Group LASSO with Asymmetric Structure Estimation for Multi-Task Learning

Saullo H. G. de Oliveira[1] , **André R. Gonçalves**[2] and **Fernando Von Zuben**[1]
[1]School of Electrical and Computer Engineering - FEEC, University of Campinas - Unicamp, Brazil
[2]Lawrence Livermore National Laboratory, USA
{shgo, vonzuben}@dca.fee.unicamp.br, goncalves1@llnl.gov

Abstract

Group LASSO is a widely used regularization that imposes sparsity considering groups of covariates. When used in Multi-Task Learning (MTL) formulations, it makes an underlying assumption that if one group of covariates is not relevant for one or a few tasks, it is also not relevant for all tasks, thus implicitly assuming that all tasks are related. This implication can easily lead to negative transfer if this assumption does not hold for all tasks. Since for most practical applications we hardly know a priori how the tasks are related, several approaches have been conceived in the literature to (i) properly capture the transference structure, (ii) improve interpretability of the tasks interplay, and (iii) penalize potential negative transfer. Recently, the automatic estimation of asymmetric structures inside the learning process was capable of effectively avoiding negative transfer. Our proposal is the first attempt in the literature to conceive a Group LASSO with asymmetric transference formulation, looking for the best of both worlds in a framework that admits the overlap of groups. The resulting optimization problem is solved by an alternating procedure with fast methods. We performed experiments using synthetic and real datasets to compare our proposal with state-of-the-art approaches, evidencing the promising predictive performance and distinguished interpretability of our proposal. The real case study involves the prediction of cognitive scores for Alzheimer's disease progression assessment. The source codes are available at GitHub.

1 Introduction

Multi-task learning (MTL) deals with the problem of learning multiple related tasks simultaneously in such a way that similar tasks can share information with each other. By using this interplay between tasks we can improve the overall performance of learning models [Caruana, 1997; Baxter, 1997; Thrun and O'Sullivan, 1996].

In real world scenarios where the tasks present groups of coupled features, the Group LASSO regularization has been widely used to encourage group sparsity across tasks [Liu *et al.*, 2009; Wang *et al.*, 2012; Liu *et al.*, 2018]. The drawback so far is that these methods do not estimate a transference structure among tasks, using regularization techniques to enforce a priori knowledge into the transference scheme.

In the structure estimation literature, several proposals to capture the tasks interplay have been presented: estimating a transference structure imposing a shared prior over the precision matrix of tasks parameters [Zhang and Yeung, 2010; Gonçalves *et al.*, 2016]; clustering/grouping tasks in a space [Kumar and Daumé, 2012]; using local learning methods in a k-nearest-neighbor fashion [Zhang, 2013]; and sharing information regarding tasks losses [Lee *et al.*, 2016]. These structure estimation mechanisms in MTL have not only improved the overall performance on individual tasks, but the estimated task relationship has also proven to be helpful on the comprehension of underlying processes expressed in the data. Despite estimating a transference structure, these methods learn a symmetric task relationship structure, imposing the amount of information transferred from task A to B to be equal to that transferred from task B to A, which might not be a valid assumption. It is also likely that two tasks might only be related at a particular group of covariates and completely unrelated at other groups.

In an attempt to acquire the best from those formulations, more specifically: (i) properly capture the task transference structure, (ii) account for different task relations for each group of covariates, and (iii) promote asymmetric sharing between tasks; our proposal is the first initiative in the literature to conceive a Group LASSO formulation for MTL with an asymmetric structure estimation. A relationship matrix is learned for each group of covariates, allowing a more flexible and possibly more realistic model.

2 Related Work

The Group LASSO regularization (standard and latent versions [Yuan and Lin, 2006; Jacob *et al.*, 2009]) was proposed to allow sparse solutions for applications where the feature set is composed of grouped features. For instance, suppose we want to map a brain imaging dataset to some condition, a classification task indicating if the condition is present or not. We also know that features representing nearby areas of the brain are related and can be tagged into Regions of Interest (ROI). Group LASSO allows us to embody this information inside the model by treating each ROI as a group of features.

In MTL literature, several approaches have employed Group LASSO as a way to deal with grouped features. [Liu *et al.*, 2009] proposed a model that encourages group-structured sparsity across all tasks, exploring potential parameter coupling between tasks. Extending [Liu *et al.*, 2009], [Wang *et al.*, 2012] also enforces sparsity within each group. MT-SGL [Liu *et al.*, 2018] uses a group-based approach as in [Wang *et al.*, 2012] but decoupling the tasks and encouraging sparsity in a feature level across all tasks. In other words, each task is free to find its own sparsity pattern at a group level, but each feature is coupled among all tasks.

Other methods do not consider group information but estimate a structure that relates tasks to each other. MTRL [Zhang and Yeung, 2010] propose a convex formulation in which a matrix-variate prior distribution is placed on the task coefficients to model task relationship. In [Gonçalves *et al.*, 2016] a sparse precision matrix is learned from the data to capture tasks relationship and help to isolate unrelated tasks. A LASSO penalty is also applied to task parameters for automatic feature selection. The model uses a semi-parametric Copula distribution as prior for the tasks parameter matrix, thus also capturing non-linear correlation among tasks.

Another direction is to model transference among tasks using a latent basis, where each task is represented by a linear combination of the basis vectors. In MTFL [Kang *et al.*, 2011], tasks are grouped into a pre-defined number of disjoint groups and each feature is coupled with all tasks of the same group using an $l_{(2,1)}$-norm [Argyriou *et al.*, 2008]. Both [Kumar and Daumé, 2012] and [Kang *et al.*, 2011] recover a latent basis with no direct interpretation. AMTL [Lee *et al.*, 2016] estimates an asymmetric transference matrix where more confident tasks may transfer more information to less confident ones than the converse.

We propose a Group LASSO formulation for MTL that estimates an asymmetric transference structure at a group level: for each group of features, we learn an asymmetric task relationship matrix. Considering that tasks may relate to each other in different ways for different groups, our model brings more flexibility than the presented methods that can estimate a transference structure. Compared to other Group LASSO models for MTL, we explicitly learn the relationship of the tasks without any strong assumption.

Notation: Matrices are represented using uppercase letters, while scalars are represented by lowercase letters. Vectors are lowercase in bold. For any matrix A, $\mathbf{a}_{\bar{i}}$ is the i-th row of A, and $\mathbf{a}_j$ is the j-th column. Also, a_{ij} is the scalar at row i and column j of A. The i-th element of any vector $\mathbf{a}$ is represented by $(\mathbf{a})_i$. $\mathbf{I}_n$ is the identity matrix of size $n \times n$. For any two vectors $\mathbf{x}, \mathbf{y}$ the Hadamard product is denoted by $(\mathbf{x} \odot \mathbf{y})_i = (\mathbf{x})_i(\mathbf{y})_i$.

3 The GAMTL Formulation

The Group Asymmetric Multi-Task Learning (GAMTL) formulation is presented in what follows. Let T be the number of tasks and $\mathcal{T} = \{1, \cdots, T\}$ the set of task indices. For each task t, the data consists of the design matrix $X_t \in \mathbb{R}^{m_t \times n}$ and the vector of labels $y_t \in \mathbb{R}^{m_t}$. Let $\mathcal{G} = \{1, \cdots, G\}$ be the set of groups. A group $g \in \mathcal{G}$ defined as $g \subseteq \{1, \cdots, n\}$ is a group of covariates of $X_t, \forall t \in \mathcal{T}$, with cardinality $|g|$ containing related covariates that should be penalized together. As an example, consider again the case where our dataset is composed of brain images annotated with Regions Of Interest. Each feature could be a pixel in the image, and a group of covariates contains several pixels of the same ROI.

Let $W \in \mathbb{R}^{n \times T}$ be the parameter matrix, where each column $\mathbf{w}_t$ represents the parameters of task t. $W^g \in \mathbb{R}^{n \times T}$ is the parameter matrix restricted to group g, where $(\mathbf{w}^g)_i = (\mathbf{w})_i$ when $i \in g$, and $(\mathbf{w}^g)_i = 0$ otherwise. When the groups overlap, we assume that the adequate columns of X_t are duplicated, and W is set accordingly [Jacob *et al.*, 2009].

To model the relationship among tasks in an explainable manner, we assume that the parameters of task t can be represented by a sparse linear combination of the parameters of the other tasks, considering each group of attributes independently, i. e., $\mathbf{w}_t^g \approx W^g \mathbf{b}_t^g, \forall g$. Let $B^g \in \mathbb{R}^{T \times T}$ be the relationship matrix of group g, where b_{ij}^g represents how much task i contributes to task j in group g. A column $\mathbf{b}_i^g$ indicates how much all tasks contribute to task i, while a row $\mathbf{b}_{\bar{i}}^g$ indicates how much task i contributes to all other tasks. Let $\mathcal{L} : \mathcal{R}^n \to \mathcal{R}$ be a suitable task specific convex loss function, e.g., squared loss for regression or logistic loss for classification, the optimization problem associated with GAMTL is:

$$
\begin{aligned}
\min_{W, B^g} \quad & \sum_{t \in \mathcal{T}} \frac{1}{m_t} (1 + \lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_{\bar{t}}^g\|_1) \mathcal{L}(\mathbf{w}_t) + \\
& \frac{\lambda_2}{2} \|\mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g\|_2^2 + \lambda_3 \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2 \qquad (1) \\
\text{subject to} \quad & \mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{w}_t^g \\
& \mathbf{b}_t^g \geq 0, \forall g \in \mathcal{G} \text{ and } t \in \mathcal{T}
\end{aligned}
$$

where λ_1, λ_2, and λ_3 are regularization hyper-parameters. The normalizing factor $\frac{1}{m_t}$ avoids that tasks with a large number of samples dominate the entire cost function. d_g is usually set to $\sqrt{|g|}$ to account for group sizes in the overall function.

The first term of Eq. (1) considers the loss function and uses it to weight all transferences from t to other tasks (row t of B^g). It learns the task parameters while avoiding transference from tasks with a higher cost to tasks with lower cost. The l_1 penalization in each row of B^g enforces a sparse subset of tasks on the combination. The loss also strengthens this penalization: the higher the loss of a task, the higher the penalization. The second term enforces the transference between tasks at the group level. This is achieved by penalizing the Euclidean distance between a task parameter vector and its estimate given by the linear combination of the parameters of the other tasks. The third term and the constraint on $\mathbf{w}_t$ account for the latent Group LASSO regularization [Jacob *et al.*, 2009]. The second restriction ensures that all values in our transference structure are positive. Figure (1) shows the structural configuration of the model parameters, which will be estimated from data (X, y) from all T tasks.

Problem (1) integrates our goals into one formulation: estimating task parameters with a transference structure among

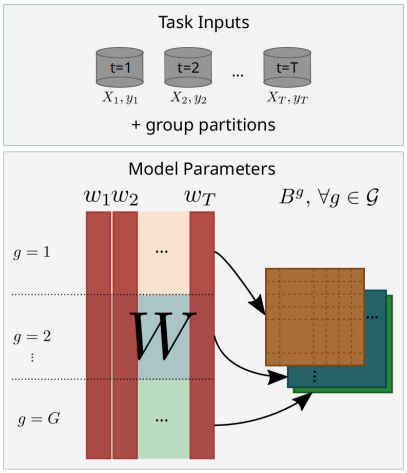


Figure 1: Tasks data input (covariates/labels for each task, and group information), and GAMTL model parameters (W and $B^g \forall g \in \mathcal{G}$).

all tasks, at the group level. Considering the simultaneous adjustment of all parameters, the problem is not jointly convex. However, when optimizing (1) in terms of $\mathbf{w}_t$, while holding $\mathbf{b}_t^g$ fixed, and vice versa, the overall non-convex problem becomes two easier-to-handle convex problems. The resulting problems are solved in an alternating optimization manner in $\mathbf{w}_t$ and $\mathbf{b}_t^g, \forall g \in \mathcal{G}, t \in \mathcal{T}$. The complete process is presented in Algorithm (1).

When $\lambda_1 = 0$, $\lambda_2 = 0$, and $\lambda_3 = 0$, independent Single Task Learning (STL) linear models are recovered. If only $\lambda_3 \neq 0$ we still have independent linear models per task but with Group LASSO regularization active. When $\lambda_2 \neq 0$ transference between tasks will occur, with λ_1 controlling the sparsity of the transference. Eq. (1) allows two variants: with and without the second constraint. Restricting or not the values of B^g will depend on the application and on the meaning of a task being negatively related with other tasks. Compared to other MTL algorithms such as MTFL, MTRL, and AMTL, GAMTL has only one additional parameter while providing an explainable transference structure for each group.

Algorithm 1 GAMTL

```
Initialize W ~ N(0, I_|T|) and set B^g = 0, ∀g ∈ G
while convergence not reached do
    for t = 1, ··· , T do
        w_t ← argmin_{w_t} Eq. 2
    end for
    for t = 1, ··· , T do
        for g ∈ G do
            b_t^g ← argmin_{b_t^g} Eq. 4
        end for
    end for
end while
```

3.1 Solving for $\mathbf{w_t}$

Isolating Eq. (1) in terms of $\mathbf{w}_t, t = 1, 2, \cdots, T$, we have:

$$\min_{\mathbf{w}_t} \frac{1}{m_t}(1+\lambda_1 \sum_{g\in\mathcal{G}} \|\mathbf{b}_t^g\|_1)\mathcal{L}(\mathbf{w}_t) + \frac{\lambda_2}{2}\|\mathbf{w}_t - \sum_{g\in\mathcal{G}} W^g \mathbf{b}_t^g\|_2^2 + \frac{\lambda_2}{2} \sum_{s\in\mathcal{T}\setminus t} \|\tilde{\mathbf{w}}_s - \sum_{g\in\mathcal{G}} \mathbf{w}_t^g b_{ts}^g\|_2^2 + \lambda_3 \sum_{g\in\mathcal{G}} d_g \|\mathbf{w}_t^g\|_2, \tag{2}$$

where

$$\tilde{\mathbf{w}}_s = \mathbf{w}_s - \sum_{u\in\mathcal{T}\setminus\{s,t\}} \sum_{g\in\mathcal{G}} \mathbf{w}_u^g b_{us}^g.$$

To solve Eq. (2) we use the accelerated proximal method FISTA [Beck and Teboulle, 2009]. We decompose our objective function into $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, both closed proper convex functions, f being L-Lipschitz continuous while h being non-differentiable:

$$f(\mathbf{w}_t) = \frac{1}{m_t}(1 + \lambda_1 \sum_{g\in\mathcal{G}} \|\mathbf{b}_t^g\|_1)\mathcal{L}(\mathbf{w}_t) + \frac{\lambda_2}{2}\|\mathbf{w}_t - \sum_{g\in\mathcal{G}} W^g \mathbf{b}_t^g\|_2^2 + \frac{\lambda_2}{2} \sum_{s\in\mathcal{T}\setminus t} \|\tilde{\mathbf{w}}_s - \sum_{g\in\mathcal{G}} \mathbf{w}_t^g b_{ts}^g\|_2^2. \tag{3}$$

Function h is the group LASSO regularization

$$h(\mathbf{w}_t) = \lambda_3 \sum_{g\in\mathcal{G}} d_g \|\mathbf{w}_t^g\|_2.$$

The proximal operator for the group LASSO regularization is

$$\mathbf{prox}_{\lambda h}(\mathbf{w}^g) = \begin{cases} \sum_{g\in\mathcal{G}} \mathbf{w}^g \frac{(\|\mathbf{w}^g\|_2 - d_g)}{\|\mathbf{w}^g\|_2} & \|\mathbf{w}^g\|_2 \geq \lambda d_g \\ 0 & \text{otherwise.} \end{cases}$$

We estimate the L constant with a backtracking procedure.

3.2 Solving for $\mathbf{b}_t^g$

Since a task cannot be represented by itself, $b_{tt}^g = 0$. Isolating Eq. (1) in terms of $\mathbf{b}_t^g$, let $\tilde{\mathbf{w}}_t = \mathbf{w}_t - \sum_{\tilde{g} \in \mathcal{G}\setminus g} W^{\tilde{g}} \mathbf{b}_t^{\tilde{g}}$, and let $\overline{W}^g = [\mathbf{w}_1^g/\mathcal{L}(\mathbf{w}_1), \cdots, \mathbf{w}_T^g/\mathcal{L}(\mathbf{w}_T)]$. The resulting problem is:

$$\begin{aligned} \min_{\mathbf{b}_t^g} \quad & \frac{1}{2}\|\tilde{\mathbf{w}}_t - \overline{W}^g \mathbf{b}_t^g\|_2^2 + \frac{\lambda_1}{\lambda_2}\|\mathbf{b}_t^g\|_1 \\ \text{subject to} \quad & \mathbf{b}_t^g \geq 0, \forall g \in \mathcal{G} \text{ and } t \in \mathcal{T}. \end{aligned} \tag{4}$$

This problem is similar to the Adaptive LASSO [Zou, 2006]. Without the constraints in Eq. (4), it can be solved using any standard method for LASSO. Here we will derive the case where the constraints are required, using the Alternating Direction Method of Multipliers (ADMM) [Boyd *et al.*, 2011]. In the ADMM framework, the inequality constraint can be transformed by means of an indicator function:

$$\begin{aligned} \min \quad & f(\mathbf{x}) + h_1(\mathbf{z}_1) + h_2(\mathbf{z}_2) \\ \text{subject to} \quad & \mathbf{x} = \mathbf{z}_1 \\ & \mathbf{x} = \mathbf{z}_2 \end{aligned} \tag{5}$$

where $h_1 = h$, and $h_2(\mathbf{z}_2)$ is defined as

$$h_2(\mathbf{z}_2) = \mathbf{1}_{\mathbb{R}_+}(\mathbf{z}_2) = \begin{cases} 0 & , \mathbf{z}_2 \geq 0 \\ +\infty & , \text{otherwise.} \end{cases}$$

The augmented Lagrangian of Formulation (5) is then, $L_{\rho_1,\rho_2} = L_{\rho_1,\rho_2}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{u}_1, \mathbf{u}_2)$:

$$\begin{aligned} L_{\rho_1,\rho_2} =& f(\mathbf{x}) + h_1(\mathbf{z}_1) + h_2(\mathbf{z}_2) \\ &+ \frac{\rho_1}{2}\left(\|\mathbf{x} - \mathbf{z}_1 + \mathbf{u}_1\|_2^2 - \|\mathbf{u}_1\|_2^2\right) \\ &+ \frac{\rho_2}{2}\left(\|\mathbf{x} - \mathbf{z}_2 + \mathbf{u}_2\|_2^2 - \|\mathbf{u}_2\|_2^2\right) \end{aligned}$$

The ADMM updating steps are:

$$\mathbf{z}_i^{k+1} := \underset{\mathbf{z}_i}{\text{argmin}} \left(h_i(\mathbf{z}_i) + \frac{\rho_i}{2}\|\mathbf{x}^k - \mathbf{z}_i + \mathbf{u}_i^k\|_2^2\right) , i = \{1, 2\}$$

$$\mathbf{x}^{k+1} := \underset{\mathbf{x}}{\text{argmin}} \left(f(\mathbf{x}) + \sum_{j=1}^{2} \frac{\rho_j}{2}\|\mathbf{x} - \mathbf{z}_j^{k+1} + \mathbf{u}_i^k\|_2^2\right)$$

$$\mathbf{u}_i^{k+1} := \mathbf{u}_i^k + \mathbf{x}^{k+1} - \mathbf{z}_i^{k+1} \quad , i = \{1, 2\}$$

Notice that the two steps in $\mathbf{z}_i$-update are executed in parallel. The same occurs for $\mathbf{u}_i$. The $\mathbf{z}_i$-update steps are solved by the proximal operators: soft-thresholding, $S_\kappa(\mathbf{a}) = (1 - \kappa/|\mathbf{a}|)_+\mathbf{a}$; and projection onto the non-negative orthant $\mathbb{R}_+$, $S(\mathbf{a}) = (\mathbf{a})_+ = \max(0, \mathbf{a})$. The $\mathbf{x}$-update step is a convex problem with a differentiable function f plus quadratic terms, which can be solved in closed-form via Cholesky decomposition or by any gradient-based method. The Python code associated with GAMTL is available online [1].

3.3 Complexity Analysis

The complexity of an iteration of GAMTL is driven by the steps 4 and 8 of the Algorithm (1), which involve a FISTA and an ADMM execution, respectively.

For step 4, we compute ∇f and $\mathbf{prox}_{\lambda g}$. The overall cost of the proximal operator is $G[g_{max}]^2 n$, where g_{max} is the size of the largest group; and to compute the derivative of Eq. 3 we need $T^2 G g_{max}$ flops. Bigger costs involved in the gradient computation are in order of $T^2 G n$, with other negligible costs. The overall cost of the full computation of ∇f is then $\mathcal{O}(T^2 G n)$. Therefore, a FISTA iteration has then a total cost of $\mathcal{O}(T^2 G n)$.

In step 8, we prepare $\tilde{\mathbf{w}}_t$ using $GTn + n$ flops. For $\overline{W}^g$, we compute the loss function of each task with cost of $n^2 + mn$, and it is reused for all iterations over the same g. ADMM requires the computation of a soft-thresholding operator, the projection of z, and the update of u. All with negligible costs. Solving the $\mathbf{x}$-update in closed-form via Cholesky decomposition uses T^3 flops, with a back-solve cost of n^2. This results in a overall cost of Tn^2 when considering $n > T$. The cost of a complete ADMM iteration is on order of $\mathcal{O}(Tn^2)$.

In summary, one iteration of GAMTL consists of T FISTA and GT ADMM executions. Therefore, setting a fixed number of iterations, the overall GAMTL time complexity of $\mathcal{O}(T^3 G n + T^2 G n^2)$.

[1] https://github.com/shgo/gamtl

4 Experiments and Discussion

For all experiments we denote GAMTLnr as GAMTL without considering the constraints on $B^g, \forall g \in \mathcal{G}$.

4.1 Artificial Dataset

To illustrate the components of our proposal and validate the model, we designed an artificial dataset as follows. We generate 8 regression tasks with 50 attributes partitioned into groups $g_1 = [1, \ldots, 25]$ and $g_2 = [26, \cdots, 50]$. For tasks $t = [1, 2]$, $\mathbf{w}_t^1 \sim \mathcal{N}(0, \mathbf{I}_{25})$ while $\mathbf{w}_t^2 = 0$. The opposite holds for tasks $t = [3, 4]$, where $\mathbf{w}^2 \sim \mathcal{N}(0, \mathbf{I}_{25})$ while $\mathbf{w}^1 = 0$. For the last four tasks $t = [5, \cdots, 8]$, $\mathbf{w}_t^1 = W_{[1,2]}^1 \mathbf{b}_t^1$ and $\mathbf{w}_t^2 = W_{[3,4]}^2 \mathbf{b}_t^2$, where each $\mathbf{b}_t^g$ is sampled from a truncated Gaussian distribution, having positive values. The first column of Figure 3 depicts these vectors concatenated as B^g matrices. For each task t, $X_t \sim \mathcal{N}(0, \mathbf{I}_{50})$, and $\mathbf{y}_t = X_t \mathbf{w}_t + \sigma$, where $\sigma = 0.3$ in the first four tasks, and $\sigma = 0.9$ in the last four tasks. This difference in the amount of noise makes the derived tasks more difficult to be solved. In this case, we expect the transference to occur from tasks with low cost to the tasks with a higher cost, thus recovering the transference structure.

The number of samples varied from 30 to 100, by steps of 10 samples. We split the dataset so that 70% of the samples are used for training and 30% for testing. For each amount of samples, the parameters of all methods were chosen by cross-validation using 30% of the training set. The best performing parameters are selected, and we repeat the training process 30 times. As λ_3 directly impacts the group sparsity, we can use results from the parameter tuning of Group LASSO to aid this selection: as λ_1 and λ_2 are related, it is possible to express one as a function of the other, resulting in just one parameter to choose in the end. However, in practice, setting each parameter independently led to better performance. Our recommendation is to choose initial values for λ_1 and λ_2 in a similar range but independently from λ_3.

The performance of all methods are compared by the normalized mean squared error (NMSE) metric, defined as

$$NMSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{t=1}^{T}(\|\mathbf{y}_t - \hat{\mathbf{y}}_t\|_2^2)/\sigma(\mathbf{y}_t)}{\sum_{t=1}^{T} m_t}.$$

where $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ are the true and predicted labels for task t, respectively. We considered LASSO [Tibshirani, 1996] and Group LASSO [Jacob *et al.*, 2009] as STL contenders; AMTL [Lee *et al.*, 2016] as the MTL contender that can recover a similar structure, and GAMTL with the squared loss.

Figure 2 shows the NMSE of all methods when varying the total number of samples. Mean and standard deviation from 30 independent runs are reported. Since we start the experiment with an ill conditioned scenario due to small training sample size, all methods perform poorly. But even in this case, GAMTL achieves better performance when $30 \leq m \leq 60$. As m increases, all methods start to perform similarly.

The gains of GAMTL can be possibly explained by the flexibility of its transference structures ($\mathbf{b}_t^g$) that reduces negative transfer often introduced by symmetric information

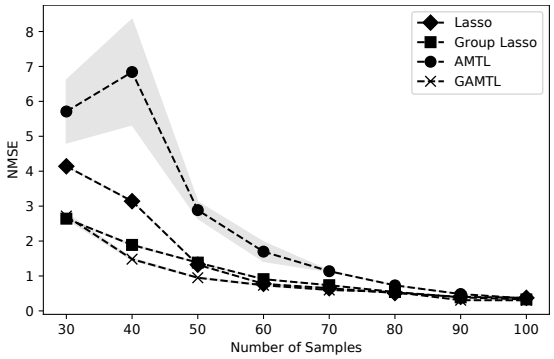


Figure 2: Mean of NMSE for all methods when varying the quantity of training samples. The shaded area is the standard deviation. When $m \leq 70$ GAMTL has the best generalization performance. With more samples all methods show similar performance.

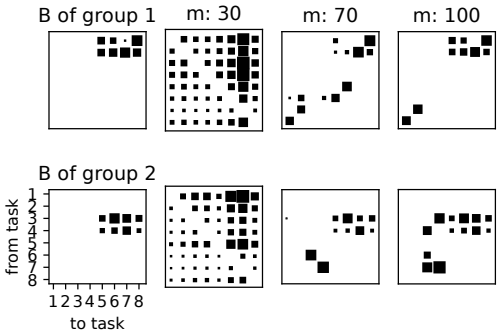


Figure 3: Hinton Diagram of the task relationship recovered by GAMTL in the first run. The size of the squares are proportional to the values of the entry of the matrix. The first column shows the conceived structures for both groups of attributes. The other columns show the structure recovered by GAMTL when $m = \{30, 70, 100\}$ respectively.

sharing across tasks. In contrast to AMTL, that considers all attributes to transfer, our model fits local relationships. Figure 3 shows the generated B matrices for the two groups, and the estimated transferences of GAMTL with $30, 70$, and 100 samples. A column $\mathbf{b}_t^g$ in B^g contains the coefficients of the approximation of task t parameters, and thus its components represent how other tasks affect $\mathbf{w}_t^g$. A row $\mathbf{b}_{\bar{t}}^g$ represents how task t affects the parameters of other task on group g only. Notice that the last four tasks are related with the first four tasks in both groups, but not in the same way.

When the sample size is small ($m = 30$), we observe that the last columns of the relationship matrices of both groups have higher values than their transposed coordinates. This indicates that all tasks are related but tasks with smaller costs are influencing tasks with higher costs more than the opposite. Since the B^g matrices regularize the task parameters in the direction of tasks with smaller costs, even when the sample size is small, GAMTL makes use of this structure to improve performance on all tasks. When $m = 70$ the relationship matrices are sparser, with only the more meaningful relations between tasks remaining; and when $m = 100$ the structures are close to the true matrices.

The estimated transferences from tasks with lower cost are close to the generated values, and some small transference occurs back as tasks are linearly dependent, with results gradually converge to the last column of Figure (3). With enough samples, GAMTL does not transfer between unrelated tasks. Nonetheless, only GAMTL is capable of providing the asymmetric structural relationship required for this problem.

4.2 Real Dataset

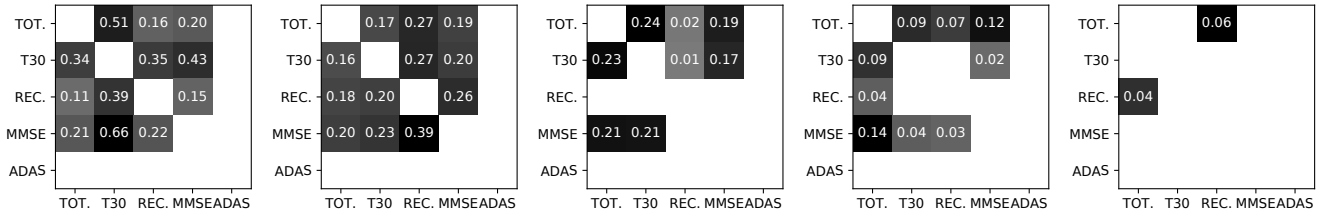
The ADNI dataset was collected by the Alzheimer's Disease Neuroimaging Initiative (ADNI) and pre-processed by a team from University of California at San Francisco, as described in [Liu *et al.*, 2018], who performed cortical reconstruction and volumetric segmentation with the FreeSurfer image analysis suite. It consists of information from 816 subjects. There are 116 groups of features in this application corresponding to ROI in the brain. From the total group set, 46 of these groups have a single covariate and 70 groups have four covariates.

The tasks consist of the prediction of 5 cognitive scores based on physical characteristics of each individual's brain extracted from structural MRI images. Note that all tasks use the same input matrix (X). The cognitive scores used in this study are: Rey Auditory Verbal Learning Test (RAVLT) Total score (TOTAL), RAVTL 30 minutes delay score (T30), RAVLT recognition score (RECOG), Mini Mental State Exam score (MMSE), and Alzheimer's Disease Assessment Scale cognitive total score (ADAS). Those are important tasks in the domain of research related to AD, since the use of these scores impacts on drug trials, assessments of the severity of symptoms of AD, the progressive deterioration of functional ability, and deficiencies in memory, as highlighted in [Liu *et al.*, 2018]. Note that for this experiment, understanding how certain areas of the brain impact the outcome of each cognitive score and how they share this impact amongst each other is of high relevance. Our model presents explainable transference structures that can aid researchers to explore further relationships.

GAMTL and GAMTLnl used the squared loss for regression tasks, and the contenders are LASSO, Group LASSO, and AMTL. We add other related MTL formulations: MTSGL, that is also based on group sparsity; MTRL that includes transference structure; and MTFL that accounts for task grouping but has no transference structure estimation.

Following [Liu *et al.*, 2018], the dataset is partitioned into training (95%) and test (5%) sets. All performance comparisons used NMSE as metric. Regularization parameters for the methods are chosen by a 5-fold cross-validation procedure using training data. Then we train each method using the training set and evaluate on the test set. To account for variability in the data, 30 independent executions were performed. The limits of the search grid used to tune parameters for MTRL were $\rho_1 \in [0.06, \cdots, 5]$ and $\rho_2 \in [0.08, \cdots, 5]$. For MTFL we had $2, 3$ as the number of task groups, and $\rho_1, \rho_2 \in [0.001, \cdots, 10]$. For AMTL we used $\mu \in [0.001, \cdots, 1], \lambda \in [0.01, \cdots, 1]$. All variants of GAMTL used $\lambda_1 \in [10e^{-5}, \cdots, 0.03]$, $\lambda_2 \in [0.01, \cdots, 0.5]$, and $\lambda_3 \in [0.008, \cdots, 0.15]$.

Table (1) summarizes the performance of all methods, pre-



(a) Left Inferior Lateral Ventricle (b) Corpus Callosum Posterior (c) Left Choroid Plexus (d) Right Inferior Parietal (e) Right Ventral DC

Figure 4: B^g for the 5 ROIs with most transference activity in GAMTL, ordered by Frobenius norm from (a) to (e). Each row represents how a task affects other tasks, and each column represents how a task is affected by other tasks.

Method	NMSE
LASSO	0.787 (0.000)
Group LASSO	1.005 (0.262)
MT-SGL	0.809 (0.000)
MTFL	0.814 (0.000)
MTRL	0.798 (0.000)
AMTL	0.887 (0.057)
GAMTL	**0.774 (0.001)**
GAMTLnr	0.787 (0.002)

Table 1: NMSE of all methods in the ADNI dataset (mean and standard deviation over all 5 folds). GAMTL had the best result, highlighted in bold.

	GAMTL	LASSO
TOTAL	0.888 (0.001)	**0.864 (0.000)**
T30	0.620 (0.000)	**0.604 (0.000)**
RECOG	**0.744 (0.001)**	0.812 (0.000)
MMSE	**0.576 (0.001)**	0.580 (0.000)
ADAS	**0.505 (0.000)**	0.524 (0.000)

Table 2: MSE (mean and standard deviation over 30 runs) of methods with best performance per task in the ADNI dataset. Best results are highlighted in bold.

senting mean and standard deviation of NMSE over all runs. We can see that GAMTL obtained the best score. We used a Mann-Whitney U test with $p \leq 0.05$ to determine whether there was a statistically significant difference between the scores, and it resulted positive when comparing GAMTL scores with the results of other methods.

Table 2 summarizes the MSE (mean and standard deviation over all runs) of the methods with best performance on each task. We highlighted the best MSE of each task. GAMTL exceeded in all but RAVLT TOTAL and T30, where LASSO outperformed all methods.

Note that GAMTL allows the relationship between tasks to be independent for each group of active attributes. We allow the practitioner to understand how the groups of variables are relating to each other by choosing a group and looking straight onto the specific relationship matrix. This turns GAMTL into a more explainable model, as task parameters can be interpreted with all procedures to understand linear regression tasks, and transferences on each ROI are of direct interpretation. In 25 of 30 run, only 5 of 116 ROIs had $\|B^g\|_2 \geq 0.01$. Figure (4) shows the B^g recovered for the 5 ROIs with most transference activity. GAMTL was able to estimate the transference on ROIs of interest on AD literature research. For instance: rates of ventricular enlargement were found to increase over time in both subjects with mild cognitive impairment (MCI) and AD, representing a feasible short-term marker of disease progression for multi-centre studies [Leung *et al.*, 2013]; measurement of corpus callosum size allows in vivo mapping of neocortical neurodegeneration in AD over a wide range of clinical dementia severities and may be used as a surrogate marker for evaluation of drug efficacy [Teipel *et al.*, 2002].

5 Conclusion and Future Work

GAMTL is a flexible and explainable model for MTL, suitable for domains where features can be partitioned into a predefined overlapping group structure. Without any strong assumption, we can estimate an asymmetric transference structure involving all tasks in a way that each group of covariates has its own relationship matrix and can properly isolate unrelated tasks. This leads to an easy interpretation of the underlying relationship supported by the tasks, which is desired in several domains. We validated our model on an artificial dataset and also on the ADNI dataset, whose tasks are the prediction of 5 cognitive scores related to the progress and symptoms of Alzheimer's disease. GAMTL not only obtained competitive performance but also estimated a meaningful relationship structure on results supported by the AD research literature. The next research steps include the exploration of new applications, the inspection of other restrictions on the relationship matrices, and the investigation of other visual representations for the estimated structure.

Acknowledgements

We acknowledge the grants #1418812015-1 and #3072282018-5 from the Brazilian National Council for Scientific and Technological Development (CNPq), grant #201307559-3 from São Paulo Reseach Foundation (FAPESP), and the Coordination for the Improvement of Higher Education Personnel (CAPES).

References

[Argyriou *et al.*, 2008] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[Baxter, 1997] Jonathan Baxter. A Bayesian/Information Theoretic Model of Learning to Learn via Multiple Task Sampling. *Machine Learning*, 28(1):7–39, 1997.

[Beck and Teboulle, 2009] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1), 2011.

[Caruana, 1997] Rich Caruana. Multitask learning. *Machine Learning*, 28(1), July 1997.

[Gonçalves *et al.*, 2016] André R. Gonçalves, Fernando J. Von Zuben, and Arindam Banerjee. Multi-task Sparse Structure Learning with Gaussian Copula Models. *Journal of Machine Learning Research*, 17(33):1–30, 2016.

[Jacob *et al.*, 2009] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group Lasso with Overlap and Graph Lasso. In *International Conference on Machine Learning*, pages 433–440, 2009.

[Kang *et al.*, 2011] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521–528, 2011.

[Kumar and Daumé, 2012] Abhishek Kumar and Hal Daumé. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning*, 2012.

[Lee *et al.*, 2016] Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Asymmetric Multi-task Learning Based on Task Relatedness and Loss. In *International Conference on Machine Learning*, pages 230–238, 2016.

[Leung *et al.*, 2013] Kelvin K. Leung, Jonathan W. Bartlett, Josephine Barnes, Emily N. Manning, Sebastien Ourselin, and Nick C. and Fox. Cerebral atrophy in mild cognitive impairment and Alzheimer disease. *Neurology*, 80(7):648–654, 2013.

[Liu *et al.*, 2009] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2,1-norm minimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 339–348, 2009.

[Liu *et al.*, 2018] Xiaoli Liu, André R. Goncalves, Peng Cao, Dazhe Zhao, and Arindam Banerjee. Modeling Alzheimer's Disease Cognitive Scores Using Multi-Task Sparse Group Lasso. *Computerized Medical Imaging and Graphics*, 66:100 – 114, 2018.

[Teipel *et al.*, 2002] Stefan J. Teipel, Wolfram Bayer, Gene E. Alexander, York Zebuhr, Diane Teichberg, Luka Kulic, Marc B. Schapiro, Hans-Jürgen Möller, Stanley I. Rapoport, and Harald Hampel. Progression of Corpus Callosum Atrophy in Alzheimer Disease. *Archives of Neurology*, 59(2):243–248, 02 2002.

[Thrun and O'Sullivan, 1996] Sebastian Thrun and Joseph O'Sullivan. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In *International Conference on Machine Learning*, pages 489–497, 1996.

[Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[Wang *et al.*, 2012] Hua Wang, Feiping Nie, Heng Huang, Sungeun Kim, Kwangsik Nho, Shannon L. Risacher, Andrew J. Saykin, Li Shen, and For the Alzheimer's Disease Neuroimaging Initiative. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the ADNI cohort. *Bioinformatics*, 28(2):229–237, 2012.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68:49–67, 2006.

[Zhang and Yeung, 2010] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 733–742, 2010.

[Zhang, 2013] Yu Zhang. Heterogeneous-neighborhood-based multi-task local learning algorithms. In *Advances in Neural Information Processing Systems*, pages 1896–1904, 2013.

[Zou, 2006] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.