j.ejor.2015.06.081

28. T. Sato, Y. Takano, R. Miyashiro, A. Yoshise, Feature subset selection for logistic regression via mixed integer optimization, *Comput. Optim. Appl.*, **64** (2016), 865–880. http://dx.doi.org/10.1007/s10589-016-9832-2

29. C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, W. Gao, Multi-task learning with low rank attribute embedding for person re-identification, In: *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, 3739–3747. http://dx.doi.org/10.1109/ICCV.2015.426

30. J. Torres, G. Bai, J. Wang, L. Zhao, C. Vaca, C. Abad, Sign-regularized multi-task learning, 2021, arXiv:2102.11191.

31. A. Tsanas, M. Little, P. McSharry, L. Ramig, Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests, *IEEE Trans. Biomed. Eng.*, **57** (2010), 884–893. http://dx.doi.org/10.1109/TBME.2009.2036000

32. H. Wang, F. Nie, H. Huang, S. Risacher, C. Ding, A. J. Saykin, et al., Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance, In: *2011 International Conference on Computer Vision*, 2011, 557–562. http://dx.doi.org/10.1109/ICCV.2011.6126288

33. H. Wang, F. Nie, H. Huang, J. Yan, S. Kim, S. Risacher, et al., High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction, In: *Proceedings of the 25th International Conference on Neural Information Processing Systems* , 2012, 1277–1285.

34. J. Wang, L. Zhao, L. Wu, Multi-convex inequality-constrained alternating direction method of multipliers, 2019, arXiv:1902.10882.

35. C. Williams, E. V. Bonilla, K. M. Chai, Multi-task gaussian process prediction, In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007, 153–160.

36. Z. Wu, C. Valentini-Botinhao, O. Watts, S. King, Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis, In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, 4460–4464. http://dx.doi.org/10.1109/ICASSP.2015.7178814

37. Q. Xu, S. J. Pan, H. H. Xue, Q. Yang, Multitask learning for protein subcellular location prediction, *IEEE/ACM Trans. Comput. Bi*, **8** (2010), 748–759. http://dx.doi.org/10.1109/TCBB.2010.22
38. T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, 2042–2049. http://dx.doi.org/10.1109/CVPR.2012.6247908
39. Y. Zhang, Q. Yang, A survey on multi-task learning, *IEEE Trans. Knowl. Data Eng.*, in press. http://dx.doi.org/10.1109/TKDE.2021.3070203
40. Y. Zhang, D.-Y. Yeung, A regularization approach to learning task relationships in multitask learning, *ACM Trans. Knowl. Discov. Data*, **8** (2014), 1–31. http://dx.doi.org/10.1145/2538028
41. J. Zheng, L. M. Ni, Time-dependent trajectory regression on road networks via multi-task learning, In: *Proceedings of the twenty-seventh AAAI conference on artificial intelligence*, 2013, 1048–1055.
42. J. Zhou, J. Chen, J. Ye, Clustered multi-task learning via alternating structure optimization, *Adv. Neural Inf. Process. Syst.*, **24** (2011), 702–710.
43. Q. Zhou, Q. Zhao, Flexible clustered multi-task learning by learning representative tasks, *IEEE Trans. Pattern Anal.*, **38** (2015), 266–278. http://dx.doi.org/10.1109/TPAMI.2015.2452911

## A. Extensions

In this Appendix, we show possible extensions of the basic models (2.2) and (2.3) that mixed-integer modeling makes possible to handle.

Firstly, we highlight that the CMTL and the wcMTL models might be combined in several ways:

- Trivially, corresponding weights could be simultaneously forced to share the sign and to be close to each other if the tasks belong to the same cluster. We would have model (3.1) with the addition of the constraints of (3.2).
- Some weights could be forced to be close while others to only share the polarity. This requirement can be modeled by the following formulation:

$$\begin{aligned}
\min_{\substack{w_1,\dots,w_m\in\mathbb{R}^n\\ z_1,\dots,z_K\in\mathbb{R}^{n_c}\\ \delta\in\{0,1\}^{m\times K}, s\in\mathbb{R}^{m\times K\times n_c}\\ y\in\{0,1\}^{n_p}}} \quad & \sum_{i=1}^{m}\left(\mathcal{L}^i(w_i)+\lambda\|w_i\|^2+\nu\sum_{k=1}^{K}\|s_{ik}\|^2\right)\\
\text{s.t.}\ & \sum_{k=1}^{K}\delta_{ik}=1 \quad \forall i=1,\dots,m,\\
& w_i=(w_{i,C},w_{i,P}), \quad w_{i,C}\in\mathbb{R}^{n_c},\ w_{i,P}\in\mathbb{R}^{n_p} \quad \text{for all } i=1,\dots,m\\
& s_{ik}\geq -M(1-\delta_{ik})e+(w_{i,C}-z_k) \quad \text{for all } i=1,\dots,m,\ k=1,\dots,K,\\
& s_{i_k}\leq M(1-\delta_{ik})e+(w_{i,C}-z_k) \quad \text{for all } i=1,\dots,m,\ k=1,\dots,K,\\
& -M(1-y_j)\leq w_{i,j}\leq My_j \quad \text{for all } j=n_c+1,\dots,n_c+n_p,\ i=1,\dots,m.
\end{aligned}$$

- Sign constraints could be based on the clusters structure. The constraints would have the following form:

$$\sum_{k=1}^{K} \delta_{ik} = 1 \quad \forall i = 1, \dots, m,$$
$$-M(2 - y_{jk} - \delta_{ik}) \leq w_{i,j} \leq M(1 + y_{jk} - \delta_{ik}) \quad \text{for all } j = 1, \dots, n,\ i = 1, \dots, m,\ k = 1, \dots, K.$$

A second modeling element that could be handled by mixed-integer approaches concerns sparsity and feature selection. Indeed, the best feature subset selection task in linear regression problems has been tackled by MIQP formulations in many works in the recent years [9, 15, 19, 27].

Now, in the multi-task setting, we can either:

- force a feature selection which is in common for all tasks by adding to the model binary variables $\nu$ and the following constraints:

$$-M\nu_j \leq w_{i,j} \leq M\nu_j \quad \text{for all } i = 1, \dots, m,\ j = 1, \dots, n,$$
$$\sum_{j=1}^{n} \nu_j \leq S;$$

- select a different set of relevant features for each cluster:

$$-M(1 + \nu_{jk} - \delta_{ik}) \leq w_{i,j} \leq M(1 + \nu_{jk} - \delta_{ik}) \quad \text{for all } i = 1, \dots, m,\ j = 1, \dots, n,$$
$$\sum_{j=1}^{n} \nu_{jk} \leq S \quad \text{for all } k = 1, \dots, K;$$

- select a different set of relevant features for each individual task:

$$-M\nu_{ij} \leq w_{i,j} \leq M\nu_{ij} \quad \text{for all } i = 1, \dots, m,\ j = 1, \dots, n,$$
$$\sum_{j=1}^{n} \nu_{ij} \leq S \quad \text{for all } i = 1, \dots, m.$$

Note that the latter approach introduces $n \times m$ binary variables, which may excessively increase the computational cost of the approach.

Employing analogous mechanisms, it is also possible to select variables to which impose the polarity constraints and those to consider when forcing cluster compactness.

## B. Scalability of the proposed approach

In this Appendix we focus on scalability issues regarding our proposed approaches. The hardness of mixed-integer mathematical programming problems is well known: the computational cost of exact

solvers for mixed-integer problems grows significantly with the number of integer variables. This still holds true with the powerful modern software developed in recent years.

We are therefore interested in providing readers with an insight on the computational resources demanded by the mixed-integer approaches and the extent it may be practically sustainable.

To this aim, we generated two new problems benchmarks. The first one is designed for testing the CMTL model: we generated MTL problems with $n = 8$ features, 4 examples per task and a variable number of tasks $m = 2, \dots, 30$. The number of clusters $K$ was fixed to 3, so that the problems have a minimum of 6 and a maximum number of 90 binary variables. We also set $M = 1000$, $\lambda = 0.01$, $\nu = 1$.

As for the wcMTL case, we generated problems with $m = 15$ tasks, $n = 3t$ features for $t = 3, \dots, 30$ and $N = \lfloor n/2 \rfloor$ examples per task; the number of binary variables is determined by $n$, so we have again problems with up to 90 integer variables. Here, we set $M = 500$ and $\lambda = 0.01$.

For both benchmarks, we extracted samples $x_j^i$ from the uniform distribution $\mathcal{N}_n(0, 1)$ whereas labels are generated computing $y_j^i = (w_{\text{common}} + 0.1 w_j)^T x_j^i + \mathcal{N}(0, 0.2)$, where $w_{\text{common}}, w_j \sim \mathcal{N}_m(0, 1)$.

The results of the experiments, performed running Gurobi 9.1.0 on a computer with Ubuntu Server 20.04 LTS OS, Intel Xeon E5-2430 v2 @ 2.50GHz CPU, 12 cores and 16GB RAM, are reported in Figures 1 and 2.

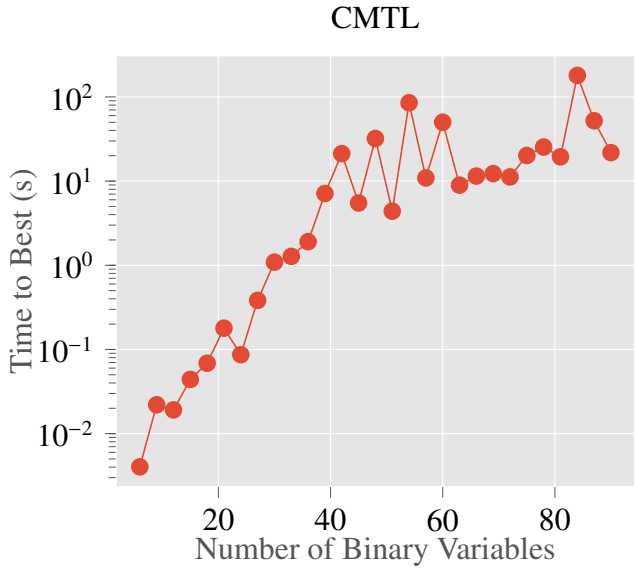


**Figure 1.** Computing time required by Gurobi to find the globally optimal solution to problem (3.1) for a number of binary variables from 6 up to 90. Note that we report here the time required to get to the global optimum, not taking into account the additional time required to certify optimality.

We can observe that, for problems of the considered size, globally optimal solutions can be obtained in a reasonable amount of time (~$10^2$-$10^3$s for the hardest problems); nonetheless, the order of magnitude of the cost, measured by CPU time, not surprisingly grows quite fast with the number of integer variables. This fact makes us guess that problems with hundreds of integer variables may become computationally unsustainable to be solved by the proposed approaches; we also shall note that we did not test problems of larger size, as certifying global optimality of the solutions becomes impractical.

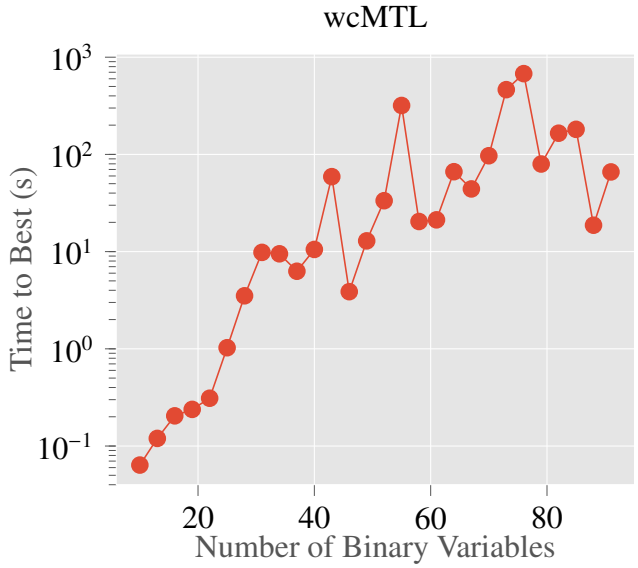


**Figure 2.** Computing time required by Gurobi to find the globally optimal solution to problem (3.2) for a number of binary variables from 9 up to 90. Note that we report here the time required to get to the global optimum, not taking into account the additional time required to certify optimality.

In conclusion, the proposed methods appear to be reasonably employable in the typical MTL use case where few data are available and we want to obtain the most accurate possible model out of them. As the size of the problem grows, the computational cost likely becomes significant and the approach may become unsustainable in very large scale scenarios. However, we shall note that in our experiments we employed a very efficient, yet general purpose off-the-shelf solver; specific branching or bounding strategies for our problem may be able to further improve the performance of the training procedure.