UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Saullo Haniell Galvão de Oliveira

Aprendendo Múltiplas Tarefas e Estimando Relacionamentos Locais entre Tarefas Relacionadas

Learning Multiple Tasks and Estimating Local Relationships Among Related Tasks

Campinas

2022

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Saullo Haniell Galvão de Oliveira

Aprendendo Múltiplas Tarefas e Estimando Relacionamentos Locais entre Tarefas Relacionadas

Learning Multiple Tasks and Estimating Local Relationships Among Related Tasks

Thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Computer Engineering.
Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Fernando José Von Zuben

Co-supervisor Dr. André Ricardo Gonçalves

Este exemplar corresponde à versão final da tese defendida pelo aluno Saullo Haniell Galvão de Oliveira, e orientada pelo Prof. Dr. Fernando José Von Zuben

_____

Campinas
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

OL4e

Oliveira, Saullo Haniell Galvao de, 1988-
Learning multiple tasks and estimating local relationships among related tasks / Saullo Haniell Galvão de Oliveira. – Campinas, SP : [s.n.], 2022.

Orientador: Fernando José Von Zuben.
Coorientador: André Ricardo Gonçalves.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado de máquina. 2. Análise estrutural (Engenharia) - Processamento de dados. 3. Esparsidade. 4. Sistemas de aprendizado. 5. Processamento de sinais. I. Von Zuben, Fernando José, 1968-. II. Gonçalves, André Ricardo, 1986-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações Complementares

Título em outro idioma: Aprendendo múltiplas tarefas e estimando relacionamentos locais entre tarefas relacionadas
Palavras-chave em inglês:
Machine learning - Statistical methods
Structural analysis (Engineering) - Data processing
Sparse matrices
Learning systems
Signal processing
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Levy Boccato
Marcos Medeiros Raimundo
Denis Fernando Wolf
Rafael Izbicki
Data de defesa: 22-11-2022
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-9432-8688
- Currículo Lattes do autor: http://lattes.cnpq.br/2409061107012809

COMISSÃO JULGADORA – TESE DE DOUTORADO

Candidato: Saullo Haniell Galvão de Oliveira RA: 144466
Data da Defesa: 22 de novembro de 2022
Título da Tese: "Learning Multiple Tasks and Estimating Local Relationships Among Related Tasks".

Prof. Dr. Fernando José Von Zuben
Prof. Dr. Levy Boccato
Prof. Dr. Marcos Medeiros Raimundo
Prof. Dr. Denis Fernando Wolf
Prof. Dr. Rafael Izbicki

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Dedico esta tese à minha mãe Iris, a minha avó Nelcy, aos meus irmãos Júnior e Shayra, e a minha sobrinha Júlia.

AGRADECIMENTOS

Que jornada! É impossível começar essa seção sem ser inundado por uma infinidade de memórias.

Agradeço ao Prof. Dr. Fernando J. Von Zuben pela orientação e pelas inúmeras discussões riquíssimas ao longo desse processo que se iniciou em meu mestrado. Você foi e é um espelho! Também ao André R. Gonçalves, pela mistura perfeita entre co-orientação e amizade, regada a boas cervejas e discussões diversas.

Agradeço a toda a minha família, que direta e indiretamente moldou minha forma de interpretar o mundo, me ofereceu força quando as batalhas se apresentaram, me confortou nos dias difíceis, e me ensinou resiliência para nunca desistir perante qualquer desafio. **Esse título é nosso!** Fafá, minha companheira de todas as horas, que me aturou ao longo de grande parte dessa jornada. Seu sorriso iluminou meu caminho e tornou tudo mais leve!

A todos os professores e funcionários da Unicamp, em especial da FEEC, meu muito obrigado! Vocês compõem um ambiente incrível e inspirador! Eu jamais poderia imaginar que estar perto de vocês teria um impacto tão proeminente em minha vida. A todos os colegas do LBiC pelas marcas profundas de companheirismo e pelas inúmeras discussões técnicas e não técnicas. Meu senso crítico e minhas definições de amizade não seriam as mesmas sem vocês. Aos colegas da Associação de Pós-Graduandos da Faculdade de Engenharia Elétrica e de Computação (APOGEEU), e também do Capítulo Estudantil da Sociedade de Inteligência Computacional do IEEE na Unicamp (IEEE-CIS), pelos diversos eventos que organizamos juntos!

A Elisabete Mergulhão, por me mostrar possibilidades e ajudar a encontrar rumos quando minha visão era sombria demais para avançar. Nossas sessões me inspiram muito. Obrigado! A todas as pessoas com quem compartilhei um teto ao longo desses anos como estudante de pós: Benoît, Pedro, Sarah, Prioli (Fera), Bera, Taynan, Victor, e Tom. Obrigado por aturar meu mau-humor nos momentos complicados e oferecer força e companhia ao longo desse processo. Não poderia deixar de agradecer também a todas as pessoas que conheci nos bares de Barão Geraldo. Com seus currículos tão variados (músicos e musicistas, dançarinos e dançarinas, economistas, geógrafos e geógrafas, arquitetos e arquitetas, engenheiros e engenheiras, e por aí vai), vocês me ajudaram a perceber nuances e detalhes no mundo que eu jamais veria sozinho. A todas as pessoas que conheci durante essa jornada, meu muito obrigado!

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) através do processo (141881/2015-1).

ABSTRACT

Multi-Task Learning focuses on the simultaneous learning of multiple tasks - classification or regression tasks, for example - expecting to improve performance on each task individually, by exploring relationships among them.

To benefit from the joint learning procedure, we model a representation structure that is shared by all tasks and can be used in multiple ways. It is possible to encode prior knowledge about the tasks into the shared representation by thoroughly using regularization terms, thus imposing the desired beliefs on the model's parameters. Nevertheless, by imposing our beliefs about task relationship, we risk forcing relationships among unrelated tasks, or simply imposing regularities not supported by the available datasets of the tasks, thus promoting negative transference. An alternative is to estimate how tasks are related during the learning process, thus avoiding some pitfalls of the previous strategy. But even in this way we may assume fragile premises, such as: *i*) the relationship between tasks A and B is symmetrical, so that the influence of task A on task B is taken to be the same as the influence of task B on task A; or *ii*) all features of the dataset are equally responsible for the relationship among tasks.

In this thesis, we present Group Asymmetric Multi-Task Learning (GAMTL), a model capable of: *i*) estimating how tasks are related in an interpretable manner; *ii*) considering asymmetric relationships among tasks; and *iii*) considering local relationships that are based on subsets of features, instead of imposing the involvement of all features. Experimental results demonstrate that the flexibility added by GAMTL mitigates negative transference, while recovering significant relationships among tasks in an interpretable transference structure. The efficiency of the method is also demonstrated in a real scenario whose goal is to predict scores of cognitive tests associated with the progress of Alzheimer's disease (AD), taking as input pre-processed data based on cerebral imaging. Besides obtaining a high performance on the scores predictions, GAMTL was able to capture regions of interest in the brain that are part of the estimated relationships, in accordance with independent results found in medical literature.

Keywords: Multi-Task Learning; Asymmetric Structural Learning; Sparsity; Structural Sparsity; Local Feature Transference.

RESUMO

A área de aprendizado multi-tarefa se preocupa em aprender simultaneamente múltiplas tarefas - de classificação ou regressão, por exemplo - buscando melhorar o desempenho de cada tarefa individualmente ao explorar as relações entre elas. Para se beneficiar do aprendizado em conjunto, modela-se uma representação compartilhada entre as tarefas que pode ser utilizada de diversas formas. É possível embutir conhecimento prévio sobre o domínio das tarefas nessa representação compartilhada, aplicando minuciosamente termos de regularização, impondo assim características desejadas nos parâmetros do modelo. Contudo, ao impor características que acreditamos serem verdadeiras na relação entre as tarefas, corre-se o risco de forçar relações entre tarefas não relacionadas ou simplesmente forçar características que não estão presentes nos dados. Quando isso ocorre, o desempenho individual das tarefas é prejudicado ao invés de melhorar, o que é conhecido como transferência negativa. Outra abordagem possível consiste em estimar as relações existentes entre as tarefas durante o processo de aprendizagem, evitando assim as armadilhas da estratégia anterior. Mesmo assim, podemos assumir premissas frágeis, como por exemplo: *i*) a relação entre uma tarefa A e uma tarefa B é simétrica, isto é, a influência de A em B é idêntica a influência que B tem sobre A; ou *ii*) todos os atributos incluídos na base de dados participam igualmente na influência entre as tarefas.

Nesta tese apresentamos Group Asymmetric Multi-Task Learning (GAMTL), um modelo capaz de: *i*) estimar como as tarefas estão relacionadas de maneira interpretável; *ii*) considerar relações assimétricas entre tarefas; e *iii*) considerar relações entre tarefas que não incluem necessariamente todos os atributos presentes nos dados, podendo se restringir a relações localizadas em sub-conjuntos desses atributos. Resultados experimentais demonstram que a flexibilidade adicionada é capaz de mitigar fortemente a transferência negativa, além de recuperar relações significativas entre as tarefas em uma estrutura interpretável. A eficiência do método tambem é demonstrada na predição de *scores* de testes cognitivos relacionados ao progresso da doença de Alzheimer (AD), considerando como entrada dados pré-processados a partir de imagens cerebrais. Além de obter uma bom desempenho na predição dos *scores*, o método foi capaz de estimar quais regiões de interesse no cérebro fazem parte das relações entre tarefas, em concordância com resultados independentes na literatura médica.

Palavras-chaves: Aprendizado Multi-Tarefa; Aprendizado Estrutural Assimétrico; Esparsidade; Esparsidade Estrutural; Transferência Local de Atributos.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The science of learning plays a key role in the fields of statistics, data mining and artificial intelligence (HASTIE *et al.*, 2001). Among the many possible configurations of learning - such as learning from data (BISHOP, 2006), learning from the environment through reinforcement signals (SUTTON; BARTO, 2018), or learning rules (LIU *et al.*, 2015) - Machine Learning (MURPHY, 2012; HASTIE *et al.*, 2001; BISHOP, 2006) is enjoying an increasing popularity as more data becomes available for applications in both scientific and commercial scenarios.

Definition 1. We define Machine Learning (ML) as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision-making under uncertainty (MURPHY, 2012).

Machine Learning can be further divided in more specific approaches, for instance: the *predictive* or *supervised learning*, and the *descriptive* or *unsupervised learning*. In the supervised learning approach, we are interested on predicting the outcome/label of data points that the model has never seen, based on what it could learn from a labeled dataset. To that end, first we need a set of labeled data points where an annotator goes through the sample and categorizes it in the labels. This will compose the labeled dataset. The features that describe each data point can be of many types (discrete, continuous, categorical or ordinal, for example) and their main function is to encode information in a useful way to the learning model. After that, we split the labeled dataset into training and test sets. We use the training dataset to tune the parameters of the model, and the test set is left out of the training, being used to measure the quality of the trained model. Finally, when the model is trained, it can be used to predict the labels of unlabeled data points. The characteristics of the label space determine the type of problem we have to solve. For example, when labels are categorical we call the learning task a *classification task*; when labels are continuous we call the learning task a *regression task*.

Definition 2 (Supervised Learning)**.** Let $X \in \mathbb{R}^{m \times n}$ be a set of m data points with n features, and $\mathbf{y} = [y_1, \cdots, y_m]$ be a list of labels for each data point. The labeled dataset consists of the tuple $(X, \mathbf{y})$ and a supervised learning task involves the estimation of a mapping function $f(\cdot) : \mathbb{R}^n \mapsto \{labels\}$ in case of a classification task, or $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$ in case of a regression task.

A classical example of supervised learning is the task of spam classification. Suppose we are interested in building an automated system to classify our e-mails in SPAM versus regular e-mails, i.e., $\mathbf{y} \in \{\text{SPAM}, \text{Not SPAM}\}$. In order to train the model, first

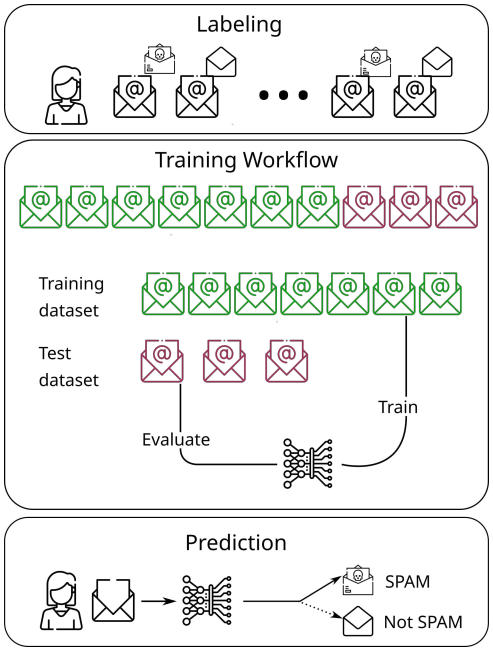


Figure 1 – Supervised learning workflow: on the top we see the labeling step, where a human annotator labels some e-mails into SPAM or Not SPAM; on the middle the model is trained and its generalization capability is measured; and on the bottom the trained model is used to predict if new incoming e-mail messages are SPAM or Not SPAM.

we need a labeled set of e-mails to compose our labeled dataset. This is depicted in the top of Fig. 1, where a human annotator labels some e-mails as SPAM or Not SPAM. For the training procedure we transform the text of each e-mail message into a numeric feature space, extracting features to represent X in a $\mathbb{R}^{m \times n}$ domain. The labels in this case can assume a value from two discrete outcomes, that characterizes the binary classification nature of our problem. For the training workflow, the labeled dataset is split into the training and test sets (middle of Fig. 1). We use the training dataset to fit the model's parameters and existing hyper-parameters. The test set is used to measure the *generalization capability* of the model, i.e., the capability to properly predict the label of unlabeled data points that were not used to fit parameters or to tune hyper-parameter values. The generalization capability is measured using some appropriate metric over the test set, since we have the labels available. For example, accuracy is a widely used metric for binary classification problems, and mean squared error for regression problems. Both metrics indicate how close the estimation provided by the learning model is from the desired output values. Once the training is completed and we are able to trust in the generalization capability of the model, we can use it to predict if new incoming e-mails are SPAM or Not SPAM (bottom of Fig. 1).

For the regression case of supervised learning, let us suppose that we have sensors that capture some wheater measurements such as air pressure, air humidity, air temperature, among other variables, and we are interested in predicting the air temperature of

the next hour, given the measurements of our sensor in the current hour. In this case, the desired outcome is a continuous variable, indicating that we have a regression task.

On the unsupervised approach of ML, the dataset is composed only of the sample $X \in \mathbb{R}^{m \times n}$ and the goal is to find "interesting patterns" in the data. "This is a much less well-defined problem, since we are not told what kinds of patterns to look for, and there is no obvious error metric to use" (MURPHY, 2012). The goal of clustering is to separate a finite unlabeled dataset into a finite and discrete set of 'natural' hidden data structures, rather than provide an accurate characterization of unobserved samples generated from the same probability distribution (XU; WUNSCH, 2005). Although not complete, a classic definition for clustering is described as follows (JAIN; DUBES, 1988):

- Instances, in the same cluster, must be similar as much as possible;
- Instances, in the different clusters, must be as distinct as possible; and
- Measurement for similarity and dissimilarity must be clear and have a practical meaning.

On both approaches, supervised and unsupervised learning, each data point is described by a set of n features.

1.1 THE PRESENCE OF MULTIPLE TASKS

Suppose now that instead of classifying our own e-mails, we are an e-mail provider that wants to identify SPAM e-mails for all accounts using our service. Each user can be seen as a classification task, where $\mathbf{y} \in \{\text{SPAM}, \text{Not SPAM}\}$. As the provider hosts for many accounts, we have now multiple training datasets. How can we solve this problem?

In one extreme, we can train a classifier in isolation for each account, what is called Single Task Learning (STL) approach and is represented in Fig. 2. In this approach, each user has a learning model that uses its own data for training. We may require from each user to label enough data for the training procedure. When the user has labeled enough e-mails, their SPAM detector will probably be very suited to their needs, and will not be affected by SPAM e-mails from other users. But how many data points should each user label manually in order to have a classifier with acceptable accuracy? If we require a great amount of data from each user, we will not have a practical solution for the problem. Another problem occurs when new users arrive. Since in this setting every user must label their e-mails, we will not have e-mails to train a classifier for new users, which is known as the 'cold start' problem. Finally, depending on the number of hyper-parameters we need to set for each classifier, the computational cost of finding a good configuration and the amount of information required to enable a proper tuning may impose practical

Figure 2 – The Single Task Learning approach for handling multiple tasks, where one model is trained per user with labeled data, is depicted on the left. The Pooled Model Learning approach for handling multiple tasks simplifies everything by using a single model and is depicted on the right.

challenges. An important limitation of the single-task approach is that *no information is shared among related tasks*, i.e., the models are not able to help each other even when their respective users have similar SPAM e-mails.

On the other extreme, we can train a single classifier for all accounts, which is known as the Pooled Model approach, represented in Fig. 2. This solution has the benefit that new users will already have a trained model to classify their e-mails. Even if some users do not label their e-mails, some of their SPAM may be represented in the training set with data from other users and the classifier may work for them. However, there are some aspects that may contribute to reduce the performance of pooled model learning. The content of some SPAM campaigns may be general enough to be sent to as many users as possible. Other campaigns may segment their audience based on other collected information about the users. These specialized SPAM e-mails can be sent only to a small fraction of users that meet their segmentation criteria, which leads to a severe sub-representation in the training dataset when compared to the more popular SPAM e-mails. This imbalance is a problem for most off-the-shelf classifiers and usually requires more effort in tuning hyper-parameters or employing sampling strategies to alleviate the imbalance during training. Since we only have a single model, it would probably have an unacceptable classification performance for some of the tasks. Unlike the STL, in this strategy information is shared among all users. The drawback now is that even unrelated tasks share information and this may negatively affect the overall performance (CARUANA, 1997).

Multi-Task Learning (MTL) (VANDENHENDE *et al.*, 2021; ZHANG; YANG, 2017) is an attempt devoted to solve this problem by exploring the potential relationship among the tasks. The goal is to train all tasks simultaneously while leveraging information among related tasks, such that all tasks have better generalization performances when compared to a process that trains each task independently (i.e., STL). This approach is illustrated in Figure 3. In our SPAM detection example, we still train a classifier for each user, but now all models are trained jointly and an information-sharing mechanism in the MTL model will handle the information flow among the tasks that generally leads to increased performance. This approach shares with STL the commonality that each

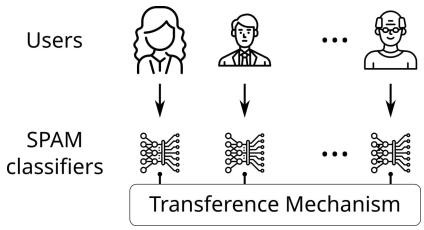


Figure 3 – In the Multi-Task Learning approach, each task has its own model, but all models are trained together and can share information among tasks that are related in some way. Notice that each user has a dedicated model, and the structural relationship among the tasks may be previously informed or even estimated during the learning process.

account will have a learner that is dedicated to it. When compared to the pooled model approach, MTL allows tasks to handle specific behavior of the accounts while sharing information with related tasks. But even if we need to retrain the model, that can be done in isolation for one or more tasks, with minimal impact on the other tasks.

Some MTL methods allow us to leverage prior knowledge about the structural relationship of the tasks, that will guide information sharing to relate tasks known in advance to be related (LIU *et al.*, 2019; LIU *et al.*, 2018; OLIVEIRA *et al.*, 2019; KOLAR *et al.*, 2011; HAN; ZHANG, 2015). MTL methods are also more robust to sample complexity: when the number of data points is small, MTL is empirically proven to enhance the generalization capability of the tasks, supported by consistent theoretical results (CARUANA, 1997; VANDENHENDE *et al.*, 2021; ZHANG; YANG, 2017).

1.2 NEGATIVE TRANSFERENCE: A MAJOR CHALLENGE

One of the main challenges for MTL methods is that information sharing assumptions of the model may not match the underlying tasks relationships. In this case, instead of improving generalization performance on all tasks, the MTL method will probably deteriorate performance. This phenomenon is know as *negative transference* (CARUANA, 1997) and conceiving new MTL approaches capable of alleviating negative transference is a great motivation of this work.

Negative transference can occur for multiple reasons. A good example is when unrelated tasks are forced to be related: in this case, the influence of one task into unrelated tasks will probably have a harmful effect on the performance. But even sharing among related tasks may reduce performance (CARUANA, 1997): if the assumptions of the model are not satisfied by the data, it is likely that negative transference will occur.

As we will see in detail in Chapter 2, designing how tasks can be related is a challenging effort. Different assumptions about how tasks are related can be encoded in the MTL approach in many ways, popularly through the usage of regularization terms

(ZHANG; YANG, 2017). Some methods assume that tasks are related in clusters (EVGENIOU; PONTIL, 2004; ZHOU *et al.*, 2011a), while others assume that related tasks share a latent space (ANDO; ZHANG, 2005; JALALI *et al.*, 2010), to mention a few. MTL methods that leverage regularization terms to promote information sharing among tasks have the advantage of encoding a priori domain knowledge into the model formulation. By choosing proper regularization terms we can drive a set of parameters to zero due to the "sparsity" property of such terms. As fewer variables are active in the final solution, models with sparsity-inducing regularization terms are simpler to interpret. Other regularization terms are able to structure the way sparsity is induced in patterns of overlapping groups of variables. There is a great number of MTL methods that combine such regularization terms disposed in clever arrangements, each one making different assumptions about tasks relationships.

Given those scenarios, two major challenges remain: i) choosing a method is difficult, as the problem needs to meet too many a priori assumptions in order to benefit from a given model; ii) models usually oversimplify the possible ways the tasks can be related.

A different approach is taken by the Structure Estimation strategy (ZHANG; YEUNG, 2010a; GONÇALVES *et al.*, 2016; OLIVEIRA *et al.*, 2019). Instead of imposing hard a priori assumptions, they estimate how tasks are related to each other during training, while using the estimated relationships to guide the learning procedure. They usually require less a priori knowledge about tasks relationships, which alleviates the odds of making the wrong assumptions. The drawback of many methods in this category is that they also oversimplify how tasks can be related. This oversimplification typically includes the following assumptions:

- tasks are symmetrically related - that is, task A affects task B in the same way that task B affects task A (LEE *et al.*, 2016; ZHANG; YEUNG, 2010a; JALALI *et al.*, 2010; LIU *et al.*, 2019); and
- if two tasks are related they must influence each other on the entire set of features, to which we will call *global feature transference* (LEE *et al.*, 2016; ZHANG; YEUNG, 2010b; ZHOU *et al.*, 2011a).

There have been attempts to flexibilize these models in order to circumvent the observed downsides (LEE *et al.*, 2016; GONÇALVES *et al.*, 2016), but no candidate simultaneously accounts for both the local feature transference and asymmetrical structural learning, as we will see in Chapter 2. It is also worth mentioning that the optimization challenges introduced by MTL formulations commonly include non-smooth terms, and may not even pose convex formulations.

1.3 AVOIDING NEGATIVE TRANSFERENCE BY LEARNING HOW TASKS ARE RELATED

The goal of this thesis is to develop an MTL method that avoids the mentioned pitfalls. Namely, our intent is to design a more general MTL method that estimates how tasks are related, while being capable of learning asymmetrical and local relationships among the tasks.

In Chapter 5, we present the Group Asymmetric Multi-Task Learning (GAMTL) (OLIVEIRA *et al.*, 2019) approach, which was specially tailored to meet three goals:

1. estimate how tasks are related in an interpretable way;
2. consider that the relationship between two tasks may not involve all features, i.e., can be restricted to a subset of features, an approach we will call *local feature transference*;
3. allow two tasks to be asymmetrically related, i.e., the influence a task A has on a task B may differ from the influence task B has on task A.

GAMTL proposes a flexible mechanism to the problem of learning the complex relationship tasks may exhibit, possibly more in tune with realistic scenarios by putting forward local and asymmetric structural relations both among features and tasks.

As the Multi-Task Learning term is growing in popularity, it assumes different meanings depending on the research area it is being used. Our scope was restricted to regularization based sub-fields (ZHANG; YANG, 2017): Regularized Multi-Task Learning, and Structure Learning. More on that discussion is presented in Chapter 2. In summary, here we leverage the solid usage of regularization terms while reducing the assumptions usually required by existing MTL formulations, together with ideas from the Structure Estimation, to design flexible relationships among the tasks.

1.4 STRUCTURE OF THE THESIS

This work is organized in two parts. The first part lays down the fundamental concepts of Multi-Task Learning, Sparse Models, and Non-Convex Optimization; which stand as the building blocks of this thesis.

- Chapter 2 gives a formal definition of MTL, and highlights the benefits of this approach in more detail. Related areas are discussed, and the usage of the term Multi-Task Learning in other research areas is also exposed to better position our work. We provide a landscape of the common terminology of MTL based on recent

surveys. We discuss how some MTL methods are able to handle groups of features, and proceed to an explanation of how Structure Estimation works.

- Chapter 3 explains how regularization works in Machine Learning and how it can induce sparsity into the models. We present the l_p-norm family and the LASSO, together with many variations. A geometric explanation of the sparsity properties of l_p-norms is provided. We also discuss how to induce sparsity considering groups of correlated features with the $l_{(p,q)}$-norm family, also with a geometric explanation. This leads to the Group LASSO regularization, and the Latent Group LASSO extension that handles overlapping groups of features.

- Chapter 4 is dedicated to solving non-convex optimization problems. We develop tools to handle non-smooth functions, mostly based on proximal operator functions, and use them in optimization algorithms. We explain two common algorithms of this class: (i) ISTA (BECK; TEBOULLE, 2009), a simple proximal method and (ii) FISTA (NESTEROV, 1983; BECK; TEBOULLE, 2009), a fast algorithm with improved convergence in the first-order methods; and (iii) ADMM (BOYD *et al.*, 2011), a flexible and parallelizable proximal algorithm.

- Chapter 5 focuses on our main contribution, the Group Asymmetric Multi-Task Learning (GAMTL) method. GAMTL is an MTL proposal that estimates how tasks are related considering groups of features in an independent way. The resulting non-convex optimization problem is solved by deriving smaller convex sub-problems that can be solved with an alternating optimization procedure and methods presented in Chapter 4. The source code of this proposal is available using the Python programming language [1].

- In Chapter 6 we provide empirical results of GAMTL. An artificial setting is devised with challenging transference assumptions to evaluate how the algorithm performs against multiple related contenders. We investigate how the accuracy of our proposal evolves as more data is available for training, and how a varying degree of noise in the tasks labels can affect the estimated relationship structure. We also compare GAMTL with state-of-the-art methods for the task of predicting cognitive scores related to Alzheimer's Disease, based on feature-processed brain images collected on distinct stages of the disease. The results highlight the performance of the method and the interpretability of the explainable relationship structure. Another experiment is devised to validate the robustness of the structure of transference estimated by GAMTL with respect to data sampling and hyper-parameter settings. This experiment reveals how the flexibility of GAMTL allows the method to capture the distinct roles the features can play on related tasks.

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

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 of A. 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 a_i. For any two vectors $\mathbf{x}, \mathbf{y}$ the Hadamard product is denoted by $(\mathbf{x} \odot \mathbf{y})_i = x_i y_i$.

2 THE CHALLENGES OF LEARNING MULTIPLE TASKS

It may be intuitive for us humans to think that learning one task may help us learning other related and more complex tasks. For instance, let us consider the case of an undergraduate course. The course usually divides certain areas of knowledge into a set of disciplines. Each discipline has it's own agenda, covering the most important topics of that area of knowledge. As a student, we enroll in multiple disciplines at the same time, which may be related or not with other disciplines. When we take related disciplines at the same time, we begin to understand connections between the related disciplines. This helps us to develop a better understanding of the involved subjects and also allows us to take multiple perspectives of each subject. Learning one task helps us learning the other by leveraging what is common among them, and recognizing these connections is valuable.

Although this imagination exercise may not be useful to draw a mathematical formalism, it helps us set a common ground of intuition about our main subject, Multi-Task Learning. In the introduction we could see how the presence of multiple learning tasks brings new challenges to the traditional ML setting, and we also discussed some options on how to tackle the problem. This setting gives rise to a new learning paradigm that is the main subject of this thesis, named Multi-Task Learning (MTL) (CARUANA, 1997). In this paradigm we train all tasks simultaneously, leveraging information from related tasks to improve generalization performance in all tasks.

In this chapter, we formally introduce Multi-Task Learning (MTL) and discuss how it relates to other machine learning areas. Also, we present how different MTL methods can be categorized based on the assumptions made by the models, particularly about how tasks are believed to be related. We present in detail the most common strategies to model transference between tasks in MTL, discussing the advantages and drawbacks of each approach through the lens of a regularization formulation. We also explore some sources of negative transference in the way the methods consider how transference occurs. By the end of the chapter, the reader will be familiar with the current literature of the topic and have a clear view of the main decisions involved in MTL methods.

The chapter is structured as follows: in Section 2.1 we give a formal definition of MTL. We explain how transference among tasks can be modeled by choosing what will be shared, and how this information will be shared. Closing the section, we discuss how MTL is related to other research areas. Section 2.2 presents the most common transference strategies in a regularization formulation. We reproduce the categorization made by Zhang e Yang (2017) and explore it to better outline our scope. Section 2.3 focuses on Structure

Estimation, an approach that estimates how tasks are related to avoid forced relationships among unrelated tasks. We also highlight the main advantages and drawbacks of this approach, especially the global transference assumption. Finally, we summarize the main remarks of the chapter and offer the motivation of the next chapter in section 2.4.

2.1 MULTI-TASK LEARNING

Let us introduce MTL formally, considering first a broad definition.

Definition 3 (Multi-Task Learning)**.** Let $\mathcal{T}$ be a set of T tasks, where all or some of the tasks are related. In MTL, we want to improve the learning of the individual models by leveraging the knowledge contained in the entire set $\mathcal{T}$.

In this definition, our learning models can be of many types: supervised learning (OLIVEIRA *et al.*, 2019; GONÇALVES *et al.*, 2016), unsupervised learning (ZHANG; ZHANG, 2010; ZHANG; ZHANG, 2013), or reinforcement learning (LI *et al.*, 2009; LAZARIC; GHAVAMZADEH, 2010), for each category. All these fields have seen some work under the MTL umbrella, but as noticed by (ZHANG; YANG, 2017), most of the current work in the field is focused on supervised learning.

In the supervised learning setting each task $t \in \mathcal{T}$ is composed of a dataset $X_t \in \mathbb{R}^{m_t \times n_t}$, and a label vector $y_t \in \mathbb{R}^{m_t}$ for a regression task t, or $y_t \in [0,1]^{m_t}$ for a binary classification task t, for example. The dataset of each task can have its specific domain $\mathbb{R}^{m_t \times n_t}$. If the domains are different, we call it a *heterogeneous* setting, (HAN *et al.*, 2012; ZHANG; YEUNG, 2011) depicted in Fig. 4. On the left we have X_t, for $t \in \mathcal{T}$, each dataset having its own number of data points m_t and features n_t. Each X_t also has its vector of labels $\mathbf{y}_t$ with an equal number of data points as X_t, depicted on the right. When the dataset of all the tasks share the same domain, we have a *homogeneous* setting - most of the literature explored in this chapter - where $n_t = n$ (the same for all tasks), resulting in $X_t \in \mathbb{R}^{m_t, n}$ for all $t \in \mathcal{T}$. Notice that Fig. 4 does not present the learning models of each task.

Let $\mathbf{w}_t$ be the vector of parameters for a model dedicated to task t. Without loss of generality, we will assume linear models for the homogeneous setting, where features of all tasks share the same feature set, i.e. the same number of features with the same meaning for all tasks, $\mathbf{w}_t \in \mathbb{R}^n$ for all $t \in \mathcal{T}$. The homogeneous setting is shown in Figure 5. Now all datasets have the same number of features, $X_t \in \mathbb{R}^{m_t \times n} \forall t \in \mathcal{T}$. The parameters of tasks models are depicted in the middle of the figure. We call $W \in \mathbb{R}^{n \times T}$ the matrix having the parameters of the tasks arranged as columns of W. Each row of W corresponds to a feature, while each column corresponds to a task. Other settings where the models are not linear, or labels are not represented by vectors, are discussed in Section 2.1.3.

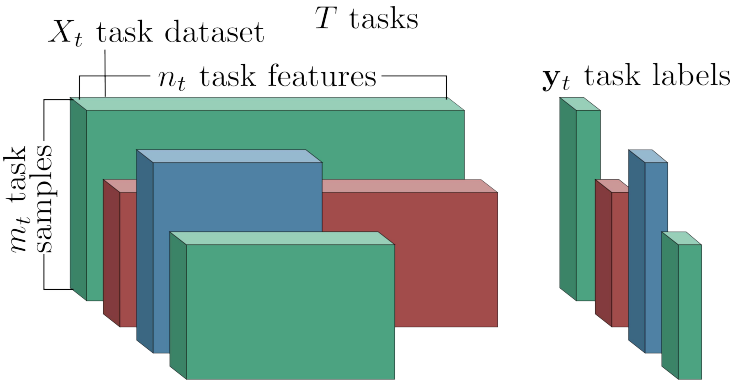


Figure 4 – A general depiction of multi-task learning. Each task can have its own dataset with their own data points from a task-specific domain on the left, and labels on the right.

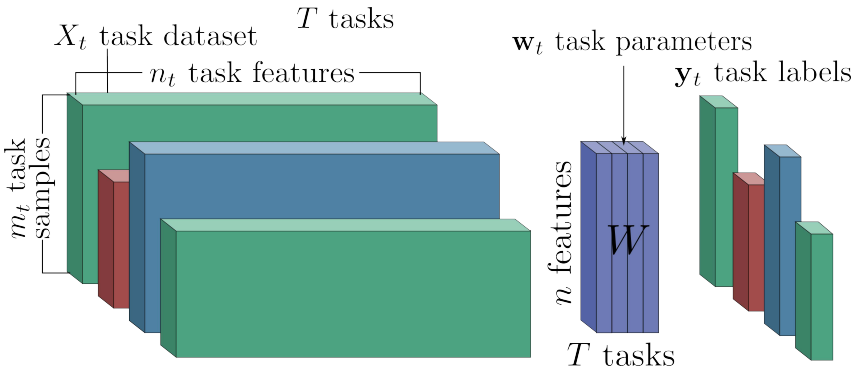


Figure 5 – MTL for supervised learning tasks in the homogeneous setting with linear models. Each task can have its dataset (on the left) but now all tasks share a common domain and $n_t = n$, hence the homogeneous setting. The linear model's parameters are at the center of the figure, and on the right we have the labels for each task.

In definition 3, the concept of "knowledge contained in $\mathcal{T}$" is not properly defined. What we already know is that it is only through transference of knowledge that tasks can affect each other to promote performance improvement. This leads to two important questions that we answer in the next section: i) "what to share?"; and ii) "how to share?".

2.1.1 SPECIFYING WHAT TO SHARE AND DETERMINING HOW TO SHARE

Machine Learning models can be designed to reflect how tasks share information in multiple ways, usually through (i) sample (BICKEL *et al.*, 2008), (ii) features (CARUANA, 1997; EVGENIOU; PONTIL, 2004; LIAO; CARIN, 2005), (iii) parameters (JALALI *et al.*, 2010; KANG *et al.*, 2011; KUMAR; DAUMÉ, 2012), or (iv) loss (OLIVEIRA *et al.*, Article no. 99, pp. 1-30, 2022.; GONÇALVES *et al.*, 2016; LEE *et al.*, 2016) . This is illustrated in Figure 6.

When **sharing is based on sample**, models try to identify data points that

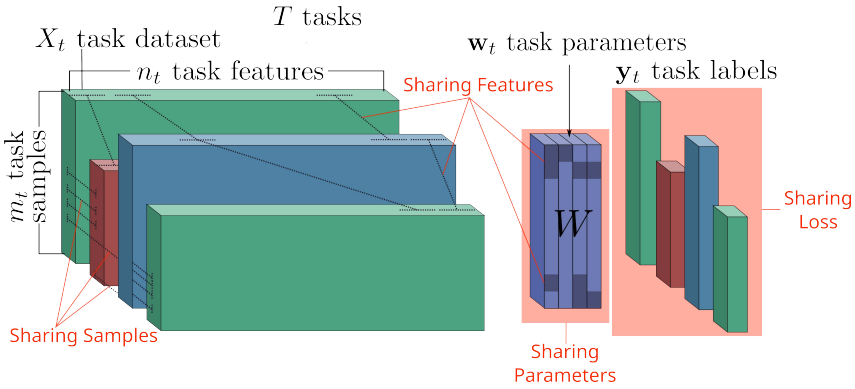


Figure 6 – What can be shared in MTL. We can share sample (highlighted in red on the left), features (highlighted in red on the center), task parameters (red rectangle around tasks parameters), and tasks losses (red rectangle around $\mathrm{y}_t vectors$).

can be useful to more than one task, which is depicted on the left by the connections between data points of distinct tasks. When **sharing is based on features**, models try to learn common feature representations that are useful to all tasks. In the figure, this is represented by the connection between the features of different tasks. When **sharing is based on parameters**, task models use their parameters to share information with other tasks, also called *inductive bias*. Sharing will involve the parameter matrix W, and for linear models they will indirectly involve tasks features, as parameters are directly associated with features of the dataset. Finally, when **sharing is based on loss**, models use the loss of each tasks to estimate relationships among them.

An example of **sharing based on sample** is (BICKEL *et al.*, 2008). Here the model first estimates a density ratio between probabilities that each data point/label belongs to their task, against belonging to a mixture of tasks. Once this estimation is finished, the parameters of the tasks are trained using a weighted training dataset based on the previous step. However, this is not a common approach.

When **sharing is based on features**, we try to learn common feature representations that are useful to all tasks. For example, the initial layers of neural networks with multiple outputs act as a feature representation layer for the downstream tasks (represented by each component in the output layer) (CARUANA, 1997). More on neural networks and deep learning for MTL methods in Section 2.1.4. Another possible way to enable sharing of features is the usage of regularization terms to impose relationships among parameters of the MTL formulation, arranging them according to a priori knowledge of the problem. This is discussed in Section 2.2.

When **sharing is based on loss** we assume that tasks with similar losses are related to each other, such as in (GONÇALVES *et al.*, 2016) where the residual of each task is used to establish relationships, or in (LEE *et al.*, 2016; OLIVEIRA *et al.*, 2019) where the loss of each task is used as a measure of confidence of each learning model and penalize transference: tasks with higher loss are less encouraged to influence tasks with

lower loss value.

Most publications in the field are dedicated to the last approach, where **sharing is based on tasks parameters**. In this category, the methods explicitly model how tasks are related in the parameters domain, depending on the assumptions each model has of how tasks are related. According to these assumptions, methods that rely on this approach can be sub-categorized as follows:

- **Low-Rank** Assumes that the parameters of the tasks share a low-rank representation (ARGYRIOU *et al.*, 2008; KANG *et al.*, 2011; KUMAR; DAUMÉ, 2012). The latent basis can also be regularized to enforce more specific premises;
- **Task Clustering** Assumes that tasks parameters are clustered into one or more groups (EVGENIOU; PONTIL, 2004; JACOB *et al.*, 2009a). If we consider overlapping groups of tasks, low-rank models can also achieve this goal;
- **Structure Estimation** Learns a task relationship structure while imposing it into the parameters of the tasks (ZHANG; YEUNG, 2010b; GONÇALVES *et al.*, 2016; LEE *et al.*, 2016; OLIVEIRA *et al.*, 2019);
- **Dirty Models** Decompose tasks parameters into a sum of two components, where different regularizers are applied to each component (JALALI *et al.*, 2010; CHEN *et al.*, 2011; CHEN *et al.*, 2012; GONG *et al.*, 2012);
- **Multi-Level** Decompose tasks parameters into a sum of more than two components, each one with a distinct regularization term (JAWANPURIA; NATH, 2012). Similar to the dirty models, but able to express more complex relationships among tasks.

We will see how different models of each of these categories leverage regularization terms. But before that, it is important to see why MTL can improve the generalization capacity of tasks, and also see how other areas of research are related to MTL. For a broader overview of the MTL field, please refer to the survey on (ZHANG; YANG, 2017).

2.1.2 WHY MULTI-TASK LEARNING HELPS?

The motivation for MTL may be sound, but we still have not explored the main reasons that make this approach beneficial. Here we list the main hypothesis for the effectiveness of MTL, as observed by multiple authors.

Preventing overfitting: overfitting is the phenomenon where a learning model is exaggeratedly fitted to the training dataset, so that the model will exhibit poor performance when dealing with unseen data points, as the intrinsic noise of the training dataset was also learned. In MTL, as tasks are learned jointly, the presence of multiple related tasks acts as a regularization mechanism and prevents overfitting (CARUANA,

1997; EVGENIOU; PONTIL, 2004; ZHOU; ZHAO, 2016; ZHONG *et al.*, 2016; RUDER, 2017);

Feature selection: as features that are relevant to more than one task will usually have a smaller impact on the cost function that involves all tasks, MTL will promote representation sparsity by identifying features that have little to no impact in the generalization performance of multiple tasks (CARUANA, 1997; HERNÁNDEZ-LOBATO; HERNÁNDEZ-LOBATO, 2013; LIU *et al.*, 2009);

Data augmentation: when tasks are related, MTL can help by leveraging the signal from data of one task to the other related tasks, as noticed by (CARUANA, 1997; THRUN; O'SULLIVAN, 1996; FELDMAN *et al.*, 2014; JALALI *et al.*, 2010; JACOB *et al.*, 2009b). As all tasks are noisy and different tasks have different noise patterns, leaning them simultaneously helps the model to find a more general representation (RUDER, 2017). This benefit is especially pronounced when few data points are available, with relation to the dimensionality of the datasets (CILIBERTO *et al.*, 2017);

Representation bias: MTL biases the model to use representations preferred by multiple tasks (RUDER, 2017). On neural networks in general, since we have multiple local minima, MTL can help by guiding the search for a solution to a local minimum that is exploited by multiple tasks simultaneously, promoting a representation bias in the iterative weight adjustment (CARUANA, 1997);

It is important to keep in mind that MTL can also negatively impact on models' performance. If the data does not support the assumptions made by the chosen model, it will impose inadequate biases into the learning process that will lead to worse performance. This can also happen when all tasks are related but the model assumptions are not capable of handling the latent relationship among tasks. Whenever performance decreases when compared to STL training procedure, we say that **negative transference** occurred.

2.1.3 RELATED AREAS

Other machine learning problems share similarities with MTL. Here we draw some lines on research areas that are related to MTL, providing key references for the interested reader. Most importantly, we expose the main distinctions between MTL and the related areas.

Some of these related areas can be seen as special cases of MTL, such as Multi-Class, Multi-Label classification, and Multiple-Output regression. Although guarding similarities with MTL, these related areas have the specific characteristic that all tasks share a single dataset.

2.1.3.1 Multi-Class Classification

Multi-class classification occurs when the labels can assume one value from a set of T discrete values. Without loss of generality, this problem can be stated as follows: map X to $y \in [1, \cdots, T]$. Since labels can have multiple values, we can imagine that each possible value is a different task. However, notice that in multi-class the label values are not independent: if one value is assumed, all the others are not. In MTL the tasks are not intertwined like that. Also, in multi-class, we only have a sample X, while in MTL each task can have its dataset. The Handwritten Digit Recognition problem, known by the popular MNIST challenge and benchmark dataset (LECUN; CORTES, 2010), is a good example of multi-class. In this dataset, each data point is an image that contains a hand-written number between 0 and 9. The challenge is to properly map each data point to the correct written number. Since each data point can belong to a single class, we have a multi-class classification problem. A more detailed explanation of multi-class classification is provided by (ALY, 2005).

2.1.3.2 Multi-Label Classification

In Multi-Label Classification we want to predict T binary labels for each data point. This problem can be stated as the following: map data points from X to $y_t \in [0, 1]$, for $t \in [0, \cdots, T]$, where each task t represents a label. Different from Multi-Class classification, labels are independent now, as in MTL. But again, in Multi-Label Classification we only have a single dataset, while in MTL we can have multiple. A problem that perfectly matches this paradigm is the Object Detection in Computer Vision research. In this case, for a single data point (image) we want to identify one or more objects (classes), which characterizes the multi-label classification definition. Surveys with comprehensive coverage of this approach can be found in (LIU *et al.*, 2020; ZHANG; ZHOU, 2014; SOROWER, 2010).

2.1.3.3 Multiple-Output Regression

In Multiple Output Regression, we also want to predict multiple outputs for each data point, as in Multi-Label Classification, but now our labels are continuous. The goal is to find a map from X to $y \in \mathbb{R}^T$. The comparative analysis made for Multi-Label Classification, in the previous subsection, is valid here. Refer to surveys in (BORCHANI *et al.*, 2015; XU *et al.*, 2020).

2.1.3.4 Transfer Learning and Domain Adaptation

In this configuration, we have one or more models already pre-trained to some tasks (*source tasks*), and some new tasks that need new models, called *target tasks*. The goal is to benefit from previously trained tasks using them as an initialization to the target

tasks parameters, to accelerate model training and improve generalization performance. Transfer Learning leverages Domain Adaptation, an area that focuses on learning a source task on a biased domain that may be different at inference time.

One of the main differences between Transfer Learning and Domain Adaptation to MTL is that in the former we start with a subset of models that are already trained, while in the latter all tasks are trained jointly: in MTL all tasks are source and target. Another main difference is that in MTL we are interested in increasing the performance of all tasks, not only of a subset of target tasks.

Among the many successful applications of Transfer Learning, Natural Language Processing (NLP) gives a famous example. BERT (DEVLIN *et al.*, 2019) is a famous pre-trained neural network model consisting of 345 million-parameters. This pre-trained model is used as a language representation model that can be fine-tuned to a varied number of tasks, such as named entity recognition (SOUZA *et al.*, 2019; LIU *et al.*, 2021; LI *et al.*, 2022), sentence textual similarity (REIMERS; GUREVYCH, 2019), and recognizing textual entailment (CABEZUDO *et al.*, 2020; POLIAK, 2020). Transfer learning surveys can be found in (ZHUANG *et al.*, 2021; PAN; YANG, 2010).

2.1.4 MULTI-TASK LEARNING IN DEEP NEURAL NETWORK MODELS

From the conception of the term in (CARUANA, 1997), MTL is easily achieved in neural networks, automatically occurring with the presence of multiple outputs. With the advance of *deep learning* methods and the multiple developments in their architecture, MTL acquired a special place in more applied research fields, such as Natural Language Processing and Computer Vision.

Ruder (2017) discusses MTL in Deep Neural Networks. The author categorizes MTL methods based on how tasks share parameters. In the *hard parameter sharing* approach, networks explicitly share layers of parameters across multiple tasks. When tasks have some shared parameters, but also have their exclusive parameters and transference is induced by regularization, the approach is called *soft parameter sharing*. The author categorizes the recent works that employ MTL in deep neural networks in groups of related strategies, based on architectural decisions.

In (VANDENHENDE *et al.*, 2021) the authors review MTL methods in the context of Dense Prediction Tasks. According to their definition, dense prediction tasks are tasks that produce pixel-level predictions in Computer Vision applications, such as semantic segmentation. The authors present an alternative to the classification of methods made by (RUDER, 2017), arguing that for dense prediction tasks the architectural choices of deep learning models are not the same as deep networks in general. As models in this domain are usually based on an encoder-decoder architecture, it makes more sense to organize the MTL contributions in this field as encoder-based and decoder-based methods.

Now that we have a better definition of MTL, and a more clear picture of how it relates with other machine learning areas, it is time to dive into more technical detail. We are interested in the methods that transfer information through tasks parameters. This strategy is the most popular in MTL (ZHANG; YANG, 2017) and allows us to encode all kinds of assumptions into the model.

2.2 PARAMETER-BASED TRANSFERENCE WITH REGULARIZATION

This section is particularly based on Oliveira *et al.* (2019). Here we will see that, on parameter-based formulations, regularization terms provide a powerful tool to encode assumptions about how tasks are related. Here we visit the most common assumptions in the literature, such as task clustering, low-rank, and dirty/multi-level models, and also present a discussion of how to model groups of related features.

Multiple MTL models use regularization terms to encourage information sharing among the related tasks during training. In this case, regularization terms are applied directly to the tasks' parameters, encouraging specific types of transference among multiple tasks.

Definition 4 (Cannonical Formulation for Regularized MTL Models)**.** The canonical formulation for the Regularized Multi-Task Learning problem is given by:

$$\min_{W} \quad \sum_{t=1}^{T} \mathcal{L}_t(\mathbf{w}_t) + \mathcal{R}(W), \tag{2.1}$$

where $\mathcal{L}_t(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a suitable convex loss function for task t ($\mathbf{w}_t \in \mathbb{R}^n, \quad \forall t \in \mathcal{T}$), and $\mathcal{R}$ is a regularization term over all tasks parameters. When the tasks weights are stacked as columns of a matrix, we represent the task parameters as $W \in \mathbb{R}^{n \times T}$.

Examples of loss functions for regression problems are Mean Squared Error and Mean Absolute Error, while classification problems may use a cross-entropy loss. The regularization term $\mathcal{R}(W)$ is usually convex and possibly non-differentiable. In this term, we may consider additional aspects such as i) using prior knowledge to act as bias on tasks parameters; and ii) adding a mechanism that enforces and/or captures the relationship among tasks.

We present now a summary of the most common approaches used by models that share information through tasks parameters, named: i) Task Clustering; ii) Low-Rank Decomposition; iii) Dirty Models; and iv) Grouping Features. Given the connection of the *structure estimation* approach with our work, we treat it with more detail in Section 2.3.

2.2.1 TASK CLUSTERING

Task clustering is the assumption that all tasks belong to one or more groups of related tasks, usually considering all features in the clustering mechanism.

Evgeniou e Pontil (2004) started from the general premise that the parameter vectors of tasks should be close to each other. Following the intuition of Hierarchical Bayes (HESKES, 2000; ALLENBY; ROSSI, 1998), assuming that all $\mathbf{w}_t \in \mathbb{R}^n$ can be decomposed as $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$, where $\mathbf{v}_t$ should be small for all tasks, their model penalizes the deviation from the "average model" $\mathbf{w}_0$ using slack variables. The dual problem of their optimization problem is linked with the dual problem of standard SVM.

Jacob *et al.* (2009a) also resort to the task clustering assumption and start their proposal by assuming that task clusters are known a priori. They use three regularization penalties, one penalizing the magnitude of tasks parameters; a second one that penalizes the between-cluster variance for each group of tasks; and a third regularization term that penalizes the within-cluster variance. After defining the regularization term that encodes these assumptions, they relax the requirement of task assignment to clusters, turning it into another variable of the optimization problem. To make their proposal tractable, they use a convex relaxation approach. Elaborating further, Zhou *et al.* (2011a) considered that tasks parameters are grouped into K groups, being K a hyper-parameter of the model. They employed a k-means inspired strategy to find the task clusters, but also needed to relax the original problem to have a tractable formulation.

Despite allowing tasks to share information, the aforementioned methods induce all the parameter vectors to pursue the average behavior of their corresponding group of tasks. Another implication is that if tasks are related only through a subset of features, both methods will fail to capture the relationship and will enforce the relationship on features that are not supposed to be related.

2.2.2 LOW-RANK DECOMPOSITION

In this approach, we assume that tasks parameters share a latent space and their parameters can be retrieved through a combination of vectors of the latent basis, as illustrated in Fig. 7.

A well-known method of this category is GO-MTL (KUMAR; DAUMÉ, 2012), that considers a latent space where task parameters can be linearly decomposed. Let $L \in \mathbb{R}^{n \times k}$, where k is the dimension of the latent basis, and $S \in \mathbb{R}^{k \times T}$ be a matrix with the weights of a linear combination of tasks. Assuming that $W = LS$, the associated optimization problem is defined as:

$$\min_{W} \quad \sum_{t=1}^{T} \mathcal{L}_t(L\mathbf{s}_t) + \lambda_1 \|S\|_1 + \lambda_2 \|L\|_F^2 \tag{2.2}$$

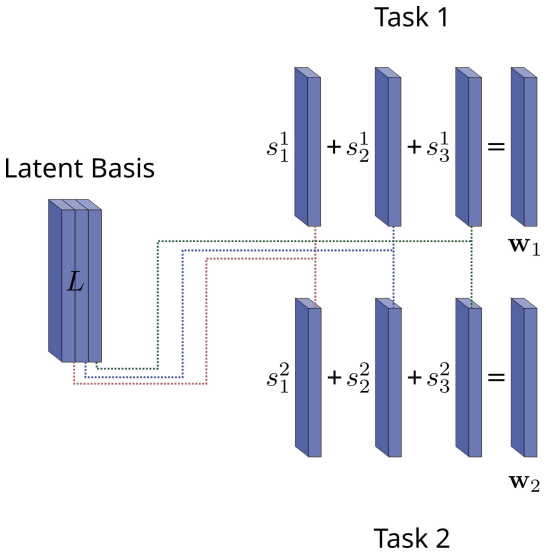


Figure 7 – The Low Rank Decomposition strategy for MTL. In this class, tasks parameters lie on a shared latent basis L and their actual values may be recovered through a linear combination of this basis. The parameters of the linear combination of this basis are arranged in the matrix S.

The regularization term is composed of two norms on matrices S and L, $\|S\|_1$ being the entry-wise l_1-norm, and $\|L\|_F^2 = Tr(LL^T)^{\frac{1}{2}}$ being the Frobenius norm of a matrix. The norm on L restrains the magnitude of tasks parameters, while the sparsity term on S enforces the tasks to derive from a small subset of the latent basis L. The relationship between tasks occurs when two tasks share components in L, based on their decomposition coded in S. If a task does not share basis vectors on S with any other task, it may be interpreted as an outlier task.

Notice that by assuming that tasks parameters share the latent basis, GO-MTL considers that tasks are related in possibly overlapping groups, i.e., one task can be part of several groups. However, when two tasks are related, all parameters are involved in this relationship since each component of the latent basis represents all tasks parameters.

Argyriou *et al.* (2008) present MTFL, a method that learns sparse feature representations for the tasks on an orthogonal basis. This proposal also allows features to be discarded when they are not beneficial to the performance of any task. In this case, tasks are grouped based on the usage of the same feature representations. The proposal of Kang *et al.* (2011) is based on Argyriou *et al.* (2008) with the difference that they apply a low-rank constraint on the latent basis, and an $l_{2,1}$-norm is used to enforce a relationship on all tasks for each feature. In this work, tasks are assigned to a group of related tasks using an integer programming approach (thus the title: learning with whom to share), and the tasks' parameters are optimized independently by each group of related tasks.

Other methods in this category take advantage of the trace norm as a regularization term, which can force a matrix to have a low rank. This approach is explored with detail

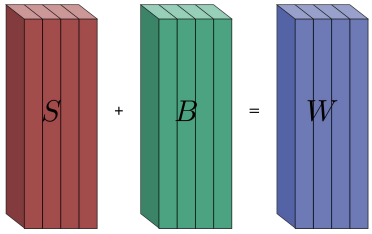


Figure 8 – The Dirty Model strategy for MTL. The parameter matrix W is decomposed into a sum of two matrices. Different regularization terms can be applied to the components of the decomposition, encoding a priori assumptions of the problem.

by (PONG *et al.*, 2010), where the authors derive both primal and dual reformulations of the related optimization problem, prove the existence and uniqueness of stationary points on an augmented formulation and compare multiple solutions to this problem. A variation of this formulation is proposed by (HAN; ZHANG, 2016), using a capped trace regularizer, that minimizes only the singular values that are smaller than some threshold.

2.2.3 DIRTY MODELS

The dirty models approach is based on a decomposition of the tasks parameter matrix W into a sum of matrices. For example, in (JALALI *et al.*, 2010) it is assumed that $W = S + B$, as depicted in Fig. 8. This allows us to encode more flexible assumptions about task relatedness by employing different regularization terms on each component of the decomposition. In this work, the authors use the $l_{1,1}$-norm (the sum of l_1-norm of all columns of the matrix) in one component, to induce sparsity over the parameters of all tasks; and the $l_{1,\infty}$-norm (the sum of l_{inf}-norm of all columns of the matrix) in the other component, to enforce a relationship among all tasks considering each feature in isolation.

The referred optimization problem becomes:

$$\min_{W} \quad \sum_{t=1}^{T} \mathcal{L}_t(\mathbf{s}_t + \mathbf{b}_t) + \lambda_1 \|S\|_{1,1} + \lambda_2 \|B\|_{1,\infty}. \tag{2.3}$$

In this formulation, one feature can be active for all tasks but each task is free to avoid this feature with the regularization of the second term. The factorization strategy adds the flexibility of sharing features between tasks when convenient, but we still are left with two limiting properties: i) the $l_{1,\text{inf}}$-norm encourages similar values for each feature across all tasks, implying that one feature has the same impact on the outcome of all related tasks; and ii) the model does not consider the case of grouped features. Other works apply different regularization terms to the components of the parameter matrix decomposition (CHEN *et al.*, 2011; CHEN *et al.*, 2012; GONG *et al.*, 2012).

In (ZHANG; YANG, 2017), when the number of components in this decomposition is greater than two, it is referred to as a *multi-level approach*. Increasing the number

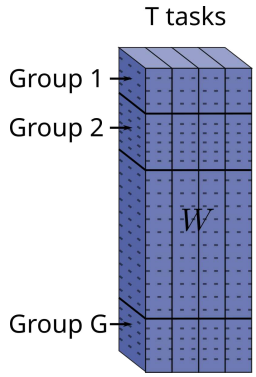


Figure 9 – Grouped Features in the parameters of a linear MTL model. Each group of related features corresponds to a group of rows in the parameter matrix W.

of components allows for more flexibility in using different norms to encode multiple assumptions that increase the complexity of the tasks' structure. On the other hand, more components increase the number of terms in the cost function, which increase the challenges associated with the optimization. This approach is used by works such as (JAWANPURIA; NATH, 2012) and (HAN; ZHANG, 2015).

2.2.4 GROUPING FEATURES

Let us consider the l_1-norm, a common choice for a regularization term. When applied to the parameters of each task individually, it encourages a sparse activation of its features. When applied to different arrangements of parameters, it can encourage information sharing over all tasks on each feature independently. This regularization has the nice property of being able to recover the exact support - i.e., the set of active variables of a vector - of a given model when using data generated from this model, if the parameters are not too correlated (NEGAHBAN; WAINWRIGHT, 2009). However, if the parameters are structured into correlated groups, it loses support recovery guarantees and interpretability of results (OBOZINSKI *et al.*, 2011). The family of $l_{p,q}$-norms is suited to enforce that groups of features are jointly active or absent. If one feature of a given group is active, all features in the same group should be active; and if one feature is not active, the whole group of features should not be active.

The Group LASSO regularization (YUAN; LIN, 2006) is a regularization that accounts for grouped features. Let the tasks features be partitioned into G groups of correlated features. Each group $g \in \mathcal{G} = \{1, \cdots, G\}$ consists of a subset of features in X_t for all tasks $t \in \mathcal{T}$. Let X_t^g be the design matrix restricted to the features present in the group g for task t, and $\mathbf{w}_t^g$ be the task parameter with the same dimension as $\mathbf{w}_t$ but admitting non-null values only at locations associated with features belonging to group g, and having null values at the remaining positions. The Group LASSO regularization

term is defined as:

$$\mathcal{R}_{GL}(W) = \sum_{t \in T} \sum_{g \in \mathcal{G}} \|\mathbf{w}_t^g\|_2.$$

Notice that the partition of features into the same groups is the same for all tasks, as represented in Fig. 9.

When used as the regularization term of the canonical regularized multi-task learning formulation in Eq. (2.1), the MTL optimization problem becomes:

$$\min_W \quad \sum_{t \in \mathcal{T}} \mathcal{L}(\mathbf{w}_t) + \lambda \sum_{g \in \mathcal{G}} \|\mathbf{w}_t^g\|_2. \tag{2.4}$$

In this penalty, each feature must belong to one group, although isolated features can be put into a singleton group. As it penalizes the l_1-norm of a vector of G l_2 norms (one per group), when one element is forced to zero, all features of this group are forced to zero, in other words: $\mathbf{w}_t^g = 0$ for some $g \in \mathcal{G}$. However, when two groups overlap and only one group is active in the final solution, the group that is not active will have all its features zeroed, even the features that are shared with the active group. The recovered support of this norm is then the complement of the union of the overlapping groups (JACOB *et al.*, 2009b; KOLAR *et al.*, 2011; VOGT; ROTH, 2010). One appealing property of the $l_{2,1}$-norm regularization in MTL is that it can help encouraging multiple predictors from different tasks to share similar parameter sparsity patterns, as in a variation of MTFL proposed by Liu *et al.* (2009).

Jacob *et al.* (2009b) proposed an extension to the Group LASSO where the feature vector is decomposed as a sum of representations for each group $\mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{u}_t^g$, applying the l_2-norm (or l_{inf}) on each group. The difference from the Group LASSO is that $\mathbf{u}_t^g \forall g \in \mathcal{G}$ are latent representations of $\mathbf{w}_t$, instead of being a direct partition. This regularization is called the Latent Group LASSO, or Overlapping Group LASSO, and can be posed as:

$$\mathcal{R}_{OGL}(W) = \sum_{t \in T} \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2,$$

where $\mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{w}_t^g, \quad \forall t \in [1, \cdots, T]$, and d_g are independent weights for each group, accounting for the cardinality of each group. The MTL optimization problem using the Latent Group LASSO independently over all tasks is represented as:

$$\min_W \quad \sum_{t \in \mathcal{T}} \mathcal{L}(\mathbf{w}_t) + \lambda \sum_{g \in G} d_g \|\mathbf{w}_t^g\|_2 \tag{2.5}$$

$$\text{s.t.} \quad \mathbf{w}_t = \sum_{g \in G} \mathbf{w}_t^g, \forall t \in [1, \cdots, T]. \tag{2.6}$$

In this case, the support for each task is a union of groups, not the complement, as a feature shared by two groups will have its value preserved for the active group and be zeroed in the inactive group.

Both sparsity inducing l_1 and $l_{p,q}$-norms present strong support guarantees when appropriate conditions are met. However, the performance of methods based on $l_{p,q}$-norms, depends on how features are shared across tasks. For the $l_{1,q}$ norm, (NEGAHBAN; WAINWRIGHT, 2009) showed that if the number of tasks sharing a group of features is less than a threshold, or even if the parameter values of features of the same group are highly uneven, the regularization could perform worse than the l_1 norm. Ideally, each group of features should be free to play distinct roles depending on the task, i.e., each task may have its support (number of non-null elements in $\mathbf{w}_t^g$). In this case, we still need a mechanism to select which tasks should transfer to each group independently.

Several MTL methods that use the Group LASSO regularization find applications in the medical sciences. G-SMuRFS (WANG *et al.*, 2012) finds groups of related variables using correlation, and applies the Group LASSO to enforce the estimated feature grouping. The authors apply their method to identify quantitative trait loci - associations between genetic variations and imaging measures - to better understand the underlying biological etiology of Alzheimer's Disease (AD) on the Alzheimer's Disease Neuroimaging Initiative dataset. The ADNI was launched in 2003 to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of AD in early stages or when there already is a mild cognitive impairment (MCI) [1]. MT-SGL (LIU *et al.*, 2018) uses the Group LASSO and the $l_{2,1}$-norm to encourage individual feature selection that will be used on all tasks, and also models regions of interest (ROI) in the brain with the Group LASSO. In FL-SGL (LIU *et al.*, 2019) several LASSO extensions such as the Group LASSO and the Fused LASSO (TIBSHIRANI *et al.*, 2005) were used, accounting for ordered features to consider temporal smoothness.

All methods presented so far allow tasks to transfer in different ways, but exhibit several limitations, namely i) the presence of non-convex and / or non-smooth terms in the optimization formulation; and ii) the need to meet strong assumptions beforehand. Since gradients are not always available, vanilla gradient-descent methods are not able to solve such formulations. A different set of optimization tools is required to handle such problems, which are presented in Chapter 4. Since all tasks are considered to be related, they may not be robust to the presence of unrelated tasks as they force them to be related; most of them do not account for grouped features and when they do, all tasks must share the same sparsity pattern, which implies that each group of features has the same influence on all tasks outcomes. To avoid imposing strong assumptions beforehand, and better handle isolated tasks, we proceed to a distinct family of MTL methods that estimates how tasks are related while learning the tasks' parameters.

[1] Information from <https://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Manuscript_Citations.pdf>

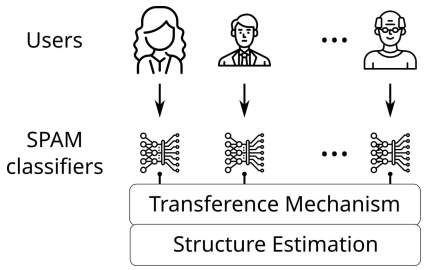


Figure 10 – Multi-Task Structure Learning. In this setting, a mechanism responsible for estimating how tasks are related is included as an additional module.

2.3 MODELING TASK RELATIONSHIPS: STRUCTURE ESTIMATION FOR MULTI-TASK LEARNING

In MTL, *Structure Estimation* is the process of not only estimating all tasks' parameters but also how transference occurs from one task to another. Instead of meeting strong assumptions of how tasks are related, it is possible to estimate statistically relevant transference patterns by encoding them into variables that are not directly mapped into the predictive loss function. In Fig. 10 we can see a depiction of this strategy. We learn all tasks simultaneously, with the help of a coupling mechanism that estimates the relationships among tasks. As we will see, this mechanism can take many forms.

MTRL (ZHANG; YEUNG, 2010b) uses a probabilistic framework and places a matrix-variate prior distribution on tasks coefficients to model their relationship. They estimate a precision matrix Ω as the transference mechanism. The precision matrix is the inverse of the covariance matrix, $\Omega = \Sigma^{-1}$. It is important to highlight that if one of its elements is set to 0, the involved variables are conditionally independent. Similar to MTRL, MSSL (GONÇALVES *et al.*, 2016) relies on a probabilistic framework, in which a sparse precision matrix is learned from the data to capture tasks relationship and to help in isolating unrelated tasks. A semi-parametric copula distribution is used as prior for task parameter matrix, capturing non-linear correlation among tasks. They also use a LASSO penalty on the task parameters for automatic feature selection, which adds non-smooth terms in the associated optimization problem. Since the transference structure is encoded in a precision matrix, both methods share the property that transference between two tasks is symmetric, i.e., transference from task t to task s is the same as transference from task s to task t. Moreover, as the precision matrix relates two tasks over all tasks parameters, these methods do not account for groups of features.

AMTL (LEE *et al.*, 2016) assumes that parameters of a task t can be approximated by a sparse linear combination of the parameters of all other tasks. In other words, $\mathbf{w}_t \approx W\mathbf{b}_t$, where $\mathbf{b}_t \in \mathbb{R}^T$ is a vector with the coefficients of the linear combination. A task cannot participate in its own formulation, thus $\mathbf{b}_{tt} = 0 \; \forall t \in \mathcal{T}$. In this case, the tasks' parameters serve as a latent basis. The authors also use tasks losses to weight trans-

ference: relationships must flow from tasks with lower cost (easier to learn) to tasks with higher cost (harder to learn). Let $B \in \mathbb{R}^{T \times T}$ be a task relationship matrix where $\mathbf{b}_t$ is the t-th column, and $\mathbf{b}_{\bar{t}}$ its t-th row. Each column t indicates how the parameters of the other tasks participate in the linear combination that approximates the parameters of task t, and a row t indicates the degree with which the parameters of a task t participate in the approximation of the parameters of other tasks. Therefore, B encodes the relationships among tasks in an asymmetric scheme: the transference from task t to task s may not be the same as that from task s to task t.

The associated optimization problem is written as follows:

$$\min_{W} \quad \sum_{t=1}^{T}(1 + \lambda_1 \|\mathbf{b}_{\bar{t}}\|_1)\mathcal{L}_t(\mathbf{w}_t) + \lambda_2 \|\mathbf{w}_t - W\mathbf{b}_t\|_2^2. \tag{2.7}$$

In the first term, the cost of a task t weights the l_1-norm applied to $\mathbf{b}_{\bar{t}}$ (t-th row of B), i.e., the transferences from task t to all other tasks. (λ_1, λ_2) are regularization hyper-parameters. The asymmetrical transference is encoded in a set of variables that are distinct from the variables involved in prediction, which allows AMTL to achieve a flexible regularization of related tasks. Nevertheless, AMTL also enforces the transference considering all features, which we will call from now on as **global feature transference**. The formulation also contains non-convex and non-smooth terms.

Now we are familiar with the main challenges involved with the presence of multiple learning tasks. Not only that, we reviewed a portion of the literature exploring what and how information is shared. The main advantages and drawbacks of each approach were discussed and the common ground is set with relation to ML research.

2.4 FINAL REMARKS

In this chapter, we formally defined Multi-Task Learning and reviewed a relevant portion of the current literature. We could see how complex can become the relationships among tasks. The transference between two tasks can include only one feature or a subset of features of arbitrary size. This implies a combinatorial search on the powerset of the features, for all possible pairs of tasks. If we consider that the two tasks may be related asymmetrically, the search space is much larger. None of the proposals can properly handle such complex scenarios. If not supported by tasks data, many assumptions made by each category of methods can lead to negative transference. The usage of regularization terms became a common option to model a priori knowledge about tasks relationships, to overcome the risk of negative transference. Another common trend is to estimate how tasks are related, avoiding the imposition of wrong a priori assumptions.

As presented, although the current state-of-the-art MTL methods can model and learn the tasks relationship information from the data, two major drawbacks are present:

(i) most of the methods assume that tasks are symmetrically related, that is, task A affects task B in the same way that task B affects task A; and (ii) if two tasks are related, they must influence each other on the entire set of features. There have been attempts to enhance the flexibility of these models and to alleviate the mentioned downsides, but no candidate simultaneously accounts for both the local feature transference and asymmetrical structural learning. Filling this gap is precisely the main theme of GAMTL, the main contribution of this work - presented in Chapter 5.

Most presented methods leverage properties of some norms to induce sparsity in their associated cost functions. The choice of norms to compose a regularization term presents new challenges, as we need to understand how the norms promote sparsity and how much we can trust on the few variables that remain active in the final solution. The norms are used in distinct arrangements of the variables, allowing us to encode complex relationships among tasks and/or features in the MTL formulation. For the models that we presented in this chapter, sparsity by itself is not enough if we want to handle groups of correlated variables. We may need to structure how we enforce sparsity properties in the variables. In the next chapter, we present how some norm-families promote sparsity in optimization problems, and detail specific properties that are useful for ML and statistics models.

3 INDUCING SPARSE ACTIVATION OF VARIABLES IN MACHINE LEARNING MODELS AS A WAY TO ENCODE PRIOR KNOWLEDGE

The previous chapter covered the main ideas behind Multi-Task Learning and exhibited how regularization is a key component in multiple strategies used by MTL methods to model the transference of information among tasks. The usage of multiple regularization terms is widely adopted, and many of them leverage sparsity properties in many ways, such as a feature selection mechanism, sparse activation of a latent basis, and so on. In this chapter, we study the class of sparse models in ML. We discuss how some norm families promote sparsity when used as regularization terms of a learning problem. At the end of this chapter, the reader will know why some norms induce sparse solutions, in which scenarios this property is useful, and how to promote sparsity in a structured way.

3.1 REGULARIZATION AND SPARSITY INDUCING NORMS

Regularization terms are important in machine learning because of multiple factors. By penalizing the magnitude of the parameters of the model being trained, these terms prevent features of the optimization problem of assuming large values and turn the output of the model too sensitive to their values, helping to alleviate overfitting problems. Regularization terms can turn the optimization problem better conditioned, or enhance smoothness in the error surface, turning the optimization easier. As we could see in Chapter 2, regularization terms are also used to impose restrictions and prior knowledge into the structure of the estimated parameters, in the presence of multiple tasks.

The formulation for a general regularized ML model (of a single task) is defined as:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \lambda \mathcal{R}(\mathbf{w}), \tag{3.1}$$

where $\mathcal{L}$ is a proper loss function, $\mathcal{R}$ is a regularization possibly norm-based, and $\lambda > 0$ is a hyper-parameter that weights the regularization term in the cost function.

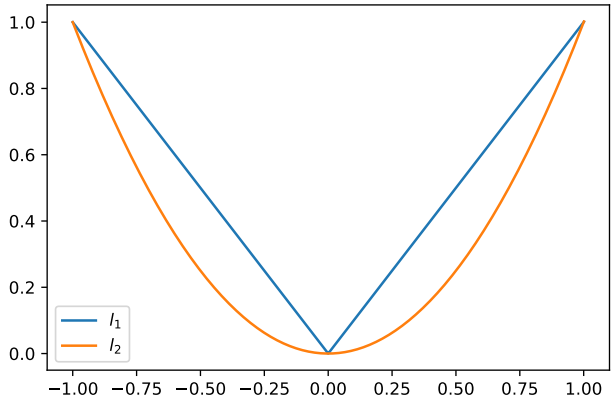


Figure 11 – The norm-ball of l_p norms, when $p = [1, 2]$. Notice that when $p = 1$ we have a singularity at $x = 0$.

Let $p \geq 1$ be a real number. The l_p-norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is given by

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}.$$

When $p = 1$, we have the **Manhattan or Taxicab norm**; when $p = 2$, we have the Euclidean or **Frobenius norm**; and if p approaches infinity, we have the **Maximum norm**.

Notice that the formulation in 3.1 is the Lagrangian formulation of the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathcal{L}(\mathbf{w}) \\ \text{s. t.} \quad & \mathcal{R}(\mathbf{w}) \leq C, \end{aligned} \tag{3.2}$$

where C is a hyper-parameter. There is a one-to-one correspondence between the formulations in Eq. (3.1) and Eq. (3.2). This alternative formulation has the nice property that the constraint is exactly the definition of a norm ball.

Let us examine the norm ball of l_p-norms, for some values of p in Figure 11. When $p = 1$, the value of the norm is zero if $x = 0$ and increases linearly with the absolute value of x. This norm ball is convex but not smooth at $x = 0$, where its derivative can take infinite values. When $p = 2$, the value of the norm follows a smooth quadratic function of the absolute value of x. Notice that in this case, the norm ball is convex and smooth for all values of x. If p approaches infinity, the norm always assumes the maximum value of x.

The geometry of the norm balls highlights important properties of the norm and carries an easy intuition of how the norm will impact the surface of the cost function in a regularized ML model. For example, for any smooth convex non-constrained problem,

a local optimum is a global optimum and the solution can be found using any gradient-based optimization method. When the chosen norm introduces non-smoothness into the problem, gradients may not even exist for all elements of the domain. But it is precisely at the introduced discontinuities (or singularities) that sparsity is achieved.

As a didactic exercise, let $\mathbf{x} \in \mathbb{R}^2$ and $f(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 5)^2$, the area of a circle with center at $x_1 = 3, x_2 = 5$. Let us minimize the area of this circle, with the constraint that $\|\mathbf{x}\|_p = 1$, for $p = [1, 2]$. This can be expressed as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & (x_1 - 3)^2 + (x_2 - 5)^2 \\ \text{s. t.} \quad & \|\mathbf{x}\|_p = 1. \end{aligned}$$

Let us examine this problem for some values of p. Figure 12 depicts the problem for $p = 2$. The contour levels of $f(\mathbf{x})$ show all possible values for $\mathbf{x}$ with f value associated with the color of the lines. The l_2-norm ball that corresponds to the constraint is shown in blue. The solution of our problem must lie in the intersection of f with the norm ball in order to meet our constraint. Since we want to minimize the area of the circle, the first contour line that touches norm ball contains our solution. The optimal solution is depicted with an 'X' mark, at the point $\mathbf{x}^* = \left[\frac{3}{\sqrt{34}}, \frac{5}{\sqrt{34}}\right]$. Notice that the values for both x_1 and x_2 are different than zero. The features could only have zero as their optimal value if the f circle was centered at zero in one of the coordinates.

In Figure 13, we show the same optimization problem, but now with $p = 1$. The solution that meets our constraint also lies at the intersection of a contour line of f with the l_1-norm ball, and is marked with an 'X' mark at $\mathbf{x}^* = [0, 1]$, even though f is not centered at any axis. As noticed by (HASTIE *et al.*, 2015), this does not occur for norms with $p > 1$; while for $p < 1$, although solutions are also sparse, the problem is not convex.

There are a few important reasons that make sparse solutions desirable, if not in general, for plenty of applications. Hastie *et al.* (2015)[p. 13] highlights three main motivations for sparsity. The first is the **simplicity assumption**: as few features are active in the final solution, the resulting model is simpler to understand and thus more interpretable. The second motivation is the **computational convenience** brought by sparse models. As many features are zeroed, sparse models have significantly less memory requirements. Sparse solutions lie on smaller sub-spaces and can handle many more features than a dense counterpart. This allows us to handle high-dimensionality problems with less memory requirements in a reasonable amount of time. The third motivation is what the authors termed as the **bet on sparsity principle**:

> "Use a procedure that does well in sparse problems, since no procedure does well in dense problems." (HASTIE *et al.*, 2015).

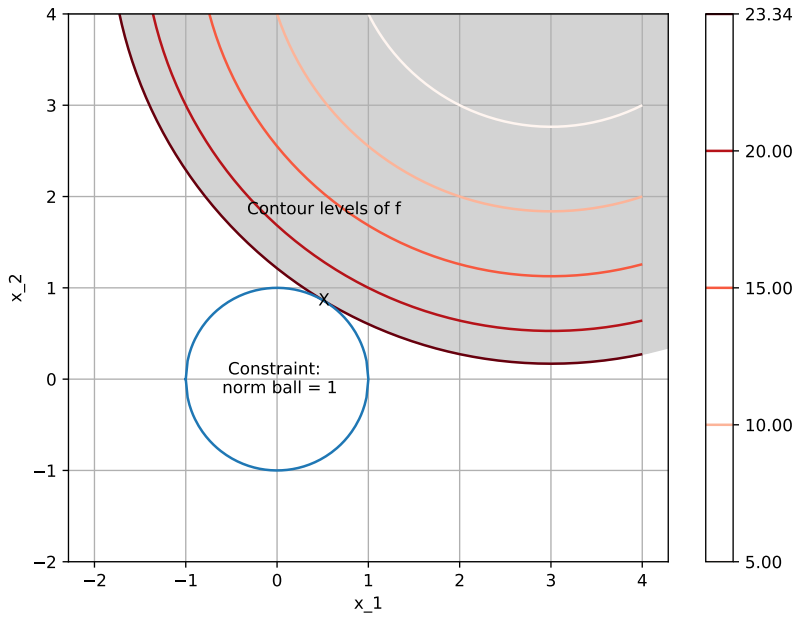


Figure 12 – Depiction of the optimization problem where we minimize the area of the circle, constrained to the l_2-norm ball, in two dimensions.

In cases where the number of features in a problem is much larger than the number of data samples, it is usually safer to use sparse models even if the underlying model is not sparse, as the low number of samples does not allow us to find the optimal solution.

Sparsity plays a key role in other areas of research as well. For instance, in Compressed Sensing (DONOHO, 2006) the goal is to measure n general linear functionals of $\mathbf{x} \in \mathbb{R}^m$, a digital image or signal, where $n < m$, and reconstruct the original signal as precisely as possible. $\mathbf{x}$ is assumed to have a sparse representation in some orthonormal basis (DONOHO, 2006; CANDES *et al.*, 2006). When the samples of $\mathbf{x}$ are noisy, Orthogonal Matching Pursuit (CAI; WANG, 2011; TROPP; GILBERT, 2007) is a popular algorithm based on sparsity properties. In this case, sparsity is seen as a compression mechanism.

3.1.1 THE l_1-NORM AND THE LASSO

Originally proposed for regression problems, the LASSO (TIBSHIRANI, 1996) is a regularized linear model that is based on the l_1-norm. It is defined as follows.

Definition 5 (LASSO)**.** The LASSO model finds the solution to the following optimization problem (TIBSHIRANI, 1996):

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|X\mathbf{w} - \mathbf{y}\| \\ \text{s. t.} \quad & \|\mathbf{w}\|_1 \leq C, \end{aligned} \tag{3.3}$$

where C is a hyper-parameter.

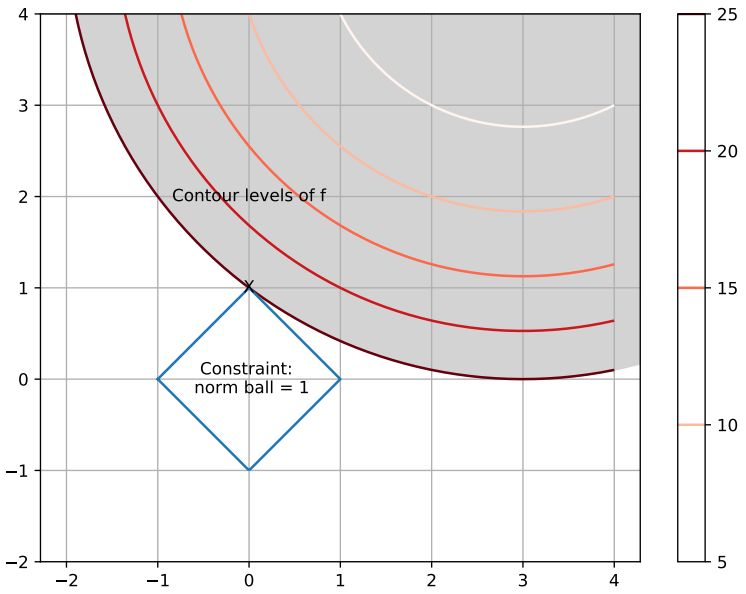


Figure 13 – Depiction of the optimization problem where we minimize the area of the circle, constrained to the l_1-norm ball, in two dimensions.

As we did with Eq. (3.1), it is convenient to rewrite the problem in the Lagrangian form, where the constraint is incorporated into the objective function (Eq. (3.1)).

Definition 6 (LASSO in the Lagrangian Form)**.** The LASSO formulation is as follows:

$$\min_{\mathbf{w}} \quad \|X\mathbf{w} - \mathbf{y}\| + \lambda\|\mathbf{w}\|_1, \tag{3.4}$$

where $\lambda > 0$ is the Lagrangian multiplier.

The hyper-parameter λ allows us to tune the weight of the regularization in the solution, imposing more or less sparsity in the estimated parameters.

As highlighted in (HASTIE *et al.*, 2015)[p. 9] by Lagrangian duality, there is a one-to-one correspondence between the constrained problem and the Lagrangian form: for each value of C in the range where the constrain $\|\mathbf{w}\|_1 \leq C$ is active, there is a corresponding value of C that yields the same solution from the Lagrangian form.

Tibshirani (1996) analyzes the solution of Eq. 3.6 when X is orthonormal, i.e., $X^TX = \mathbf{I}$. Let $\mathbf{w}_j^o$ be the related least-squares solution of Eq. 3.5 for a feature j, without considering the LASSO constraint. The solution for Eq. 3.5 is:

$$\mathbf{w}_j = \text{sign}\left(\mathbf{w}_j^o\right)\left(|\mathbf{w}_j^o| - \gamma\right)^+, \tag{3.5}$$

where γ is determined by the condition that $\sum_j |\mathbf{w}_j| = \lambda$, and $(\cdot)^+ = max(0, \cdot)$. This solution is called the *soft-thresholding operator*.

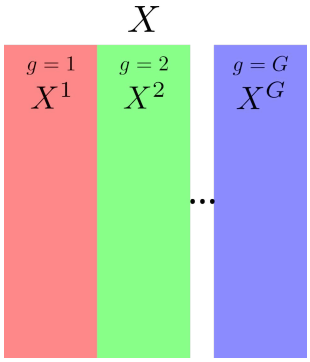


Figure 14 – Dataset X partitioned in g groups of correlated features. This imposes extra challenges to the l_1-norm regularization.

Using this operator in a coordinate descent iterated procedure to solve the LASSO can be reasonably fast because the coordinate-wise minimizers are explicitly available in Eq. (3.5), and thus we do not need to search along each coordinate. The sparsity property of the problem also avoids multiple computational steps, as for large enough values of λ, most of the features are set to zero. More information on numerical options to solve Eq. 3.5 are available at (HASTIE *et al.*, 2015).

Notice, however, that the LASSO is not well suited to handle groups of correlated features. As noticed in (JACOB *et al.*, 2009a; OBOZINSKI, 2011), in the presence of groups of correlated features, the LASSO model loses important properties such as the recovery of the true support even if given enough samples. This is an important limitation, as groups of features naturally arise in many statistical and learning applications. For instance, in linear models: i) a categorical feature is commonly transformed to a group of dummy variables; ii) a continuous feature may be represented by a group of basis functions that incorporate non-linear relationships; or iii) some specific domain prior knowledge can be introduced into the model in a way that features are arranged in structured groups - example: genes in the same biological pathway may be part of the same group (XU; GHOSH, 2015).

Let us see how we can impose sparsity properties when features are correlated in groups.

3.2 STRUCTURED SPARSITY: GROUPING FEATURES WITH THE GROUP LASSO

Features correlated in groups arise in many applications. For example, it may happen if groups of different features are collected from the same source, may belong to a nearby area or to the same time interval, or may even represent a single feature collected by multiple sources. In all these cases subsets of features are correlated.

Assume that $X \in \mathbf{R}^{m \times n}$ can be partitioned into G groups of correlated features with $|G| < n$, as illustrated in Fig. 14. When estimating a linear model to predict the corresponding labels of samples in X, we are interested in sparse solutions that can penalize all features of the same group in the same way. In other words, if a feature in a group g is active, a reasonable assumption is that all other features of the same group are active. On the other hand, we expect that if a feature of a group g is not active, meaning that it is not relevant for predicting the label, all features of the same group are not relevant as well. This is usually called *structured sparsity*, since we are structuring the sparsity pattern retrieved by the regularized ML model.

If we know a priori that our features are correlated, we want to be capable of inserting this information into the regularization terms of our model. The Group LASSO is a widely used norm for this case. It is a special case of the mixed $l_{p,q}$-norm family where $p = 2$ and $q = 1$, and is defined as follows.

Definition 7 (Group LASSO)**.** Let the features of the design matrix be partitioned into G groups of correlated features. Each group $g \in \mathcal{G} = \{1, \cdots, G\}$ consists of a subset of features in X. Let $\mathbf{w}^g \in \mathbb{R}^{|g|} = [\mathbf{w}_i \text{ if } i \in g]^T$. The Group LASSO solves the following problem (YUAN; LIN, 2006):

$$\min_{\mathbf{w}} \quad \|X\mathbf{w} - \mathbf{y}\| + \lambda \sum_{g \in \mathcal{G}} \|\mathbf{w}^g\|. \tag{3.6}$$

In this penalty, each feature must belong to one group, although isolated features can be put into a singleton group. As it penalizes the l_1-norm of a vector of G l_2 norms (one per group), when one element is forced to zero, all features of this group are forced to zero. In other words, $\mathbf{w}_t^g = 0$ for some $g \in \mathcal{G}$. However, when two groups overlap and only one group is active in the final solution, the group that is not active will have all its features zeroed, even the features that are shared with the active group.

As a regularization, the Group LASSO is defined as:

$$\mathcal{R}_{GL}(\mathbf{w}) = \|\mathbf{w}\|_{2,1} = \sum_{g \in \mathcal{G}} \|\mathbf{w}^g\|_2.$$

The recovered support of this norm is then the complement of the union of the overlapping groups (JACOB *et al.*, 2009b; KOLAR *et al.*, 2011; VOGT; ROTH, 2010).

In order to handle the overlapping among groups of features, the Overlapping Group LASSO (JACOB *et al.*, 2009a) proposes to manage the groups in a latent space, so the support of the recovered parameters is the union of possibly active groups, as opposed to the complement of the union when groups overlap. Let X^g be the design matrix restricted to the features present in a group g, and $\mathbf{w}^g$ be the task parameter with the same dimension as $\mathbf{w}$ but admitting non-null values only at locations associated with features belonging to group g, and having null values at the remaining positions. The Overlapping Group LASSO is defined as follows.

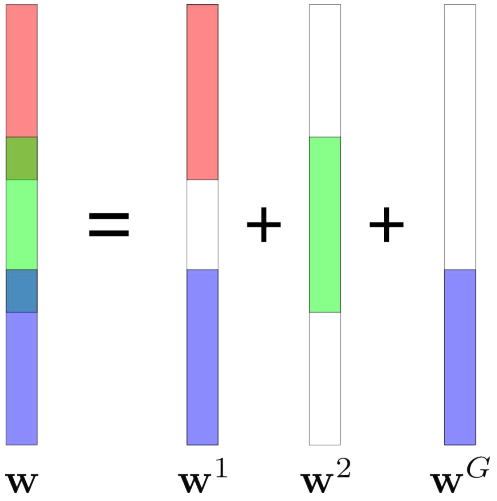


Figure 15 – Behavior of the Overlapping Group LASSO when X contains g groups of correlated features. Notice that when a group of features is not active, the features that are also present in other groups of features can still be active.

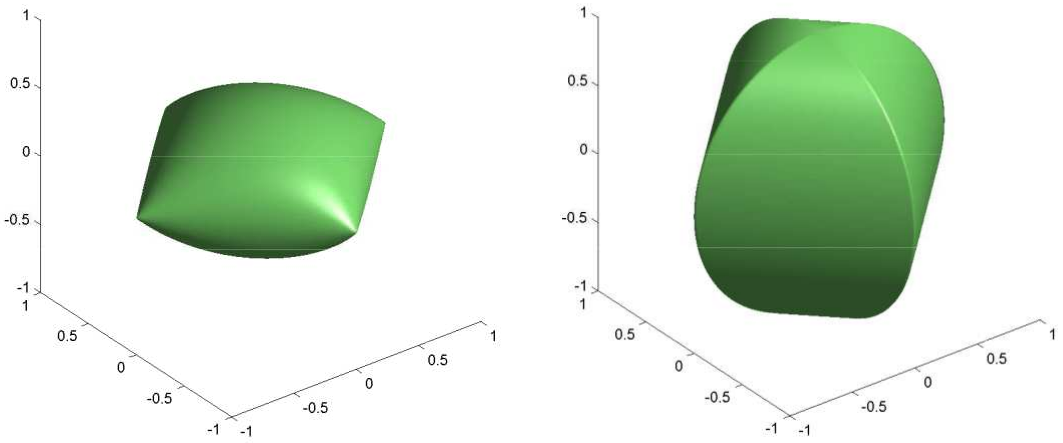
Definition 8 (The Overlapping Group LASSO)**.** Assume that $\mathbf{w} = \sum_g \mathbf{w}^g$ where each w^g represents the portion of $\mathbf{w}$ that is included in the group $g \in \mathcal{G}$. The Overlapping Group LASSO regularization term is defined as (JACOB *et al.*, 2009a):

$$\mathcal{R}_{OGL}(\mathbf{w}) = \sum_{g \in \mathcal{G}} \|\mathbf{w}^g\|_2, \text{ where } \mathbf{w} = \sum_g \mathbf{w}^g.$$

Fig. 15 shows how the Overlapping Group LASSO setting can handle the overlapping features. In this formulation, the features belonging to more than one group are considered independently on each group. When the parameter vector is decomposed, the regularization can set a group of features to zero without affecting the other overlapping groups. This implies that the support for each task is the union of the groups now, not the complement of the union, as features shared by more than one group will have their values preserved for the active group and be zeroed in the inactive group.

Again, to enhance our intuition of these regularization terms, let us compare the norm-balls of both norms, the Group LASSO and the Overlapping Group LASSO. Figure 16 shows both norm-balls in $\mathbb{R}^3$ (Figures from (JACOB *et al.*, 2009a)). For both cases we have the groups of features $\mathcal{G} = \{\{1, 2\}, \{2, 3\}\}$. For the Group LASSO (left), the singularities that appear correspond to cases when only w_1 or w_3 is non-zero. The Overlapping Group LASSO on the other hand (right), has two circular sets of singularities when (w_1, w_2) or (w_2, w_3) is non-zero.

To solve the latent version of the Group LASSO regularization we can rearrange the features that belong to the overlapping groups by replicating them in the design matrix. Then we can use any solver for the original Group LASSO regularization. But depending on the number of groups this solution is far from optimal in terms of memory



(a) Group LASSO. (b) Overlapping Group LASSO.

Figure 16 – Norm-ball of the Group LASSO (left) and Overlapping Group LASSO (right) in $\mathbb{R}^3$, with $\mathcal{G} = \{\{1, 2\}, \{2, 3\}\}$. Figure from (JACOB *et al.*, 2009a).

usage, as we are replicating data to match our representation.

3.3 VARIATIONS OF THE LASSO

The properties of the LASSO were used to construct many variations of this algorithm, considering more specific knowledge from the application.

The Relaxed LASSO (MEINSHAUSEN, 2007), for instance, first trains a LASSO model and then uses only the active features in a least-squared model. The LASSO here acts as a feature selection step. Fused-LASSO (TIBSHIRANI *et al.*, 2005) is a generalization designed for problems with features that can be ordered in some meaningful way. This variation penalizes the l_1-norm of each feature and their successive differences, promoting sparsity of the features and their successive differences jointly. The Fused-LASSO is especially useful when the number of features is much greater than the sample size.

The Adaptive LASSO (ZOU, 2006) is a weighted version of the LASSO, where each feature is weighted according to some rule.

Definition 9 (Adaptive LASSO)**.** Let $\hat{\mathbf{w}} \in \mathbb{R}^n$ be a weight vector with a weight for each feature of our linear model. Under certain assumptions on $\hat{\mathbf{w}}$, the Adaptive LASSO solves the following optimization problem(ZOU, 2006):

$$\text{minimize} \quad \|X\mathbf{w} - \mathbf{y}\| + \lambda \|\hat{\mathbf{w}} \otimes \mathbf{w}\|_1, \tag{3.7}$$

where $\lambda > 0$ is a hyper-parameter.

The usual configuration for $\hat{\mathbf{w}}$ is to solve an Ordinary Least Squares problem with the available data and set $\hat{\mathbf{w}} = \frac{1}{\mathbf{w}_{OLS}^{\gamma}}$ where $\gamma = 1$. This variation of the LASSO preserves

important oracle properties, related to the consistency of the selection of active features.

The LASSO also has some Bayesian variations, where linear regression parameters can be interpreted as a Bayesian posterior mode estimate when the regression parameters have independent Laplace (i.e., double-exponential) priors. Park e Casella (2008) propose a Gibbs sampling from this posterior using an expanded hierarchy with conjugate normal priors for the regression parameters and independent exponential priors on their variances. Casella *et al.* (2010) extended this model and proposed a fairly general fully Bayesian formulation which could accommodate various LASSO variations, including the Group LASSO, the Fused LASSO and the Elastic Net. A Bayesian Group LASSO model with spike and slab priors for problems that only require feature selection at the group level is found in (XU; GHOSH, 2015).

All these variations gain from the main advantages of the LASSO:

- less is more: as fewer parameters are active, the resulting model is more interpretable.
- statistical efficiency: if the signal is sparse the bet justifies; if the signal is not sparse and the results are poor, other methods will not improve much more.
- computational efficiency: sparsity may lead to a dramatic reduction of computation in the optimization process.

A common problem for all these sparsity-inducing regularization terms is that gradients may not exist at the introduced non-smoothness, which prevents us from using vanilla gradient methods. In this case, we need efficient optimization procedures that can solve these challenges, which is the main topic of the next chapter.

3.4 FINAL REMARKS

In this chapter, we presented how sparsity plays an important role in ML methods. We formally introduced the l_p-norm family, focusing on the l_1-norm and the LASSO model. The geometric interpretation of the LASSO showed how the singularities imposed by the l_1-norm promote sparsity. We also saw how groups of correlated features can be handled with a specific instance of the mixed l_{pq}-norm family presented by the Group LASSO regularization. Although being able to handle groups of features, if features belong to more than one group (i.e. groups overlap), the recovered support is the complement of the union of the active groups, which can be counter-intuitive. The Overlapping Group LASSO is an extension of the Group LASSO that is able to handle overlapping groups of correlated features while having its support as the union of the active groups, instead of the complement of the union of active groups. Again, by analyzing the geometry of the norm-balls we could see how these norms promote different sparsity patterns.

By exploring sparsity, we can achieve simpler solutions and handle high-dimensionality models, for example. A common drawback of sparsity inducing regularization is the introduction of non-smoothness in the cost function. This prevents us from using vanilla gradient-based optimization methods that are widely adopted in the ML community. As highlighted by Nesterov (2005), although sub-gradient methods are popular for such problems, they do not have an optimal convergence rate. The next chapter is devoted to optimization procedures that can handle non-smooth terms with optimal convergence rates, or in an easily parallelizable fashion.

4 SOLVING OPTIMIZATION PROBLEMS WITH NON-SMOOTH TERMS

The previous chapters presented how to design regularization terms that induce sparsity into the variables of the optimization problem and can be used to encode a priori knowledge on learning formulations. The variables of the optimization problem are generally associated with free parameters of the learning models and also parameters of the structural relationship established by the learning tasks. Considering all the multiple ways that tasks can be related to each other we end up with a combinatorial search that can easily become intractable in high-dimensional settings, as we would have to take into account all possible subsets of features, as well as all subsets of tasks (that share information) and transference in more than one direction. By leveraging sparsity inducing norms, the tasks relationships is automatically inferred by the learning process.. Notice that the mismatch of assumptions made by the MTL model and the effective structural relationship among tasks may lead to negative transference. Sparsity inducing terms also introduce non-smoothness into the related cost function, which requires specific optimization methods.

This chapter is dedicated to solving convex optimization problems with non-smooth terms. Proximal operators are presented as a tool to handle non-smooth terms that commonly appear in ML and statistical applications. Three algorithms are presented, namely: i) Iterative Shrinkage-Thresholding Algorithm (ISTA) in Section 4.2, ii) Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) in Section 4.3, and iii) the Alternating Direction Method of Multipliers (ADMM) in Section 4.4.

4.1 PROXIMAL METHODS

Gradient-based methods are commonly used to solve unconstrained convex optimization problems, specially when no closed-form analytical solution is available and memory requirements are not prohibitive. These methods propose an iterative procedure where the variables are updated until convergence is achieved. In this case, the objective function needs to be smooth and convex to achieve fast convergence. As presented in the previous chapters, regularization terms that impose non-smoothness in the objective function or in the constraint of the optimization problem prevent us from using vanilla gradient descent.

Let us generalize this procedure by considering proximal operators and proximal-based optimization methods, taking the LASSO as an example that uses the l_1-norm in

the regularization term. The singularity introduced by this term can be easily handled by the proximal operator function associated with the l_1-norm regularization, as it assumes a closed-form analytical solution of cheap computation. The same is valid for many other non-smooth regularization terms commonly used in ML, when the proximal operator is available (POLSON *et al.*, 2015). These methods are specially useful when handling decomposable convex functions, which is the case of most regularized ML models.

Parikh e Boyd (2014) compares this class of methods with Newton's method as follows:

> "[C]onsidering the importance that Netwon's method has in solving smooth, non-constrained minimization problems, the proximal methods are for non-smooth, constrained, large-scale, or distributed optimization problems. "

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function. Thus, f is a function whose epigraph

$$\mathbf{epi}f = \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} | f(\mathbf{x}) \leq t\}$$

is a nonempty closed convex set, where $t \in \mathbb{R}$. The numerically tractable domain of f is the set where f takes on finite values, $\mathbf{dom}f = \{x \in \mathbb{R}^n | f(x) < +\infty\}$.

Definition 10 (Proximal Operator of f)**.** The proximal operator $\mathbf{prox}_f : \mathbb{R}^n \to \mathbb{R}^n$ of f is defined by

$$\mathbf{prox}_f(\mathbf{v}) := \underset{\mathbf{x}}{\text{argmin}} \quad \left(f(\mathbf{x}) + \left(\frac{1}{2} \|\mathbf{x} - \mathbf{v}\|^2 \right) \right), \tag{4.1}$$

where $\|.\|$ is the usual Euclidean norm (PARIKH; BOYD, 2014, p. 124).

It is common to parameterize the proximal operator by $\lambda > 0$, resulting in:

$$\mathbf{prox}_{\lambda f}(\mathbf{v}) := \underset{\mathbf{x}}{\text{argmin}} \quad \left(f(\mathbf{x}) + \left(\frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|^2 \right) \right). \tag{4.2}$$

Since the proximal operator is strongly convex, for any $\mathbf{v} \in \mathbb{R}^n$ there is a unique global optima. This definition can also be seen as a suitably defined envelope function of f. For instance, the Moreau envelope $f^\lambda(\mathbf{v})$ is expressed as:

$$f^\lambda(\mathbf{v}) = \inf_{\mathbf{x}} \quad \left(f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2 \leq f(\mathbf{v}) \right). \tag{4.3}$$

It approximates f from below, having the same minimizing values. The proximal operator computes the value that solves the minimization problem defined by the Moreau envelope (POLSON *et al.*, 2015), providing a trade-off between minimizing f and staying near to the point $\mathbf{v}$ that is controlled by λ.

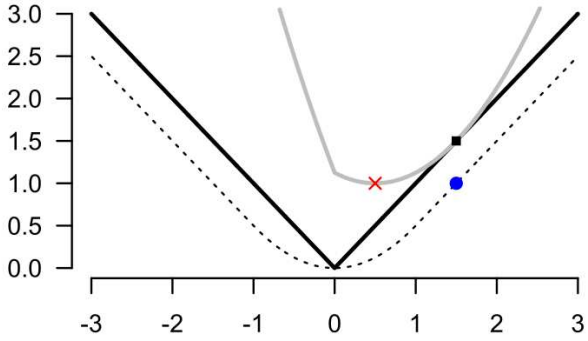


Figure 17 – The black line shows $f(x) = |x|$, while the dotted line shows the corresponding Moreau envelope with $\lambda = 1$. Let $v = 1.5$, the function $|x|+\frac{1}{2}(xv)2$ is shown as a gray line. Its minimum, located at $(0.5, 1)$ and depicted as a red cross, defines the Moreau envelope and proximal operator, being closer to the minimum $(0, 0)$ than $(v = 1.5, f^{\lambda=1}(v))$ depicted as a blue dot. Figure extracted from (POLSON *et al.*, 2015).

The proximal operator is especially useful when it can be evaluated in closed form or at a modest computational cost. A didactic example given in (POLSON *et al.*, 2015) is the proximal operator for $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$. As f is separable, for a given component x the proximal operator solves the problem

$$\min_{x\in\mathbf{R}} \quad \left[\lambda|x| + \frac{\lambda}{2}(x-v)^2\right]. \tag{4.4}$$

The solution is given by

$$x^* = \mathbf{prox}_{\lambda f}(x) = \text{sign}(x)(|x| - \lambda)^+ = S_\lambda(x). \tag{4.5}$$

Notice that this solution is the soft-thresholding operator with parameter λ.

Figure 17 depicts a geometric interpretation of the connection between the proximal operator and the Moreau envelope. The black line shows $f(x) = |x|$, while the dotted line shows the corresponding Moreau envelope with $\lambda = 1$. Let $v = 1.5$, the function $|x|+\frac{1}{2}(xv)2$ is shown as a gray line. Its minimum, located at $(0.5, 1)$ and depicted as a red cross, defines the Moreau envelope and proximal operator. Notice that this point is closer than $(v = 1.5, f^{\lambda=1}(v))$ (blue dot) to the minimum $(0, 0)$, emphasizing the point-wise construction of the Moreau envelope in terms of a simple optimization problem.

Three interpretations stated in (PARIKH; BOYD, 2014) are important for this work. First is the notion that the proximal operator behaves similarly to a gradient step of the function f. For instance, computing the optimization problem related to the derivative of the Moreau envelope $\partial f^\lambda(\mathbf{v})$ results that $\mathbf{prox}_{\lambda f}(\mathbf{v}) = \mathbf{v} - \lambda\partial f^\lambda(\mathbf{v})$, which means that computing the proximal operator is equivalent to computing a gradient-descent step for the Moreau envelop, with λ as a step-size. Secondly, the proximal operator can be seen as a generalized projection. For instance, when f is the set indicator function of a convex set

C, the proximal operator is the Euclidean projection of $\mathbf{x}$ onto C. Lastly, when applied to $\mathbf{x}^*$ - the optimal solution of f - the proximal operator returns $\mathbf{x}^*$, which means that the optimal solution for minimizing f is a fixed-point of the proximal operator and reveals a close connection between proximal operators and fixed-point theory. Please refer to Parikh e Boyd (2014) for an extensive presentation of proximal methods, and Polson *et al.* (2015) for a didactic exposition of proximal methods in ML and statistical applications.

A **proximal algorithm** leverages the proximal operators of the objective terms to solve a convex optimization problem (PARIKH; BOYD, 2014, p. 126). Let $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{\infty\}$ be a closed proper convex function, k be the iteration counter, and $\mathbf{x}^k$ be the k-th iteration of the algorithm. As defined in (PARIKH; BOYD, 2014, p. 142), the proximal minimization algorithm is presented in Algorithm 1. The main advantage of such algorithm is its simplicity: it successively applies the proximal operator of f, until convergence.

4.1.1 PROXIMAL GRADIENT

Let us now consider the problem

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) + g(\mathbf{x}),$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth convex function that is continuously differentiable with Lipschitz continuous gradient $L(f)$:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L(f)\|\mathbf{x} - \mathbf{y}\| \text{ for every } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,$$

L denoting the Lipschitz constant of ∇f, and $g : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function. The differentiable terms in f and the non-differentiable terms in g are isolated. The extended-value definition of the domain of g allows us to encode any convex constraints on the variable $\mathbf{x}$ through this function.

Let λ^k be a step size, the *proximal gradient method* consists of the updates on the variable $\mathbf{x}$ according to Algorithm 2. Compared to Algorithm 1, instead of successively applying the proximal operator to f, the proximal operator is applied to the gradient descent step of f.

Algorithm 1 Proximal Minimization Algorithm

1: Initialize $\mathbf{x}^0; k = 1$
2: **while** convergence not reached **do**
3: $\quad \mathbf{x}^{k+1} := \mathbf{prox}_{\lambda f}(\mathbf{x}^k)$
4: $\quad k := k + 1$
5: **end while**

Algorithm 2 Proximal Gradient Algorithm

1: Initialize $\mathbf{x}^0; k = 1$
2: **while** convergence not reached **do**
3: $\quad \mathbf{x}^{k+1} = \mathbf{prox}_{\lambda^k g}(\mathbf{x}^k - \lambda^k \nabla f(\mathbf{x}^k))$
4: $\quad k := k + 1$
5: **end while**

4.2 ITERATIVE SHRINKAGE-THRESHOLDING ALGORITHMS

The procedure called *iterative shrinkage-thresholding algorithm*, or ISTA (BECK; TEBOULLE, 2009) is based on a convex approximation model of the problem in Eq. (4.6). For any $L(f) > 0$, consider the quadratic approximation of $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ at $\mathbf{y}$:

$$Q_L(\mathbf{x}, \mathbf{y}) := f(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla f(\mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x}). \tag{4.6}$$

Let the unique minimizer of Q_L be $\rho_L(\mathbf{y}) := \text{argmin}\{Q_L(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbb{R}^n\}$. By ignoring the constant terms with relation to $\mathbf{y}$, $\rho_L(\mathbf{y})$ can be rewritten as:

$$\rho_L(\mathbf{y}) := \text{argmin}_{\mathbf{x}} \left\{ g(\mathbf{y}) + \frac{L}{2} \left\| \mathbf{x} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|^2 \right\}. \tag{4.7}$$

Applying the proximal minimization solution to solve this problem leads to the Iterative Shrinkage-Thresholding Algorithm (ISTA), stated in Algorithm 3 (BECK; TEBOULLE, 2009, p. 191).

Algorithm 3 ISTA with Constant Step-Size

Initialize $\mathbf{x}^0; L := L(\nabla f), k = 1$.
while convergence not reached **do**
$\quad \mathbf{x}^{k+1} = \rho_L(\mathbf{x}^k)$.
$\quad k := k + 1$
end while

When L is unknown, Beck e Teboulle (2009) proposes a backtracking procedure to find a good value for L, leading to Algorithm 4.

The main advantage of the ISTA algorithm is its simplicity. It is simple to compute, simple to implement, and can encompass many usefull problems in the machine learning domain. The main drawback is that this algorithm demonstrates a sublinear global rate of convergence. With the backtracking operation, this procedure results in a worst-case complexity result of $\mathcal{O}(\frac{1}{k})$ (BECK; TEBOULLE, 2009). The next section introduces an accelerated version of this algorithm.

4.3 ACCELERATING ISTA

An accelerated version of the ISTA algorithm was first introduced in (NESTEROV, 1983), and is presented in (BECK; TEBOULLE, 2009) as FISTA. It consists of the application of the proximal operator at each iteration, as in ISTA. Instead of computing it on the sole $\mathbf{x}_{k-1}$ point, it uses a specific combination of the previous two points of the optimization procedure. For the details of how this combination of points is reached, please refer to (BECK; TEBOULLE, 2009). FISTA is presented in Algorithm 5.

The additional computation is the point y, that has a marginal cost with respect to the cost of computing the proximal operator. The very specific linear combination of the previous two points x_{k-1} and x_{k-2}, is explained in (BECK; TEBOULLE, 2009). This algorithm keeps the advantages of ISTA and converges at the optimal rate of $\frac{1}{k^2}$, as proven by (NESTEROV, 1983).

As in ISTA, the value of L is a parameter and if its value is unknown, the same backtracking procedure used for ISTA can be employed here. Algorithm 6 shows FISTA with Backtracking.

4.4 ALTERNATING DIRECTION OF MULTIPLIERS METHOD

Suppose that the variable of our optimization problem can be decomposed into $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{z} \in \mathbb{R}^n$. The *Alternating Direction of Multipliers Method* (ADMM) (BOYD

Algorithm 4 ISTA with Backtracking

Initialize $\mathbf{x}^0; L_0 > 0; \eta > 1;$ and $k = 1$.
while convergence not reached **do**
    Find the smallest positive i_k, such that with $\bar{L} = \eta^{i_k} L_k$, the following condition is still met:
    $F(\rho_{\bar{L}}(\mathbf{x}_{k-1})) \leq Q_{\bar{L}}(\rho_{\bar{L}}(\mathbf{x}_{k-1}), \rho_{\bar{L}}(\mathbf{x}_{k-1}))$.
    Set $L_k = \eta^{i_k} L_{k-1}$, and compute
    $\mathbf{x}^{k+1} = \rho_L(\mathbf{x}^k)$.
    $k := k + 1$
end while

Algorithm 5 FISTA with Constant Stepsize

Initialize $\mathbf{x}^0; L := L(\nabla f), k = 1$.
Let $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{R}^n$ and $t_1 = 1$.
while convergence not reached, compute **do**
    $\mathbf{x}_k = \rho_L(\mathbf{y}_k)$.
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
    $\mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1})$.
    $k := k + 1$
end while

Algorithm 6 FISTA with Backtracking

Initialize $\mathbf{x}^0; L_0 > 0; \eta > 1;$ and $k = 1$.
Let $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbf{R}^n$ and $t_1 = 1$.
while convergence not reached **do**
    $(k \geq 1)$Find the smallest positive i_k, such that with $\bar{L} = \eta^{i_k} L_k$, the following condition is still met:
    $F(\rho_{\bar{L}}(\mathbf{x}_{k-1})) \leq Q_{\bar{L}}(\rho_{\bar{L}}(\mathbf{x}_{k-1}, \mathbf{x}_{k-1}))$.
    Set $L_k = \eta^{i_k} L_{k-1}$, and compute
    $\mathbf{x}_k = \rho_L(\mathbf{y}_k)$.
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
    $\mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1})$.
    $k := k + 1$
end while

et al., 2011) solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & A\mathbf{x} + B\mathbf{z} = \mathbf{c}, \end{aligned} \tag{4.8}$$

where f and g are convex functions, $A \in \mathbb{R}^{p \times m}$, $B \in \mathbb{R}^{p \times n}$, and $\mathbf{c} \in \mathbb{R}^p$, accounting for p equality constraints. Notice that splitting the variable $\mathbf{x}$ leads to a separable objective function.

The ADMM method solves this problem by mixing two strategies: the dual ascent, and the method of multipliers. In a few words, the dual ascent is an optimization method that maximizes the dual function related with the original problem using the convex conjugate of f. The method then consists of two operations, one updating the value of the primal variables, and a second that is based on the gradient of the dual formulation together with a proper step-size that performs a 'cost adjustment' step. This procedure can be used even if g is not differentiable. But the main advantage is that this procedure can optimize separable functions in a highly parallelizable algorithm.

To solve this problem, ADMM starts by forming the augmented Lagrangian for Eq. (4.8) as follows:

$$Lag_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) + \left(\frac{\rho}{2}\right)\|A\mathbf{x} + B\mathbf{z} - c\|_2^2. \tag{4.9}$$

Then it proceeds minimizing this function with respect to the primal variables $\mathbf{x}$ and $\mathbf{z}$, and then updating the dual variable $\mathbf{y}$.

Definition 11 (Alternating Direction of Multipliers Method)**.** The ADMM algorithm

consists of the following updates (BOYD *et al.*, 2011):

$$\mathbf{x}^{k+1} := \underset{\mathbf{x}}{\operatorname{argmin}}\ Lag_{\rho}(\mathbf{x}, \mathbf{z}^k, \mathbf{y}^k) \tag{4.10}$$

$$\mathbf{z}^{k+1} := \underset{\mathbf{z}}{\operatorname{argmin}}\ Lag_{\rho}(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{y}^k) \tag{4.11}$$

$$\mathbf{y}^{k+1} := \mathbf{y}^k + \rho(A\mathbf{x}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c}), \tag{4.12}$$

where $\rho \geq 0$ (BOYD *et al.*, 2011, p. 14).

ADMM iterates as in the dual ascent: first it optimizes the primal variable (steps 1 and 2 in Equations (4.10) and (4.11)), and then it optimizes the dual variable (step 3 in Eq. (4.12)) performing an adjustment on the primal variables. Notice that the augmented Lagrangian parameter ρ is used as the step-size of the dual variable update. The addition of the penalty term in the augmented Lagrangian formulation enhances the convexity properties of the problem, helping the method of multipliers to converge faster and under less restrictive assumptions, such as f being not strictly convex, or having an extended domain (when f assumes infinite values when $\mathbf{x}$ is out of f domain). It is also important to notice that "the roles of $\mathbf{x}$ and $\mathbf{z}$ are almost symmetric, but not quite, since the dual update is done after the $\mathbf{z}$-update but before the $\mathbf{x}$-update" (BOYD *et al.*, 2011).

The most common representation of ADMM method is slightly different from what was shown. It is more convenient to write the algorithm by combining the linear and quadratic terms and scaling the dual variable. Let the residual be $\mathbf{r} = A\mathbf{x} + B\mathbf{z} - c$, so that:

$$\begin{aligned}\mathbf{y}^T\mathbf{r} + \left(\frac{\rho}{2}\right)\|\mathbf{r}\|_2^2 &= \left(\frac{\rho}{2}\right)\|\mathbf{r} + \left(\frac{1}{\rho}\right)\mathbf{y}\|_2^2 - \left(\frac{1}{2\rho}\right)\|\mathbf{y}\|_2^2 \\ &= \left(\frac{\rho}{2}\right)\|\mathbf{r} + \mathbf{u}\|_2^2 - \left(\frac{\rho}{2}\right)\|\mathbf{u}\|_2^2,\end{aligned}$$

where $\mathbf{u} = \left(\frac{1}{\rho}\right)\mathbf{y}$ is the *scaled dual variable*.

Definition 12 (Scaled Form of ADMM)**.** Using the scaled form of the dual variable, ADMM can be expressed with the following updates (BOYD *et al.*, 2011):

$$\mathbf{x}^{k+1} := \underset{\mathbf{x}}{\operatorname{argmin}}\ f(\mathbf{x}) + \left(\frac{\rho}{2}\right)\|A\mathbf{x} + B\mathbf{z}^k - \mathbf{c} + \mathbf{u}^k\|_2^2 \tag{4.13}$$

$$\mathbf{z}^{k+1} := \underset{\mathbf{z}}{\operatorname{argmin}}\ g(\mathbf{z}) + \left(\frac{\rho}{2}\right)\|A\mathbf{x}^{k+1} + B\mathbf{z} - \mathbf{c} + \mathbf{u}^k\|_2^2 \tag{4.14}$$

$$\mathbf{u}^{k+1} := \mathbf{u} + A\mathbf{x}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c}. \tag{4.15}$$

Assuming that i) f and g are closed proper convex functions with real-extended domain, and that ii) the unaugmented Lagrangian has a saddle point, ADMM convergence satisfy the following properties:

- *residual convergence*: $\mathbf{r}^k$ approaches 0, as k approaches infinity, which is known as feasibility;
- *objective convergence*: $f(\mathbf{x}^k)+g(\mathbf{z}^k)$ approaches the optimal value p^*, as k approaches infinity; and
- *dual variable convergence*: $\mathbf{y}^k$ approaches $\mathbf{y}^*$ as k approaches infinity. $\mathbf{y}^*$ is the dual optimal point.

See the Appendix A of (BOYD *et al.*, 2011) for detailed proofs.

In practice, ADMM is not known to be a method of fast convergence to high accuracy. Instead, it converges to modest accuracy within a few tens of iterations, which is more than reasonable for most machine learning applications. The optimality conditions of ADMM help us to decide on the stopping criteria when implementing ADMM. Basically, some conditions on the magnitude of the primal and dual residuals can be imposed. The optimization procedure is stopped if they reach a small enough value. Another interesting characteristic of ADMM is its flexibility to handle parallelism. The updates of the primal variables can occur in parallel, needing to synchronize only for the dual update, that is a cheap operation. The structure on f, g, A, and B is also of great advantage if the application suits this framework. For more in-depth exposition of ADMM and how many applications can leverage its properties, please refer to (BOYD *et al.*, 2011).

4.5 FINAL REMARKS

Regularization terms that induce sparsity are commonly used in ML to encode a priori knowledge into the variables of the optimization problem. This is especially true when the goal is to incorporate knowledge about how tasks are assumed to be related in MTL settings. Some of the norm regularization terms presented in the previous chapter introduce non-smooth components in the objective function. To solve optimization problems containing non-smooth terms requires approaches other than vanilla first-order gradient descent or Newton's Method. This chapter presented the necessary tools to solve non-smooth optimization problems, common in the MTL setting. Proximal methods allow us to efficiently handle the non-smoothness and to use multiple regularization terms.

5 GROUP ASYMMETRIC MULTI-TASK LEARNING

The previous chapters of this thesis discussed the problem of learning in the presence of multiple tasks. Over the many possible ways to tackle this problem, our focus is in the Multi-Task Learning approach. The main assumption of MTL is that by jointly learning the tasks, each task can benefit from information of other related tasks to improve their own performance. This strategy brings some interesting benefits such as preventing overfitting and allowing interpretability of the learning structure of the involved tasks.

The possible choices of how tasks may be related guide to a combinatorial challenge, considering all the subsets of features and tasks as possible paths of information transference. Most of the presented literature still enforces limiting assumptions in this sense. Restricting ourselves to models with interpretable parameters, two characteristics of the tasks's transference that are most limiting in the current literature are:

- the *global transference* assumption, where the relationship between tasks includes the entire set of features; and
- *symmetric relationships*, i.e., the influence of task A on task B is the same as the influence of task B on task A.

These assumptions can become a limiting factor as they do not encompass relationships that may be present in the data. Another motivation for this work is that it is easy to assume that negative transference occurs when unrelated tasks are forced to be related by the method's assumptions, but this is not entirely true. Negative transference may occur even when all tasks are related.

It may be natural to some applications to consider that the relationship between tasks can involve only subsets of features, while including two or more tasks; and also that this relationship may not occur both-ways. Based on the intrinsic characteristics of the problem, choosing a model that enforces the wrong assumptions on the data may deteriorate the performance instead of promoting gains. This aspect evinces the difficulty of choosing a model.

In this chapter we present Group Asymmetric Multi-Task Learning (GAMTL), an MTL that overcomes these assumptions. By considering that tasks are related asymmetrically based on the losses of the tasks, and that the relationship between any two tasks is based on groups of features, instead of on the entire set of features, GAMTL offers a flexible alternative to model more sophisticated groups of tasks. GAMTL estimates how

tasks are related in a fine-grained way that is also easy to interpret, as it is based on linear relationships among tasks parameters.

This Chapter is organized as follows: In Section 5.1 we present the model and its formulation. Section 5.2 explores some variants of GAMTL. Section 5.3 decomposes the non-convex problem of GAMTL into smaller convex problems of easier solution, and proposes an optimization procedure that solves the proposed optimization problem and the presented variants. Section 5.4 presents a complexity analysis of the computational solution that reassures the competitive performance of the solution.

5.1 GROUP ASYMMETRIC MULTI TASK LEARNING (GAMTL) FORMULATION

Group Asymmetric Multi-Task Learning (GAMTL) - presented in (OLIVEIRA *et al.*, 2019) with expanded analysis in (OLIVEIRA *et al.*, Article no. 99, pp. 1-30, 2022.) - is an MTL method that accounts for grouped features on the tasks design matrix of linear models while estimating how tasks share information. The learning models are taken as linear models, and each vector of parameters is a column of the design matrix. So, the number of columns equals the number of tasks. The estimated relationship structure considers each group of features independently, enabling a bidirectional transference between any two tasks. In this flexible formulation, tasks can transfer differently depending on how a group of features is benefitial to their predictions: if a group of features is relevant for some tasks, transference occurs. On the other hand, transference refrains when the same group of features is not relevant for a different set of tasks.

Considering a set of linear models as tasks in an MTL problem, let the tasks features be partitioned into $\mathcal{G} = \{1, \cdots, G\}$ groups of correlated features in X_t for all tasks $t \in \mathcal{T}$. X_t^g is the design matrix restricted to the features present in group g for task t, and $\mathbf{w}_t^g$ is the parameter vector of task t whose values are zeroed outside of g. Let us assume that the parameters of a task can be decomposed into a sparse linear combination of the parameters from the other tasks when considering each group of features independently. In this case $\mathbf{w}_t^g \approx \sum_{s \in T \setminus t} b_{st}^g \mathbf{w}_s^g$, where b_{st}^g is a scalar that encodes the influence of task s on task t restricted to the group of features g. The $b_{st}^g \; \forall \; s, t \in \mathcal{T}$ variables compose G matrices $B^g \in \mathbb{R}^{T \times T}$, where a row $\mathbf{b}_t^g$ encodes the influence of task t on all other tasks, and a column $\mathbf{b}_t^g$ encodes the influence of the other tasks on task t when considering the group g. Each one can be seen as the adjacency matrix of a directional graph transference structure. Nodes are tasks and directional weighted edges indicate transference from one task to another. Based on the latent representation of each task parameter vector, $\mathbf{w}_t \approx \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g$, where W^g is the task parameter matrix with values restricted to the group

g, and zeros elsewhere. Eq. (5.1) shows the resulting MTL optimization problem.

$$\begin{aligned} \min_{W, B^g \forall g \in \mathcal{G}} \quad & \sum_{t \in \mathcal{T}} \frac{1}{m_t} \left(1 + \lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_t^g\|_1 \right) \mathcal{L}(\mathbf{w}_t) + \frac{\lambda_2}{2} \left\| \mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g \right\|_2^2 + \lambda_3 \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2 \\ \text{s.t.} \quad & \mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{w}_t^g \\ & \mathbf{b}_t^g \geq 0, \forall g \in \mathcal{G} \text{ and } t \in \mathcal{T} \end{aligned} \tag{5.1}$$

The first term computes the loss function of each task weighted by the number of samples. Therefore, it takes into account sample imbalance among tasks, while also using the loss to weight transference from task t to the other tasks. The l_1-norm applied to $\mathbf{b}_t^g$ is used to enforce sparsity on the estimated relationship among the tasks. This helps us pruning the search space while keeping only the more relevant transferences per group of features. As in (LEE *et al.*, 2016), GAMTL considers the noise of each task as a weight when transferring to the other tasks in order to enforce that a task with a higher cost should be less inclined to influence tasks with smaller costs, while tasks with smaller costs are encouraged to transfer more. This increases the asymmetry on the transference between tasks.

The second term penalizes the difference between the parameters of a specific task t and the linear combination of parameters from the tasks with which task t is grouped. Notice that this term considers how the task t is related to possibly different tasks for each group of features independently. Together with the equality constraint on each $\mathbf{w}_t$, the last term corresponds to the Overlapping Group LASSO regularization. The constraint on B^g variables restricts the way tasks relate by allowing only non-negative values in the linear combination. However, in case this restriction is not suitable for the application, an optimization procedure for the more relaxed variant (without the restriction on B^g values) is presented. GAMTL uses the transference matrices B^g in a way that allows us to employ the Group LASSO while estimating how tasks share information, instead of forcing transference involving all tasks on each group of features.

GAMTL contains three hyper-parameters that impact how transference occurs. When $\lambda_1 = 0$, $\lambda_2 = 0$, and $\lambda_3 = 0$, independent linear models are recovered, leading to a Single Task Learning (STL) approach. If only $\lambda_3 > 0$, we still have independent linear models per task but regularized by Overlapping Group LASSO. $\lambda_2 > 0$ controls the transference flexibility from many groups of related tasks - one per group of features - to $\mathbf{w}_t$. With $\lambda_1 > 0$ the sparsity of the transference is activated.

Eq. (5.1) allows some variants: with and without the second constraint; and considering or not the loss to penalize transference between tasks. This choice is based on the particularities of the application and whether a negative relationship between tasks makes sense or not.

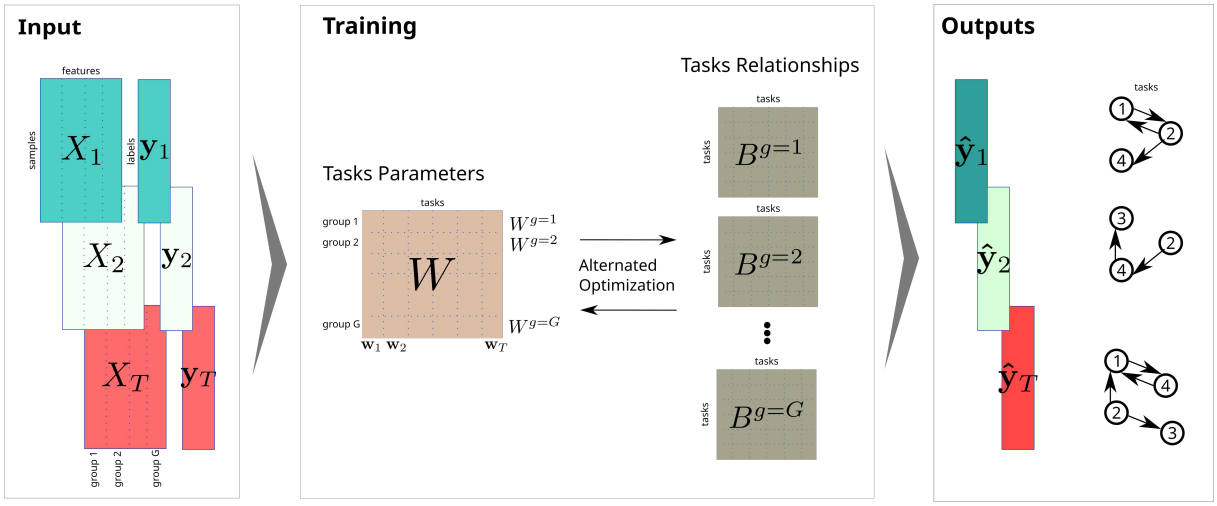


Figure 18 – Input data representation is depicted on the left: a design matrix and labels for each task along with a possibly overlapping partition of the input feature set into the same groups for all tasks. The training procedure is depicted in the middle, where an alternating optimization takes place. One step involves the optimization of tasks parameters so that each task is free to find its own features sparsity pattern and the relationship between any pair of tasks is enforced locally to each group. The second step estimates how tasks are related considering each group of features. The resulting relationship matrices are shown as the adjacency matrix of a multi-digraph, where each level corresponds to a group of features, recursively used at the first step as the structural relationship among tasks, thus implementing the asymmetric local transference. The output is shown on the right, consisting of the predicted labels for each task, and an asymmetrical relationship among tasks estimated per group of features.

Figure 18 shows a flowchart presenting the training process for GAMTL. The input consists of a labeled training set for each task, with the tasks features structured into groups. The grouped partition of features must be the same for all tasks design matrix. However, the partition is arbitrary allowing non-contiguous groups of features to overlap, despite Figure 18 induces contiguity of features. An alternating optimization procedure performs the training process, switching between the estimation of tasks parameters and the relationship among tasks. The relationship among tasks is encoded into G matrices, a transference structure that enables local transference and is equivalent to a multi-digraph. In this multi-digraph, each level of the graph corresponds to a group of features where tasks can be related. Tasks are related independently for each group of features in an asymmetrical fashion. Finally, the output for each task is shown on the right: predictions for each task, and structural information about how tasks are related at the level of groups of features.

By representing the relationship among tasks via multiple matrices, and considering the parameters of the tasks as a latent space for relationship, GAMTL promotes unique flexibility for the transference:

- Tasks may be related only on subsets of features.
- Groups of features can play distinct roles on different groups of related tasks.
- Transference is asymmetric: the influence of task t on task s may differ from the influence of task s on task t, at the group level.

Using the categorization in the survey of (ZHANG; YANG, 2017), GAMTL belongs to the parameter-based transference category of MTL models. Another important aspect of our formulation is that GAMTL is designed for linear models. The structure that encodes the relationship of the tasks is based on linear combinations that can be easily interpreted. The assumption that the parameters of one task can be decomposed as a linear combination of the parameters of other tasks, on each group of features, may be too restrictive for multi-layer nonlinear models, such as neural networks.

5.2 VARIANTS OF GAMTL

Let us call GAMTL the formulation that uses the loss to refrain tasks with higher costs to transfer to other tasks, and that restricts the values of all B^g to be equal to or greater than zero, as shown in Eq. (5.1). Consider this as the *standard formulation*, but all experiments report all variants of GAMTL. These variants arise by putting aside one or both of the following aspects that are present in the standard formulation of (5.1): (i) using the loss to regularize how much a task can transfer to other tasks; and (ii) using the non-negativity restriction on the elements of B^g.

5.2.1 GAMTL-NL: NO LOSS

The standard formulation of GAMTL considers the loss of each task as a weight that is multiplied by the regularization parameter of the tasks transference. In this case, the transference of a task t to all other tasks is proportionally penalized, based on the value of the loss function of task t. On the contrary, tasks with a low value of the loss function are encouraged to transfer to other tasks.

GAMTL-nl is the variation that disables the loss-weighting behavior on the formulation, which yields the following optimization problem:

$$\begin{aligned} \min_{W,B^g} \quad & \sum_{t\in\mathcal{T}} \frac{1}{m_t}\mathcal{L}(\mathbf{w}_t) + \lambda_1 \sum_{g\in\mathcal{G}} \|\mathbf{b}_t^g\|_1 + \frac{\lambda_2}{2}\left\|\mathbf{w}_t - \sum_{g\in\mathcal{G}} W^g\mathbf{b}_t^g\right\|_2^2 + \lambda_3 \sum_{g\in\mathcal{G}} d_g\|\mathbf{w}_t^g\|_2 \\ \text{s.t.} \quad & \mathbf{w}_t = \sum_{g\in\mathcal{G}} \mathbf{w}_t^g \\ & \mathbf{b}_t^g \geq 0, \forall g \in \mathcal{G} \text{ and } t \in \mathcal{T} \end{aligned} \tag{5.2}$$

The difference lies in the first term of Eq. (5.1) that is now expanded on the first two terms of Eq. (5.2). After expanding, the product between the loss function and the l_1-norm regularization on the $\mathbf{b}_t^g$ variables is removed. Now the regularization on the variables that encode how a task transfers to the other tasks depends only on the value of the hyper-parameter λ_1.

5.2.2 GAMTL-NR: NO RESTRICTION

GAMTL-nr is the variation where $B^g \in \mathbb{R}^{T \times T}$, as shown in Equation (5.3).

$$\begin{aligned} \min_{W,B^g} \quad & \sum_{t \in \mathcal{T}} \frac{1}{m_t} \left(1 + \lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_t^g\|_1 \right) \mathcal{L}(\mathbf{w}_t) + \frac{\lambda_2}{2} \left\| \mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g \right\|_2^2 + \lambda_3 \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2 \\ \text{s.t.} \quad & \mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{w}_t^g. \end{aligned} \tag{5.3}$$

Compared to the standard version presented in Equation 5.1, the difference lies in the removal of the constraints on B^g.

5.2.3 GAMTL-NLNR: NO LOSS, NO RESTRICTION

Based on the choices presented above, another variant of GAMTL is obtained by not using the loss function on the transferences from tasks with higher costs while also not considering the constraints on the values of $B^g \quad \forall g \in \mathcal{G}$. This variation is called GAMTL-nlnr and is associated with the following optimization problem:

$$\begin{aligned} \min_{W,B^g} \quad & \sum_{t \in \mathcal{T}} \frac{1}{m_t} \mathcal{L}(\mathbf{w}_t) + \lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_t^g\|_1 + \frac{\lambda_2}{2} \left\| \mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g \right\|_2^2 + \lambda_3 \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2 \\ \text{s.t.} \quad & \mathbf{w}_t = \sum_{g \in \mathcal{G}} \mathbf{w}_t^g. \end{aligned} \tag{5.4}$$

Notice that this formulation combines the changes introduced in Equations (5.2) and (5.3).

Considering all optimization variables at the same time, Eq. (5.1) ends up being a non-convex optimization problem, possibly with the presence of local minima (GORSKI *et al.*, 2007). The next section derives smaller convex sub-problems that allow the use of an alternating optimization procedure.

5.3 SOLVING THE GAMTL FORMULATION

Considering all parameters at the same time, Eq. (5.1) constitutes a non-convex problem. Let $\mathbf{w}_t \in \mathbb{R}^n$ and $\mathbf{b}_t^g \in \mathbb{R}^T$ for all $g \in \mathcal{G}, t \in \mathcal{T}$ compose the partitions of the objective function variables. However, by considering each partition at a time while fixing

the other variables, we attain smaller convex sub-problems that characterize a multi-convex optimization problem (XU; YIN, 2013; SHEN *et al.*, 2017).

GAMTL uses an alternating strategy in terms of each $\mathbf{w}_t$, while keeping $\mathbf{w}_s \forall s \in \mathcal{T} \setminus t$, and $B^g \ \forall g \in \mathcal{G}$ fixed, and then optimizing with respect to each $\mathbf{b}_t^g, \forall g \in \mathcal{G}, t \in \mathcal{T}$, as shown in Algorithm 7. As commonly used in alternating optimization strategies, the procedure is carried out until there is a sufficiently small change in the values of the variables between successive iterations (BEZDEK; HATHAWAY, 2002; XU; YIN, 2013; GORSKI *et al.*, 2007).

5.3.1 OPTIMIZING TASK PARAMETERS

Isolating Eq. (5.1) in terms of $\mathbf{w}_t, t = 1, 2, \cdots, T$ with all remaining variables fixed results in:

$$\begin{aligned} \min_{\mathbf{w}_t} \quad & \frac{1}{m_t}(1+\lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_t^g\|_1)\mathcal{L}(\mathbf{w}_t) + \frac{\lambda_2}{2}\|\mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g\|_2^2 \\ & + \frac{\lambda_2}{2} \sum_{s \in \mathcal{T} \setminus t} \|\tilde{\mathbf{w}}_s - \sum_{g \in \mathcal{G}} \mathbf{w}_t^g b_{ts}^g\|_2^2 + \lambda_3 \sum_{g \in \mathcal{G}} d_g \|\mathbf{w}_t^g\|_2, \end{aligned} \tag{5.5}$$

where

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

The first term is composed of a convex loss function, as the l_1-norm on $\mathbf{b}_t^g$ is constant. The second term is the projection of $\mathbf{w}_t$ onto the other task parameters, which is also a convex term. The third term computes the interference of $\mathbf{w}_t$ on the projections of the other task parameters, being a sum of convex terms. The last term is the Group LASSO regularization, a convex and non-differentiable term.

Eq. (5.5) can be solved using an accelerated proximal method, such as FISTA (BECK; TEBOULLE, 2009). Let us decompose the objective function into $f : \mathbb{R}^n \to \mathbb{R}$ and

Algorithm 7 GAMTL Alternating Minimization

Require: $(X_t, \mathbf{y}_t) \quad \forall t \in \mathcal{T}, \mathcal{G}$
1: Initialize $W \sim \mathcal{N}(0, \mathbf{I}_\mathcal{T})$ and set $B^g = 0, \forall g \in \mathcal{G}$
2: **while** convergence not reached **do**
3: **for** $t = 1, \cdots, T$ **do**
4: update $\mathbf{w}_t$ optimizing task parameters - Eq. (5.5)
5: **end for**
6: **for** $t = 1, \cdots, T$ **do**
7: **for** $g \in \mathcal{G}$ **do**
8: update $\mathbf{b}_t^g$ optimizing task relationships - Eq. (5.7)
9: **end for**
10: **end for**
11: **end while**
return $W, B^g \quad \forall g \in \mathcal{G}$

$h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, both closed proper convex functions, f being L-Lipschitz continuous — L can be found with a backtracking procedure — while h being non-differentiable:

$$\begin{aligned} f(\mathbf{w}_t) = & \frac{1}{m_t}(1 + \lambda_1 \sum_{g \in \mathcal{G}} \|\mathbf{b}_t^g\|_1)\mathcal{L}(\mathbf{w}_t) \\ + & \frac{\lambda_2}{2}\|\mathbf{w}_t - \sum_{g \in \mathcal{G}} W^g \mathbf{b}_t^g\|_2^2 + \frac{\lambda_2}{2} \sum_{s \in \mathcal{T} \setminus t} \|\tilde{\mathbf{w}}_s - \sum_{g \in \mathcal{G}} \mathbf{w}_t^g b_{ts}^g\|_2^2, \end{aligned} \tag{5.6}$$

and 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 given by

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

5.3.2 OPTIMIZING TASK RELATIONSHIPS

The matrices B^g encode the relationship between tasks. Since a task cannot be represented by itself, we fix $b_{tt}^g = 0$. The strategy used by GAMTL is to isolate Eq. (5.1) in terms of $\mathbf{b}_t^g$, with all remaining variables fixed. 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:

$$\begin{aligned} \min_{\mathbf{b}_t^g} \quad & \frac{1}{2}\|\tilde{\mathbf{w}}_t - \overline{W}^g \mathbf{b}_t^g\|_2^2 + \frac{\lambda_1}{\lambda_2}\|\mathbf{b}_t^g\|_1 \\ \text{subject to} \quad & \mathbf{b}_t^g \geq 0, \forall g \in \mathcal{G} \text{ and } t \in \mathcal{T}. \end{aligned} \tag{5.7}$$

This problem is similar to the Adaptive LASSO (ZOU, 2006) and thus is convex but not differentiable at all points. Without the constraints in Eq. (5.7), it can be solved using any standard method for the LASSO. To handle the constraints ($b_t^g \geq 0 \quad \forall g \in \mathcal{G}, t \in \mathcal{T}$), GAMTL uses the ADMM (BOYD *et al.*, 2011). In this framework, the inequality constraint can be transformed into an indicator function:

$$\begin{aligned} \min \quad & f(\mathbf{x}) + h_1(\mathbf{z}_1) + h_2(\mathbf{z}_2) \\ \text{subject to} \quad & \mathbf{x} = \mathbf{z}_1 \\ & \mathbf{x} = \mathbf{z}_2 \end{aligned} \tag{5.8}$$

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 (5.8) is then, $L_{\rho_1,\rho_2} = L_{\rho_1,\rho_2}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{u}_1, \mathbf{u}_2)$:

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

resulting in the following ADMM update steps:

$$\begin{aligned} \mathbf{z}_i^{k+1} &:= \underset{\mathbf{z}_i}{\operatorname{argmin}} \left(h_i(\mathbf{z}_i) + \frac{\rho_i}{2}\|\mathbf{x}^k - \mathbf{z}_i + \mathbf{u}_i^k\|_2^2\right), i = \{1, 2\} \\ \mathbf{x}^{k+1} &:= \underset{\mathbf{x}}{\operatorname{argmin}} \left(f(\mathbf{x}) + \sum_{j=1}^{2} \frac{\rho_j}{2}\|\mathbf{x} - \mathbf{z}_j^{k+1} + \mathbf{u}_i^k\|_2^2\right) \\ \mathbf{u}_i^{k+1} &:= \mathbf{u}_i^k + \mathbf{x}^{k+1} - \mathbf{z}_i^{k+1} \quad , i = \{1, 2\} \end{aligned}$$

The two steps in $\mathbf{z}_i$-update can run in parallel, with the same occuring for $\mathbf{u}_i$. The $\mathbf{z}_i$-update steps are solved with 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. GAMTL implementation using the Python programming language is available on Github [1].

5.4 COMPUTATIONAL COMPLEXITY

The existence of many transference matrices tends to offer better results and interpretability to the model while including some extra computational effort. The cost of each GAMTL iteration is mostly driven by steps 4 and 8 of Algorithm 7, which involve a FISTA and an ADMM execution, respectively.

Step 4 computes ∇f and $\mathbf{prox}_{\lambda g}$. The cost of the proximal operator is $G[g_{max}]^2 n$, where g_{max} is the size of the largest group. The derivative of Eq. (5.6) needs $T^2 G g_{max}$ flops. Higher costs involved in the gradient computation are $\mathcal{O}(T^2 G n)$, with other negligible costs. The full computation of ∇f is then $\mathcal{O}(T^2 G n)$. Therefore, a FISTA iteration has an overall cost of $\mathcal{O}(T^2 G n)$.

Step 8 prepares $\tilde{\mathbf{w}}_t$ using $GTn + n$ flops. For $\overline{W}^g$, the computation of the loss function of each task costs $\mathcal{O}(n^2 + mn)$, and is reused for all iterations over the same g. ADMM computes a soft-thresholding operator, the projection of $\mathbf{z}$, and the update of $\mathbf{u}$, all with negligible costs. Solving $\mathbf{x}$-update in closed-form with Cholesky decomposition uses T^3 flops, with a back-solve cost of n^2. When considering $n > T$ other theoretical results

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

(BOYD *et al.*, 2011) allow us to perform this computation at a cost of Tn^2. Therefore, the cost of a complete ADMM iteration is at the order of $\mathcal{O}(Tn^2)$.

As one iteration of GAMTL consists of T FISTA and GT ADMM executions, with a fixed number of iterations, GAMTL presents a time complexity of order $\mathcal{O}(T^3Gn + T^2Gn^2)$ when considering $n > T$. There is an overhead on learning how tasks are related for each group of related features. Computing gradients is expensive, but the number of relationship matrices also involves all tasks in a bi-directional way.Computing gradients is expensive, but the number of relationship matrices also involves all tasks in a bi-directional way.

However, as we expect tasks to have a sparse activation of their parameters, most of the computation involved with the relationship of tasks can be skipped when related to groups of features that are not active.

Let us consider (EVGENIOU; PONTIL, 2004) as a baseline comparison, which assumes that all tasks are related over all parameters by penalizing their deviation of the mean (basically clustering the tasks into a single group), and presents a time complexity of order $\mathcal{O}(T^3m^3)$. Comparing the worst time complexity of order $\mathcal{O}(T^3Gn + T^2Gn^2)$ presented by GAMTL, it does not increase the complexity related to the number of tasks T, but adds a quadratic cost on the number of features. As the method is designed to handle data of high dimensionality, the usage of sparsity in both the grouped features of the tasks and in the estimated relationships is essential to mitigate this increase in computational cost. As most groups of features will be set to zero, a great portion of the computation can be skipped. Considering that GAMTL adds a detailed structure of transference among all tasks for each group of features, the additional computational burden is counterbalanced by the gain in flexibility, as demonstrated by the experiments in the next chapter.

5.5 FINAL REMARKS

This chapter proposed GAMTL, a method that estimates how multiple tasks are related considering that:

- *local transference*: transference can occur in subsets of features considering two or more tasks; and
- *asymmetric transference*: transference from task A to task B can be different from the transference from task B to task A.

The method uses the Group LASSO regularization to consider the sparse activation of groups of related features. The latent version of this regularization also accounts for

features belonging to more than one group. GAMTL estimates linear relationships among the tasks considering each group of features independently, which leads to a fine-grained relationship among tasks that can be asymmetric.

Although the presented formulation is not convex, the problem is split into smaller convex sub-problems where the resulting problem can be solved with an alternating procedure. This leads to an algorithm that adds little computational burden when compared with simpler MTL methods, while adding more flexibility to information transference among the tasks, as demonstrated in Section 5.4.

The next chapter demonstrates the effectiveness of GAMTL in a set of experiments. It investigates how GAMTL performs with a crescent number of samples available for training, starting with an ill-conditioned scenario and ending with enough samples to train the models for all tasks. Additionally, the chapter proceeds to a real application where GAMTL is used to predict cognitive scores related to the progression of Alzheimer's Disease, comparing its performance with the state-of-the-art models. Finally, the robustness of the method when noise is present in the data, and the sensitivity to the setting of hyper-parameters is empirically explored.

6 EXPERIMENTS AND RESULTS

This chapter presents empirical results of GAMTL in three experimental settings and compare the results with existing MTL methods. Section 6.1 investigates the performance of GAMTL when varying the number of samples available for training, assessing the quality of fit and capacity to recover the structure of how tasks are related. We designed an artificial dataset with known structure of relationship in the parameters of the tasks. Section 6.2 uses GAMTL to predict cognitive scores that are related with the progression of Alzheimer's Disease, based on pre-processed medical imaging data. Section 6.3 examines the sensitivity of the parameters estimated by GAMTL with relation to noise in the data, and the sensitivity of the active variables with a varied setting of hyper-parameters.

6.1 VARYING THE NUMBER OF DATA POINTS

By leveraging information among related tasks, MTL is known to show improved performance when there are few data points available for training. This is important as many applications may not have plenty of data available. This section explores how STL and MTL methods perform with a varying number of data points, given a fixed number of features in each task. Our goal is to verify which methods present pronounced gains when few data points are available, and also verify how many data points are enough so that the methods stop improving performance.

For all experiments, the variants of GAMTL are denoted as follows:

- GAMTL - standard formulation (Section 5.1);
- GAMTL-nl - without considering loss as weighting coefficient (Section 5.2.1);
- GAMTL-nr - removing the constraint that $B^g \geq 0 \quad \forall g \in \mathcal{G}$ (Section 5.2.2); and
- GAMTL-nlnr - removing the constraint that $B^g \geq 0 \quad \forall g \in \mathcal{G}$, while not considering loss as weighting coefficient (Section 5.2.3).

6.1.1 SYNTHETIC DATASET

To validate GAMTL we designed a synthetic data with the following characteristics: 8 regression tasks are generated with 50 attributes partitioned into two groups $g_1 = [1, \ldots, 25]$ and $g_2 = [26, \cdots, 50]$. For the first two tasks, the true values of the parameters of the first group of attributes were sampled from a standard Gaussian distribution, $\mathcal{N}(0, \mathbf{I}_{25})$, and the second group of parameters was set to zero. As for the third

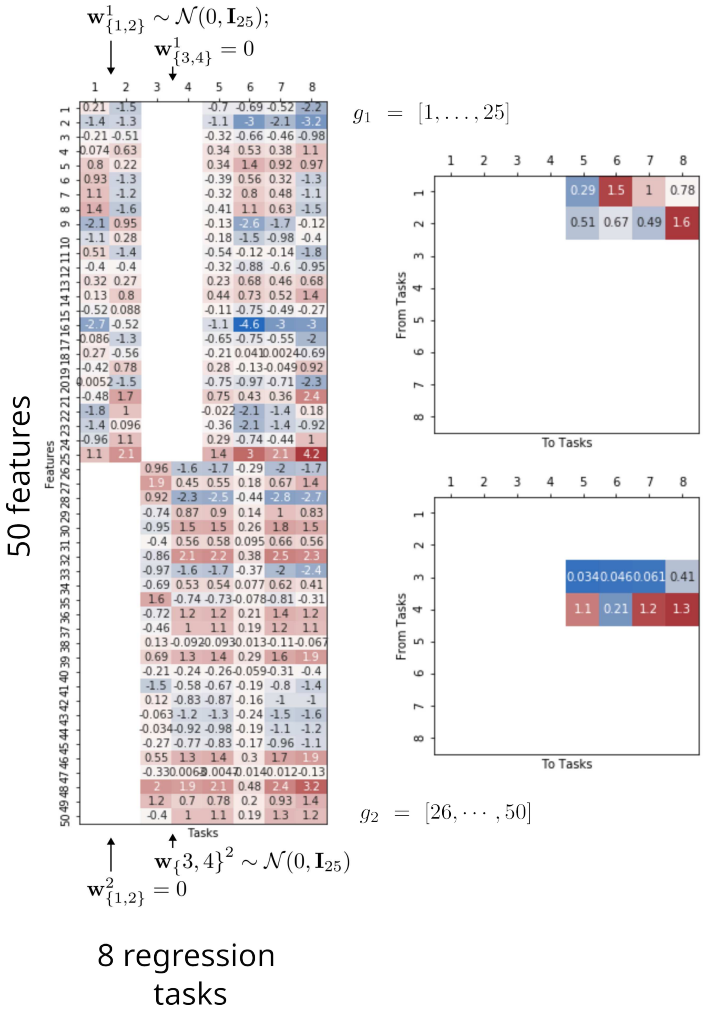


Figure 19 – Instance of generated parameters for the artificial experiment. On the left, we can see an example of W, on the right we see the relationship matrices of the two feature groups g_1, and g_2.

and fourth tasks, the parameters of the first group were set to zero, while the parameters of the second group were sampled from a standard Gaussian distribution. The last four tasks are based on the previous ones, as their parameters are generated as a linear combination of the parameters of the previous tasks. The linear combination parameters were sampled from a truncated Gaussian distribution, ensuring that all values were positive. An instance of generated parameters is shown in Fig. 19. On the left, we can see an example of W, on the right we see the relationship matrices of the two feature groups g_1, and g_2.

The dataset of each task was sampled from a standard Gaussian distribution with 300 data points and 50 variables. After that, a Gaussian noise with $\sigma = 0.4$ was added on the first four tasks, and with $\sigma = 2.9$ on the remaining tasks. This difference in the amount of noise is related to our assumption of asymmetric transference based on loss. Transference is expected to occur from tasks with lower costs to tasks with higher costs, recovering the transference structure among all tasks. If all tasks have the same

level of noise, all transferences will be penalized similarly and the last four tasks will be equally encouraged to transfer back to the first tasks, resulting in quasi-symmetric matrices $B^g, \quad \forall g \in \mathcal{G}$.

6.1.2 EXPERIMENTAL SETUP

The number of data points available for training and testing the models varied from 30 to 100 per task, as all methods converge to similar performances from this value on. The synthetic dataset is split so that 70% of the sample are used for the training and 30% for the test. For each number of data points, the hyper-parameters of all methods are chosen by a holdout procedure in which the training set is split in 70% for training and 30% for validation. The best parameters are used to train the models for 30 independent runs.

The results of GAMTL are compared with the LASSO (TIBSHIRANI, 1996) and Group LASSO (JACOB *et al.*, 2009a) as STL contenders. MTFL (KANG *et al.*, 2011), and MTRL were considered as MTL contenders, along with the GAMTL variants. Hyper-parameters were chosen using the Python library Optuna (AKIBA *et al.*, 2019), which implements a relational sampling strategy to search for the optimal values of some function in a given interval. For each method, 200 trials are used for the search of hyper-parameter values, and the values with the lowest normalized mean squared error (NMSE) in the validation portion of the training data are chosen to the experiment. The NMSE metric is 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.

For the LASSO, the search limits were $\lambda \in [10^{-5}, 4]$, while for the Group LASSO we adopted $\lambda \in [10^{-5}, 15]$. MTFL had $2, 3, 4$ as the quantity of task groups, and $\rho_1, \rho_2 \in [0.001, \cdots, 10]$. MTRL values were chosen as $\rho_1, \rho_2 \in [0.0001, \cdots, 10]$. All GAMTL variants used $\lambda_1, \lambda_2, \lambda_3 \in [10^{-5}, 5]$. All reports include the mean and standard deviation of the normalized mean squared error (NMSE) on the test set, over all 30 runs.

6.1.3 RESULTS ON THE SYNTHETIC DATASET

Figure 20 shows the behavior of the average NMSE for all methods as the size of the training sample is increased. Considering first the STL methods, notice how Group LASSO outperforms the LASSO at the given interval, highlighting that ignoring the group structure of the features is sub-optimal, even if no transference is modeled. For LASSO all features are independent, while for Group LASSO there is a group-structured feature dependence in place. All MTL methods improve upon STL, specially when the number of data points available for training is small, as in the interval between 30 and 70 data

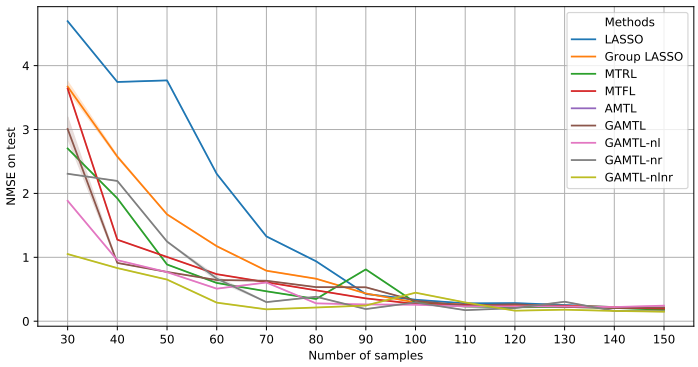


Figure 20 – Normalized Mean Squared Error (NMSE) of all methods in this experiment, with a varying number of data points available for training. MTL methods outperform STL methods specially when there are only small sets of data points for training.

points. Notice that AMTL presents sub-optimal results: when $m \leq 60$ it performs better than the LASSO, while for $m > 60$ its performance is worse than all other methods (MTFL, MTRL, and GAMTL variants). We believe that despite having a transference mechanism, this happens because it doesn't leverage the group structure information. GAMTL variants are able to improve even further when compared to the other MTL methods (AMTL, MTFL, and MTRL) on this synthetic setting. GAMTL-nl presented the best results in this experiment. Even when $m = 30$ it is able to achieve a NMSE that Group LASSO reaches only when twice the number of data points is available. As m increases, for instance when $m \geq 90$, both MTL and STL methods achieve similar performance as all tasks have enough data points for a successful training procedure with respect to the number of parameters estimated by each method.

Figure 21 shows the relationship matrices B^g estimated by GAMTL when $m = \{40, 80, 90, 100\}$ sided by the original B^g used on the generative process. With few data points, such as when $m = \{40, 80\}$, the recovered relationship structure estimated by GAMTL is not similar to the original, but is informative enough to improve performance in an ill-conditioned scenario. Since GAMTL formulation is subject to local optima, we conjecture that this is a local minima. With enough data points, when ($m \geq 90$), the relationship structure is sufficiently close to the generative structure.

6.2 PREDICTING COGNITIVE SCORES RELATED TO THE PROGRESSION OF ALZHEIMER'S DISEASE WITH GAMTL

Alzheimer's Disease (AD) is the most common form of dementia in the world (KHACHATURIAN, 1985). As people live longer and we improve our capabilities of identifying and diagnosing dementia, we expect the number of people living with demen-

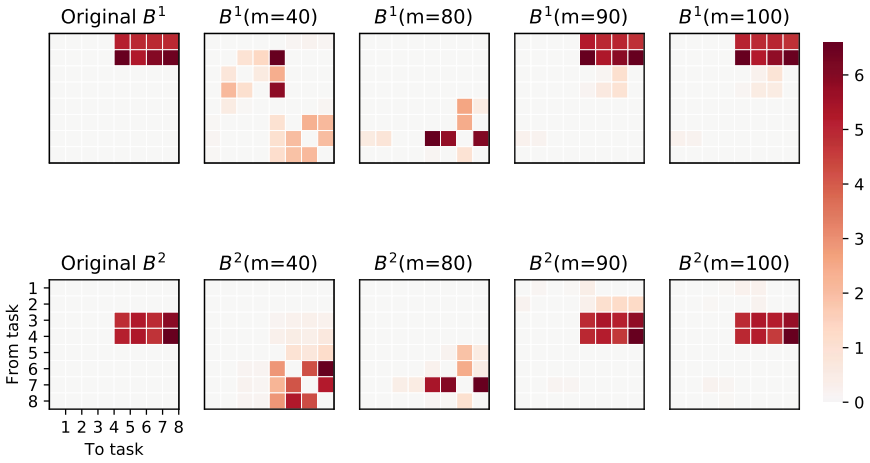


Figure 21 – Relationship matrices of the two groups of features, estimated by GAMTL on the synthetic dataset with different training / testing set sizes. The estimated relationships between tasks are used to regularize tasks parameters. The more data points available for training, the closer the relationship matrices are to their true values.

tia will more than triple by 2050 when compared with the estimates of 2018, according to the World Alzheimer Report of 2018 (Alzheimer's Disease International, 2018). As a recognition of the need for global actions to mitigate and further investigate dementia, in May 2017, the World Health Assembly endorsed a global action plan [1] on a public health response to dementia, directed to policy-makers, international, regional and national partners. The absence of treatment to reverse the progression of this neurodegenerative disease fuels plenty of current research in the hope to understand the underlying mechanisms of AD. (LIU *et al.*, 2018) and (ZHOU *et al.*, 2011b) have already shown that MTL can contribute in modeling the connection between cognitive scores (representing multiple regression tasks) and the progression of AD, considering multiple distinct Regions Of Interest (ROI) in the brain, with each ROI representing a group of features.

6.2.1 ADNI DATASET

The performance of GAMTL is tested in a real scenario on the ADNI dataset [2]. This dataset of medical images was collected by the Alzheimer's Disease Neuroimaging

[1] <https://www.who.int/mental_health/neurology/dementia/action_plan_2017_2025/en/>

[2] Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in this experiment or in writing the subsequent analysis. A complete listing of ADNI investigators can be found at: <http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf>

Initiative (ADNI) and was 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 contains information from 816 subjects that are the same for all tasks and are divided into three stages: those cognitively normal (CN) (228), with mild cognitive impairment (MCI) (399), and with Alzheimer's disease (AD) (189). There is a total number of 327 features including cortical thickness average, cortical volume, and sub-cortical volume. In this dataset, regions of interest (ROI) in the brain are represented by groups of features. The labels available in this dataset and used as tasks in our formulation include five cognitive measures. 3 of these scores are based on Rey Auditory Verbal Learning Test (RAVLT), a test that is used as a neuropsychological instrument for evaluating episodic declarative memory. It provides scores for assessing immediate memory, new verbal learning, susceptibility to interference (proactive and retroactive), retention of information after a period of time, and memory recognition (MAGALHÃES; HAMDAN, 2010). The other two scores are the Mini Mental State Exam Score (MMSE), a test used to screen for cognitive impairment; and the Alzheimer's Disease Assessment Scale (ADAS) cognitive total score, used to assess the severity of cognitive symptoms of dementia. Here is a list of the tasks:

- RAVTL 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).

The usage of these scores is widespread, impacting on drug trials, assessment of the severity of symptoms of AD, the progressive deterioration of functional ability, and deficiencies in memory, as highlighted in (LIU *et al.*, 2018), thus emphasizing the importance of this type of modeling. Our goal is to estimate the values of the scores obtained by the individuals, while estimating how scores are related through ROI activation.

6.2.2 EXPERIMENTAL SETUP

This experiment considers all STL and MTL methods used in the previous experiment, but add other state-of-the-art contenders. We added some other MTL methods for completion: AMTL (LEE *et al.*, 2016) is also based on using task parameters as a latent basis, but does not account for groups of features; MT-SGL (LIU *et al.*, 2018), a method that instead of learning a shared representation from the level of feature and groups of features across all the tasks simultaneously, it encourages i) individual feature selection

based on the utility of the features across all tasks, and ii) task specific group selection based on the utility of the group. In this way, the method decouples the sparsity of groups of features from individual features, although it is not designed to estimate the relationships among tasks; GO-MTL (KUMAR; DAUMÉ, 2012) that is based on a latent basis to model related tasks, thus estimating transference structure; MSSL (GONÇALVES *et al.*, 2016) which accounts for unrelated tasks and estimates a precision matrix as the learning model structure for transference among tasks; MTRL (ZHANG; YEUNG, 2010b), that uses a probabilistic framework and places a matrix-variate prior distribution on tasks coefficients to model their relationship; and MTFL (KANG *et al.*, 2011) that groups tasks based in an orthogonal-complement sub-spaces decomposition where features are shared among tasks. We obtained implementations of the contenders with their respective authors. AMTL, GO-MTL, MSSL, and MTRL were implemented in Matlab, while MT-SGL was implemented in Python.

As in the previous experiment, Optuna (AKIBA *et al.*, 2019) is used to search for the value of the hyper-parameters of all methods, using 200 samples for the search of each method. For this experiment we used a 5-fold cross-validation procedure, where each fold contains the same proportion of participants from the stages CN, MCI, and AD. The search limits used to tune the methods hyper-parameters are defined as follows: For the LASSO $\lambda \in [10^{-5}, \cdots, 4]$, while for the Group LASSO $\lambda \in [10^{-5}, 15]$. For AMTL $\mu, \lambda \in [10^{-5}, 5]$. GO-MTL has the number of groups set to $2, 3, 4$, while $\rho_1 \in [10^{-4}, 10]$, $\rho_2 \in [10^{-4}, 10]$. MSSL had $\rho_1, \rho_2 \in [10^{-5}, 10]$. For MTFL, $[2, 3, 4]$ are the quantity of task groups, and $\rho_1, \rho_2 \in [10^{-5}, 10]$. MT-SGL used $r \in [10^{-5}, 15]$. MTRL hyper-parameters were chosen as $\rho_1, \rho_2 \in [10^{-4}, 10]$. For AMTL, $\mu, \lambda \in [10^{-5}, 5]$. All variants of GAMTL used $\lambda_1, \lambda_2, \lambda_3 \in [10^{-5}, 5]$. The hyper-parameters values with the best results in this step are selected to train the methods for 30 runs, accounting for the initial randomness of the parameters of the tasks.

6.2.3 PERFORMANCE RESULTS

Table 1 shows the overall performance of all methods using the NMSE metric. Values are the mean and standard deviation of the 30 runs and the best result is highlighted in bold.

Among the STL methods, LASSO is the one presenting the best score, but most MTL methods achieved better results when compared with the LASSO. GAMTL variants achieved better results than all methods, but presented more deviation than most of them on the results. As GAMTL estimates more parameters, it is at certain extent an expected outcome. A Mann-Whitney U test with $p \leq 0.05$ was used to verify that the score difference between GAMTL-nr and all other methods was statistically significant.

As for each task individually, the mean-squared error (MSE) is used to compare

Table 1 – NMSE of all methods in ADNI dataset (mean and standard deviation over 30 runs). GAMTL-nr had the best results (highlighted in bold), closely followed by the other GAMTL variants, and MTRL method. A Mann-Whitney U non-parametric test was run, assuring the significance of score improvement when comparing GAMTL-nr with all other methods.

	Method	NMSE
STL	LASSO	0.840 $(2.2 \cdot 10^{-16})$
	Group-LASSO	0.977 $(2.0 \cdot 10^{-1})$
MTL	GO-MTL	0.896 $(1.1 \cdot 10^{-16})$
	MSSL	0.818 $(1.1 \cdot 10^{-16})$
	MTFL	0.810 $(2.2 \cdot 10^{-16})$
	MT-SGL	0.801 $(1.5 \cdot 10^{-13})$
	MTRL	0.791 $(2.2 \cdot 10^{-16})$
	AMTL	0.898 (0.0)
	GAMTL	0.781 $(1.4 \cdot 10^{-5})$
	GAMTL-nl	0.787 $(4.3 \cdot 10^{-5})$
	GAMTL-nr	**0.780** $(2.2 \cdot 10^{-5})$
	GAMTL-nlnr	0.789 $(2.8 \cdot 10^{-5})$

the methods, with the results presented in Table 2, and the mean absolute error (MAE) is reported in Table 3. For visual interpretation, the same information is depicted in Figure 22 on a bar plot. Each sub-figure presents a bar plot of the MSE obtained by all methods in the experiment for each task.

It is important to highlight that in terms of MAE, the LASSO results were competitive with GAMTL variants. On tasks TOTAL and T30 it obtained better performance, while closely following GAMTL variants on tasks MMSE and ADAS. But notice that beyond sole performance, GAMTL also returns a complex structure mapping how tasks are related in a flexible and interpretable way. In this type of application, the interpretation of the active variables is often more important than pure metric performance. Although Group LASSO has access to the group structure of the features, this information is not enough to improve performance on an STL setting where no information is shared among tasks. Considering each task individually, AMTL presented the smaller MSE for the task TOTAL but showed poor performance for the other tasks. For the same task Group LASSO showed wide variance in their results. For the task T30, the LASSO presented the best result, closely followed by MT-SGL. For all other tasks, GAMTL variants had the most competitive performance. In contrast with the task TOTAL, the tasks RECOG and MMSE, AMTL showed a poor performance. GO-MTL also showed a similar behavior in this task: it showed competitive performance for some tasks, but presented poor results for the task MMSE. GAMTL got the best result for both RECOG and ADAS tasks. As for the ADAS task, the variation of performance among the methods is small. Each task benefited the most from a different strategy of transference, but still, task T30 could not

Table 2 – MSE of all methods per task in ADNI dataset. The best results for each task are highlighted in bold.

	Method	TOTAL	T30	RECOG	MMSE	ADAS
STL	LASSO	0.857 $(2.2 \cdot 10^{-16})$	**0.617**$(\mathbf{1.1 \cdot 10^{-16}})$	0.962 $(4.4 \cdot 10^{-16})$	0.618 $(5.5 \cdot 10^{-16})$	0.556 $(1.1 \cdot 10^{-16})$
	Group-LASSO	1.190 $(9.9 \cdot 10^{-1})$	0.705 $(1.3 \cdot 10^{-1})$	1.019 $(9.6 \cdot 10^{-2})$	0.736 $(1.0 \cdot 10^{-1})$	0.635 $(7.1 \cdot 10^{-2})$
MTL	GO-MTL	0.837 (0.0)	0.643 $(2.2 \cdot 10^{-16})$	0.842 $(3.3 \cdot 10^{-16})$	0.856 $(1.1 \cdot 10^{-16})$	0.584 $(2.2 \cdot 10^{-16})$
	MSSL	0.846 $(2.2 \cdot 10^{-16})$	0.648 $(3.3 \cdot 10^{-16})$	0.856 $(0.0 \cdot 10^{0})$	0.597 $(4.4 \cdot 10^{-16})$	0.566 $(1.1 \cdot 10^{-16})$
	MTFL	0.851 $(2.2 \cdot 10^{-16})$	0.648 $(0.0 \cdot 10^{0})$	0.839$(1.11 \cdot 10^{-16})$	0.588$(2.22 \cdot 10^{-16})$	0.554$(1.11 \cdot 10^{-16})$
	MT-SGL	0.885 $(5.9 \cdot 10^{-13})$	0.619 $(5.6 \cdot 10^{-13})$	0.760 $(7.0 \cdot 10^{-13})$	0.612 $(3.4 \cdot 10^{-13})$	0.551 $(4.7 \cdot 10^{-13})$
	MTRL	0.848 $(1.1 \cdot 10^{-16})$	0.674 $(1.1 \cdot 10^{-16})$	0.786 $(3.3 \cdot 10^{-16})$	0.579 $(1.1 \cdot 10^{-16})$	0.520 $(0.0 \cdot 10^{0})$
	AMTL	**0.769**$(2.2 \cdot 10^{-16})$	0.700 $(4.4 \cdot 10^{-16})$	0.994 (0.0)	0.753 $(1.1 \cdot 10^{-16})$	0.547 $(2.2 \cdot 10^{-16})$
	GAMTL	0.914 $(2.7 \cdot 10^{-5})$	0.653 $(6.0 \cdot 10^{-5})$	**0.744**$(\mathbf{1.5 \cdot 10^{-5}})$	0.560 $(3.2 \cdot 10^{-5})$	**0.506**$(\mathbf{1.8 \cdot 10^{-5}})$
	GAMTL_nl	0.860 $(9.3 \cdot 10^{-5})$	0.646 $(6.9 \cdot 10^{-5})$	0.794 $(1.7 \cdot 10^{-4})$	0.563 $(4.5 \cdot 10^{-5})$	0.528 $(4.2 \cdot 10^{-5})$
	GAMTL_nr	0.870 $(5.0 \cdot 10^{-5})$	0.654 $(6.5 \cdot 10^{-5})$	0.775 $(6.0 \cdot 10^{-5})$	**0.555**$(\mathbf{4.3 \cdot 10^{-5}})$	0.513 $(3.3 \cdot 10^{-5})$
	GAMTL_nlnr	0.857 $(3.1 \cdot 10^{-5})$	0.645 $(5.6 \cdot 10^{-5})$	0.801 $(9.8 \cdot 10^{-5})$	0.566 $(6.1 \cdot 10^{-5})$	0.531 $(2.8 \cdot 10^{-5})$

Table 3 – MAE of all methods per task in ADNI dataset. The best results for each task are highlighted in bold.

	Method	TOTAL	T30	RECOG	MMSE	ADAS
STL	LASSO	0.714 (0.0)	**0.629** (0.0)	0.819 (0.0)	0.635 (0.0)	0.560 (0.0)
	Group-LASSO	0.796 $(2.4 \cdot 10^{-1})$	0.661 $(5.2 \cdot 10^{-2})$	0.831 $(3.8 \cdot 10^{-2})$	0.695 $(5.0 \cdot 10^{-2})$	0.599 $(4.2 \cdot 10^{-2})$
MTL	GO-MTL	0.715 (0.0)	0.650 (0.0)	0.736 (0.0)	0.790 (0.0)	0.575 (0.0)
	MSSL	**0.709** (0.0)	0.662 (0.0)	0.757 (0.0)	0.642 (0.0)	0.556 (0.0)
	MTFL	0.711 (0.0)	0.665 (0.0)	0.748 (0.0)	0.639 (0.0)	0.549 (0.0)
	MT-SGL	0.725$(3.4 \cdot 10^{-14})$	0.654$(1.3 \cdot 10^{-13})$	0.700$(4.3 \cdot 10^{-13})$	0.653$(2.5 \cdot 10^{-14})$	0.558$(1.6 \cdot 10^{-13})$
	MTRL	0.720 (0.0)	0.676 (0.0)	0.712 (0.0)	0.640 (0.0)	0.537 (0.0)
	AMTL	0.721 (0.0)	0.669 (0.0)	0.857 (0.0)	0.758 (0.0)	0.543 (0.0)
	GAMTL	0.743$(9.2 \cdot 10^{-6})$	0.659$(2.3 \cdot 10^{-5})$	**0.692**$(7.7 \cdot 10^{-6})$	0.631$(1.8 \cdot 10^{-5})$	0.549$(4.6 \cdot 10^{-5})$
	GAMTL_nl	0.720 $(4.9 \cdot 10^{-5})$	0.669 $(5.0 \cdot 10^{-5})$	0.723 $(1.0 \cdot 10^{-4})$	0.626 $(5.2 \cdot 10^{-5})$	0.537 $(3.5 \cdot 10^{-5})$
	GAMTL_nr	0.725 $(2.3 \cdot 10^{-5})$	0.672 $(3.4 \cdot 10^{-5})$	0.713 $(1.6 \cdot 10^{-5})$	**0.623**$(4.2 \cdot 10^{-5})$	**0.531**$(2.2 \cdot 10^{-5})$
	GAMTL_nlnr	0.718 $(1.5 \cdot 10^{-5})$	0.668 $(4.6 \cdot 10^{-5})$	0.727 $(4.4 \cdot 10^{-5})$	0.627 $(5.6 \cdot 10^{-5})$	0.539 $(3.0 \cdot 10^{-5})$

benefit from MTL. As each method holds distinct premises for the transference among tasks, this result indicates that no single transference mechanism is capturing all nuances of information transference among the tasks. Most importantly, when not improving performance, some MTL methods incur in poorer performance.

A linha da primeira figura está errada. na última figura tem um AMTL3 To see how GAMTL improves upon their results, we focus now on methods that account for grouped features. Choosing Group LASSO as the main reference, the difference of MSE between Group LASSO and GAMTL variants for each run is taken and the results are shown in Figure 23. Positive values indicate the method had a smaller MSE than the Group LASSO (positive transference), while negative values indicate negative transference. GAMTL variants improved the generalization performance on all tasks when compared with Group LASSO. Strong improvements are exhibited for RECOG, MMSE, and ADAS tasks, while not incurring negative transference for the most challenging tasks (TOTAL and T30). RECOG is the task that benefits the most from GAMTL models.

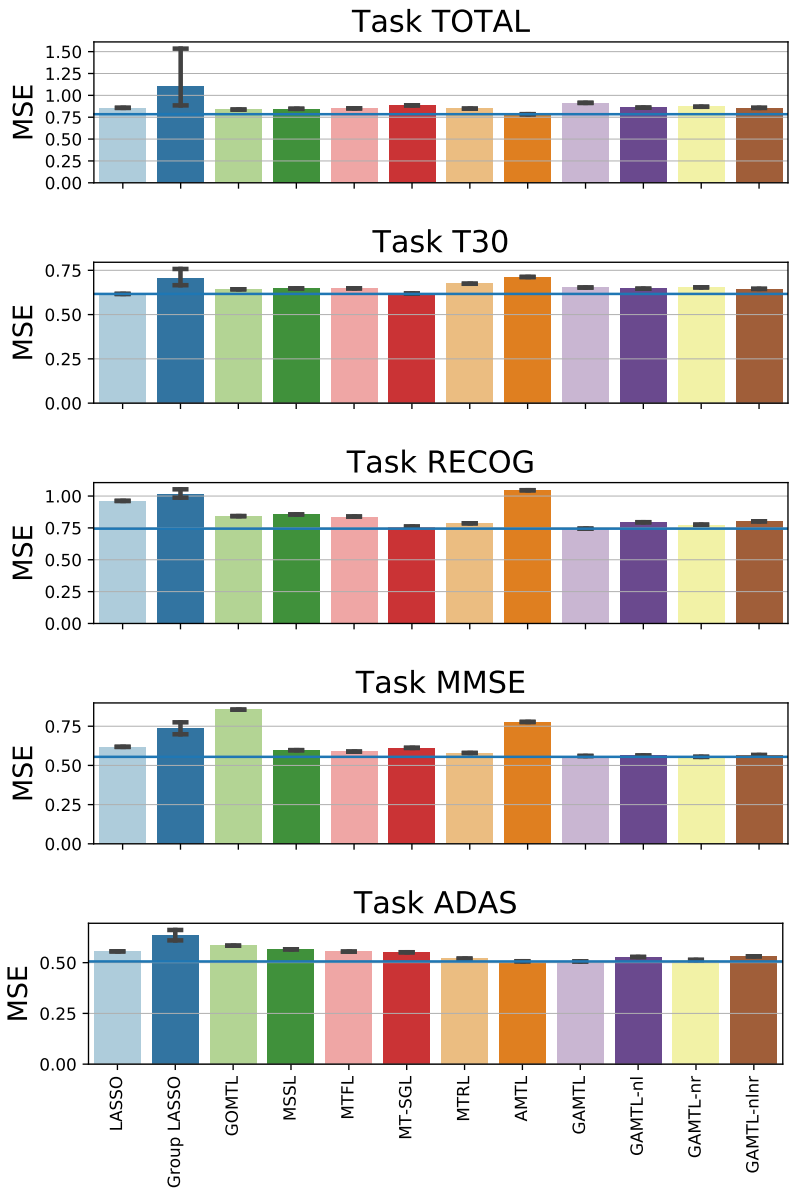


Figure 22 – MSE of all methods, for each task, with a blue horizontal line highlighting the best performance. AMTL had the best performance for the TOTAL task, with Group LASSO showing a great variance in their results. For the task T30, the LASSO presented the best result, closely followed by MT-SGL. For the other tasks GAMTL variants had the most competitive performance.

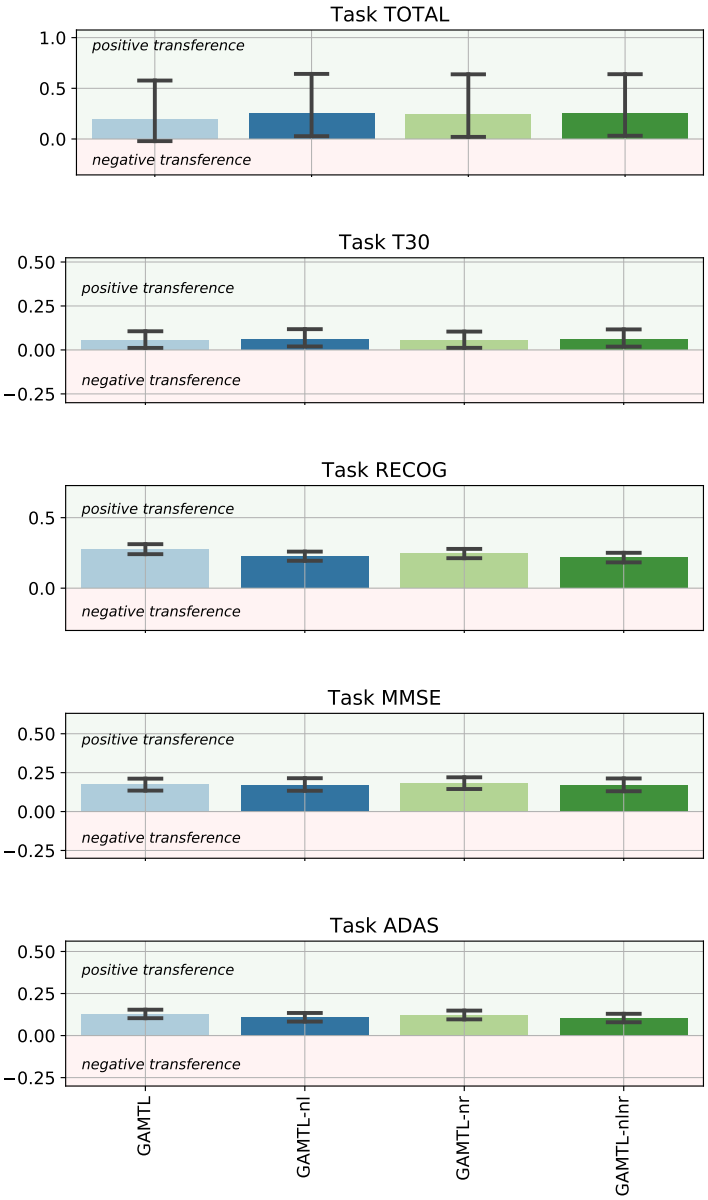


Figure 23 – GAMTL gains over STL Group LASSO for each task. For the task TOTAL the gains vary due to the unstable performance of Group LASSO on that task. For the task T30 a consistent small gain was obtained, and GAMTL variants presented an expressive gain for tasks RECOG, MMSE, and ADAS.

6.2.4 RECOVERED SPARSITY PATTERNS

Figure 24 presents a heatmap of the structural sparsity produced by each method that achieved the best result on at least one task. The mean of the parameter values for each group of parameters is taken and if the value is greater than zero, we consider it an active group, represented by a darker color. LASSO (Figure 24a) is used as a reference for the STL methods. AMTL obtained the best result for the task TOTAL and is represented in Figure 24b. Notice the presence of two groups of related tasks: TOTAL, T30, RECOG, and MMSE as part of one group, while ADAS was isolated in a singleton group. It is also noticeable that when tasks belong to the same group, they show a strongly related sparsity pattern on all tasks features.

GAMTL variants show sparser results (Figures 24c and 24d). The ADAS task also seems unrelated to the other tasks by presenting a different sparsity behavior on GAMTL results. Both GAMTL methods allow the tasks to relate in different ways when sharing, thus yielding a more flexible structural sparsity pattern for related tasks. In this case, GAMTL allows ADAS task to be related to other tasks only in a few groups of features.

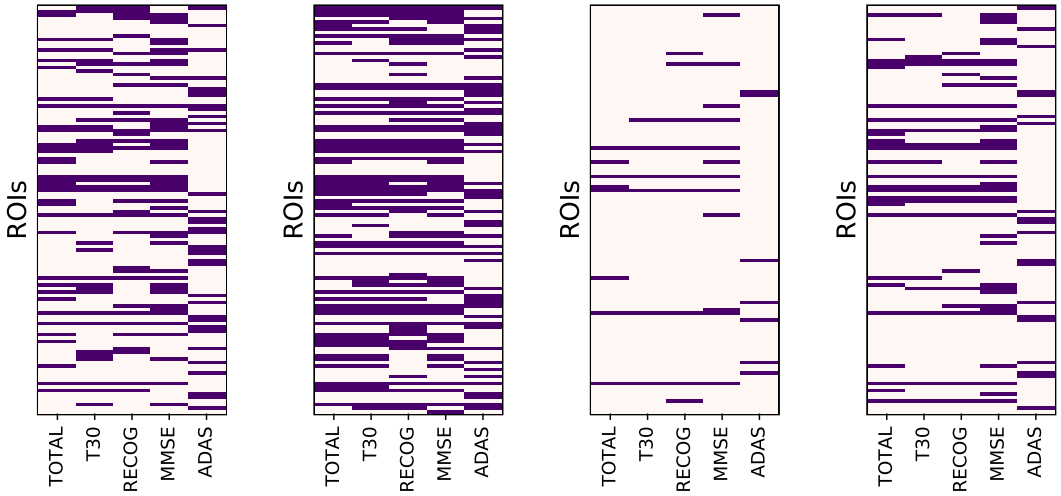
The transference scheme encoded on $B^g, \forall g \in \mathcal{G}$ matrices is responsible for regularizing the parameters of the tasks to fit into the estimated relationships. As these matrices present interpretable information, a Stability Selection (MEINSHAUSEN; BüHLMANN, 2010) procedure was performed (see the next section) to examine how robust are the active variables of a learning model with relation to noise on the training data and arbitrary settings of hyper-parameter values.

6.3 STABILITY SELECTION ON ADNI

Besides learning the parameters of the tasks, GAMTL also estimates GT^2 parameters for the relationship among the tasks. On one hand, these extra parameters encode a distinct source of information retrieved from data; on the other hand, three hyper-parameters are added and need to be fine-tuned. This raises a question: is the set of active variables robust to hyper-parameterization processes and data noise? As in (LIU *et al.*, 2009; LIU *et al.*, 2018; HE; YU, 2010), a Stability Selection is used on the ADNI dataset both to validate the robustness of GAMTL and also to highlight the interpretative capabilities of the model.

6.3.1 EXPERIMENTAL SETUP

Stability Selection (MEINSHAUSEN; BüHLMANN, 2010) is a feature selection procedure that i) employs a sampling procedure on both the data that is used to train the method and hyper-parameter values; and ii) computes the marginal probability of a feature being active by the total number of runs in the procedure.



(a) The STL LASSO parameters shows a visual similarity involving the parameters of all but the ADAS task.

(b) AMTL leverages the relationship among tasks and shows a clearer shared pattern, but less sparse.

(c) GAMTL does not enforce groups to be active for ADAS when the task is not related to the others. This is the sparser solution.

(d) GAMTL-nr also allows different groups of features to share differently among tasks. The results are also quite sparse.

Figure 24 – Sparsity pattern estimated by the methods with best performance on at least one task. Darker cells indicate groups of attributes where the mean of their parameters is greater than zero. All methods show a distinct sparsity pattern on the ADAS task, when compared to the other tasks. When comparing results of AMTL with the LASSO, we see groups of features that became active for ADAS task, but play no role in the STL result. GAMTL variants (showed on 24c and 24d) present even sparser results, with the benefit of not enforcing groups to be active for ADAS when the task is not related to the others, preserving the flexibility of tasks to share only on the groups of features that are valuable for transference.

Given a set of hyper-parameter values Γ, a subset of the available dataset is chosen randomly and without replacement, then the model is trained for N times. After that, the frequency that a variable was active in the obtained solutions is computed and the variables are filtered with a threshold. The overall process is described in Algorithm 8. For each variable i of our problem and a certain configuration of hyper-parameters $\lambda = \lambda_1, \lambda_2, \lambda_3 \in \Gamma$, τ_i represents the percentage of times that variable i was active over all runs. Let $\hat{S}^\lambda = \{\tau_i, | i \in W \cup B^g, \forall g \in \mathcal{G}\}$ be the set of percentages, and $\hat{S} = \{\hat{S}^\lambda | \lambda \in \Gamma\}$ be the set of percentages over all hyper-parameter values. A variable i is considered stable when the mean over all elements of $\hat{S}^\lambda$ are greater than a certain threshold. A ROI (which corresponds to a group of features) is stable if the mean of the percentages of all its features is greater than the threshold.

Algorithm 8 Stability Selection

Require: $\Gamma, (X_t, \mathbf{y}_t) \forall t \in \mathcal{T}$, and a threshold.
1: $\hat{S} = \{\emptyset\}$
2: **for** $\lambda_1, \lambda_2, \lambda_3 \in \Gamma$ **do**
3: **while** $run \leq N$ **do**
4: **for** $t \in \mathcal{T}$ **do**
5: $\tilde{X}_t \in \mathbb{R}^{m_t/2,n}, \tilde{y} \in \mathbb{R}^{m_t/2} \sim X_t, \mathbf{y}_t \forall t \in \mathcal{T}$
6: **end for**
7: Initialize GAMTL with $\lambda_1, \lambda_2, \lambda_3$
8: Train GAMTL on $\tilde{X}, \tilde{y}$
9: **end while**
10: $\hat{S}^{\lambda} = \{\tau_i | i \in W \cup B^g, \forall g \in \mathcal{G}\}$
11: $\hat{S} = \hat{S} \cup \hat{S}^{\lambda}$
12: **end for**
13: Compute the mean over $\hat{S}$ and apply threshold.
return $\hat{S}, \hat{S}^{\lambda} \forall \lambda \in \Gamma$

The model hyper-parameters are chosen from the set $\Gamma = \{\lambda_1, \lambda_2, \lambda_3 | \lambda_1, \lambda_2 \in [0.001, \cdots, 5], \lambda_3 \in [0.0001, \cdots, 1]\}$ and present results using a threshold of 80%, which is commonly used in the literature.

6.3.2 STABILITY SELECTION RESULTS

For each ROI, the mean of the stability percentages of its features is taken and compared against the pre-defined threshold of 80%, resulting in a binary matrix $W_{stab} \in \mathbf{Z}^{G \times T}$ whose entries indicate which groups are active for which tasks. For visualization purposes, we clustered the rows of W_{stab}, considering the number of clusters as 2 - after a comparison of the Silhouette Score (ROUSSEEUW, 1987) of the samples after experimenting with values in the range between 2 and 10. K-means is applied in 30 distinct runs to alleviate the effects of the random initialization, keeping the result with the best results in terms of within cluster sum of squares. Figure 25 presents W_{stab} split in those two groups. Notice that the clustering procedure here aims to help us visualize the stability of ROIs that are similar to each other. GAMTL estimated transference structure does not depend on this procedure to be meaningful, but since we have a great number of ROIs, it may be difficult to visualize sets of ROIs are active in a stable manner on the tasks.

On the first cluster (left) almost all ROIs are stable on the ADAS task, while almost no ROI is stable for the other tasks. The second cluster (right) shows stable ROIs for all tasks but ADAS. However, each cluster contains a few distinct active features depending on each task, which shows the flexible transference among tasks. This is a key point in GAMTL models: the distinct behavior of features is an important characteristic of the MTL problem setting. If the model does not account for the distinct roles that features can play on related tasks, negative transference may occur. A relationship that is

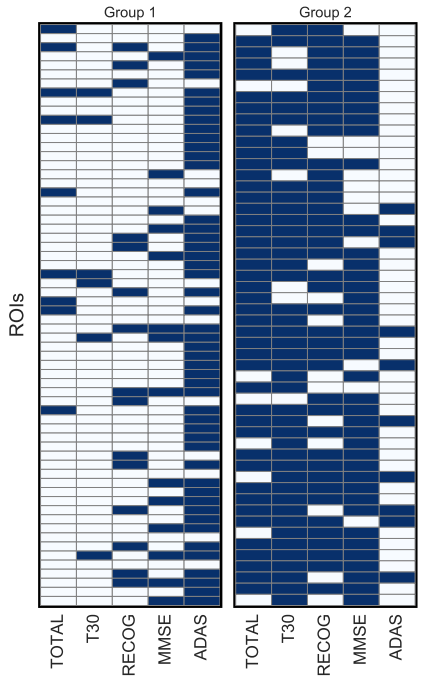


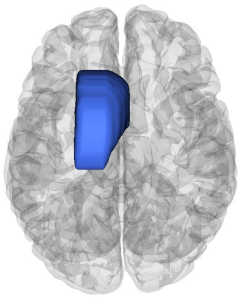
Figure 25 – ROIs clustered by similar stability among all cognitive tests (tasks). The cluster on the left shows high activity for ADAS task, with a sparse presence on the other tasks. On the other hand, the second cluster is highly active for all tasks but ADAS, clearly showing the flexible transference possibilities of GAMTL.

highly expressed in a set of features among two tasks should not be imposed on a different set of features.

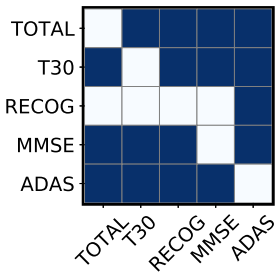
Choosing some ROIs to further explore the transference among tasks, we picked two ROIs that are active for all tasks on the second cluster: the Left Caudate and Left Inferior Temporal. Figure 26a shows the illustrative anatomical location of the Left Caudate on a template brain, and Figure 26b shows the estimated relationship among tasks considering this ROI. The task RECOG is influenced by all other tasks (RECOG column) but influences only the task ADAS (see the row for task RECOG), while all other tasks are fully connected on this ROI. The Left Inferior Temporal ROI is depicted anatomically in Figure 26c. In this case, the ADAS task is not related to any other task; TOTAL and MMSE influence all other tasks, while receiving their influence as well; and RECOG influences TOTAL and MMSE tasks while is influenced by TOTAL, T30, and MMSE. Even by choosing ROIs that are active on the solution of all tasks, different relationship schemes among tasks are expressed, stressing the benefits of a flexible mechanism that learns how transference occurs.

Considering now the estimated relationship matrices, the average of stability scores is computed for each $B^g \forall g \in \mathcal{G}$, and these are the 6 ROIs with the highest average value:

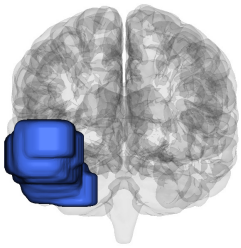
- Left Cerebral Cortex;



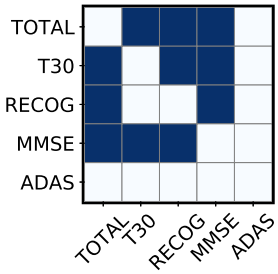
(a) Left Caudate illustrative anatomical position.



(b) Relationship among all tasks for the Left Caudate ROI.



(c) Left Inferior Temporal illustrative anatomical position.

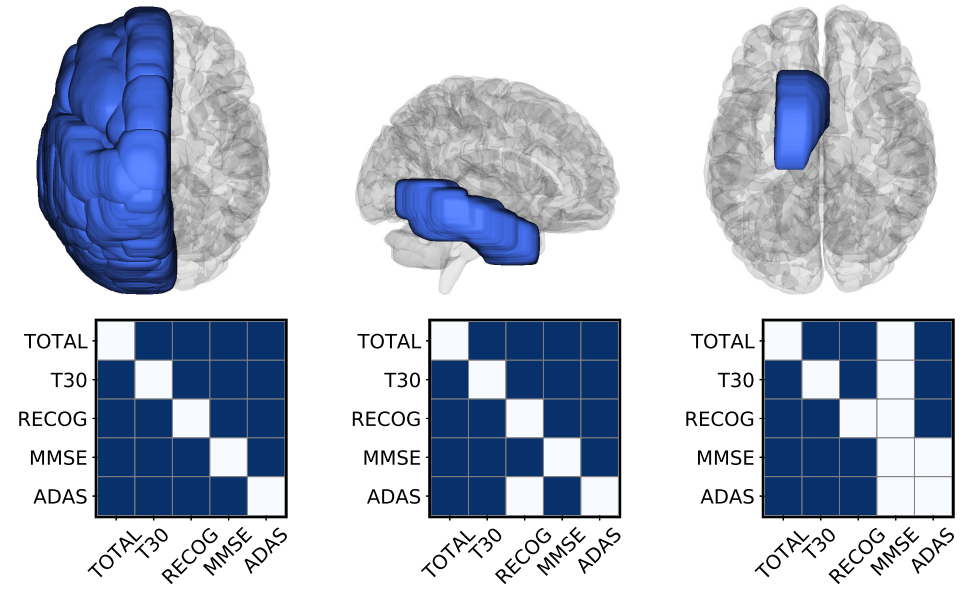


(d) Relationship among all tasks for the Left Inferior Temporal ROI.

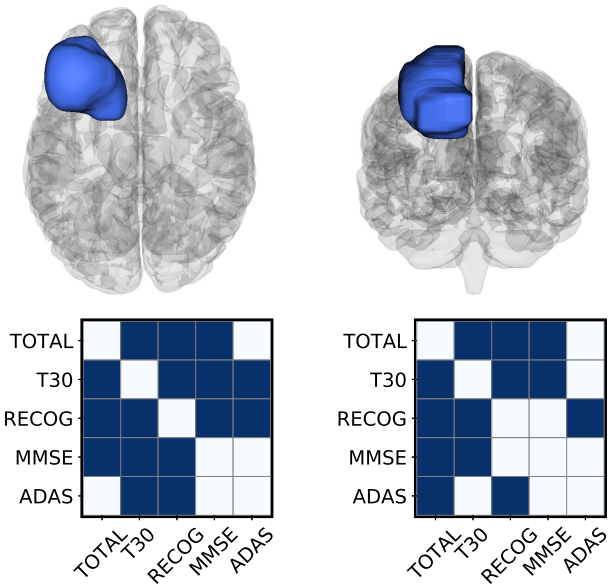
Figure 26 – Left Caudate and Left Inferior Temporal ROIs belonging to the second cluster were stable on all tasks. Their illustrative anatomical position is depicted on the left, while the right depicts the estimated relationship among the tasks. Despite being part of the same cluster, those two ROIs present distinct transference among tasks.

- Right Inferior Temporal;
- Left Caudate;
- Left Accumbens Area;
- Left Pars Orbitalis;
- Left Superior Parietal.

Since the Left Caudate was already explored, its presentation is skipped. Figure 27a illustrates the Left Cerebral Cortex, which is the ROI with most stable transference among the tasks in all directions. This is the outermost layer surrounding the brain, that serves as a connection for several ROIs. We can see strong relationships among tasks in this analysis.



(a) Left Cerebral Cortex. (b) Right Inferior Temporal. (c) Left Accumbens Area.



(d) Left Pars Orbitalis. (e) Left Superior Parietal.

Figure 27 – ROIs with highest stability: the Left Cerebral Cortex, the Right Inferior Temporal, the Left Accumbens Area, the Left Pars Orbitalis, and the Left Superior Parietal. Each sub-figure shows the illustrative anatomical position of the ROI, together with the respective estimated relationship matrix.

Figure 27b shows the Right Inferior Temporal ROI, also presenting stable connections among all tasks, with the only exception when transferring from ADAS to RECOG.

The Accumbens Area is a small part of the Left Caudate ROI, being depicted in Figure 27c. The relationship matrix shows fewer stable connections when compared to the results of the previous ROIs. The MMSE task is not influenced by any other tasks,

influencing all but the ADAS task. The Left Pars Orbitalis is shown in Figure 27d. It is visible that the pairs of tasks ADAS and TOTAL, RECOG and MMSE, do not influence each other. Notice that coincidentally this ROI shows a symmetric relationship among tasks. Finally, the Left Superior Parietal (Figure 27e) presents sparser relationships among the tasks.

These results agree with findings in the literature. For example, it is known that the Left gray matter suffers greater loss than its symmetric counterpart in the presence of Alzheimer (THOMPSON *et al.*, 2003). It is also known that the left hemisphere as a whole is impacted by AD, especially the Temporal and Parietal areas (THOMPSON *et al.*, 2001; CANU *et al.*, 2011). In this case, GAMTL could find a stable solution where the ROIs with the most transference activity are known to be related to the progression of Alzheimer's Disease.

6.4 FINAL REMARKS

This chapter provided the empirical results of the GAMTL formulation in a set of experiments. Section 6.1 investigated the performance of GAMTL with a varying number of samples available for training, in an artificial dataset with known structure of relationship in the parameters of the tasks. Compared to state-of-the-art contenders, GAMTL gains were enhanced when the number of samples available for training was small. Section 6.2 compared the performance of GAMTL in the prediction of cognitive scores related with Alzheimer's Disease progression. GAMTL showed improved performance when predicting the cognitive scores according to the RMSE and MSE metrics. Moreover, the method also estimated an interpretable structure of how regions of interest in the brain are related through the cognitive scores. Section 6.3 tested the robustness of the parameters estimated by GAMTL formulation with relation to noise present in the training data, and also in the settings of the hyper-parameters of the model. When applied to the same dataset of Section 6.2, the structure of transference that was estimated by GAMTL and encoded how ROIs (groups of features) were related when considering the different cognitive scores (modeled as multiple tasks), was backed by the medical literature.

7 CONCLUSIONS AND FUTURE DIRECTIONS

7.1 SUMMARY

This thesis explored how Multi-Task Learning (MTL) is used to tackle Machine Learning (ML) problems in the presence of multiple tasks. Chapter 1 introduced the presence of multiple ML tasks and showed some naive solutions to handle the new challenges. Chapter 2 presented MTL, an approach to handle multiple tasks where ML models for all tasks are trained jointly to promote gains of performance on the tasks by transferring information among related tasks. Chapter 3 exposed how sparse models play a key role in MTL by allowing us to impose a priori knowledge in the cost function using regularization terms. Chapter 4 presented the class of proximal methods as a means to solve optimization problems with non-smoothness in the objective function or constraints of a convex optimization problem.

The main contribution of this work was depicted in Chapter 5, with the proposal of Group Asymmetric Multi-Task Learning (GAMTL), an MTL model that circumvents important limitations of the current MTL literature, namely:

- *global transference*: transference of information between two tasks involve all features; and
- *symmetric relationships*: transference between two tasks occurs both ways with the same intensity..

The distinguished flexibility of GAMTL's estimated transference structure among tasks leads to the following main properties:

- Tasks may be related considering only a few groups of features.
- Groups of features can play distinct roles in different groups of related tasks.
- Transference is asymmetric: the influence from a task t on a task s may be different from the influence of s on t.

The method allows tasks to transfer in a highly flexible way and learns a rich set of local relationship structures, which also admit interpretable parsing. Chapter 6 provided the empirical results of the thesis, and GAMTL was tested in two experimental settings:

Sample complexity: this setting measured the performance of GAMTL with a varying number of samples available during the training phase, and compared the performance of GAMTL with related contenders in an synthetic setting, measuring its predictive performance and ability to recover the true structure of the relationships among the tasks.

Prediction of cognitive scores related to AD progression: GAMTL was contrasted with state-of-the-art contenders on the problem of predicting cognitive scores to estimate Alzheimer's Disease progression. In this experiment, each cognitive score was taken as a learning task. GAMTL showed the best results to predict the majority of the scores while unveiling relationships among them that have been found in the medical literature.

The same chapter included a stability selection analysis, which assessed the robustness of the estimated parameters of GAMTL with relation to noise in the data and the values of hyper-parameters. The stability selection procedure applied on GAMTL parameters highlighted statistically robust relationships among cognitive scores, conditioned on regions of the brain taken as groups of features.

7.2 IMPLICATIONS FOR THE MTL COMMUNITY

Among the many conclusions stated in the body of the chapters, we highlight a few that we believe to be the most relevant for the MTL community.

Structural estimation: Choosing methods that estimate how tasks are related is preferred over methods that trust on strong a priori assumptions. But keep in mind that the way we model how tasks are related needs to be flexible in order to match the information found in data. GAMTL was demonstrated to properly explore the premises that tasks are related considering subsets of features that can overlap, in an asymmetric fashion. These are two degrees of flexibility added in the way tasks can be related, with a low computational complexity overhead.

Transference among tasks can be complex: as demonstrated in this work, transference can occur in multiple groups of features, possibly overlapping, in an asymmetric fashion. There are potentially other ways to capture nuances of relationships among tasks, but this was a direction with positive results and low computational overhead. It is important to consider flexibility in the transference among tasks to avoid negative transference. The flexibility of transference added by GAMTL helped the method to almost nullify the effect of negative transference in a variety of experiments, while improving predictive performance.

MTL may be even more advantageous in the context of scarce data: the gains of MTL methods when compared with STL counterparts are more expressive when a smaller number of data points is available for training. Notice that this is a characteristic of many real-world applications, as many scenarios in the daily life suffer from *small data*.

Proximal methods are suited for non-convex regularization: Proximal methods are effective solutions to a variety of formulations to promote regularization. They are adequate to solve non-smooth problems that are widely common in MTL; they provide fast convergence rates and are easy to parallelize.

7.3 FUTURE DIRECTIONS

- Extending GAMTL to the heterogeneous features case, where features may not be the same for all tasks. It may be possible that with the current formulation, GAMTL is able to handle such scenarios if we consider that all tasks share the same domain composed of features of all tasks, and the dataset of each task would be pre-processed to set to zero the features that belong to other tasks but are not present in this task. In this case, there is no need to modify the formulation or to impose new constraints on the problem. It would be the easiest path forward, requiring only to find a suitable application and experimenting with the method. In case of success, it is possible to modify the formulation adding more constraints to block transference on groups of features that do not exist in a given task. This may lead to improvements on the optimization procedure.

- The Group LASSO requires a definition of how features are correlated in groups. In case the group information is not known beforehand, it is possible to consider each feature as a singleton group. But a path forward would be searching the group information automatically, without requiring this information from the user. This would expand the applicability of this method to new domains.

- The assumption that the parameters of the tasks are linearly dependent among themselves leads to an easy interpretation of how the tasks are related, but implicates that the features must lie in the same domain. We are going to investigate how the advantages brought by GAMTL can be expanded to the heterogeneous case, where features are not on the same domain for all tasks.

- Extending GAMTL to non-linear models seems to be a relevant initiative. Multiple Kernel Learning (GÖNEN; ALPAYDN, 2011), tree-based models or even Neural Networks may benefit from GAMTL's assumptions.

- The multi-convex nature of the formulation of GAMTL is practical and handy to combine multiple regularization terms in ML and Statistics. Solving these problems usually requires a reformulation of the optimization problem into a set of smaller convex problems with an easier solution, and then applying an alternating minimization procedure. Alternating procedures are easy to use, are simple to parallelize, and converge for many cases, but still are subject to local optima. This work lacks more theoretical analyses about the convergence of such schemes. For example, what properties are desired to achieve faster convergence or find better local optima in problems like that? Does the number of partitions of the original problem affect the convergence rate of the alternating procedure or the quality of the local minima? Is it preferable to have a few partitions of the problem or lots of them?

BIBLIOGRAPHY

AKIBA, T.; SANO, S.; YANASE, T.; OHTA, T.; KOYAMA, M. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2019. Cited 2 times on pages 76 and 80.

ALLENBY, G. M.; ROSSI, P. E. Marketing models of consumer heterogeneity. *Journal of Econometrics*, v. 89, n. 1, p. 57–78, 1998. ISSN 0304-4076. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0304407698000554>. Cited on page 34.

ALY, M. Survey on multiclass classification methods. *Neural Networks*, p. 1–9, 2005. Cited on page 31.

Alzheimer's Disease International. *World Alzheimer Report*. [S.l.], 2018. Disponível em: <https://www.alzint.org/resource/world-alzheimer-report-2018/>. Cited on page 78.

ANDO, R. K.; ZHANG, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, JMLR.org, v. 6, p. 1817–1853, December 2005. ISSN 1532-4435. Cited on page 21.

ARGYRIOU, A.; EVGENIOU, T.; PONTIL, M. Convex multi-task feature learning. *Machine Learning*, v. 73, n. 3, p. 243–272, 2008. Cited 2 times on pages 29 and 35.

BECK, A.; TEBOULLE, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, SIAM, v. 2, n. 1, p. 183–202, 2009. Cited 4 times on pages 23, 58, 59, and 69.

BEZDEK, J. C.; HATHAWAY, R. J. Some notes on alternating optimization. In: PAL, N. R.; SUGENO, M. (Ed.). *Advances in Soft Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 288–300. ISBN 978-3-540-45631-5. Cited on page 69.

BICKEL, S.; BOGOJESKA, J.; LENGAUER, T.; SCHEFFER, T. Multi-task learning for hiv therapy screening. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2008. (ICML '08), p. 56–63. ISBN 9781605582054. Disponível em: <https://doi.org/10.1145/1390156.1390164>. Cited 2 times on pages 27 and 28.

BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738. Cited on page 16.

BORCHANI, H.; VARANDO, G.; BIELZA, C.; LARRAÑAGA, P. A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery*, v. 5, n. 5, p. 216–233, 2015. Disponível em: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1157>. Cited on page 31.

BOYD, S.; PARIKH, N.; CHU, E.; PELEATO, B.; ECKSTEIN, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method

of Multipliers. *Foundations and Trends in Machine Learning*, v. 3, n. 1, 2011. Cited 6 times on pages 23, 60, 61, 62, 70, and 72.

CABEZUDO, M. A. S.; INÁCIO, M.; RODRIGUES, A. C.; CASANOVA, E.; SOUSA, R. F. de. Natural language inference for portuguese using bert and multilingual information. In: *Computational Processing of the Portuguese Language: 14th International Conference, PROPOR 2020, Evora, Portugal, March 2–4, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2020. p. 346–356. ISBN 978-3-030-41504-4. Disponível em: <https://doi.org/10.1007/978-3-030-41505-1_33>. Cited on page 32.

CAI, T. T.; WANG, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, v. 57, n. 7, p. 4680–4688, 2011. Cited on page 46.

CANDES, E.; ROMBERG, J.; TAO, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, v. 52, n. 2, p. 489–509, 2006. Cited on page 46.

CANU, E.; MCLAREN, D. G.; FITZGERALD, M. E.; BENDLIN, B. B.; ZOCCATELLI, G.; ALESSANDRINI, F.; PIZZINI, F. B.; RICCIARDI, G. K.; BELTRAMELLO, A.; JOHNSON, S. C.; FRISONI, G. B. Mapping the structural brain changes in Alzheimer's disease: The independent contribution of two imaging modalities. *Journal of Alzheimer's Disease*, v. 26, n. Suppl 3, p. 263–274, 2011. ISSN 1387-2877. 00034. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3267543/>. Cited on page 91.

CARUANA, R. Multitask learning. *Machine Learning*, v. 28, n. 1, July 1997. Cited 7 times on pages 19, 20, 25, 27, 28, 30, and 32.

CASELLA, G.; GHOSH, M.; GILL, J.; KYUNG, M. Penalized Regression, Standard Errors, and Bayesian Lassos. *Bayesian Analysis*, International Society for Bayesian Analysis, v. 5, n. 2, p. 369 – 411, 2010. Disponível em: <https://doi.org/10.1214/10-BA607>. Cited on page 52.

CHEN, J.; LIU, J.; YE, J. Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data*, Association for Computing Machinery, New York, NY, USA, v. 5, n. 4, February 2012. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2086737.2086742>. Cited 2 times on pages 29 and 36.

CHEN, J.; ZHOU, J.; YE, J. 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*. New York, NY, USA: Association for Computing Machinery, 2011. (KDD '11), p. 42–50. ISBN 9781450308137. Disponível em: <https://doi.org/10.1145/2020408.2020423>. Cited 2 times on pages 29 and 36.

CILIBERTO, C.; RUDI, A.; ROSASCO, L.; PONTIL, M. Consistent multitask learning with nonlinear output relations. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/b24d516bb65a5a58079f0f3526c87c57-Paper.pdf>. Cited on page 30.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://aclanthology.org/N19-1423>. Cited on page 32.

DONOHO, D. Compressed sensing. *IEEE Transactions on Information Theory*, v. 52, n. 4, p. 1289–1306, 2006. Cited on page 46.

EVGENIOU, T.; PONTIL, M. Regularized multi–task learning. In: ACM. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004. (KDD '04), p. 109–117. ISBN 1-58113-888-1. Disponível em: <http://doi.acm.org/10.1145/1014052.1014067>. Cited 6 times on pages 21, 27, 29, 30, 34, and 72.

FELDMAN, S.; GUPTA, M. R.; FRIGYIK, B. A. Revisiting stein's paradox: Multi-task averaging. *Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 3441–3482, 2014. Cited on page 30.

GONÇALVES, A. R.; ZUBEN, F. J. V.; BANERJEE, A. Multi-task sparse structure learning with gaussian copula models. *Journal of Machine Learning Research*, v. 17, n. 33, p. 1–30, 2016. Disponível em: <http://jmlr.org/papers/v17/15-215.html>. Cited 7 times on pages 21, 26, 27, 28, 29, 40, and 80.

GONG, P.; YE, J.; ZHANG, C. Robust multi-task feature learning. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2012. (KDD '12), p. 895–903. ISBN 9781450314626. Disponível em: <https://doi.org/10.1145/2339530.2339672>. Cited 2 times on pages 29 and 36.

GORSKI, J.; PFEUFFER, F.; KLAMROTH, K. Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, v. 66, n. 3, p. 373–407, 2007. Disponível em: <https://EconPapers.repec.org/RePEc:spr:mathme:v:66:y:2007:i:3:p:373-407>. Cited 2 times on pages 68 and 69.

GÖNEN, M.; ALPAYDN, E. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, JMLR.org, v. 12, p. 2211–2268, July 2011. ISSN 1532-4435. Cited on page 94.

HAN, L.; ZHANG, Y. Learning multi-level task groups in multi-task learning. In: . [s.n.], 2015. Disponível em: <https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9510>. Cited 2 times on pages 20 and 37.

HAN, L.; ZHANG, Y. Multi-stage multi-task learning with reduced rank. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2016. (AAAI'16), p. 1638–1644. Cited on page 36.

HAN, S.; LIAO, X.; CARIN, L. Cross-domain multitask learning with latent probit models. In: *Proceedings of the 29th International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2012. (ICML'12), p. 363–370. ISBN 9781450312851. Cited on page 26.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics). Cited on page 16.

HASTIE, T.; TIBSHIRANI, R.; WAINWRIGHT, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*. [S.l.]: Chapman Hall/CRC, 2015. ISBN 1498712169. Cited 3 times on pages 45, 47, and 48.

HE, Z.; YU, W. Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, v. 34, n. 4, p. 215 – 225, 2010. ISSN 1476-9271. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1476927110000502>. Cited on page 85.

HERNÁNDEZ-LOBATO, D.; HERNÁNDEZ-LOBATO, J. M. Learning feature selection dependencies in multi-task learning. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2013. p. 746–754. Cited on page 30.

HESKES, T. Empirical bayes for learning to learn. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (ICML '00), p. 367–374. ISBN 1558607072. Cited on page 34.

JACOB, L.; OBOZINSKI, G.; VERT, J.-P. Group Lasso with Overlap and Graph Lasso. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2009. p. 433–440. Cited 8 times on pages 10, 29, 34, 48, 49, 50, 51, and 76.

JACOB, L.; VERT, J.-p.; BACH, F. Clustered multi-task learning: A convex formulation. In: KOLLER, D.; SCHUURMANS, D.; BENGIO, Y.; BOTTOU, L. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009. v. 21. Disponível em: <https://proceedings.neurips.cc/paper/2008/file/fccb3cdc9acc14a6e70a12f74560c026-Paper.pdf>. Cited 3 times on pages 30, 38, and 49.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. USA: Prentice-Hall, Inc., 1988. ISBN 013022278X. Cited on page 18.

JALALI, A.; SANGHAVI, S.; RUAN, C.; RAVIKUMAR, P. K. A dirty model for multi-task learning. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2010. p. 964–972. Cited 5 times on pages 21, 27, 29, 30, and 36.

JAWANPURIA, P.; NATH, J. S. A convex feature learning formulation for latent task structure discovery. In: *Proceedings of the 29th International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2012. (ICML'12), p. 1531–1538. ISBN 9781450312851. Cited 2 times on pages 29 and 37.

KANG, Z.; GRAUMAN, K.; SHA, F. Learning with whom to share in multi-task feature learning. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2011. p. 521–528. Cited 5 times on pages 27, 29, 35, 76, and 80.

KHACHATURIAN, Z. Diagnosis of alzheimer's disease. *Archives of Neurology*, American Medical Association, v. 42, n. 11, p. 1097–1105, 1985. ISSN 0003-9942. Cited on page 77.

KOLAR, M.; LAFFERTY, J.; WASSERMAN, L. Union support recovery in multi-task learning. *Journal of Machine Learning Research*, v. 12, n. 72, p. 2415–2435, 2011. Disponível em: <http://jmlr.org/papers/v12/kolar11a.html>. Cited 3 times on pages 20, 38, and 49.

KUMAR, A.; DAUMÉ, H. Learning task grouping and overlap in multi-task learning. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2012. Cited 4 times on pages 27, 29, 34, and 80.

LAZARIC, A.; GHAVAMZADEH, M. Bayesian multi-task reinforcement learning. In: *Proceedings of the 27th International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010. (ICML'10), p. 599–606. ISBN 9781605589077. Cited on page 26.

LECUN, Y.; CORTES, C. MNIST handwritten digit database. 2010. Disponível em: <http://yann.lecun.com/exdb/mnist/>. Cited on page 31.

LEE, G.; YANG, E.; HWANG, S. J. Asymmetric Multi-task Learning Based on Task Relatedness and Loss. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2016. p. 230–238. Cited 7 times on pages 21, 27, 28, 29, 40, 65, and 79.

LI, H.; LIAO, X.; CARIN, L. Multi-task reinforcement learning in partially observable stochastic environments. *Journal of Machine Learning Research*, JMLR.org, v. 10, p. 1131–1186, June 2009. ISSN 1532-4435. Cited on page 26.

LI, J.; SUN, A.; HAN, J.; LI, C. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, v. 34, n. 1, p. 50–70, 2022. Cited on page 32.

LIAO, X.; CARIN, L. Radial basis function network for multi-task learning. In: WEISS, Y.; SCHöLKOPF, B.; PLATT, J. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2005. v. 18. Disponível em: <https://proceedings.neurips.cc/paper/2005/file/c7558e9d1f956b016d1fdba7ea132378-Paper.pdf>. Cited on page 27.

LIU, H.; GEGOV, A.; COCEA, M. *Rule Based Systems for Big Data: A Machine Learning Approach*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319236954. Cited on page 16.

LIU, J.; JI, S.; YE, J. Multi-task feature learning via efficient l2,1-norm minimization. In: *Conference on Uncertainty in Artificial Intelligence*. [S.l.: s.n.], 2009. p. 339–348. Cited 3 times on pages 30, 38, and 85.

LIU, W.; SHEN, X.; WANG, H.; TSANG, I. The emerging trends of multi-label learning. *ArXiv*, abs/2011.11197, 2020. Cited on page 31.

LIU, X.; CAO, P.; WANG, J.; KONG, J.; ZHAO, D. Fused group lasso regularized multi-task feature learning and its application to the cognitive performance prediction of alzheimer's disease. *Neuroinformatics*, v. 17, n. 2, p. 271–294, April 2019. ISSN 1559-0089. Disponível em: <https://doi.org/10.1007/s12021-018-9398-5>. Cited 3 times on pages 20, 21, and 39.

LIU, X.; GONÇALVES, A. R.; CAO, P.; ZHAO, D.; BANERJEE, A. Modeling Alzheimer's Disease Cognitive Scores Using Multi-Task Sparse Group Lasso. *Computerized Medical Imaging and Graphics*, v. 66, p. 100 – 114, 2018. Cited 5 times on pages 20, 39, 78, 79, and 85.

LIU, Z.; JIANG, F.; HU, Y.; SHI, C.; FUNG, P. *NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging*. ArXiv, 2021. Disponível em: <https://arxiv.org/abs/2112.00405>. Cited on page 32.

MAGALHÃES, S.; HAMDAN, A. The rey auditory verbal learning test: Normative data for the brazilian population and analysis of the influence of demographic variables. *Psychology Neuroscience*, v. 3, June 2010. Cited on page 79.

MEINSHAUSEN, N. Relaxed lasso. *Computational Statistics Data Analysis*, v. 52, n. 1, p. 374–393, 2007. ISSN 0167-9473. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167947306004956>. Cited on page 51.

MEINSHAUSEN, N.; BüHLMANN, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, v. 72, n. 4, p. 417–473, 2010. Disponível em: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00740.x>. Cited on page 85.

MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. [S.l.]: The MIT Press, 2012. ISBN 0262018020. Cited 2 times on pages 16 and 18.

NEGAHBAN, S.; WAINWRIGHT, M. J. Phase transitions for high-dimensional joint support recovery. In: KOLLER, D.; SCHUURMANS, D.; BENGIO, Y.; BOTTOU, L. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009. v. 21. Disponível em: <https://proceedings.neurips.cc/paper/2008/file/aa169b49b583a2b5af89203c2b78c67c-Paper.pdf>. Cited 2 times on pages 37 and 39.

NESTEROV, Y. A method for solving the convex programming problem with convergence rate o(1/k^2).*Proceedings of the USSR Academy of Sciences*, *v.* 269, *p.*543 − −547, 1983.*Cited* 2 *times on pages* 23*and* 59.

NESTEROV, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, Springer-Verlag, Berlin, Heidelberg, v. 103, n. 1, p. 127–152, May 2005. ISSN 0025-5610. Disponível em: <https://doi.org/10.1007/s10107-004-0552-5>. Cited on page 53.

OBOZINSKI, G. *Group Lasso with Overlaps: the Latent Group Lasso Approach*. 2011. Cited on page 48.

OBOZINSKI, G.; WAINWRIGHT, M. J.; JORDAN, M. I. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 39, n. 1, p. 1–47, 2011. ISSN 00905364, 21688966. Disponível em: <http://www.jstor.org/stable/29783630>. Cited on page 37.

OLIVEIRA, S. H. G.; GONÇALVES, A. R.; ZUBEN, F. J. V. Group LASSO with asymmetric structure estimation for multi-task learning. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 2019. p. 3202–3208. Disponível em: <https://doi.org/10.24963/ijcai.2019/444>. Cited 8 times on pages 20, 21, 22, 26, 28, 29, 33, and 64.

OLIVEIRA, S. H. G.; GONÇALVES, A. R.; ZUBEN, F. J. V. Asymmetric multi-task learning with local transference. *ACM Transactions on Knowledge Discovery from Data*, Association for Computing Machinery, New York, NY, USA, v. 16, n. 5, Article no. 99,

pp. 1–30, 2022. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/3514252>. Cited 2 times on pages 27 and 64.

PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, p. 1345–1359, 2010. Cited on page 32.

PARIKH, N.; BOYD, S. Proximal algorithms. *Foundations and Trends in Optimization*, Now Publishers Inc. Hanover, MA, USA, v. 1, n. 3, p. 127–239, 2014. Cited 3 times on pages 55, 56, and 57.

PARK, T.; CASELLA, G. The bayesian lasso. *Journal of the American Statistical Association*, Taylor Francis, v. 103, n. 482, p. 681–686, 2008. Disponível em: <https://doi.org/10.1198/016214508000000337>. Cited on page 52.

POLIAK, A. A survey on recognizing textual entailment as an NLP evaluation. In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Online: Association for Computational Linguistics, 2020. p. 92–109. Disponível em: <https://aclanthology.org/2020.eval4nlp-1.10>. Cited on page 32.

POLSON, N. G.; SCOTT, J. G.; WILLARD, B. T. Proximal Algorithms in Statistics and Machine Learning. *Statistical Science*, Institute of Mathematical Statistics, v. 30, n. 4, p. 559 – 581, 2015. Disponível em: <https://doi.org/10.1214/15-STS530>. Cited 4 times on pages 10, 55, 56, and 57.

PONG, T. K.; TSENG, P.; JI, S.; YE, J. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, Society for Industrial and Applied Mathematics, USA, v. 20, n. 6, p. 3465–3489, December 2010. ISSN 1052-6234. Disponível em: <https://doi.org/10.1137/090763184>. Cited on page 36.

REIMERS, N.; GUREVYCH, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 3982–3992. Disponível em: <https://aclanthology.org/D19-1410>. Cited on page 32.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377042787901257>. Cited on page 87.

RUDER, S. An overview of multi-task learning in deep neural networks. *ArXiv*, 2017. Cited 2 times on pages 30 and 32.

SHEN, X.; DIAMOND, S.; UDELL, M.; GU, Y.; BOYD, S. Disciplined multi-convex programming. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2017. p. 895–900. Cited on page 69.

SOROWER, M. S. *A Literature Survey on Algorithms for Multi-Label Learning*. [S.l.], 2010. Cited on page 31.

SOUZA, F.; NOGUEIRA, R. F.; LOTUFO, R. de A. Portuguese named entity recognition using BERT-CRF. *CoRR: Computing Research Repository*, abs/1909.10649, 2019. Disponível em: <http://arxiv.org/abs/1909.10649>. Cited on page 32.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Second. [S.l.]: The MIT Press, 2018. Cited on page 16.

THOMPSON, P. M.; HAYASHI, K. M.; ZUBICARAY, G. de; JANKE, A. L.; ROSE, S. E.; SEMPLE, J.; HERMAN, D.; HONG, M. S.; DITTMER, S. S.; DODDRELL, D. M.; TOGA, A. W. Dynamics of gray matter loss in alzheimer's disease. *Journal of Neuroscience*, Society for Neuroscience, v. 23, n. 3, p. 994–1005, 2003. ISSN 0270-6474. Disponível em: <https://www.jneurosci.org/content/23/3/994>. Cited on page 91.

THOMPSON, P. M.; MEGA, M. S.; WOODS, R. P.; ZOUMALAN, C. I.; LINDSHIELD, C. J.; BLANTON, R. E.; MOUSSAI, J.; HOLMES, C. J.; CUMMINGS, J. L.; TOGA, A. W. Cortical Change in Alzheimer's Disease Detected with a Disease-specific Population-based Brain Atlas. *Cerebral Cortex*, v. 11, n. 1, p. 1–16, January 2001. ISSN 1047-3211. Disponível em: <https://doi.org/10.1093/cercor/11.1.1>. Cited on page 91.

THRUN, S.; O'SULLIVAN, J. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In: *International Conference on Machine Learning*. [S.l.: s.n.], 1996. p. 489–497. Cited on page 30.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, [Royal Statistical Society, Wiley], v. 58, n. 1, p. 267–288, 1996. ISSN 00359246. Disponível em: <http://www.jstor.org/stable/2346178>. Cited 3 times on pages 46, 47, and 76.

TIBSHIRANI, R.; SAUNDERS, M.; ROSSET, S.; ZHU, J.; KNIGHT, K. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, v. 67, n. 1, p. 91–108, 2005. Disponível em: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00490.x>. Cited 2 times on pages 39 and 51.

TROPP, J. A.; GILBERT, A. C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, v. 53, n. 12, p. 4655–4666, 2007. Cited on page 46.

VANDENHENDE, S.; GEORGOULIS, S.; GANSBEKE, W. V.; PROESMANS, M.; DAI, D.; GOOL, L. V. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021. Cited 3 times on pages 19, 20, and 32.

VOGT, J. E.; ROTH, V. The group-lasso: 1, regularization versus 1,2 regularization. In: GOESELE, M.; ROTH, S.; KUIJPER, A.; SCHIELE, B.; SCHINDLER, K. (Ed.). *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 252–261. ISBN 978-3-642-15986-2. Cited 2 times on pages 38 and 49.

WANG, H.; NIE, F.; HUANG, H.; KIM, S.; NHO, K.; RISACHER, S. L.; SAYKIN, A. J.; SHEN, L.; INITIATIVE, F. the A. D. N. Identifying Quantitative Trait Loci via Group-Sparse Multitask Regression and Feature Selection: An Imaging Genetics Study of the ADNI Cohort. *Bioinformatics*, v. 28, n. 2, p. 229–237, 2012. Disponível em: <http://dx.doi.org/10.1093/bioinformatics/btr649>. Cited on page 39.

XU, D.; SHI, Y.; TSANG, I. W.; ONG, Y.-S.; GONG, C.; SHEN, X. Survey on multi-output learning. *IEEE Transactions on Neural Networks and Learning Systems*, v. 31, n. 7, p. 2409–2429, 2020. Cited on page 31.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v. 16, n. 3, p. 645–678, 2005. Cited on page 18.

XU, X.; GHOSH, M. Bayesian Variable Selection and Estimation for Group Lasso. *Bayesian Analysis*, International Society for Bayesian Analysis, v. 10, n. 4, p. 909 – 936, 2015. Disponível em: <https://doi.org/10.1214/14-BA929>. Cited 2 times on pages 48 and 52.

XU, Y.; YIN, W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences [electronic only]*, v. 6, July 2013. Cited on page 69.

YUAN, M.; LIN, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, v. 68, p. 49–67, 2006. Cited 2 times on pages 37 and 49.

ZHANG, J.; ZHANG, C. Multitask bregman clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 24, n. 1, p. 655–660, July 2010. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/7674>. Cited on page 26.

ZHANG, M.-L.; ZHOU, Z.-H. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, v. 26, n. 8, p. 1819–1837, 2014. Cited on page 31.

ZHANG, X.; ZHANG, X. Smart multi-task bregman clustering and multi-task kernel clustering. In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2013. (AAAI'13), p. 1034–1040. Cited on page 26.

ZHANG, Y.; YANG, Q. A survey on multi-task learning. *CoRR: Computing Research Repository*, abs/1707.08114, 2017. Disponível em: <http://arxiv.org/abs/1707.08114>. Cited 10 times on pages 19, 20, 21, 22, 25, 26, 29, 33, 36, and 67.

ZHANG, Y.; YEUNG, D.-Y. A convex formulation for learning task relationships in multi-task learning. In: *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, USA: AUAI Press, 2010. (UAI'10), p. 733–742. ISBN 9780974903965. Cited on page 21.

ZHANG, Y.; YEUNG, D.-Y. A convex formulation for learning task relationships in multi-task learning. In: *Conference on Uncertainty in Artificial Intelligence*. [S.l.: s.n.], 2010. p. 733–742. Cited 4 times on pages 21, 29, 40, and 80.

ZHANG, Y.; YEUNG, D.-Y. Multi-task learning in heterogeneous feature spaces. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 25, n. 1, p. 574–579, August 2011. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/7909>. Cited on page 26.

ZHONG, S.; PU, J.; JIANG, Y.-G.; FENG, R.; XUE, X. Flexible multi-task learning with latent task grouping. *Neurocomputing*, v. 189, p. 179–188, 2016. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231216000035>. Cited on page 30.

ZHOU, J.; CHEN, J.; YE, J. Clustered multi-task learning via alternating structure optimization. In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2011. (NIPS'11), p. 702–710. ISBN 9781618395993. Cited 2 times on pages 21 and 34.

ZHOU, J.; YUAN, L.; LIU, J.; YE, J. A multi-task learning formulation for predicting disease progression. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2011. (KDD '11), p. 814–822. ISBN 9781450308137. Disponível em: <https://doi.org/10.1145/2020408.2020549>. Cited on page 78.

ZHOU, Q.; ZHAO, Q. Flexible clustered multi-task learning by learning representative tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 38, n. 2, p. 266–278, February 2016. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2015.2452911>. Cited on page 30.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, v. 109, p. 43–76, 2021. Cited on page 32.

ZOU, H. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, v. 101, p. 1418–1429, 2006. Disponível em: <https://EconPapers.repec.org/RePEc:bes:jnlasa:v:101:y:2006:p:1418-1429>. Cited 2 times on pages 51 and 70.