Multitask and Transfer Learning for Multi-Aspect Data

Bernardino Romera Paredes

UCL

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy of **University College London**.

I, Bernardino Romera Paredes, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Supervised learning aims to learn functional relationships between inputs and outputs. Multitask learning tackles supervised learning tasks by performing them simultaneously to exploit commonalities between them. In this thesis, we focus on the problem of eliminating negative transfer in order to achieve better performance in multitask learning. We start by considering a general scenario in which the relationship between tasks is unknown. We then narrow our analysis to the case where data are characterised by a combination of underlying aspects, e.g., a dataset of images of faces, where each face is determined by a person's facial structure, the emotion being expressed, and the lighting conditions. In machine learning there have been numerous efforts based on multilinear models to decouple these aspects but these have primarily used techniques from the field of unsupervised learning. In this thesis we take inspiration from these approaches and hypothesize that supervised learning methods can also benefit from exploiting these aspects. The contributions of this thesis are as follows:

1. A multitask learning and transfer learning method that avoids negative transfer when there is no prescribed information about the relationships between tasks.
2. A multitask learning approach that takes advantage of a lack of overlapping features between known groups of tasks associated with different aspects.
3. A framework which extends multitask learning using multilinear algebra, with the aim of learning tasks associated with a combination of elements from different aspects.
4. A novel convex relaxation approach that can be applied both to the suggested framework and more generally to any tensor recovery problem.

Through theoretical validation and experiments on both synthetic and real-world datasets, we show that the proposed approaches allow fast and reliable inferences. Furthermore, when performing learning tasks on an aspect of interest, accounting for secondary aspects leads to significantly more accurate results than using traditional approaches.

Acknowledgments

I would like to thank all those people who have supported me during the last three and a half years. Firstly, thank you to my supervisors, Massimiliano Pontil and Nadia Bianchi-Berthouze. I have been extremely lucky to have benefited from their knowledge, advice, motivation, and openness to consider even the most senseless of my suggestions. Thank you to Hane Aung whose interesting conversations have filled my mind with thoughts both research-related and beyond. I am indebted to Andrew McDonald for his research discussions and conversations, and for proof-reading this thesis. Thanks also to Chris Joyce for giving me the motivation to keep writing, and for diligently proof-reading parts of this thesis. Thanks to Andreas Maurer for sharing his ever-brilliant thoughts and for his contagious enthusiasm for learning theory. Thanks to Luca Baldasarre for his support in the first stages of my PhD. Thanks to the examiners of this thesis, David Barber and Tijl De Bie, as well as Mark Herbster who helped me prepare the viva, for their insightful comments and questions which made this a stronger thesis.

Thank you to the Emo & Pain Group: Aneesha Singh, Harry Griffin, Ana Tajadura, Temi Olugbade, Hongying Meng, and all the others at UCLIC. It has been a pleasure to share all those meals and conversations with you during the years.

I am very thankful to Cha Zhang, who gave me the opportunity to work in an amazing project at Microsoft Research. Thanks also to the people I met there, especially to Piotr Bilinski, Mar González, and Danai Koutra.

Finally, I would like to dedicate this thesis to my family: María del Mar, her parents and mine. Their love and support has been a constant source of encouragement to me during this time.

Contents

List of Figures

List of Tables

1. Introduction

Over the past decade the field of machine learning has undergone major developments, gradually maturing into a prominent area of computer science. This growth has led to more effective and efficient ways to automatically learn properties from data and to tackle different kinds of scenarios where learning is needed.

One of the scenarios which has been well studied is that of learning a set of supervised learning tasks with the assumption that they are somehow related. A reasonable approach is to learn these tasks at the same time so that the model can leverage the commonalities between them. This strategy is the core of multitask learning (MTL) [34, 33, 187, 19] and transfer learning (TL) [47, 110, 157, 164, 176]. MTL consists of learning tasks simultaneously, taking advantage of their commonalities, so that the accuracy of each task is increased, whereas in TL, knowledge is gained from a set of source tasks to improve the accuracy of new tasks. These frameworks have successfully been applied in many different scenarios, often providing improved performance over single task learning. Furthermore, when the data available for each task is scarce, single task learning may not be a valid alternative, as there are uniform lower bounds on the performance of single task learning, e.g. see [129].

One significant problem within MTL and TL is negative transfer. In a survey carried out on transfer learning [157], the authors pointed out both the importance of controlling negative transfer, and the little attention it has received. Negative transfer occurs whenever these frameworks not only fail to improve performance, but actually reduce it. An obvious example would be the naive application of multitask learning to a scenario in which tasks are not related in any way. More generally, negative transfer arises when the model used fails to reflect the specific kinds of relationships held among the tasks, making strong, or simply wrong, assumptions. Thus, all available information about the relationships between tasks should be used whenever possible.

In our research we follow an incremental approach regarding the information available about how tasks are related. Our research starts by studying negative transfer in the broad case where nothing is known about the relationships between tasks. In this sce-

nario, there may be closely related groups of tasks, as well as tasks which are not related at all. To avoid negative transfer, the model should not presume relations between tasks, however it should be able to positively transfer knowledge between tasks that are related.

In a second stage of this research, we consider using extra information that may be useful to uncover the relationships between tasks. Several approaches based on this key principle have previously been proposed and studied. They target the cases where information about relationships takes the form of clusters and hierarchies [58, 99, 201, 225]. Nevertheless, there are important real cases when these structures lack the necessary expressiveness to encode the relationships between tasks. One such instance arises when data can be looked at from several different angles. In this case, it would be preferable to learn representations of the data that are tailored to the learning tasks at hand, filtering out or leveraging other viewpoints of the data. Hereafter, we will refer to those viewpoints as aspects, and to the resultant datasets as multi-aspect datasets. In this thesis we focus on modelling relationships between tasks in multi-aspect data situations because they appear naturally in many real learning problems, yet they have received little attention.

1.1. Multi-aspect data

Multi-aspect datasets are those composed of instances that can be simultaneously categorized according to different category systems (aspects). Let us consider for example a dataset composed of many images of faces, as in Fig. 1.1. Each image is determined by the person who appears in it (aspect 1), the emotion (aspect 2) and perhaps other characteristics such as the viewpoint (aspect 3) and the illumination (aspect 4). Multi-aspect datasets are very common, as they arise whenever there are different conditions involved in the data gathering process. A representative example occurs when data are obtained from multiple people, as is the case in many social and medical scientific experiments, and in data gathered by companies about their clients.

In many supervised learning problems only one of those aspects is generally of interest, for example learning the set of affective states conveyed in the facial expression. The remaining aspects pose particular challenges in developing recognition models that focus on the specific aspect of interest. In such cases the biases caused by the secondary aspects could be significant if they are ignored or not treated carefully.

Human perception systems are able to decouple different aspects in data to extract in-

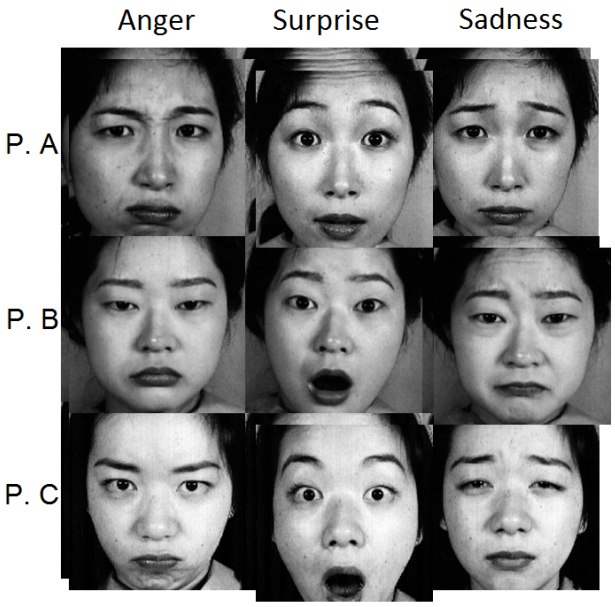

Figure 1.1.: Samples of a multi-aspect dataset, in which each instance (facial image) is characterised by the identity of the person (P.A, P.B, and P.C), and the emotion being expressed (Anger, Surprise, and Sadness). The images belong to the JAFFE dataset [124].

formation about an aspect of interest. Continuing the example, humans are able to recognize affective states in other people's faces regardless of the specific facial features of a person as well as other ambient conditions such as light and viewpoint.

In the field of machine learning several approaches have been proposed to deal with this kind of data, most of them based on multilinear algebra models. These use tensors, multidimensional generalizations of vectors and matrices, to extend concepts from linear algebra. Even though those approaches are useful to analyze multi-aspect datasets in an unsupervised learning setting, few multilinear approaches have been investigated in the context of supervised learning. Furthermore, they impose strong assumptions on the data [194]. The paucity of interest in this area may be due to the belief that the labels available in supervised learning problems are sufficient to discriminate useful information about the aspect of interest, hence the multi-aspect information can be discarded. In this thesis, we hypothesize that explicitly accounting for all aspects of the data in supervised learning problems leads to better performance.

1.2. Problem statement and contributions

A main goal of this thesis is to investigate how to exploit commonalities in learning tasks while avoiding negative transfer, placing a special emphasis on multi-aspect scenarios. We have taken an incremental approach, where we start by considering negative transfer in the general MTL and TL cases, where there is no side information about the relationships among tasks. A common case occurs when there are several unknown groups of related tasks, and tasks belonging to different groups have no relationship at all. We then focus on multi-aspect datasets and assume that the knowledge about groups of tasks is given, and that each group of tasks is associated to a different aspect of the data. Using the example of images of faces (Fig. 1.1), one could consider two groups of tasks, one for emotion recognition and another for identity recognition. We then consider an even more informative scenario in which each task is associated with a combination of elements of different aspects. In the example we would consider one task for each combination of identity/emotion.

This approach has led to the following contributions:

- Sparse coding multitask learning

  We present an extension of sparse coding to the problems of multitask and transfer learning which avoids negative transfer by imposing weak bindings between tasks. The central assumption of our learning method is that the task parameters are well approximated by sparse linear combinations of the atoms of a dictionary on a high or infinite dimensional space. This assumption, together with the large quantity of available data in the multitask and transfer learning settings, allows a principled choice of the dictionary. We provide bounds on the generalization error of this approach for both settings. Numerical experiments on one synthetic and two real datasets show the advantage of our method over the competing methods: single task learning, a previous method based on orthogonal and dense representation of the tasks and a related method learning task grouping. This scenario will be studied in Chapter 4, considering both MTL and TL settings.

- Decoupling of features

  We study the problem of learning a group of tasks related to one aspect of interest, using a group of auxiliary tasks related to a different aspect. In many applications, joint learning of unrelated tasks which use the same input data can be beneficial. The reason is that prior knowledge about which tasks are unrelated can lead to more sparse and more informative representations for each task, es-

sentially screening out idiosyncrasies of the data distribution. We propose a novel method which builds on a prior multitask methodology by favoring a shared low dimensional representation within each group of tasks. In addition, we impose a penalty on tasks from different groups which encourages the two representations to be orthogonal. We further discuss a condition which ensures convexity of the optimization problem and show that it can be solved by alternate minimization. We present experiments on synthetic and real data, which indicate that incorporating unrelated tasks can improve significantly over standard multitask learning methods. This will be presented in Chapter 5.

- Multilinear Multitask Learning

  We consider the scenario in which the information we have about how tasks are related can be described by linking each task with a combination of elements of aspects. We propose the use of multilinear algebra as a natural way to model such a set of related tasks. This framework can incorporate several prediction patterns (e.g. different emotions) and different data domains (e.g. different subjects, while performing different activities). Furthermore, it can perform zero-shot transfer learning, i.e. learning tasks even in cases where there are no training instances available. We present two learning methods. The first one is an adapted convex relaxation method used in the context of tensor completion. The second method is based on the Tucker decomposition and on alternating minimization. Experiments on synthetic and real data indicate that the multilinear approaches provide a significant improvement over other multitask learning methods. Overall, our second approach yields the best performance in all datasets. The resultant framework and experiments will be described in Chapter 6.

- Novel Convex Approach for Tensor Recovery

  The previous approach boils down to the challenging task of learning a low rank tensor, which is characterized by having simultaneously several low-dimensional structures. A prominent methodology for this problem is based on a generalization of trace norm regularization, which has been used extensively for learning low rank matrices, to the tensor setting. We highlight some limitations of this approach and propose an alternative convex relaxation on the Frobenius ball. We then describe a technique to solve the associated regularization problem, which builds upon the alternating direction method of multipliers. Experiments on one synthetic dataset and three real datasets indicate that the proposed method improves significantly over tensor trace norm regularization in terms of estimation

error, while remaining computationally tractable. This development will be presented in Chapter 7.

For each of these cases, we will describe practical situations where their application arises naturally. We will provide optimization methods to obtain good solutions and, whenever possible, we will establish links with previous approaches from the literature and provide theoretical arguments to justify when the use of the method is appropriate.

1.3. Timeliness of research

From a theoretical point of view, there are two points that make this research timely. Firstly, negative transfer has been identified as the primary general problem facing machine learning models that leverage some sort of knowledge transfer [157]. Secondly, part of the research focusing on multi-aspect data builds on multilinear models, but simultaneously tackles general problems within this field, particularly tensor recovery.

Tensors and multilinear models have gained a lot of popularity in the last years within machine learning and related fields. For example several books [70, 104, 108] and many tutorials and surveys [46, 68, 100, 122, 139] have recently been published on the topic, and this trend is expected to continue for two reasons. Firstly, many data problems involve the use of multi-aspect data. Some examples of this are context-aware recommendation [2, 97, 180, 167], statistical relational analysis [15, 16, 90, 148, 147, 149, 189], disentangling different effects on images [140, 194, 209, 208], and other applications in computer vision [117, 210], among others. Secondly, multilinear algebra is increasingly being used in machine learning approaches in traditional (non multi-aspect) learning problems, such as latent variable model estimation from higher order moments [5].

One of the direct applications of this research is for personalization of machine learning models, that is, tailoring or adapting a machine learning model to account for user specificity. This research topic has received much attention in recent years, being the central topic of several workshops in top-tier machine learning conferences such as NIPS. One important reason for this is its ubiquity in many application areas such as content (e.g. web, media, advertisement, products) recommendation, therapy personalization, and model calibration, among others. Companies are inclined to incorporate personalized machine learning models to tailor their products and services to customers, as by doing so they may gain a competitive advantage. This is also fostered by the cheap availability of processing power and the large amount of data being collected.

A specific area which can benefit from personalization is that of automatic affect recognition, particularly emotion recognition in natural, uncontrolled, settings. Automatic affect recognition from non-verbal behaviour (e.g. facial expressions or affective body expressions) needs to take into account contextual aspects. The manifest expressions are of course caused by the underlying affect, but a person's idiosyncratic tendencies are significant, as are environmental aspects. Hence, recognition performance could be improved by personalizing the affect recognition models in an appropriate way. For this reason, affect recognition datasets are often used to test the performance of the methods in this thesis.

In summary, multi-aspect data arise in many real scenarios and give rise to very challenging questions. Thus, we believe that the research presented here is very timely from both theoretical and practical viewpoints.

1.4. List of publications

The following publications were completed over the course of this thesis:

Conferences:

1. A. Maurer, M. Pontil, B. Romera-Paredes: An Inequality with Applications to Structured Sparsity and Multitask Dictionary Learning. Conference on Learning Theory, COLT 2014, Barcelona, Spain. [132].
2. B. Romera-Paredes, C. Zhang, Z. Zhang: Facial Expression Tracking from Head-Mounted, Partially Observing Cameras. IEEE International Conference on Multimedia & Expo, IEEE ICME 2014. Chengdu, China. [175].
3. B. Romera-Paredes, M. Pontil: A New Convex Relaxation for Tensor Completion. Neural Information Processing Systems, NIPS 2013. Lake Tahoe, USA. [174].
4. H. J. Griffin, M. S. H. Aung, B. Romera-Paredes, C. McLoughlin, G. McKeowny, W. Currany, N. Bianchi-Berthouze: Laughter Type Recognition from Whole Body Motion. Affective Computing and Intelligent Interaction, ACII 2013. Geneva, Switzerland. Best paper award. [69].
5. A. Maurer, M. Pontil, B. Romera-Paredes: Sparse coding for multitask and transfer learning. International Conference on Machine Learning, ICML 2013. Atlanta, USA. Best paper runner-up. (Authors contributed equally). [131].
6. B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze, M. Pontil: Multilinear Multitask Learning. International Conference on Machine Learning, ICML 2013.

Atlanta, USA. [171].

7. B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze: A One-Vs-One Classifier Ensemble with Majority Voting for Activity Recognition. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium. [170].
8. B. Romera-Paredes, H. Aung, M. Pontil, N. Bianchi-Berthouze, A.C.D.C. Williams, P. Watson: Transfer Learning to Account for Idiosyncrasy in Face and Body Expressions. Automatic Face and Gesture Recognition, IEEE FG 2013. Shangai, China. [172].
9. B. Romera-Paredes, A. Argyriou, N. Bianchi-Berthouze, M. Pontil: Exploiting Unrelated Tasks in Multi-Task Learning. Artificial Intelligence and Statistics, AISTATS 2012. La Palma, Spain. [168].

Workshops:

1. M.S.H. Aung, B. Romera-Paredes, A. Singh, S. Lim, N. Kanakam, A.C.D.C. Williams, N. Bianchi-Berthouze. Getting rid of pain-related behaviour to improve social and self perception: A Technology-Based Perspective. 14th International Workshop on Image and Audio Analysis for Multimedia Interactive Services, WIAMIS 2013. Paris, France. [13].
2. B. Romera-Paredes, A. Argyriou, A.C.D.C. Williams, N. Berthouze, M. Pontil, Automatic Recognition of Facial Expressions. 14th World Congress on Pain, IASP 2012. Milan, Italy. [169].
3. B. Romera-Paredes, M. Pontil, N. Bianchi-Berthouze: Leveraging Different Transfer Learning Assumptions: Shared Features, Hierarchical and Semi-Supervised. Challenges in Learning Hierarchical Models, NIPS Workshop 2011. Granada, Spain. [173].
4. H. Meng, B. Romera-Paredes, N. Bianchi-Berthouze: Emotion recognition by two view SVM_2K classifier on dynamic facial expression features. Automatic Face and Gesture Recognition, IEEE FG Workshop 2011. Santa Barbara, USA. [136].

Furthermore, the following two journal publications are under review and the last one is in preparation:

1. H. J. Griffin, M. S. H. Aung, B. Romera-Paredes, C. McLoughlin, G. McKeowny, W. Currany, N. Bianchi-Berthouze: Perception and Automatic Recognition of Laughter from Whole Body Motion: Continuous and Categorical Perspectives.

2. M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, A. Elkins, N. Tyler, P. Watson, A.C.D.C. Williams, M. Pantic, N. Berthouze: Detecting Chronic Pain-Related Expressions and Behaviours from Multimodal Naturalistic Data.
3. B. Romera-Paredes, M. S. H. Aung, M. Pontil, N. Bianchi-Berthouze: Multilinear Multitask Learning for Affect Recognition Across Subjects

This work has led to several awards: Best Paper Award at ACII 2013, the Best Paper Runner-up Prize at ICML 2013 and winner of a machine learning challenge on automatic recognition of human activity at ESANN 2013.

1.5. Outline

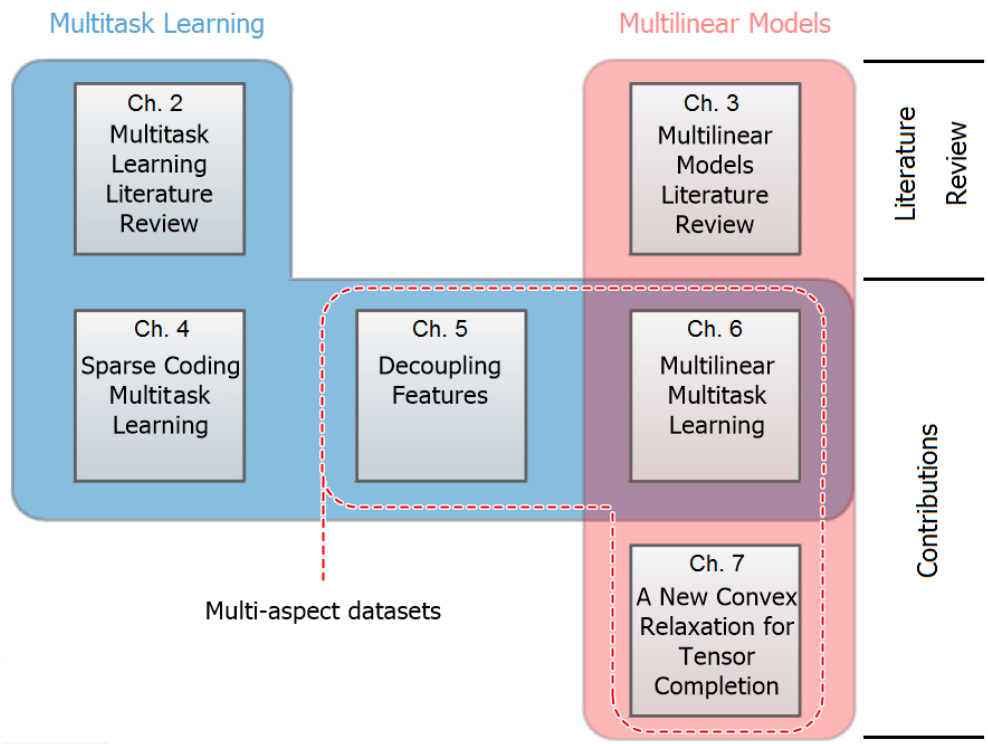

Figure 1.2.: Structure of this thesis.

In this introduction we have highlighted the two key concepts of this thesis, namely negative transfer and multi-aspect data. In the two following chapters, we explore the state of the art in those areas. In particular in Chapter 2, we review MTL approaches based on optimization, and in Chapter 3, we review multilinear models, as the predominant

models dealing with multi-aspect data. Following the literature reviews, we describe the contributions of the thesis, introduced in Section 1.2. We address sparse coding multitask learning in Chapter 4, decoupling of features in Chapter 5, multilinear multitask learning in Chapter 6, and rounding off the contributions, we describe the novel convex relaxation for tensor recovery in Chapter 7. Finally, we conclude the thesis with a summary of the achievements, and we propose research opportunities that arise from them. Fig. 1.2 summarizes the structure of this thesis.

2. Multitask Learning Literature Review

In this chapter we survey the multitask learning literature. This survey has two main purposes. Firstly, we present the state of the art in this field, as this provides the starting point of this thesis. In particular, we study models that vary the assumptions made on the data, and examine how they avoid negative transfer. Secondly, we highlight the absence of multitask learning approaches that can leverage multi-aspect datasets.

2.1. Introduction

Multitask learning (MTL) is a machine learning framework which considers several learning tasks together so that by taking advantage of the commonalities between the tasks, it can achieve a better performance than by solving them separately. Its application is particularly appropriate whenever there are a large number of related tasks to learn and/or when each task has only a small set of instances from which to learn. MTL is inspired by the fact that human beings are able to improve the learning process of a task if it is simultaneously addressed with other related tasks rather than assimilating it in isolation [21]. The literature in psychology [24, 1] show that, for example, the knowledge acquired by human beings while learning physical tasks facilitates the learning process of new motor skills.

Supervised learning problems are not usually isolated in real life, rather they are related to others. As an example we can consider the task of learning the gastronomical preferences of a customer, that is, given a description of a dish, we want to learn a function whose output will determine whether our customer will like that dish or not. This is a typical supervised learning task, however in a real situation we will have more than one customer and we will want to build one function per customer to predict her preferences. One reasonable assumption to make is that those functions share some commonalities, such as for example, a set of deciding ingredients. Therefore, the main objective of

MTL is to try to discover the commonalities between these functions in order to obtain a more accurate model. Through the course of this review, other examples will be introduced to illustrate different kinds of multitask relationships.

MTL was proposed in [19, 33, 34, 187] where neural networks, trees and other supervised leaning approaches were adapted to deal with MTL scenarios. The empirical results showed a clear improvement over the single task counterpart. Since then, a wide variety of MTL strategies have been proposed and have found applications in many different fields such as machine vision [162, 201, 223], natural language processing [203, 53], biometrics [193] and traffic flow forecasting [93], to name a few.

MTL has strong connections with other established machine learning frameworks, so that research and applications have benefited collectively in those fields. One of them is transfer learning [157] (also known as learning to learn [110] or inductive transfer [47, 164, 176]), whose aim is to extract knowledge from a set of source tasks to be applied in the learning process of target tasks. Therefore, it can be seen as an asymmetric modification of MTL where there is an explicit distinction between source and target tasks. The similarities between MTL and TL are such that several MTL approaches can directly be employed in a TL setting. A different framework is multivariate response learning [14, 28], also known as multi-output learning, which can be seen as a particular case of MTL when all tasks receive the same input data. This category also includes multi-label classification. MTL can also be seen as an extension of matrix completion (or collaborative filtering) [138, 190]. There, the objective is to predict entries in a matrix where only a small subset of them is known. Matrix completion has received increasing interest lately as a way of evaluating items through the ratings of other users. It can be considered as an MTL problem by treating users as tasks where the input data are indicators of the items and the labels are the actual values of the entries in the matrix (conversely, one can consider items to be tasks and input data to be indicators of the users).

In this literature review, we focus on MTL approaches based on regularization on the task parameters to encode the relationships assumed among the tasks. There are three important categories of MTL approaches which can be distinguished by the assumptions encoded in the regularizer:

- Modelling the proximity of the parameters of all tasks by making use of quadratic norms.
- Modelling structure sparsity by employing extensions of the Lasso.
- Modelling the assumption that all tasks share a common set of features by using

spectral norms.

As we will see, many MTL approaches are based on one of these frameworks, or build upon them introducing further constraints which incorporate more complex or more rich relationships among all tasks. Nevertheless, there are other methods that do not fall within any of those categories, they will also be briefly reviewed.

We introduce the notation which will allow us to formally define the three frameworks and study them. We then review extensions of these frameworks which consider more complex relations among tasks. Next, we address those approaches which fall outside the previous categories. Finally we conclude with a discussion of the approaches.

2.2. Notation

In the following we focus on regression problems but the generalization to classification problems is usually straightforward. In most cases we assume the following setting: the model we want to build has to be able to learn a set of T linear tasks $f_t(x) = \langle w_t, x \rangle$, $x, w_t \in \mathbb{R}^d$, $\forall t \in [T]$, where d is the dimensionality of the data, and $[N]$ denotes the set of natural numbers from 1 to N. In order to learn these tasks, a set of labeled instances is provided: $\{X^t, y^t\}$, $X^t \in \mathbb{R}^{d \times m_t}$, $y^t \in \mathbb{R}^{m_t}$, where X^t is the matrix composed of all the m_t instances provided for task t as columns and y^t is the vector containing the labels. With a slight abuse of notation, we denote $\boldsymbol{X} = \left\{X^1, X^2, \ldots, X^T\right\}$ and $\boldsymbol{Y} = \left\{y^1, y^2, \ldots, y^T\right\}$. Let W be the matrix composed of the T weight vectors w_t as columns, that is, $W = [w_1, \ldots, w_T]$. When making reference to the i-th element of w_t we use the notation $w_{i,t}$.

The notation $\|\cdot\|$ makes reference to norms of vectors or matrices. In the case of matrices we will consider the Frobenius norm, $\|\cdot\|_{\mathrm{Fr}}$, as well as other norms which will be introduced as they are needed.

We will denote as $\mathbf{1}$ the column vector whose all elements are 1. Its dimensionality should be clear from its context. The trace of a square matrix A is denoted as $\operatorname{trace}(A)$ and its determinant as $\det(A)$. Given any matrix $B \in \mathbb{R}^{d_1 \times d_2}$, its singular values are represented as $\sigma(B) = (\sigma_1(B), \sigma_2(B), \ldots, \sigma_K(B))$, where $K = \min\{d_1, d_2\}$ and $\sigma_1(B) \geq \ldots \geq \sigma_K(B) \geq 0$. We denote by $|B| \in \mathbb{R}^{d_1 \times d_2}$ as the matrix composed of the absolute values of the entries of B. Finally, $\operatorname{vec}(B) \in \mathbb{R}^{d_1 d_2}$ is obtained by stacking

the columns of B, that is $\operatorname{vec}(B) = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{d_2} \end{bmatrix}$.

2.3. MTL Regularizers

In this section we will consider MTL approaches which share the following optimization problem skeleton:

$$\min_W R(W),\ R(W) = \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 + \gamma\, \Omega(W),$$

where $\Omega : \mathbb{R}^{d\times T} \to \mathbb{R}_+$, called the regularizer, provides an intuitive mechanism to incorporate similarities among tasks weight vectors; and $\gamma > 0$ is a hyperparameter which needs to be tuned (for example by cross validation on the data) and regulates the importance of the regularizer with respect to the empirical loss. As noted in the introduction, we will consider three kinds of functions Ω, which implement three different assumptions on the structure of the relationships between the tasks. These functions are:

- Quadratic norms

$$\Omega(W) = \operatorname{vec}(W)^\top E \operatorname{vec}(W), \tag{2.1}$$

  where $E \in \mathbb{R}^{dT\times dT}$, $E \succeq 0$.

- Sparsity inducing norms

$$\Omega(W) = \omega(|W|), \tag{2.2}$$

  where $\omega : \mathbb{R}_+^{d\times T} \to \mathbb{R}_+$.

- Spectral functions:

$$\Omega(W) = \omega(\sigma(W)), \tag{2.3}$$

  where $\omega : \mathbb{R}_+^{K} \to \mathbb{R}_+$.

Note that the case of learning all tasks independently by employing a square norm as a regularizer on the parameters is a particular instance of all three frameworks.

In the following we review the approaches using these functions.

2.3.1. Quadratic norms

One way of defining relations among a set of tasks is by looking at similarities between their parameters. For example, let us consider the problem of detecting different fruits in pictures. We want to learn one task function per fruit, e.g. $f_{orange}(x)$ will return 1 if there is an orange somewhere in the picture x, or -1 otherwise. It is reasonable to accept that the process of detecting an orange in a picture is somewhat similar to the process of detecting an apple, as they have similar shapes. Thus, one may then assume that the parameters defining those functions are close to each other; therefore, one could penalize the distances among them. In this section we focus on linear functions and the use of regularizers of the form described in eq. (2.1), which are useful to model this kind of relationship. In the general case, the objective function to be minimized can be expressed as:

$$R(W) := \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 + \gamma \operatorname{vec}(W)^\top E \operatorname{vec}(W), \tag{2.4}$$

where E is defined a priori so that it captures the relations between the tasks. Since E is positive semidefinite, problem (2.4) is convex.

We study two alternative perspectives of problem (2.1): the feature space and the multitask kernel point of view. After that we will review some instances of this framework from the literature.

2.3.1.1. Feature space viewpoint

Let us start by assuming that all functions f_t can be written in terms of the same feature vector $\theta \in \mathbb{R}^p$, for some $p \in \mathbb{N}$, $p \geq dT$, that is:

$$f_t(x) = \theta^\top B_t x,\ x \in \mathbb{R}^d, \forall t \in [T], \tag{2.5}$$

or equivalently $w_t = B_t^\top \theta$, where B_t are prescribed $p \times d$ matrices which are task specific, whereas the vector θ is common to all tasks. Then, we can consider the regularization problem characterized by the following objective function:

$$S(\theta) := \sum_{t=1}^{T} \left\| X^{t\top} B_t^\top \theta - y^t \right\|_2^2 + \gamma \theta^\top \theta \tag{2.6}$$

As proved by [58], this problem is equivalent to problem (2.4). Let us consider the *fea-*

ture matrix $B \in \mathbb{R}^{p \times dT}$ formed by concatenating all B_t matrices: $B := [B_t : \forall t \in [T]]$. The equivalence between these two problems relates matrix B to matrix E in eq. (2.4). Specifically, [58, Prop. 1] establishes that:

> If the feature matrix B is full rank and we define the matrix E in eq. (2.4) to be $E = \left(B^\top B\right)^{-1}$, then we have that
>
> $$S(\theta) = R\left(B^\top \theta\right), \theta \in \mathbb{R}^p. \tag{2.7}$$
>
> Conversely, if E is a symmetric and positive definite matrix, T is a squared root of E and we set $B = T^\top E^{-1}$, then eq. (2.7) holds true. Moreover, the unique minimizers W^* of problem (2.4) and θ^* of problem (2.6) are related by the equation $\mathrm{vec}(W^*) = B^\top \theta^*$.

The above proposition requires matrix E to be positive definite and the matrix B to be full rank. Note that otherwise the functions f_t are linearly related. We can see as a trivial example the case that $B_t = B_0$, $\forall t \in [T]$, for some prescribed matrix $B_0 \in \mathbb{R}^{p \times d}$. In that case all tasks are the same task, $f_1 = f_2 = \ldots = f_T$, so effectively we are solving a single task learning problem on all the Tm data instances from the T tasks.

When the matrix B is not full rank, the equivalence in eq. (2.7) between functions (2.4) and (2.6) still holds true provided that matrix E is given by the pseudoinverse of matrix $B^\top B$ and we minimize function (2.4) on the linear subspace $\mathcal{S}$ spanned by the eigenvectors of E which have a positive eigenvalue. For example, in the above case where $B_t = B_0$, $\forall t \in [T]$, we have that $\mathcal{S} = \{(w_t : t \in [T]) : w_1 = w_2 = \cdots = w_T\}$. This observation would also extend to the circumstance where there are arbitrary linear relations amongst the tasks.

2.3.1.2. Multitask kernel viewpoint

In this section we study another viewpoint of quadratic norms based on the use of a kernel (see [179] for a review on kernel methods). Making use of the previous viewpoint where eq. (2.4) can be expressed as a single task learning problem, we can consider a kernel defined on a multitask setting. We can start by considering the functions $f = (f_t : t \in [T])$ as the real-valued functions $(x, t) \mapsto \theta^\top B_t x$ on the input space $\mathbb{R}^d \times [T]$ whose squared norm is $\theta^\top \theta$. The corresponding reproducing kernel is:

$$K((x,t),(z,s)) = x^\top B_t^\top B_s z,\ x, z \in \mathbb{R}^d,\ t, s \in [T], \tag{2.8}$$

Since eq. (2.6) is like a single task regularization functional, by making use of the representer theorem its minimizer can be expressed as:

$$\theta^* = \sum_{i\in[m]} \sum_{t\in[T]} c_{it} B_t x_i^t.$$

Consequently, the optimal task functions can be expressed as:

$$f_q^* (x) = \sum_{i\in[m]} \sum_{t\in[T]} c_{it} K\left(\left(x_i^t, t\right), (x, q)\right), \; x \in \mathbb{R}^d, \; q \in [T],$$

2.3.1.3. MTL models based on quadratic norms

Here we present two very common MTL approaches which exploit quadratic norms to model the proximity between tasks. In the following examples, we consider a special subset of quadratic regularizers of the form:

$$\Omega (W) = \sum_{t,s\in[T]} w_t^\top w_s G_{ts}, \tag{2.9}$$

where G is a prescribed positive definite matrix. Matrices G in eq. (2.9) and E in eq. (2.4) are related by the equality $E = G \otimes I_d$, where $\otimes$ denotes the Kronecker product [119].

Penalizing task's variance

Let us assume that all tasks weight vectors are close to a reference vector w_0 which also needs to be learned. This idea was proposed in [59], where the authors assumed that

$$w_t = w_0 + v_t, \forall t \in [T],$$

where v_t is what makes task t different from the others and therefore it is assumed to be small. This assumption is taken into account by adding regularization terms which penalize the square distance between any w_t and w_0 (in other words, penalizing $\|v_t\|_2^2$, $\forall t \in [T]$). The resultant approach is based on the following regularizer:

$$\Omega (W) = \min_{w_0} \frac{1}{\lambda T} \sum_{t=1}^{T} \|w_t - w_0\|_2^2 + \frac{1}{1-\lambda} \|w_0\|_2^2, \tag{2.10}$$

where $\lambda \in [0, 1]$ is a regularization parameter which needs to be tuned a priori. It controls the prior information we have about both how close w_0 is to 0, and how close

all weight vectors w_t are to w_0. The first condition is encouraged for high values of λ whereas the second condition becomes more important for values of λ close to 0.

This regularizer can be expressed directly in terms of W getting rid of w_0 in the following manner [58]:

$$\Omega(W) = \frac{1}{T}\left(\sum_{t=1}^{T}\|w_t\|_2^2 + \frac{1-\lambda}{\lambda}\sum_{t=1}^{T}\left\|w_t - \frac{1}{T}\sum_{s=1}^{T}w_s\right\|_2^2\right). \tag{2.11}$$

From this point, it can be worked out that $G = \frac{1}{T}\left(\frac{1}{\lambda}I_T - \frac{1-\lambda}{T\lambda}\mathbf{1}\mathbf{1}^\top\right)$. By applying the results in eq. (2.7) we obtain that B is characterized in the following way:

$$B_t = [\sqrt{1-\lambda}I_d, \underbrace{0_d, \ldots, 0_d}_{t-1}, \sqrt{\lambda T}I_d, \underbrace{0_d, \ldots, 0_d}_{T-t}] \tag{2.12}$$

where $0_d \in \mathbb{R}^{d\times d}$ denotes the all zeros matrix. Finally we derive the multitask kernel associated to regularizer in eq. (2.10). Applying eq. (2.8) we have that:

$$K((x,t),(z,s)) = (1-\lambda+\lambda T\delta_{ts})\,x^\top z,\ x,z\in\mathbb{R}^d,\ t,s\in[T].$$

Note that this set of kernels defined in terms of $\lambda \in [0,1]$ is a convex hull of two kernels. One of them (when $\lambda = 0$) treats all tasks as the same task whereas the other (when $\lambda = 1$) tackles all tasks independently.

Hierarchical Regularization

Let us now assume that the tasks are organized according to a prescribed hierarchical structure or tree which contains information about how tasks are related, in the sense that each task weight vector is close to the task weights of their children. A motivating example is object recognition on classes that are organized in a hierarchy. In this way, the leaves represent elementary classes and are associated to superclasses (for example, leaf nodes could correspond to classes such as “cars”, “vans”, “bikes” and could all be children of the super-class “vehicles”). Following the idea of the previous method, we assume that the tasks are close to each other according to the prescribed hierarchical structure. Therefore, we want to set out the following regularizer [58, 201]:

$$\Omega(W) = \sum_{t=1}^{T}\|w_t\|_2^2 + \frac{1-\lambda}{\lambda}\sum_{t=1}^{T}\left\|w_t - \frac{1}{|C_t|}\sum_{s\in C_t}w_s\right\|^2, \tag{2.13}$$

where C_t represents the set of children of task t, and $|C_t|$ is its cardinality. Let us introduce the operator $\text{parent} : [T] \to \{[T], 0\}$, which returns the parent of the input according to the given tree. If the input task is the root, the output is 0. We consider the symmetric matrices $S, P \in \mathbb{R}^{T \times T}$ where S indicates sibling relationships among tasks:

$$S_{ts} = \begin{cases} \frac{1}{\left|C_{\text{parent}(t)}\right|^2} & \text{if parent}\,(t) = \text{parent}\,(\text{s}) \\ 0 & \text{otherwise} \end{cases},$$

and P encodes parent-child relationships:

$$P_{ts} = \begin{cases} \frac{1}{|C_t|} & \text{if}\, t = \text{parent}\,(\text{s}) \\ \frac{1}{|C_s|} & \text{if}\, s = \text{parent}\,(\text{t}) \\ 0 & \text{otherwise} \end{cases}.$$

Then, the matrix G corresponding to the regularizer in eq. (2.13) can be expressed as:

$$G = \frac{1}{T}\left(\frac{1}{\lambda} I_T + \frac{1-\lambda}{2\lambda} S - \frac{1-\lambda}{\lambda} P\right).$$

The characteristic matrix B and the induced kernel can be calculated from this form of G.

2.3.1.4. Discussion

The MTL approaches that arise within this framework are based on the similarity between the weight vectors of the tasks. This intuitive assumption is useful to model the relationship of tasks that are positively correlated. However, the simplicity of quadratic norm methods makes them unable to capture more complex relationships. One simple example when these approaches fail is when two tasks produce uncorrelated outputs to the same inputs. Utilizing quadratic norms in such situations would lead to negative transfer. In the following sections we review approaches capable of leveraging this and other more general relationships, and which can positively transfer the knowledge between tasks.

2.3.2. Sparsity inducing functions

The approaches described in this section assume sparse structure in the task's weight vectors. This is very useful in situations where the dimensionality of the data is large.

For example, let us think about the problem of discovering how different DNA microarrays, which are high dimensional entities, lead to a related set of cancers. One can model this problem by considering one task per cancer type. An assumption made in addressing this problem is to postulate that these kinds of cancer are related because they are caused by the same sets of genes in a microarray [98]. In other words, all tasks depend only on a small subset of the attributes which describe the input data. Sparse solutions are encouraged by the regularization functions explained in this subsection. These regularizers build up on the Lasso technique [196], which encourages sparsity on a weight vector elements by penalizing its ℓ_1-norm. The method of ℓ_1-norm regularization has proven to be superior to non-sparse methods such as ℓ_2-norm regularization whenever there is a priori knowledge about the sparsity of the explanatory variables [145].

Sparse methods are also useful when the objective of the model is not only to be accurate but also to be highly interpretable. Interpretable solutions are very important in many fields such as medicine [98]. In this case, even with a lack of prior knowledge, one can sacrifice some accuracy of the model in order to get a solution which is a function of a small set of explanatory variables.

2.3.2.1. $\ell_{p,1}$-norms

Some MTL approaches extend the Lasso by assuming that all tasks use the same sparse set of attributes.

One of the first works that formulated this hypothesis and proposed a solution is [204]. Its authors propose to shrink the ℓ_1-norm of the maximum absolute value of each explanatory variable weight across all tasks. In other words, they propose the use of the $\ell_{\infty,1}$-norm,

$$\|W\|_{\infty,1} = \sum_{i=1}^{d} \max_{t=1}^{T} |w_{i,t}|.$$

The authors justify this choice by arguing that the quantity $\max_{t=1}^{T} |w_{i,t}|$ can be seen as the "simultaneous explanatory power" of variable i among all tasks. Furthermore, this regularizer is appealing since it keeps the objective function convex. In order to solve the resultant problem, the authors employ an interior-point algorithm. This approach, which is also known as Multitask Lasso [116], is theoretically studied in [143], where the authors provide some results which establish conditions under which employing the $\ell_{\infty,1}$-norm is advantageous over the use of the $\ell_{1,1}$-norm, $\|W\|_{1,1} = \sum_{t=1}^{T} \sum_{i=1}^{d} |w_{i,t}|$, that is

the ℓ_1-norm of the matrix elements. Note that the latter regularizer does not provide any shared sparsity between tasks.

Another approach is the adaptation of Group Lasso [222] to MTL. Group Lasso is a single task supervised learning method which assumes that all attributes can be organized in several disjoint groups so that some groups of attributes are important for the solution whereas some others can be ignored. These groups of attributes are known a priori so the problem consists in minimizing a loss function plus some regularizer which induces sparsity over groups but not within groups.

This strategy can be directly applied to an MTL setting by assigning each of these groups to the same attribute employed by the different tasks. This implies the use of the $\ell_{2,1}$-norm on W, which is defined as $\|W\|_{2,1} = \sum_{i=1}^{d} \left(\sum_{t=1}^{T} w_{i,t}^2 \right)^{\frac{1}{2}}$. This idea is proposed in [8, 150, 151] and is known as multitask joint covariate selection [151]. The corresponding optimization problem is then

$$\underset{W}{\operatorname{argmin}} \sum_{t=1}^{T} \left\| X_t^\top w_t - Y_t \right\|_2^2 + \gamma \|W\|_{2,1} . \tag{2.14}$$

Let us recall that matrix W contains the weight tasks vectors in columns so the $\ell_{2,1}$-norm encourages that matrix to have a few non-zero rows, or attributes weights.

As we can see, the last approaches are very similar and are all instances of the $\ell_{p,1}$-norm, defined as:

$$\|W\|_{p,1} = \sum_{i=1}^{d} \left(\sum_{t=1}^{T} \left| w_{i,t}^p \right| \right)^{\frac{1}{p}}.$$

As stated by [218], the $\ell_{\infty,1}$ regularizer encourages all non-zero parameters to have the same absolute value across all tasks, whereas when employing the $\ell_{2,1}$ regularizer, the resultant non-zero values of the parameters across tasks have usually a higher variance. Indeed we can make the generalization to any $\ell_{p,1}$-norm for $p > 1$ (note that $p = 1$ leads to the original Lasso technique implying that all tasks are assumed to be independent).

Further development based on the Group-Lasso approach has been done in several works, and different optimization strategies have been suggested to solve problem (2.14). In [117], the authors propose an optimization approach based on proximal methods [144] whereas in [8] the authors develop an alternating algorithm which minimizes an objective function whose regularizer is the square norm of the $\ell_{2,1}$-norm of W. In [150, 151] the authors also propose a method to efficiently compute the space of solutions provided by the algorithm for all values of γ (regularization path). In [41], the

authors consider the case when different attributes have different regularization parameters (values), develop a method to automatically tune them and provide some theoretical statistical guarantees on its performance. Finally, in [120] the authors provide some learning bounds on this framework, which show that there are theoretical advantages of this approach over performing the Lasso on all tasks independently.

2.3.2.2. Other sparsity inducing functions

Even though assuming that all tasks use the same subset of attributes is the most common conjecture among sparsity MTL methods, there are other scenarios where different sparsity patterns could be useful.

Exclusive Lasso In [227], the authors propose a different scenario: they assume that tasks tend to not share any attributes between them. This can be useful, for example, in multi-category document classification, where different categories of documents are characterized by different words (attributes). In order to model this, they propose the following regularizer:

$$\Omega(W) = \sum_{i=1}^{d} \left(\sum_{t=1}^{T} |w_{i,t}| \right)^2.$$

Note that the inner sum is the ℓ_1-norm of the tasks values for the same attribute and thus encourages sparsity (or heterogeneity) of attributes across the tasks. The outer sum, which is the ℓ_2 norm of the previous values, combines the weights of all attributes.

Tree-Guided Group Lasso Let us now consider the scenario where the model is provided with extra information about the tasks (as considered in Section 2.3.1.3), where this is given as a hierarchy of tasks indicating relationships among them. In [99] the authors consider this situation in the context of sparsity parameters. Their key assumption is that any common ancestor of two leaves of the tree (tasks), contains information about the attributes which are used simultaneously by both tasks. This is induced by building a regularizer composed of a weighted sum of $\ell_{2,1}$-norms on the tasks weight vectors according to the hierarchy. Particularly, the authors place an $\ell_{2,1}$-norm regularizer on each node of the hierarchy. This term is multiplied by a weight which is determined by parameters s_v and g_v, where $s_v + g_v = 1$. The vector g_v represents the joint selection of the attributes for tasks under node v, whereas s_v represents the importance given to the independent selection of the attributes for the same set of tasks.

The authors prove that the total weight related to any task weight vector is 1, which provides the framework with consistent estimators. Since knowing a priori the hierarchy among tasks (as well as the values of s_v and g_v for all nodes in the tree) can be a strong assumption in many real cases, the authors suggest the previous use of methods such as hierarchical agglomerative clustering algorithm to estimate the underlying structure.

2.3.2.3. Discussion

In this section we have described a set of MTL methods based on imposing sparsity patterns to the task's weight vectors. Hence, the set of attributes of the data which are meaningful for the tasks can be filtered from the remaining ones. These approaches are useful whenever the dimensionality of the data is high and only a few attributes of the data convey meaningful information. However in several situations this does not hold. For example when the input data are images usually the majority of pixels provide useful information. In that case it is reasonable to look for higher level features that can be meaningful for all tasks. This more general approach is taken in the following section. It is also worth remarking that the ideas used in Exclusive Lasso, which imposes heterogeneity between the attributes used in tasks, could potentially be useful for multi-aspect datasets, because different aspects of data may depend on different features.

2.3.3. Spectral functions

In this section, we consider a different kind of relation to link tasks together: all tasks share a low dimensional representation of the data. In other words, we assume that the task's vectors w_t are linear combinations of a few common basis vectors which need to be estimated from the data. Another viewpoint can be explored by comparing this set of approaches and the unsupervised approach of principal component analysis (PCA) [160]. The latter procedure obtains a set of components (linear combinations of the attributes) so that when projecting the original data onto them, the resultant projections have the largest possible variance. In contrast, in MTL spectral function approaches, the components are extracted so that they are as useful as possible for all tasks.

Let us imagine an automatic system whose purpose is to infer levels of different affective states (such as anger, boredom and happiness) in images of faces. The MTL model may consider one task per affective state. It seems reasonable to assume that there exists a small set of components of faces which are enough to discriminate any kind of information regarding affective states. Note that, unlike the previous strategy, the

features used now do not necessarily correspond to a particular part of the face (e.g: eyes, ears or noses), but to linear configurations of the whole face, which are learned in the process.

One way to encourage the above assumption is by using certain spectral regularizers. Let us recall that a spectral function of a matrix $M \in \mathbb{R}^{d_1 \times d_2}$, $\omega(M)$ is any function that only uses the singular values of M, $\sigma(M) \in \mathbb{R}^K$, where $K = \min\{d_1, d_2\}$. These functions are especially useful since the singular values of a matrix convey determinant information regarding its structure.

2.3.3.1. Rank

Let us continue with the previous affect recognition example from faces. In that scenario we have a set of faces composed of d pixels (attributes) and the objective is to learn T affective recognition tasks. The assumption that one may impose is that all tasks use a common set of K linear features from the data, where $K \ll d$. Therefore, one can model it by setting $f_t(x) = w_t^\top x = a_t^\top B^\top x$, where $B \in \mathbb{R}^{d \times K}$ is the matrix composed of K learned linear projections common across the tasks, and $a_t \in \mathbb{R}^K$ specifies the way these projections are linearly combined to obtain the weight vector for task t, w_t. Then we would like to solve the following non-convex problem:

$$\min_{A,B} \sum_{t=1}^{T} \left\| X^{t\top} B a_t - y^t \right\|_2^2, \tag{2.15}$$

where matrix $A = [a_1, a_2, \ldots a_T]$. Matrix B can be seen as the factor which limits the input data information the tasks have access to. Therefore, by learning the K features which compose B, the model may improve the performance on all tasks. Note that solutions to problem (2.15) are not unique: if we consider any nonsingular matrix $C \in \mathbb{R}^{K \times K}$, then $W = BA = (BC)(C^{-1}A)$. An equivalent problem can be obtained by expressing the objective function directly in terms of the tasks weight vectors:

$$\begin{aligned} &\min_{W} \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 \qquad \text{s.t} : \operatorname{rank}(W) \leq K \quad \text{or equivalently} \\ &\min_{W} \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 \quad + \gamma \operatorname{rank}(W) \end{aligned} \tag{2.16}$$

for some value of γ, making clear that the resultant problem is a spectral regularization problem.

Unfortunately, both formulations of the problem are non-convex, leading to NP-hard

optimization problems, where finding the optimal solution is intractable for the general case. However, we can obtain this optimal solution if we consider a multivariate response problem, that is, when all tasks share the same input data $X^1 = X^2 = \cdots = X^T = X$. Let us assume that $\Sigma_{XX} \in \mathbb{R}^{d\times d}$ is the input covariance matrix and $\Sigma_{YX} \in \mathbb{R}^{T\times d}$ is the cross covariance matrix between the outputs and the inputs. Then the optimal solution to problem (2.16) can be expressed as [14]:

$$W^* = \sum_{k=1}^{K} V_k V_k^\top \Sigma_{YX} \Sigma_{XX}^{-1},$$

where V_k is the k-th singular vector of $\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{YX}^\top$.

In the general MTL setting, problem (2.16) is quite demanding since the regularizer is neither convex nor differentiable, and in addition the resultant problem is NP-hard [11]. In the following we will cover some more feasible approximations.

2.3.3.2. Trace norm

Let us continue from eq. (2.16), where the rank of W acts as the regularization function. In order to make the problem tractable we can look for an approximation of the rank function which keeps the whole problem convex. A good candidate function for that is the trace norm, which is defined as the sum of the singular values of the matrix. It has the interesting property to be the convex envelope of the rank in the spectral ball [61]. The trace norm as a regularizer was first proposed in [61] and since then it has been employed in many problems involving the rank of a matrix to be learned [8, 166, 165, 32, 190, 191]. In our case, the resultant problem is

$$\min_{\mathrm{W}} \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 + \gamma \left\| W \right\|_{\mathrm{Tr}}. \tag{2.17}$$

In the following, we consider two alternative viewpoints of this objective function.

Square Frobenius norm viewpoint A way to interpret eq. (2.17) is by means of the approach suggested by [4]. In this paper the authors propose a method which explicitly learns a common representation of the input data which is used for all tasks. Particularly, they propose to express each task weight vector as $w_t = Ba_t$, in a similar way as in eq. (2.15). In order to avoid overfitting, the authors add the square Frobenius norm on A (the matrix composed of all a_t, $\forall t \in [T]$ as columns) and B. The resultant

optimization problem is

$$\min_{\mathrm{A,B}} \sum_{t=1}^{T} \left\| X^{t\top} B a_t - y^t \right\|_2^2 + \frac{\gamma}{2} \left(\|A\|_{\mathrm{Fr}}^2 + \|B\|_{\mathrm{Fr}}^2 \right). \tag{2.18}$$

This problem is not convex and the number K of factors, or linear projections, needs to be defined a priori. In order to account for both issues, the authors use the following equivalence [190]:

$$\|W\|_{\mathrm{Tr}} = \min_{BA=W} \tfrac{1}{2} \left(\|A\|_{\mathrm{Fr}}^2 + \|B\|_{\mathrm{Fr}}^2 \right)$$

in eq. (2.18) leading again to eq. (2.17). It is important to remark that problem (2.18) is equivalent to problem (2.17) only when $K \geq \min(d, T)$. Notice that if this is not the case, we are adding a rank constraint on W to be less or equal than K, leading to a non-convex problem.

$\ell_{2,1}$-norm viewpoint Another interpretation can be found by considering the problem proposed in [8]:

$$\min_{\mathrm{U,A}} \left\{ \sum_{t=1}^{T} \left\| X^{t\top} U a_t - y^t \right\|_2^2 + \gamma \|A\|_{2,1}^2 \,:\, A \in \mathbb{R}^{d \times T}, U \in \mathbb{R}^{d \times d}, U^\top U = I \right\} \tag{2.19}$$

In this approach the matrix U is used to rotate the data, so that there are some projections which can be useful for all tasks. Consequently, U is constrained to be an orthonormal matrix. The matrix A contains, for each column t, the weights of task t for the components learned in U. Since the original assumption was that all tasks use the same low dimensional representation of the data, an $\ell_{2,1}$-norm based regularization term is added so that A is encouraged to have only a few non-zero rows. The product Ua_t makes the problem (2.19) non convex, however the authors of [8] show that this is equivalent to the convex problem:

$$\begin{aligned} \min_{\mathrm{W,D}} \quad & \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 + \gamma \sum_{t=1}^{T} w_t^\top D^{-1} w_t \\ \mathrm{s.t}: \quad & W \in \mathbb{R}^{d \times T}, D \in \mathbb{R}^{d \times d}, D \succeq 0, \mathrm{tr}\,(\mathrm{D}) \leq 1, \end{aligned} \tag{2.20}$$

where $W = UA$. Finally, solving this problem with respect to D and plugging it back into problem (2.20) leads to a similar formulation as in eq. (2.17) but where the regularizer takes the form of the square of the trace norm. This set of equivalences is to be expected since minimizing the rank of $W = UA$, with U orthogonal, is equivalent to minimizing the number of non-zero rows of A, which leads us to the original problem.

The fact that equations (2.17) and (2.20) present different convex formulations for the same underlying problem gives rise to different strategies to design optimization algorithms to solve the problems. The formulation in eq. (2.17) allows the application of accelerated proximal methods [144], an approach that is studied in [92]. Alternatively, the authors of [8] present an iterative algorithm to optimize problem (2.20) based on optimizing alternatively over W and over D. Note that a drawback from both strategies is that they require a repeated singular value decomposition (SVD) of W, which can be expensive for large d and T.

To avoid SVD, heuristics have been proposed which lend to less demanding methods, at the expense of losing guarantees about the optimal solution. An heuristic method is to fix the rank K and solve problem (2.18) by alternate minimization.

In [150, 151] the authors propose an alternative way to provide approximate solutions to problem (2.17). The main idea is to adapt the approaches developed in Section 2.3.2 to deal with this scenario by augmenting the dimensionality by taking random projections from the input data. The authors justify the approach by relying on the Johnson-Lindernstrauss lemma [48], which implies that any set of points in a high dimensional space can be projected in a much lower dimensional space so that the original distances among the points are nearly preserved. They start from the original formulation of the problem (eq. (2.19)) but in this case, matrix $U \in \mathbb{R}^{c \times d}$ contains more columns $c \gg d$ representing the possible useful projections of the data and is fixed in advance to a set of random vectors. Solving problem (2.19) with respect to A is equivalent to solving the previous problem (2.14) based on the $\ell_{2,1}$-norm. The advantage of employing this approach is avoiding SVD, which is required in the previous approaches. The drawback is that in order to obtain an accurate approximation, c must be really large. The authors provide some theoretical guarantees about the convergence of the solution to the one obtained by exact algorithms (like [8]). The experiments show, as expected, that there is a trade off between the accuracy of the algorithm and the required time to run it, both factors being determined by the value of parameter c. Consequently, the proposed algorithm is useful in situations when a trade off between time and accuracy can be made.

2.3.3.3. Log-Determinant

In this section we study another interesting spectral regularizer:

$$\Omega_\zeta(W) := \log\left(\det\left(\zeta I + WW^\top\right)\right) = \sum_{k=1}^{K} \log\left(\sigma_k(W)^2 + \zeta\right),$$

where $\zeta > 0$ prevents the regularizer to be undefined when $\det\left(WW^\top\right)$ vanishes (recall that $\det(\cdot)$ denotes the determinant). Unlike the previous regularizer, the present function is concave so the associated regularization problem is in general non-convex; nevertheless it presents other advantages. First, $\Omega_\zeta(W)$ can be induced by the following minimization problem:

$$\Omega_\zeta(W) = \min_{D \succ 0} \log(\det(D)) + \beta \operatorname{trace}\left(\left(\zeta I + WW^\top\right) D^{-1}\right)$$

for some $\beta > 0$. To see so, we only need to differentiate with respect to D and set it to zero. If we express the whole optimization problem we have that

$$\min_{D \succ 0, W} f(W, D), \text{ where}$$

$$f(W, D) := \sum_{t=1}^{T} \left\| X^{t\top} w_t - y^t \right\|_2^2 + \gamma \log(\det(D)) + \gamma\beta \operatorname{trace}\left(\left(\Psi + WW^\top\right) D^{-1}\right) \tag{2.21}$$

It is interesting to see that this problem bears a strong resemblance to problem (2.20), in which the constraint on the trace norm on D is replaced by the log-determinant term. As pointed out in [221], the trace norm is a convex envelope of both the log-determinant and the rank so both approaches have indeed similar underlying objectives.

Another important characteristic of this problem is that its optimum is the maximum a posteriori (MAP) estimator for the following probabilistic model (see Section A.1):

$$\begin{aligned} D|V, n &\sim \mathcal{W}^{-1}(\Psi,\, \nu) \\ w_t|D &\sim \mathcal{N}(0,\, D) \\ y_t|X_t, w_t, \tau &\sim \mathcal{N}\left(X_t^\top w_t,\, \tau I\right), \end{aligned} \tag{2.22}$$

where $\mathcal{W}^{-1}(\Psi,\, \nu)$ is the Inverse-Wishart distribution, where $\Psi \succ 0$ is the inverse scale matrix and $\nu = \frac{\gamma}{\beta} - d - T - 1$, the number of degrees of freedom of the distribution.

In order to solve problem (2.21) one can follow the same strategy as in problem (2.20), that is by optimizing alternately over W and over D. Optimizing with respect to W leads to T decoupled problems. The optimization over D can be obtained by differentiating eq. (2.21) and setting it to 0 leading to:

$$\hat{D} = \frac{\beta}{\gamma}\left(\Psi + WW^\top\right). \tag{2.23}$$

It is worth noticing that optimizing over D given W leads to a simple closed form that,

unlike the previous spectral methods, does not require any SVD which gives way to an efficient computation for large d.

In [220], the authors use the Normal-Inverse-Wishart distribution as a prior for both the mean w_0 and the covariance matrix D of w, which leads to this regularizer. The authors employ an expectation-maximization method to get good solutions for the resultant MAP problem. Other authors have used it in different contexts, for example in [138] they use it for matrix completion and in [221] for link analysis, where the authors also make a connection with MTL.

2.3.3.4. Composite spectral functions

Until now, we have considered spectral functions on the weight vector matrix W. An alternative which is convenient in some scenarios is to consider the spectral function of $H_1 W H_2$ where H_1 and H_2 are some prescribed matrices. An interesting example arises when considering matrix $\Pi = I - V$, where $V = \frac{1}{T}\mathbf{1}\mathbf{1}^\top$. For example, if $H_1 = I$ and $H_2 = \Pi$, then $H_1 W H_2 = [w_1 - \bar{w},\, w_2 - \bar{w} \ldots, w_T - \bar{w}]$ can be seen as a task-centered version of W, where $\bar{w}$ is the mean of all weight vectors $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. We can examine the previous two described approaches after applying this change:

- Composite trace norm $\Omega(W) := \|W\Pi\|_{\mathrm{Tr}}$. This is studied in [60] and conveys the assumption that the divergence among tasks can be expressed in low dimensionality, that is, the tasks are a low rank perturbation of a common "mean task", $w_t = \bar{w} + v_t$ with V low rank.
- Composite log determinant: $\Omega_\zeta(W) := \log\left(\det\left(\zeta I + W\Pi W^\top\right)\right)$. This leads to a MAP solution of a probabilistic formulation similar as in eq. (2.22), however the mean of the weight vectors is not necessarily 0, but it is considered as another random vector having a Gaussian prior:

$$
\begin{aligned}
w_0|\alpha &\sim \mathcal{N}(0,\, \alpha I) \\
D|\Psi, \nu &\sim \mathcal{W}^{-1}(\Psi,\, \nu) \\
w_t|w_0, D &\sim \mathcal{N}(w_0,\, D) \\
y_t|X_t, w_t, \tau &\sim \mathcal{N}\left(X_t^T w_t,\, \tau I\right).
\end{aligned}
$$

  If no further regularization is set, the variance of w_0 is assumed to be infinity $(\alpha \to \infty)$.

In the following we study one more composite spectral regularizer of the form $\Omega = \omega(\sigma(\Pi W))$, that is, a spectral function on an attribute-centered version of W.

Clustered Multitask Learning In [84] the authors develop an approach to discover K clusters among the tasks and employ them to learn the tasks parameters. The assumption is that a group of tasks belong to the same cluster if their weight vectors are close, in a similar way as expressed in the regularizer (2.11) when there is only one group.

Let us assume that there is an underlying set of relations among tasks which are encoded in matrix $E \in \{0,1\}^{T\times K}$, $\mathbf{1}^\top E\mathbf{1} = K$, where $E_{tk} = 1$ if task t belongs to cluster k and $E_{tk} = 0$ otherwise. It is convenient to denote by $\bar{w}_k = \frac{1}{T_k} W E_{\cdot k}$, where $T_k = E_{\cdot k}^\top \mathbf{1}$, the average task weight vector for cluster k and $\bar{w} = \frac{1}{T} W\mathbf{1}$ is the total average. Finally, let us also define matrices $M = E\left(E^\top E\right)^{-1} E^\top$ and $V = \frac{1}{T}\mathbf{1}\mathbf{1}^\top$. If we were provided with matrix E, then we could create a regularization term based on it, by following the procedure in Section 2.3.1. In the first instance let us assume that this is the case. Then we can define the following regularizer:

$$\Omega_{CMTL} = \lambda_1 \Omega_{mean} + \lambda_2 \Omega_{between} + \lambda_3 \Omega_{within}, \tag{2.24}$$

where

- $\Omega_{mean} := T\left\|\bar{w}\right\|^2 = \operatorname{trace}\left(WVW^\top\right)$ penalizes the norm of the average weight vector.
- $\Omega_{between}(W) := \sum_{k=1}^{K} T_k \left\|\bar{w}_k - \bar{w}\right\|^2 = \operatorname{trace}\left(W(M-V)W^\top\right)$ encourages all cluster centers to be close to the total average.
- $\Omega_{within}(W) := \sum_{k=1}^{K}\sum_{t\in\mathcal{G}_k} T_k \left\|w_t - \bar{w}_k\right\|^2 = \operatorname{trace}\left(W(I-M)W^\top\right)$ measures how condensed the clusters are.

Therefore, the regularizer in eq. (2.24) can be expressed as

$$\Omega = \operatorname{trace}\left(WD(M)^{-1}W^\top\right),$$

$$\text{where } D(M)^{-1} = (\lambda_1 V + \lambda_2 (M-V) + \lambda_3 (I-M)).$$

Given that the actual matrix M is unknown, one can try to minimize the objective function over both W and $D(M)$:

$$\min_{D(M),W} \sum_{t=1}^{T} \left\|X^{t\top} w_t - y^t\right\|_2^2 + \gamma \operatorname{trace}\left(WD(M)^{-1}W^\top\right)$$

$$\text{s.t. } : D(M) \in \mathcal{S}_K$$

$\mathcal{S}_K$ is defined as $\mathcal{S}_K := \{D(M) : M \in \mathcal{M}_K\}$, where

$$\mathcal{M}_K := \left\{M \,:\, M = E\left(E^\top E\right)^{-1} E^\top,\, E \in \{0,1\}^{T\times K},\, \mathbf{1}^\top E\mathbf{1} = K\right\}.$$

Unfortunately, even though the objective function is convex in both W and $D(M) \succeq 0$, the resultant problem is intractable due to non-convex constraints on $D(M)$. To overcome this, the authors present a convex relaxation based on the observation that the non-convex terms, $\Omega_{between}(W)$ and $\Omega_{within}(W)$, only depend on the centered version of W. Then they suggest solving the following convex problem:

$$\begin{aligned}\min_{D_c,W} \sum_{t=1}^{T} \left\|X^{t\top} w_t - y^t\right\|_2^2 + \lambda_1 \Omega_{mean}(W) + \gamma \operatorname{trace}\left(\Pi W D_c W^\top \Pi^\top\right) \\ \text{s.t.} : D_c \in \mathcal{S}_c\end{aligned}, \tag{2.25}$$

so that $\mathcal{S}_c = \{D : \alpha I \preceq D \preceq \beta I,\, \operatorname{trace}(D) = \sigma\}$ is a convex set which encloses $\mathcal{S}_K$, where $\beta \geq \alpha \geq 0$, $\sigma > 0$ are related to the regularization parameters $\lambda_1, \lambda_2, \lambda_3$ (see [84]).

As we see, D_c only appears in the last term of eq. (2.25) so that term can be considered as:

$$\Omega_{clusters} = \min_{D_c \in \mathcal{S}_c} \operatorname{trace}\left(\Pi W D_c W^\top \Pi\right),$$

which can be calculated by a procedure explained in [84] only depending on the singular values of ΠW.

Previous approaches, which have been derived from different starting assumptions, lead to similar spectral regularizers. In [6], the authors assumed that each task function can be expressed in the following way: $f_t(x) = w_t^\top x$, where $w_t = u_t + \Theta^\top v_t$, $w_t \in \mathbb{R}^d$. Both u_t, which has the same dimensionality as the data, and $v_t \in \mathbb{R}^k$, $k < d$, are specific for task t; whereas $\Theta \in \mathbb{R}^{k\times d}$, $\Theta\Theta^\top = I$, is a matrix which has orthogonal rows and encodes the common representation of the data across tasks. Consequently the authors propose a dimensionality reduction operation on the attributes. The regularization that they propose is

$$\begin{aligned}\Omega(W,\Theta) = \; & \alpha \sum_{t=1}^{T} \left\|w_t - \Theta^\top v_t\right\|_2^2 + \beta \|W\|_{\mathrm{Fr}}^2 \\ & \text{s.t.} : \Theta\Theta^\top = I.\end{aligned} \tag{2.26}$$

The problem is non-convex but can be relaxed to achieve convexity by defining $D =$

$\Theta\Theta^\top$ and relaxing the resultant rank constraint on D [39]. The resultant regularizer is:

$$\begin{aligned}\Omega(W, D) &= \alpha\eta(1+\eta)\operatorname{tr}\left(W(\eta I + D)^{-1}W^\top\right) \\ &\text{s.t. : } D \succeq 0, D \preceq 1, \operatorname{tr}(D) = k\end{aligned} \tag{2.27}$$

which is a special case of problem (2.25) omitting the centering matrix Π. Other special cases have been studied in [135], particularly the extension of the k-support norm, [9], to matrices. A trivial special case is the trace norm itself, explained in Section 2.3.3.2, which can be obtained by setting $\alpha = 0$, $\beta = 1$, $\sigma = 1$.

2.3.3.5. Discussion

In this section we have reviewed the use of spectral regularizers to encourage tasks which use a common set of linear features learned from data. These approaches are appealing as they make general assumptions about relationships between tasks, thus diminishing the risk of negative transfer. The capability of learning features from the data makes this framework interesting to be used as a starting point for building MTL approaches that can leverage multi-aspect datasets by learning different features for different aspects.

2.4. Extensions of the previous frameworks

In this section we review approaches which extend previously reviewed methods to account for more complex relationships between tasks.

2.4.1. Prescribed hierarchy

We have reviewed a number of approaches which take advantage of available information regarding the hierarchy of the tasks in order to improve the model. One was a parametric similarity approach (Section 2.3.1.3) and another was a sparsity inducing approach (Section 2.3.2.2). Now we review a prescribed hierarchy approach for the final framework, in which different linear features are learned at different levels of the hierarchy.

In [225] the authors propose the following setting: there exists a multiclass classification problem so that these classes can be organized in a hierarchy. Consequently, if an instance belong to class c it necessarily belongs to classes $\mathcal{A}(c)$, that is, all ancestors

of c which are in the path between c and the root of the tree. In this work, the authors impose the hierarchy by means of a regularizer to encourage the classifier of each node c to use different features than those used by $\mathcal{A}(c)$. This assumption is justified by the fact that a lower level category c usually has the same characteristics as its parent $P(c)$ plus a further set of properties which make it different from its sibling's categories. Therefore, this set of properties is all the information needed to distinguish c from all other categories under $P(c)$. In order to induce the described solution, the authors employ a regularizer which encourages parent-child tasks weight vectors to be orthogonal. Furthermore, they prove that the resultant optimization problem is convex. A similar strategy and its application to visual data is studied in [82].

2.4.2. Dirty models

"Dirty models" is a term coined in [86] and makes reference to models which can obtain a high accuracy even when the data do not cleanly satisfy an a priori assumption. In other words, given a model based on a set of assumptions, a dirty approach builds upon it to introduce robustness against situations where the original a priori assumption does not hold, preventing in this way any negative transfer.

In [86] the authors suggest a dirty model for sparsity inducing norms methods like the ones reviewed in Section 2.3.2. They argue that models based on $\ell_{p,1}$-norms are very sensitive to noise, so that separate element-wise regularization could perform better in noisy scenarios. To overcome this problem, the authors proposed to express the original weight matrix W as the sum of two components $W = S + B$, leading to an inf-convolution problem [224]. The matrix B corresponds to the original clean model and therefore an $\ell_{2,1}$-norm regularizer is imposed on it. Matrix S accounts for the cases where the original assumption does not hold, that is, there exist tasks which require some further attributes other than the common set learned in B. Accordingly, an $\ell_{1,1}$-norm regularizer is imposed on S.

The same concept has been extended for spectral functions approaches. One of these methods is proposed in [38]. There the authors apply the dirty model idea to build a model which can handle situations where the tasks weight vectors cannot be expressed exactly in a common subdimensional space. Accordingly, they set $W = S + B$, where B is assumed to have a low trace norm and accounts for the original model. By means of S, the model leverages the "dirtiness" of the data. As previously proposed by [86], an $\ell_{1,1}$-norm regularizer is applied on S. The resultant optimization problem is convex and the authors propose an accelerated proximal method to solve it.

A similar in spirit approach is proposed by the same authors in [40]. They also apply inf-convolution by decomposing W as the sum of two components $W = S + B$, where B is assumed to be a low trace norm matrix. The difference lies in the assumptions imposed on S which are reflected in an $\ell_{1,2}$-norm regularizer. The underlying idea is to make a model which can be robust to outlier tasks, which can be induced by encouraging a low number of non-zero columns (tasks) on S through the $\ell_{1,2}$-norm. These non-zero columns of S represent the outlier tasks which cannot be totally represented in the low dimensional space induced by B. Since both regularizers employed here are convex, the resultant problem is also convex and is solved by using a method which is conceptually similar to the one presented in [38]. Furthermore, the authors also provide some interesting bounds on the accuracy of the model. The experiments carried out in this work show that this approach outperforms [38] as well as other MTL models such as [8] on the two datasets used.

2.4.3. Task grouping

In Section 2.4.1 we studied a few approaches which make use of a known hierarchy among tasks. However, in many cases information about the relationships is not available. In such situations, one may try to group tasks together assuming that tasks in similar groups have a stronger connection. As we saw in Section 2.3.3.4 the spectral functions framework allowed us to build a task grouping model when the relations among tasks were defined in terms of the similarity of the weight vectors. In the following, we review approaches which perform task grouping by assuming that the relation among tasks is defined through the features they share.

In [10], the authors present a natural extension of the trace norm regularizer (eq. 2.17) [8][1] to the case when there is a fixed number N of groups of tasks. To do so, they propose a measure of heterogeneity among a set of tasks that can be defined as the minimal loss obtained by partitioning tasks in N groups, learning a low dimensional representation of the data in each of those groups. In order to reduce the heterogeneity, the authors propose an algorithm that arranges tasks into groups and learns the task weight vectors so that the matrices that are composed with the weight vectors of each group have small trace norm. The resultant problem is no longer convex, and the authors employed a stochastic gradient descent algorithm to obtain local optimal solutions. Furthermore, they provide an heuristic to initialize the algorithm with the objective of obtaining good local solutions. This heuristic is based on running the algorithm for $N' = 1$ and using

[1] However in this case, the authors used the trace norm regularizer on W rather than its square.

the resultant solution as a starting point for $N' = 2$, then repeating the process until $N' = N$.

In [94] the authors propose a similar approach, where the main difference is that their regularizer is the square trace norm rather than the trace norm itself. They also employ a different optimization algorithm to find the correspondence between tasks and groups (encoded in task-group indicators) which is based on the reformulation of the original problem into a mixed integer programming problem, relaxing the corresponding task-group indicators to be continuous in the interval $[0, 1]$ so that the sum of all task-group indicators belonging to a task is 1. This is not a major relaxation since the authors prove that for each continuous solution found, there exists another binary solution which has the same objective value. The experiments done on four datasets show that this approach is comparable to and sometimes outperforms the previous one.

One drawback of the last two approaches is that the number of clusters N is assumed to be known. In order to overcome such a drawback, the authors of both papers proposed to perform cross validation on N to estimate its value. However, taking into account that there is another parameter to tune (γ), this strategy could not be satisfactory in many situations.

2.4.4. Sparse features

Several works have been proposed to deal with separating tasks in groups without the necessity to specify a priori the number of groups.

One of these approaches has been recently proposed in [107], where the authors assume that tasks are implicitly separated in different groups which are characterized by a common set of features that they use. The main underlying idea is to learn a set of features, as linear combinations of the input data, so that each task only needs a sparse subset of them. In order to do so, they express each task weight vector as $w_t = Ba_t$, where $B \in \mathbb{R}^{d \times K}$ is the matrix composed of K learned linear projections where typically $K > d$; and $a_t \in \mathbb{R}^K$ is a sparse set of weight variables which specify how the learned projections are employed for task t. In order to impose these assumptions on the model, the authors use an ℓ_1-norm regularizer on a_t, $\forall t \in [T]$, and the Frobenius norm on B. The previous requirement of specifying the number of clusters N is now replaced by the specification of the number of linear projections K. Experimental results show that the model is less sensitive to the adjustment of this parameter.

Another interesting approach is the one proposed in [177]. There, the authors define

a MAP problem by imposing a matrix-normal prior over W, so that they can model simultaneously the covariances among features (rows) and among tasks (columns). In order to avoid overfitting, they also add an $\ell_{1,1}$-norm regularizer on both inverse covariances, that is, they impose sparsity on the elements of the inverse covariance matrices. Given the Gaussianity assumption on the random variables, adding the $\ell_{1,1}$-norm regularizer on the inverse covariances leads to favouring sparsity in the relationships among tasks and among features: a 0 in position (i, j) of the features (tasks) inverse covariance matrix implies that i and j are conditionally independent given all other features (tasks). The resultant optimization problem is:

$$\underset{W,D,\Omega}{\operatorname{argmin}} \sum_{t=1}^{T} \left\| X_t^\top w_t - Y_t \right\|_2^2 + \lambda_1 \log\left(\det\left(D\right)\right) + \lambda_2 \log\left(\det\left(\Omega\right)\right) + \lambda_3 \operatorname{tr}\left(\Omega^{-1} W D^{-1} W^\top\right) + \lambda_4 \left\| D^{-1} \right\|_{1,1} + \lambda_5 \left\| \Omega^{-1} \right\|_{1,1}$$

This method is particularly valuable since it subsumes other approaches already reviewed: when $\lambda_1 = \lambda_2 = \lambda_4 = \lambda_5 = 0$, and the tasks covariance matrix is fixed to $\Omega = I$, the resultant optimization problem is equivalent to eq. (2.20) proposed by [8]. If assuming the prior setting, we allow $\lambda_1 \geq 0$, the resultant approach is equivalent to the one developed in eq. (2.21) [220]. If rather than focusing on learning the features covariance, we focus on modelling the task covariance, we can see that CMTL (eq. (2.25)) [84, 226] can be seen as a particular case of this problem just by setting $\lambda_1 = \lambda_2 = \lambda_4 = \lambda_5 = 0$ and $D = I$. The optimization problem needs the specification of five hyperparameters which would lead to a large number of combinations in order to tune them by cross-validation. To avoid this, the authors propose to fix $\lambda_1 = d\lambda_3$, $\lambda_2 = T\lambda_3$ and $\lambda_4 = \lambda_5$ leaving only 2 hyperparameters to be tuned. The experiments carried out in the paper over two real datasets show that the described method performs better than the others approaches it subsumes.

2.4.5. Discussion

In this section we have reviewed four kinds of extensions to the frameworks presented in Section 2.3.

The first extension was based on using side information about tasks in order to improve the accuracy, where the side information is specified as a hierarchy. Even though this kind of side information is not appropriate for multi-aspect datasets, the idea of constraining orthogonality between weight vectors as a way to make them using different features will prove useful in Chapter 5.

The remaining three extensions have the objective of decreasing negative transfer in different ways. Dirty models try to fit the data to a perturbation of a model that assumes that all tasks are equally related. Thus, this approach is useful to detect abnormalities, such as outlier tasks, in a scenario where everything else is homogeneously related. However, it cannot account for situations where there are several (unknown) groups of tasks with stronger relations between them. The second set of approaches, task grouping, tries to explicitly account for groups of tasks by clustering and learning them simultaneously. The last set of approaches has a similar motivation but groups of tasks are learned implicitly. For example, in the case of [107], this is done by sparsely linking each task to a pool of learned features. This last framework is less sensitive to the hyperparameters and has empirically proved to outperform the one based on task grouping. In Chapter 4 we will focus on this strategy.

2.5. Other MTL approaches

There is a set of MTL approaches which cannot be assigned to any of the categories covered so far because they are based on different assumptions. In the following, we review some of them.

Supervised sparse coding

Classical sparse coding is an unsupervised learning framework which consists in expressing any input data as a sparse linear combination of a given dictionary. A dictionary is a collection of K vectors, also called atoms, which can be useful to describe inputs drawn from a prescribed distribution. A dictionary is in general overcomplete, that is, the number of atoms is usually larger than the dimensionality of the data. This scheme has been very useful in several fields where there is enough prior knowledge to build these dictionaries (e.g: a dictionary composed of a set of wavelets can be very useful for machine vision tasks). However, when there is not such prior knowledge about the dictionary, one can try to learn it from the data together with the sparse codes. This idea was first proposed in [153] and since then several optimization problems have been postulated to obtain the best set of dictionary and sparse codes to describe a dataset. In [3] the authors consider an approach which they call K-SVD in which the ℓ_0-norm of the codes is required to be less or equal than a prescribed natural number. Another approach is proposed in [3], where the authors employ the ℓ_1-norm to encourage sparsity

in the codes:

$$\min_{A,B} \left\{ \sum_{i=1}^{m} \left\| x_i - B^\top a_i \right\|_2^2 + \lambda \left\| a_i \right\|_1 \ : \ \left\| b_i \right\|_2 = 1, \, \forall i \in [m] \right\}, \tag{2.28}$$

where $B \in \mathbb{R}^{K \times d}$ (usually $K > d$) is the learned dictionary, and $A \in \mathbb{R}^{K \times m}$ contains the codes a_i learned for each input point x_i, $\forall i \in [m]$. The first term measures how accurate the reconstruction of the data points is, whereas the second term encourages sparsity in the codes a_i, $\forall i \in \{1 \dots m\}$. The hyperparameter λ ponders the trade off between these two terms. Note that this optimization problem resembles the one described previously in [107]; in fact if we change the constraint in eq. (2.28) by the Frobenius norm on B, the resultant problem can be expressed as an instance of the MTL approach.

In [125, 126] the authors adapt problem (2.28) to a supervised learning setting. Given a set of tasks, it is easy to extend that framework by assuming that all of them use the codes of the input data that have been computed using a common dictionary. Therefore, the relationship among tasks is defined through this dictionary. The resultant problem can be formulated as follows:

$$\min_{W,A,B} \sum_{t=1}^{T} \sum_{i=1}^{m} \left(\left\| a_{t,i}^\top w_t - y_i^t \right\|_2^2 + \lambda_0 \sum_{i=1}^{m} \left\| x_i^t - B^\top a_{t,i} \right\|_2^2 + \lambda_1 \left\| a_{t,i} \right\|_1 \right) + \lambda_2 \left\| W \right\|_{\mathrm{Fr}}^2,$$

$$\text{s.t.} \ : \ \left\| b_k \right\|_2 = 1, \, \forall k \in [K]$$

The first term is the loss function, where the instance codes A are used rather than the instances themselves. The second and the third terms are similar to those in sparse coding (eq. (2.28)) and finally, the fourth term prevents overfitting on the weight vectors W. The main advantage of this approach is that it is able to capture more complex non-linear features from the data. This comes with some drawbacks. One of them is that the inference process is not straightforward because it requires optimizing problem (2.28) for each test instance, in order to apply the resultant codes with the learned weight vectors. Another disadvantage is the necessity of tuning three hyperparameters, which makes cross-validation an expensive process.

Transfer learning by borrowing examples

In [114], the authors consider a visual object recognition scenario where there are many classification tasks (one for each object) such that each task $t \in [T]$ has to discriminate the instances belonging to class C_t from a separate background class B. The way the

authors propose to perform this transfer process is by directly sharing examples from similar categories. Making use of the example that the authors mention, using positive instances for the classification task *armchairs* can be useful for learning the classification task *sofas*. The aim of their problem formulation is to make an accurate selection of instances which helps in the learning process of each task. For each class C_t, a set of n_t positive instances is provided. In the same way, a set of b background instances is available making in total $b + \sum_{t=1}^{T} n_t = b + n$ instances. The assumption made by the authors is included into the model by considering a set of T auxiliary vectors $\omega^t \in \mathbb{R}^{b+n}$ so that $\omega_{i,t} \in [0, 1]$, $\forall i \in \{1 \dots b + n\}$, where $\omega_{i,t}$ indicates if instance i is going to be employed (borrowed) for learning task t. Therefore the b background instances and the n_t positive instances for task t will be set $\omega_{i,t} = 1$ and the remaining instances will have a $\omega_{i,t} \in [0, 1]$ that needs to be learned. This learning process is made by imposing some regularization on the vectors ω_t. In particular the authors employ a mixture between an $\ell_{2,1}$ and an ℓ_1-norm regularization. The former norm performs selection at a group (class) level whereas the latter selects a sparse set of instances independently of the class. The underlying idea is appealing because it is intuitive and simple to implement. However it has some drawbacks. One is that the resultant problem is not convex. Another important drawback is that the scenario of application is somewhat limited: the model can distinguish between each object class and a background class but if our objective were to distinguish among object classes, this approach will not be useful. However, in the setting they propose, the experiments carried out show that by making the explained assumption the model obtains more accurate results.

2.6. Discussion

Through this chapter we reviewed different optimization approaches for MTL problems. We started by exploring three central MTL frameworks that vary in the way they model the relationships between tasks. The first framework, based on quadratic regularizers, models relationships as the degree of proximity between the parameters defining the functions. The second one, based on sparse regularizers, assumes that the commonality between the tasks is defined by the subset of attributes utilized. The third framework, based on spectral regularizers, assumes that all tasks use a small set of learned linear features from the input data. All these MTL frameworks are driven by attempts to correctly model the relations between tasks and avoid negative transfer. We consider the last framework to be more general in the assumptions made, as it can account for

parametric similarity and attributes utilization between tasks, but also for more general relationships. This generality is desirable to avoid negative transfer. For this reason, in this thesis we will focus on this framework.

We have studied extensions to the previous frameworks. Many of those extensions have the aim of building models robust to negative transfer. Among them, we will focus on methods that are able to learn heterogeneous tasks by assuming that each one uses a sparse set of learned features. We will do that because this approach is more general than dirty models and task grouping, as discussed in Section 2.4.5.

Another group of extensions are motivated by using side information available about the relations between tasks. The literature has mostly focused on side information in which tasks are specified in a hierarchy. This is because that situation arises in many cases, such as in multiclass classification, where classes are organized in a tree.

Finally, in order to avoid narrowing our viewpoint, we have reviewed several other MTL approaches that do not belong to any of the previous frameworks. In doing so we reviewed sparse coding as an unsupervised way to learn non-linear features from data and its connection to MTL.

All methods reviewed in this chapter only deal with data with one aspect. This is a significant limitation when dealing with multi-aspect datasets described in the introduction. Traditional MTL methods can be applied to multi-aspect data by just considering the aspect of interest and ignoring the remaining ones. However, by ignoring the influence of other aspects characterizing the data, the model may disregard very useful information. In order to tackle this problem, side information regarding the aspects should be taken into account in the learning model. However, the reviewed approaches that consider side information are not thought to be able to deal with multi-aspect datasets, thus they can only express the relationship between tasks and aspects in a limited sense. Consequently, other ways of expressing these relations and incorporate them in a MTL model must be researched.

3. Multilinear Models Literature Review

One of the issues that emerged from Chapter 2 is the lack of methodology to exploit multitask learning on multi-aspect datasets. In this chapter we review multilinear models, as they have been successfully employed in unsupervised learning methods able to disentangle factors from different aspects of the data. Our motivation is to identify ideas that can be utilized in a multitask learning setting.

3.1. Introduction

Many machine learning problems tackled in the literature are developed around one question: for example object recognition is centered about the question “what object?”, many unsupervised learning techniques are based on “what subspace/manifold contains the data?”, and so forth. However, in real scenarios, there are usually several questions to ask from the same data. For example, in patient monitoring, one may be interested in considering if subject A walked today, how she did that (fast, limping, ...), in which time slot of the day, and on what type of terrain. Each of those questions leads to a different way to classify the same data. In these cases, multilinear algebra and multilinear models come in handy as a way to represent data and extract knowledge from it.

Multilinear algebra is the mathematical field that generalizes the concepts and methods of linear algebra to mathematical objects that have multiple modes. The term mode was introduced in [202] and denotes "a set of indices by which data might be classified". Thus, vectors are objects with one mode, because each of its elements can be referenced by one index; similarly, matrices have two modes. Tensors are a generalization of these concepts, allowing for an arbitrary number of modes.

In the design of many machine learning methods, matrices have played an important role as a way to model the relationship between two aspects of the data, such as for example users and movies in a recommendation system. However, when the data are described

by more than two aspects, matrices are unable to capture the relationships between them simultaneously. Following with the recommendation system scenario, one may wish to predict which rating a user will give to a certain movie at a certain time (see Fig. 3.1). This problem can be cast as that of learning a three order tensor which associates a rating (e.g. a number in the range $1-5$) to each triplet user/movie/time. Other examples of datasets which can be represented as tensors arise in many different research areas, ranging from computer vision, bioinformatics, natural language processing, to mention but a few. Due to the multimodal nature of such datasets, there has been an increasing interest in the study of multilinear algebra, as it provides a means to model interaction between any number of aspects in a natural way.

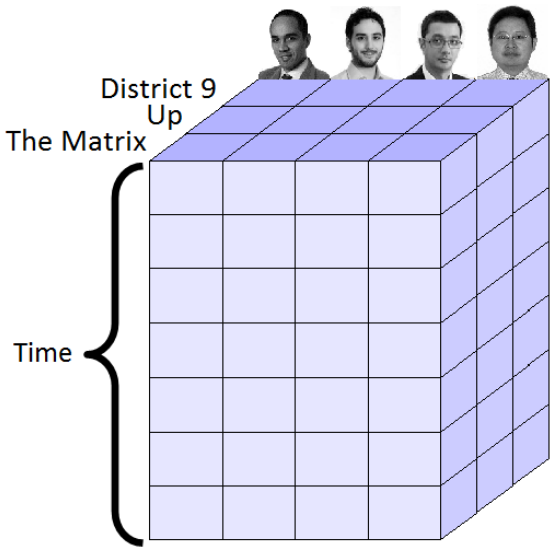

Figure 3.1.: Example of a tensor modelling the movie preferences of several subjects across time.

The outline of this literature review is as follows. In Section 3.2, after introducing some notation, we study different tensor decompositions which are generalizations of the singular value decomposition (SVD) for matrices. In Section 3.3 we focus on the Tucker decomposition, and the associated n-rank, reviewing several methods that are based on optimization problems where the n-rank of a tensor is constrained or regularized. In Section 3.4 we review some of the most important applications of tensors to machine learning problems, and finally in Section 3.5 we briefly summarize the conclusions and discuss the potential importance of multilinear algebra for this thesis.

3.2. Tensor decompositions

3.2.1. Multilinear concepts and notation

In this section we introduce several concepts and notation which are necessary to handle tensors. Let $N \in \mathbb{N}$ and let[1] $p_1, \ldots, p_N \geq 2$. An N-order tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N}$, is a collection of real numbers $(\boldsymbol{\mathcal{W}}_{i_1,\ldots,i_N} : i_n \in [p_n], n \in [N])$. Boldface Euler scripts, e.g. $\boldsymbol{\mathcal{W}}$, will be used to denote tensors of order higher than two. Vectors are 1-order tensors and will be denoted by lower case letters, e.g. x or a; matrices are 2-order tensors and will be denoted by upper case letters, e.g. W.

A mode-n fiber of a tensor $\boldsymbol{\mathcal{W}}$ is a vector composed of the elements of $\boldsymbol{\mathcal{W}}$ obtained by fixing all indices but one, corresponding to the n-th mode. This notion is a higher order analogue of columns (mode-1 fibers) and rows (mode-2 fibers) for matrices. The mode-n matricization (or unfolding) of $\boldsymbol{\mathcal{W}}$, denoted by $W_{(n)}$, is a matrix obtained by arranging the mode-n fibers of $\boldsymbol{\mathcal{W}}$ so that each of them is a column of $W_{(n)} \in \mathbb{R}^{p_n \times J_n}$, where $J_n := \prod_{k \neq n} p_k$. Note that the ordering of the columns is not important as long as it is used consistently.

A visualization of the concept of matricization is shown in Fig. 3.2.

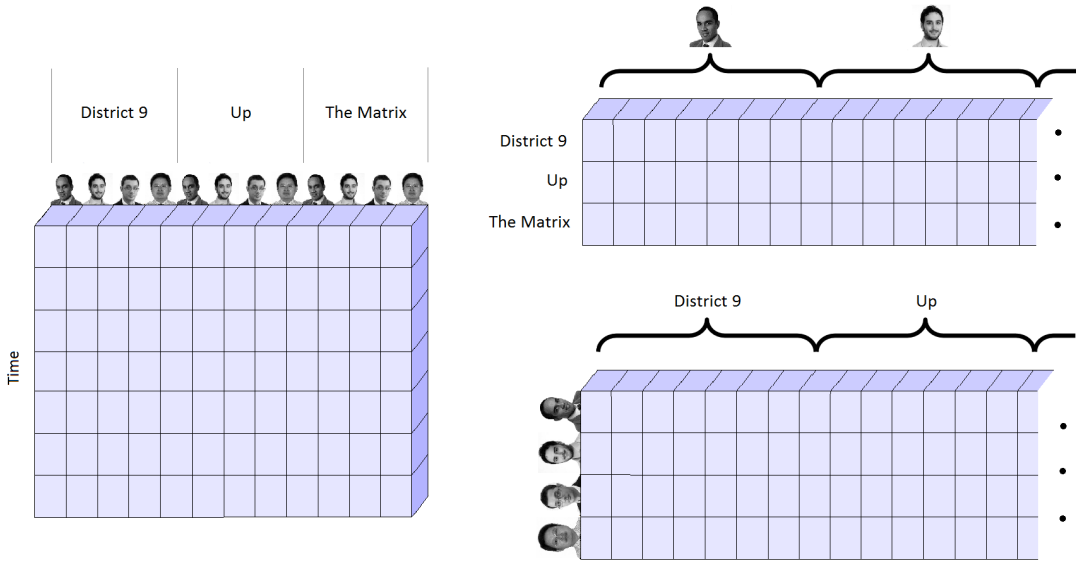

Figure 3.2.: The three matricizations of the tensor shown in Fig. 3.1.

[1] For simplicity we assume that $p_n \geq 2$ for every $n \in [N]$, otherwise we simply reduce the order of the tensor without loss of information.

Finally, the n-mode product of a tensor $\mathcal{W} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_N}$ with a matrix $A \in \mathbb{R}^{J \times p_n}$, denoted by $\mathcal{W} \times_n A$ is a new tensor of size $p_1 \times p_2 \times \ldots \times p_{n-1} \times J \times p_{n+1} \times \ldots \times p_N$, where each mode-$n$ fiber of the tensor is multiplied by A so

$$(\mathcal{W} \times_n A)_{i_1 \ldots i_{n-1}\, j\, i_{n+1} \ldots i_N} = \sum_{i_n=1}^{p_n} w_{i_1 i_2 \ldots i_N} a_{j i_n}.$$

3.2.2. Decompositions and the notion of rank

Representations and decompositions of mathematical objects are the base of many analytical methods. There are two main reasons that justify the study of decompositions for matrices and in general tensors. First, the storage of a large collection of values may become infeasible, thus it is necessary to describe them as a function of a smaller set of parameters. Second, decompositions may allow to explain or capture patterns from an a priori chaotic set of numbers, and use those patterns to perform inferences. One example that illustrates the latter point is matrix completion, a problem that has received a lot of attention in the last decade [32, 102, 165, 190]. It consists of recovering a matrix using some known linear measurements (e.g. some elements of the matrix). Given that the problem is ill-posed, it is reasonable to restrict it by looking for simple solutions, that is, matrices that have a low number of degrees of freedom. This measure can be formalized by the rank decomposition and the associated matricial rank. The rank of a matrix can be defined by any of the following three equivalent statements:

1. The minimum number of rank-one matrices required to sum to the original matrix.
2. The dimension of the space generated by the columns of the matrix.
3. The dimension of the space generated by the rows of the matrix.

They imply that any matrix $W \in \mathbb{R}^{p_1 \times p_2}$ of rank k can be represented as $W = A^\top B$, where $A \in \mathbb{R}^{k \times p_1}$, $B \in \mathbb{R}^{k \times p_2}$.

The notion of rank is used in many problems due to its convenient properties, thus it would be desirable to extend this notion when the number of modes is higher than 2. However, this is not straightforward because each of the definitions previously stated on the matricial rank leads to a different notion. In the following we will review them.

3.2.2.1. CP Rank

The CP rank (also known as the canonical polyadic, parallel factors, commonly PARAFAC, or simply tensor rank) is the generalization of the first definition of rank to tensors. Hence, it is the smallest number of rank-one tensors required to sum the original tensor.

That means that if the tensor $\mathcal{W}$ has a CP rank of K, then each of its elements can be expressed as:

$$w_{i_1,i_2,\ldots,i_N} = \sum_{k=1}^{K} u^1_{i_1,k} u^2_{i_2,k} \ldots u^N_{i_N,k},$$

or using $\circ$ to denote the outer product

$$\mathcal{W} = \sum_{k=1}^{K} u^1_k \circ u^2_k \circ \ldots \circ u^N_k \tag{3.1}$$

which is shown in Fig. 3.3.

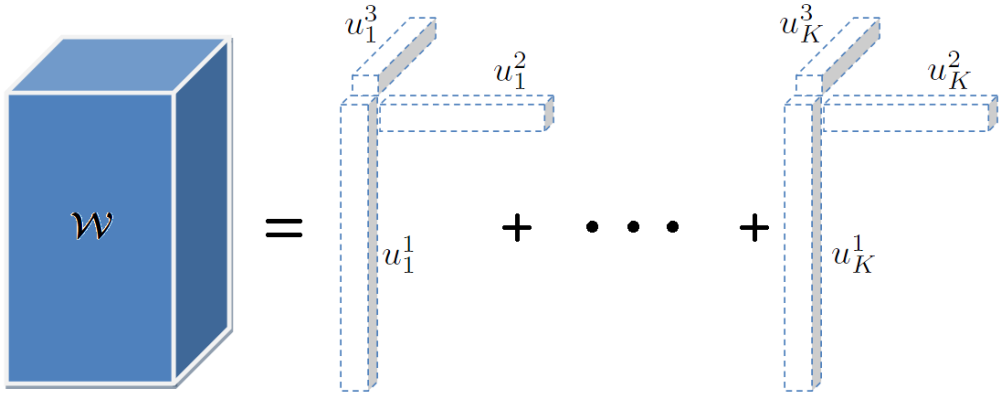

Figure 3.3.: CP decomposition of a 3 mode tensor.

The CP factorization was the first tensor decomposition proposed, and was introduced in [75, 76]. It poses an intuitive extension of the singular value decomposition of matrices, and the capacity of representing an N-mode tensor effectively in linear complexity with respect to N. However, the CP rank does not have many of the properties of the matricial rank. In particular, the following peculiarities of the CP rank are troublesome.

First, computing the rank of a tensor is an NP-hard problem [73]. In the case of matrices, the rank can be computed in polynomial time, but its use in optimization problems can be intricate because of the function being non-convex. In the general case of tensors, finding the global solution to an optimization problem involving the rank is utterly intractable.

The problem of approximating any tensor by a K-rank tensor (using any norm) is ill-posed in many cases [52], that is, finding the best K-rank approximation to a tensor has no solution in general. One notable exception to that is the approximation by a 1-rank tensor. Thus, one may try to do this by consecutively subtracting the best 1-rank

approximation, however doing this may even increase the CP rank [192]. 1-rank tensor approximation has also been studied in [73] and proved to be NP-hard.

Another unpleasant property is that the CP rank of a real tensor may be different over the set of real and complex numbers. That is, if the elements in u_k^n in eq. (3.1) are allowed to be complex, then the CP rank may be smaller than when u_k^n are restricted to be real.

One property of the CP decomposition of $N > 2$ mode tensors may be considered advantageous, that of uniqueness. Let us recall that a matrix $W \in \mathbb{R}^{p_1 \times p_2}$ of rank k can be represented as $W = A^\top B$, where $A \in \mathbb{R}^{k \times p_1}$, $B \in \mathbb{R}^{k \times p_2}$, but it can be represented as well as $W = \tilde{A}^\top \tilde{B}$, where $\tilde{A} = VA$ and $\tilde{B} = VB$, for any orthogonal matrix $V \in \mathbb{R}^{k \times k}$. In order to enforce uniqueness, orthogonal constraints should be added. In the case of higher-order tensors, the decomposition in eq. (3.1) is unique, ignoring scaling and permutation issues, under much weaker conditions. Kruskal was the first to study and prove those conditions for $N = 3$ [106]. To do so, he first defines the Kruskal rank $KR(\cdot)$ of a matrix to be the maximum number r such that any r columns of that matrix are linearly independent. Then a 3-mode tensor is uniquely decomposable if $KR(U^1) + KR(U^2) + KR(U^3) \geq 2k + 2$, where $U^n = \left[u_1^n,\, u_2^n, \ldots u_{p_i}^n\right]^\top$. The case for an arbitrary N is studied in [182], where the authors prove the general condition: $\sum_{n=1}^{N} KR(U^n) \geq 2k + N - 1$.
Minimizing the CP rank has been the base of several machine learning models. One example is found in [214], where the authors employ CP decomposition to infer a compact representation of images and videos which were seen as 3 and 4 mode tensors respectively. The algorithm they employ was based on an alternating least square (ALS) scheme. CP decomposition has also received considerable attention among the data mining and large scale machine learning community [158, 159].

3.2.2.2. Tucker rank

The last two definitions of rank of a matrix stated at the beginning of Section 3.2 make reference to the space spanned by the columns/rows of a matrix. More generally, with tensors we can think of the space generated by the mode-n fibers. It turns out that the dimensions of these spaces need no longer be the same. This observation leads to the Tucker rank, also known as the $n\text{-rank}(\boldsymbol{\mathcal{W}}) \in \mathbb{N}^N$, which is an N-element vector containing the rank of the i-th matricization in the i-th position: $n\text{-rank}(\boldsymbol{\mathcal{W}})_i = \text{rank}\left(W_{(i)}\right)$.

This definition of rank is closely related to the Tucker decomposition of a tensor, which is defined as:

$$w_{i_1,i_2,\ldots,i_N} = \sum_{j_1=1}^{k_1} \cdots \sum_{j_N=1}^{k_N} g_{j_1,\ldots,j_N} a^{(1)}_{i_1,j_1} \ldots a^{(N)}_{i_N,j_N}, \tag{3.2}$$

where $A^{(n)} \in \mathbb{R}^{p_n \times k_n}$, $n \in [N]$, are called the factor matrices, and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{k_1 \times \ldots \times k_N}$ is the core tensor and models the interaction between factors. Eq. (3.2) can be expressed more compactly using the n-mode product as

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{G}} \times_1 A^{(1)} \cdots \times_N A^{(N)}, \tag{3.3}$$

Fig. 3.4 shows the Tucker decomposition of a 3-mode tensor.

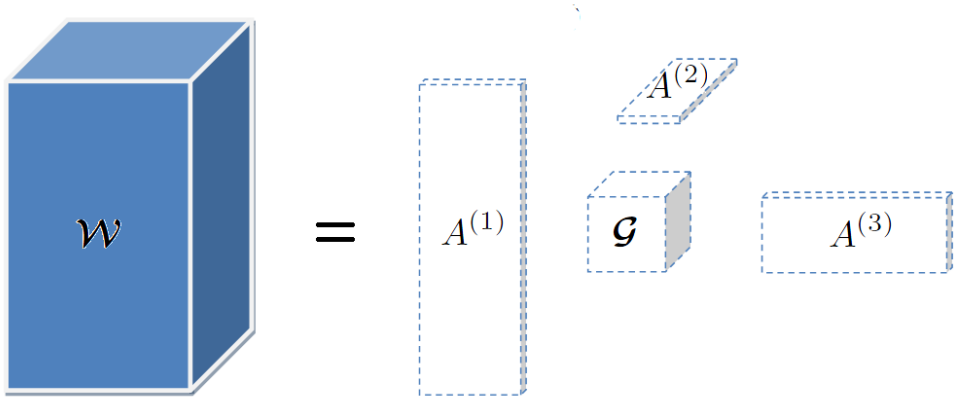

Figure 3.4.: Tucker decomposition of a 3 mode tensor.

It is clear that if a tensor $\boldsymbol{\mathcal{W}}$ can be represented using the Tucker decomposition as in eq. (3.3), then its Tucker rank is at most $[k_1, \ldots, k_N]^\top$. The reason is that the matricization of a tensor can be expressed in terms of the components of a Tucker decomposition as:

$$W_{(i)} = A^{(i)} G_{(i)} \left(A^{(N)} \otimes \cdots \otimes A^{(i+1)} \otimes A^{(i-1)} \otimes \cdots \otimes A^{(1)} \right)^\top, \tag{3.4}$$

where $\otimes$ denotes the Kronecker product, as introduced in Section 2.3.1.3. Given that the rank of $A^{(i)} \in \mathbb{R}^{p_i \times k_i}$ is at most k_i, and the rank of the product of matrices is always upper-bounded by the rank of each of the factors, we can conclude that $\text{Rank}\left(W_{(i)}\right) \leq \text{Rank}\left(A^{(i)}\right) \leq k_i$.

The converse statement, that is, that a tensor of n-rank $[r_1, \ldots, r_N]^\top$ can always be decomposed as in eq. (3.3) where $k_i = r_i$, $\forall i \in [N]$, also holds true [51].

This result leads to a corollary about the relationship between the n-ranks: none of the n-ranks can be bigger than the product of the remaining ones. We can see this by assuming that $\operatorname{rank}\left(W_{(1)}\right) > \prod_{i=2}^{N} \operatorname{rank}\left(W_{(i)}\right)$. Given that $G_{(1)} \in \mathbb{R}^{\operatorname{rank}\left(W_{(1)}\right) \times \prod_{i=2}^{N} \operatorname{rank}\left(W_{(i)}\right)}$, then $\operatorname{rank}\left(G_{(1)}\right) \leq \prod_{i=2}^{N} \operatorname{rank}\left(W_{(i)}\right)$. Then we arrive at a contradiction because $\operatorname{rank}\left(W_{(1)}\right) = \operatorname{rank}\left(A^{(1)} G_{(1)} \left(A^{(N)} \otimes \ldots A^{(2)}\right)^{\top}\right) \leq \operatorname{rank}\left(G_{(1)}\right) \leq \prod_{i=2}^{N} \operatorname{rank}\left(W_{(i)}\right) < \operatorname{rank}\left(W_{(1)}\right)$. Originally, the Tucker decomposition was proposed to extend Factor Analysis to more than two modes [202]. The objective of this framework is thus finding the factors that *explain* each of the modes of the data. In the last decade, with the growth and availability of multimodal data, this framework has been used and extended in many papers [140, 194, 207, 208, 210], in which the factors or principal components of each mode are found by applying the Tucker decomposition. Nevertheless, the Tucker decomposition and the n-rank have found many more applications than those that motivated its invention. The tractability of the computation of the n-rank of a tensor has made it a good choice in problems where computing or estimating the degrees of freedom of a tensor is required, such as in tensor completion. It is also worth noting a drawback of the Tucker decomposition with respect to the CP decomposition: the number of parameters involved is exponential with respect to the number of modes. That is, if the n-rank of a $p \times \ldots \times p$ tensor is $[r, \ldots, r]^{\top}$, then the Tucker decomposition requires a set of $r^N + Nrp$ parameters. We will focus on the optimization of problems considering the n-rank in Section 3.3.

3.2.2.3. Other notions of tensor decompositions

Even though the CP and the Tucker factorizations are arguably the most extended notions of decompositions for tensors, recently new ones have been proposed. They are the result of rethinking the previous tensor decompositions avoiding some of their drawbacks, namely the non-tractability in the case of the CP rank, and exponential number of parameters with respect to the number of modes in the Tucker decomposition. We briefly review two kinds of these decompositions and their associated definitions of ranks.

- Hierarchical singular value decomposition [67, 71, 154]. The main motivation of these approaches is to provide a representation as tractable as in the Tucker decomposition, but keeping the storage linear in N. To do so, this framework is based on building a binary tree, where the leaves represent the modes of the tensor

and each inner node represents the union of the modes in its successors, so that the root of the tree contains the set of all modes of the tensor. The hierarchical rank is defined as the set of ranks of the matricizations that are specified by each of the modes in each node of the tree. Note that in this case, matricizations may have more than one mode in both sides of the matrix. The motivation for this decomposition is that it allows one to represent a tensor by a set of smaller 3-mode tensors representing the relationship between any inner node and its two successors. Furthermore, in order to compute the new representation, it is only required to perform the SVD of auxiliary matrices that arise in a recursive process.

- Tensor-train (TT) decomposition, also known as matrix product state [155, 80]. In this decomposition, each input of the tensor $\boldsymbol{\mathcal{W}}$ is represented as:

$$w_{i_1,\ldots,i_N} = H^{1,i_1} H^{2,i_2} \ldots H^{N,i_N},$$

so that each dimension i of each mode n is represented by a matrix $H^{n,i} \in \mathbb{R}^{k_{n-1} \times k_n}$, where $k_0 = k_N = 1$, and the set of all k are known as the Tensor-Train rank. This approach resembles the hierarchical framework in that its representation is based on a set of three-modal tensors ($\boldsymbol{\mathcal{H}}^{1,\cdot}$, $\boldsymbol{\mathcal{H}}^{2,\cdot}, \ldots \boldsymbol{\mathcal{H}}^{N,\cdot}$). Furthermore, it has the advantage of avoiding recursion, making operations on tensors simpler to implement.

The main idea underlying the previous decompositions is to replace the core tensor in the Tucker decomposition by a set of interconnected three-order tensors. The nice properties that characterize these decompositions are achieved at the expense of getting rid of other desirable characteristics. For example, one objection that can be argued about these decompositions is that their complexity and the associated ranks depend on the splitting/ordering of the modes.

Given that in machine learning problems, the tensors considered have usually a low number of modes, this new notions of decompositions have not yet broken into scene, as they have in other fields such as physics and mathematics. Nevertheless, they may be an interesting alternative to consider in the future.

Other decompositions of tensors have been proposed, but they are tailored to specific applications, and for this reason they are not studied in this section. Some of them, useful for statistical relational learning, will be described in Section 3.4.3. An extended review on further tensor decompositions can be found in [100, Sec. 5], and more recently in [42, 68].

3.3. Optimization using the n-rank

Due to the advantages mentioned in Section 3.2.2.2, we will focus on the study of machine learning approaches based on the estimation and minimization of the n-rank of a tensor. In the following, we will review approaches which consider the following optimization problem:

$$\min_{\boldsymbol{\mathcal{W}}} f\left(\boldsymbol{\mathcal{W}}\right) + \left[\lambda_1, \lambda_2, \ldots \lambda_N\right] n{-}\mathrm{rank}\left(\boldsymbol{\mathcal{W}}\right), \tag{3.5}$$

or equivalently:

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{W}}} f\left(\boldsymbol{\mathcal{W}}\right) \\ &\mathrm{s.t}: n{-}\mathrm{rank}\left(\boldsymbol{\mathcal{W}}\right) \leq \left[r_1, r_2, \ldots r_N\right]^\top \end{aligned}, \tag{3.6}$$

where, $\lambda_i \geq 0$ and $r_i \in \mathbb{N}$, $\forall i \in N$, are hyperparameters that regulate the importance of the regularization of each mode of the tensor. It can be proved that for each value of λ_i in problem (3.5) there exists a value of r_i in problem (3.6) such that their solutions coincide. In that sense, both problems are equivalent.

In the following we review approaches to solve particular instances of problems (3.5, 3.6), distinguishing between non-convex and convex methods.

3.3.1. Non-convex methods

The Tucker decomposition, as many other decompositions, has the aim of representing or approximating a given mathematical object, in this case a tensor, using a lower number of parameters than in the original object. Therefore, the first question which arises when dealing with a representation is how to compute the best approximation of a given object. In particular, given a tensor $\boldsymbol{\mathcal{Y}}$, the objective is to find another tensor $\boldsymbol{\mathcal{W}}$ of n-ranks $\left[r_1,\, r_2, \ldots,\, r_N\right]^\top$ such that it is as close as possible to $\boldsymbol{\mathcal{Y}}$. This problem can be cast as problem (3.6), where f measures the distance between $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{W}}$, that is:

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{W}}} \left\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{W}}\right\|_{\mathrm{Fr}}^2 \\ &\mathrm{s.t.}: n - \mathrm{rank}\left(\boldsymbol{\mathcal{W}}\right) \leq \left[r_1, r_2, \ldots r_N\right]^\top \end{aligned}, \tag{3.7}$$

One can replace $\mathcal{W}$ by its Tucker decomposition component to reformulate the previous problem as:

$$\min_{\mathcal{G},\, A^{(1)},\ldots A^{(N)}} \left\| \mathcal{Y} - \mathcal{G} \times_1 A^{(1)} \ldots \times_N A^{(N)} \right\|_{\text{Fr}}^2, \tag{3.8}$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times \ldots \times r_N}$ is the core tensor and $A^{(i)} \in \mathbb{R}^{p_i \times r_i}\ \forall i \in [N]$ are the factor matrices of the approximation. Computing the optimal solution to this problem is extremely difficult, except for the trivial cases where $n\text{-rank}\,(\mathcal{Y}) \leq [r_1, r_2, \ldots r_N]^\top$, or where $N = 2$ leading to the matricial case in which one can apply the truncated SVD. Tucker was the first one to study this general problem and he proposed Higher-Order SVD (HOSVD) as an algorithm to approximate a sufficiently good solution [202]. HOSVD is based on computing the SVD of each n-matricization, storing in $A^{(n)}$ the r_n leading singular vectors. This process is detailed in Algorithm 3.1 and the solution obtained, $\mathcal{W}$, is sufficiently good according to the following result [67]:

$$\|\mathcal{W} - \mathcal{Y}\| \leq \sqrt{N} \left\| \mathcal{W}^{\text{best}} - \mathcal{Y} \right\|, \tag{3.9}$$

where $\mathcal{W}^{\text{best}}$ is the global solution of problem (3.8). Note that according to this bound, Algorithm 3.1 obtains the optimal solution when $n\text{-rank}\,(\mathcal{Y}) \leq [r_1, r_2, \ldots r_N]^\top$ and thus, $\mathcal{W}^{\text{best}} = \mathcal{Y}$.

Algorithm 3.1 HOSVD Algorithm to estimate the Tucker decomposition of a tensor $\mathcal{Y}$.
Input: N-mode tensor $\mathcal{Y} \in \mathbb{R}^{p_1, p_2, \ldots, p_N}$; desired rank for each mode $r_1,\ r_2, \ldots,\ r_N$
Output: N-mode (core) tensor $\mathcal{G} \in \mathbb{R}^{r_1, r_2, \ldots, r_N}$; factor matrices $A^{(1)},\ A^{(2)}, \ldots,\ A^{(N)}$

for $n = 1, \ldots, N$

- $A^{(n)} = r_n$ leading left singular vectors of $Y_{(n)}$

end
$\mathcal{G} = \mathcal{Y} \times_1 A^{(1)^\top} \times_2 A^{(2)^\top} \ldots \times_N A^{(N)^\top}$

This initial algorithm paved the way for the development of other improved methods grounded in iteratively reestimating the factor matrices [105, 96] using an alternating least square scheme. One of these methods is proposed in [109], and it is based on a power algorithm that updates at each step one factor matrix and stops when they converge. It is described in Algorithm 3.2 and it has the advantage of attaining a local optimum of problem (3.8).

Algorithm 3.2 Higher-Order Orthogonal Iteration (HOOI) Algorithm to estimate the Tucker decomposition of a tensor $\boldsymbol{\mathcal{Y}}$.

Input: N-mode tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{p_1,p_2,\ldots,p_N}$; desired rank for each mode $r_1,\ r_2,\ldots,\ r_N$
Parameters: $stop\_condition$.
Output: N-mode (core) tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1,r_2,\ldots,r_N}$; factor matrices $A^{(1)},\ A^{(2)},\ldots,\ A^{(N)}$
Initialization: Use HOSVD to initialize $A^{(n)}\ \forall n \in [N]$.

while not $stop\_condition$

- **for** $n = 1 \ldots N$
  - $\mathcal{Z} = \boldsymbol{\mathcal{Y}} \times_1 A^{(1)^\top} \times_2 A^{(2)^\top} \ldots \times_N A^{(N)^\top}$
  - $A^{(n)} = r_n$ leading left singular vectors of $Z_{(n)}$
- **end**

end
$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Y}} \times_1 A^{(1)^\top} \times_2 A^{(2)^\top} \ldots \times_N A^{(N)^\top}$

A more general problem than tensor approximation is that of tensor recovery, in which only a few linear measurements of the original tensor are known. That case corresponds to the following problem:

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{W}}} \left\| \mathcal{I}\left(\boldsymbol{\mathcal{Y}}\right) - \mathcal{I}\left(\boldsymbol{\mathcal{W}}\right) \right\|_2^2 \\ &\text{s.t} : n\text{−rank}\left(\boldsymbol{\mathcal{W}}\right) \leq \left[r_1, r_2, \ldots r_N\right]^\top \end{aligned}, \tag{3.10}$$

where $\mathcal{I} : \mathbb{R}^{p_1 \times \ldots \times p_N} \longrightarrow \mathbb{R}^m$ is a linear operator and m the number of linear measurements. A prominent example of the previous framework is tensor completion, where only a subset of the inputs of the tensor is known, and the problem consists in inferring the whole tensor assuming it is low rank in all its matricizations.

The problem in eq. (3.10) has been extensively studied in the literature for the case of matrices. One of the most common approaches is expressing the matrix to be optimized as the product of two matrices with r rows, $W = A^\top B$, where $W \in \mathbb{R}^{p_1 \times p_2}$, $A \in \mathbb{R}^{r \times p_1}$and $B \in \mathbb{R}^{r \times p_2}$, leading to the following non-convex optimization problem:

$$\min_{A,B} \left\| \mathcal{I}(A^\top B) - \mathcal{I}(Y) \right\|_2^2.$$

A solution of this problem can be found by means of alternating least squares (ALS). Extensions of this method to the Tucker decomposition are not straightforward. To see this, one can express each matricization of the tensor $W_{(n)}$ as the product of two matrices $W_{(n)} = A^{n\top} B^n$. Given that the inputs of $W_{(n)}$ and $W_{(\tilde{n})}$, $n \neq \tilde{n}$, are the

same (rearranged in a different way), the latent matrices of different matricizations are highly coupled. In a very recent work [217] the authors propose a way to overcome this by penalizing the Frobenius norm of the difference between each factorization, $A^{n\top}B^n$ and the learning tensor $\boldsymbol{\mathcal{W}}$, $\forall n \in [N]$, constraining the observations. Thus, the resulting problem is as follows.

$$\begin{array}{cc} \min\limits_{\boldsymbol{\mathcal{W}},A^n,B^n,n\in[N]} & \sum\limits_{n=1}^{N} \frac{\alpha_n}{2} \left\| W_{(n)} - A^{n\top}B^n \right\|_{\text{Fr}}^2 \\ \text{s.t}: & \mathcal{I}\left(\boldsymbol{\mathcal{Y}}\right) = \mathcal{I}\left(\boldsymbol{\mathcal{W}}\right) \end{array},$$

where α_n, $n \in [N]$ are hyperparameters. The problem is non-convex, although it is convex with respect to each of the elements, fixing the remaining ones. Hence, solutions to this problem can be obtained by using an ALS scheme.

Another framework for matrix completion constraining the rank is based on performing projected gradient descent, projecting at each step the current solution onto the set of matrices with rank smaller or equal than r. This approach, which has been called singular value projection, has been studied in [85], and although it is non-convex, theoretical bounds have been developed that guarantee its ability to recover the optimal solution provided that the linear operator $\mathcal{I}$ in problem (3.10) satisfies a set of conditions based on a restricted isometry property. Although these theoretical conditions do not usually hold in matrix completion problems, the empirical results are comparable to the state of the art, improving on it in some cases. The extension of this framework to tensors and the Tucker decomposition is again troublesome. In [163] the authors perform an approximate projection of the current solution onto the set of tensors whose n-rank is less or equal than $\boldsymbol{r}$ by applying the HOSVD algorithm. The theoretical results developed for matrices do not hold here because of the inexact projection carried out (recall the HOSVD property in eq. (3.9)).

One last non-convex methodology that has recently gained attention to solve rank constrained problems is the use Riemannian optimization methods. They are based on the fact that all matrices of fixed rank lie on a smooth manifold

$\mathcal{M}_r := \{W \in \mathbb{R}^{p_1 \times p_2} : \operatorname{rank}(W) = r\}$. Then the original rank-constrained problem can be expressed as an unconstrained one on the manifold $\mathcal{M}_r$, so that Riemannian optimization techniques can be employed. This observation was first exploited for problems involving rank constrained matrices in [137, 146, 205]. This framework has been extended to solve problem (3.6) for tensors in [103], by noticing that for any given tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N}$ of Tucker rank $[r_1, r_2, \ldots r_N]^\top$, there exists a neighbourhood

$\mathcal{D} \subset \mathbb{R}^{p_1 \times \cdots \times p_N}$ such that the HOSVD truncation of the tensors in $\mathcal{D}$ forms a smooth manifold.

3.3.2. Convex relaxations

One difficulty that arises when dealing with optimization on rank functions is due to its non-convexity. This implies that the solution found may not be optimal and is dependent on the initialization of the solver. It is therefore interesting to study a convex relaxation of the original problem. Such relaxations should be both tractable and provide a good approximation to the non-convex problems. A further advantage of convex problems is that they can be usually solved by efficient algorithms, as convex optimization methods have been and remain a deeply studied field.

In the context of matrices, convex relaxations for the rank have been much studied. In this line, many works provide theoretical and practical evidence that the trace norm (also known as nuclear norm, Schatten 1-norm and Ky-Fan r-norm) is the best convex surrogate for the rank. A useful tool to look for convex relaxations of a function is the notion of convex envelope. The convex envelope of a function f is a convex function g such that $g(x) \leq f(x)$, $\forall x \in \mathrm{dom}(f)$ so that there is no convex function h, such that both $h(x) \leq f(x)$, $\forall x \in \mathrm{dom}(f)$ and there exists some $y \in \mathrm{dom(f)}$ such that $h(y) > g(y)$. In other words, the convex envelope of a function is the tightest convex approximation from below it. One important result is shown in [61], where the author proves that the trace norm is the convex envelope for the rank of a matrix, considering the set of matrices in the unit spectral ball. In other papers, like [37], the authors prove bounds on the number of observable inputs required to recover the original low rank matrix by constraining the trace norm, and provide an alternative justification about the use of the trace norm as the best convex heuristic to recover low rank matrices. All these results have had practical use in many different areas, such as multitask learning, as shown in Section 2.3.3.2.

In the more general case of tensors, the state of the art is still in an early stage. The first paper which focused on convex relaxations to tensor problems is [117]. There, the authors study the tensor completion problem in the context of estimating missing entries in video and images. Given the success of the trace norm for matricial problems, they proposed to extend it to tensors by constraining the average of the trace norms of the

matricizations of the tensor, that is:

$$\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}} = \frac{1}{N}\sum_{n=1}^{N} \left\|W_{(n)}\right\|_{\mathrm{Tr}}, \tag{3.11}$$

which we will call tensor trace norm hereafter. The authors noted the inherent difficulty of optimizing a problem containing eq. (3.11), for all of the elements of the sum are interdependent. In order to avoid this, they propose a relaxation based on the introduction of N auxiliary tensors, $\boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^2, \ldots, \boldsymbol{\mathcal{M}}^N$, such that each $\boldsymbol{\mathcal{M}}^i$ is constrained to have a low trace norm in the i-th matricization, and the Frobenius norm of the difference between $\boldsymbol{\mathcal{W}}$ and each auxiliary tensor is penalized. The experiments reported in that paper compare this convex approach to other non-convex methods, such as HOSVD. The results show a clear advantage of the convex approach.

This initial work was followed by several papers providing better optimizations algorithms and analysis on the use of the tensor trace norm. In [64] the authors propose two optimization methods that allow one to decouple the terms in the regularizer (3.11). These algorithms are based on the Douglas-Rachford splitting method [55], and on its dual version, the Alternating Direction Method of Multipliers (ADMM) [56]. Furthermore, the authors provided convergence guarantees for each of the methods.

In [198] the authors propose a different function for approximating the average of the Tucker rank, with the objective of making the problem easier and more efficiently solvable. The regularizer is based on expressing the original N-mode tensor as a mixture of N tensors, $\boldsymbol{\mathcal{Z}}^1, \boldsymbol{\mathcal{Z}}^2, \ldots, \boldsymbol{\mathcal{Z}}^N$ where $\boldsymbol{\mathcal{Z}}^i$ is low rank in the i-th mode. The resultant regularizer looks as follows:

$$\Omega\left(\boldsymbol{\mathcal{W}}\right) = \min_{\sum_{n=1}^{N} \boldsymbol{\mathcal{Z}}^n = \boldsymbol{\mathcal{W}}} \frac{1}{N}\sum_{n=1}^{N} \left\|Z^n_{(n)}\right\|_{\mathrm{Tr}}, \tag{3.12}$$

which is an inf-convolution of N convex functions [224]. The problem is easier to optimize since the terms of eq. (3.12) are uncoupled and one can replace $\boldsymbol{\mathcal{W}}$ by $\sum_{n=1}^{N} \boldsymbol{\mathcal{Z}}^n$, and solve it using ALS type of methods. The authors made comparisons between approaches employing regularizers (3.11) and (3.12), and noted that the performance of this approach is superior when the tensor is low-rank in only one mode. In this particular case the regularizer (3.12) obtains better results, as it is able to detect the low rank mode. This statement was subsequently confirmed by theoretical analysis in [199].

Following with the study of the tensor trace norm, in [184] the authors use it as a way

to perform supervised learning, such as multitask learning, when data inputs can be represented as tensors, generalizing in this way the common case of 1-mode instances (vectors).

In [183] the authors propose the generalization of the Schatten norms to tensors by defining the Schatten-$\{p, q\}$ norm on $\mathbb{R}^{p_1 \times p_2 \times \ldots \times p_N}$ to be

$$\|\cdot\|_{\{p,q\}} : \boldsymbol{\mathcal{W}} \longmapsto \left(\frac{1}{N} \sum_{n=1}^{N} \left\| W_{(n)} \right\|_q^p \right)^{\frac{1}{p}},$$

for any $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_N}$. It is easy to see that this definition applied to 2-order tensors (matrices) is equivalent to the Schatten-q norm. The tensor trace norm corresponds to the particular case of the Schatten norm in which $p = q = 1$, $\|\cdot\|_{\{1,1\}} = \|\cdot\|_{\mathrm{Tr}}$. The authors of [183] also propose the use of ADMM as a convenient approach to optimize the resultant problem. In the analysis of the norm, the authors then focus on the study of the relationship between this quantity and the average of the n-ranks of a tensor. In particular, they prove that the tensor trace norm is a convex lower bound of the average of the $n-\text{ranks}$: $\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}} \leq \frac{1}{N} \sum_{n=1}^{N} \operatorname{rank}\left(W_{(n)}\right)$ for any $\left\{ \boldsymbol{\mathcal{W}} : \left\| W_{(n)} \right\|_{\mathrm{Sp}} \leq 1, \forall n \in [N] \right\}$, where $\|X\|_{\mathrm{Sp}}$ denotes the spectral norm of X. Let us recall that in the case of matrices, $N = 2$, the trace norm is the convex envelope of the rank, for all matrices in the unit spectral ball. However, showing whether $\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Tr}}$ is the convex envelope of $R(\boldsymbol{\mathcal{X}})$ for bigger values of N has remained an open question. In Chapter 7 of this thesis we will answer that question negatively, and provide an alternative convex method to approximate the average of the n-ranks of a tensor.

In [200] the authors analyze the capacity of the tensor trace norm regularizer to recover a low n-rank tensor. They do so by providing bounds on the difference between the estimated and the ground truth tensor under different conditions. One remarkable result obtained here is that an underlying N-mode tensor having p dimensions and rank r in each mode can be recovered from m Gaussian measurements when $m \geq \Omega\left(rp^{N-1}\right)$. The authors of [141] take over this line of research, proving that the previous statement is not only sufficient, but also necessary to recover the underlying tensor. The authors highlight that this is essentially the same bound that one would obtain by performing matrix completion on any matricization of the tensor. They point out that this phenomenon is quite common among other optimization problems which are composed of the sum of several regularizers imposing simultaneous constraints on the same variables. As a consequence, the authors propose an alternative to the tensor trace norm, which they call the Square Deal Norm. It consists in unfolding the tensor into a matrix that is *as square as possible*, that is, that the number of rows are as close as possible

to the number of columns. This is based on the observation that the matrix trace norm regularization is able to recover a $p \times p$ matrix of rank r employing $m = \Omega(rp)$ observations. Thus, this approach is able to recover the underlying tensor in $m = \Omega\left(r^{N/2}p^{N/2}\right)$ observations, becoming a better bound than the one obtained by the tensor trace norm. Note that this approach is useful when $N > 3$. In the case $N = 3$, this approach is equivalent to performing trace norm in one of the matricizations of the tensor.

The results presented in all previous papers show that convex relaxations are a powerful framework to use for tensor learning problems. Nevertheless, it is worth mentioning two important drawbacks that stem from the use of the trace norm, or in general any spectral function. First, it requires the storage of the whole tensor to be learned, which in some cases could be very large. What is more, most approaches reviewed in this section require storing $N + 1$ versions of the tensor. Second, the computation of the trace norm, which has to be done at every iteration, requires an SVD to be completed. This routine becomes the bottleneck of the previous methods, as it has a time complexity of $O(mp^2)$ for a $m \times p$ matrix [65]. Both drawbacks emerge when dealing with large tensors, leading to infeasible memory requirements and very high computational costs. There have been some attempts that start from a convex formulation of the problem and try to avoid these drawbacks [118]. However the resulting problem is no longer convex, thus the solution found may be far from the optimum.

3.4. Applications

In the final section of this review, we describe a list of applications of multilinear methods. Tensors and multilinear algebra are becoming ubiquitous in many areas of machine learning and related fields, therefore, this list is unavoidably incomplete, however it will give an overall picture of the state of the art.

3.4.1. Multilinear component analysis

One important use of tensors in the context of machine learning is to separate underlying factors in data that can be arranged as a tensor. This framework is first exploited in [194], where the authors consider the case where each data instance corresponds to a combination of a particular style and content. One example used in that work is a dataset composed of sounds of vowels (content) pronounced by different people (style). The authors arrange the instances onto a 3-mode tensor (its size is given by number

of subjects, number of vowels and dimensionality of the data) with the objective of applying bilinear approaches to untangle the underlying factors explaining both style and content. The authors describe and develop several applications to this framework, such as classification and synthesis of instances with a particular content and style.

This framework has been applied to several kinds of data. One example is the application to motion capture data in [207], where the authors set the hypothesis that each subject has a characteristic way to perform motion exercises, which they call motion signature. Their objective is therefore to extract human motion signatures among other constitutive factors inherent to human movement. Another common application of this framework is to separate modal information on faces [140]. Multilinear models in general have been broadly used to separate more than two modes in data. For example in [209] the authors decompose natural facial images according to 4 modes: identities, expressions, head poses and lighting conditions.

This framework has also been extended to account for different assumptions. One approach can be found in [211], where the authors propose multilinear independent component analysis, in which the factors extracted for each mode are not only uncorrelated but also statistically independent. Another approach is proposed in [113], where the authors develop a kernel extension for tensor decomposition algorithms.

All the previous works apply multilinear decomposition directly to the data, obtaining unsupervised learning models which can be seen as a higher-order generalization of principal (or independent) component analysis. Some of these papers, such as [194], propose algorithms to extend this model to perform classification. However the resulting algorithms present some limitations, such as being slow in the inference process, being unable to manage regression tasks, and making strong assumptions on the data (e.g. that there is one instance for each combination of factors).

3.4.2. Learning latent variable models

Latent variable models lie at the core of many machine learning methods. Their objective is to infer the value of hidden variables from the observable data. One way to do this is by applying maximum likelihood methods, but they may be computationally very expensive. A more feasible alternative for latent variable models is to apply the method of moments. This is based on computing a set of statistics, typically second, third and sometimes fourth order moments of the data and finding a model able to lead to the same statistics. It turns out that many latent variable models have some structure in their moments, which are 2, 3, and 4-mode symmetric tensors, which allows one to

infer the latent variables by decomposing them [5]. In particular, these tensors are symmetric orthogonal, which mean that they can be decomposed as in CP decomposition (eq. (3.1)), where $u_k^1 = \ldots = u_k^N = u_k$ and the vectors u_k, for $k \in [K]$ are orthonormal. These vectors are in fact the eigenvectors of the tensor, and can be obtained exactly by means of power methods.

One example of this is using tensor decomposition to perform independent component analysis [83] on some d-dimensional data. To do so, one can consider the fourth-order cumulant tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d \times d \times d \times d}$, whose entry i, j, k, l corresponds to the cross cumulant of those variables $\boldsymbol{\mathcal{W}}_{i,j,k,l} = \operatorname{cum}(x_i, x_j, x_k, x_l)$. If all d variables are independent, then all values of $\boldsymbol{\mathcal{W}}$ but the diagonal are 0. In the general case, the independent components are given by the fourth order cumulant tensor eigenvalues.

Other examples, such as Gaussian mixture models and Latent Dirichlet Allocation, are studied in [5], where the authors propose a general framework for learning latent variables using symmetric orthogonal decomposition.

3.4.3. Statistical relational learning

Statistical relational learning (SRL) is a framework which has the objective of learning the relations that may exist between a set of entities. Many problems can be categorized within this framework. Collaborative filtering, already mentioned as an example in Section 3.1 is a simple form of this, where there is a relation to learn between users and items. Other applications of SRL are finding relationships between users in social networks, learning bioinformatics ontologies by inferring relations between genes, and learning entities and relations among them from textual data, for the purpose of natural language processing.

Many models proposed for SRL are based on modelling entities by a set of latent features that need to be learned. For example collaborative filtering is usually modelled as a low rank matrix completion problem, where the two modes represent users and items, which is equivalent to learning a set of latent variables of each user and item and using the inner product to express the relation between them. This approach, which is called relational learning from latent attributes [90], is used to model the relation between only two modes, which may be limiting in many scenarios. In real recommender systems, there is usually more information available about the interaction between users and items, such as for example the time, the location, and the goal of the item consumption. In [2] the authors introduce the generalization of recommender systems that account for this information, and they call it context-aware collaborative filtering. The

authors model the scenario as a tensor, where two of the modes represent users and items, and further modes are added as contextual information becomes available. In [97], the authors propose the Tucker decomposition to model the problem. More recently, in [180] the authors focus on maximizing the mean average precision of the recommendations by learning the tensor using the CP decomposition. These and other works such as [167], introduce contextual information into the recommender system, improving significantly over conventional collaborative filtering approaches.

As introduced before, one of the aims of natural language processing is learning the relations between entities. Relational data is usually represented as triplets $\langle \text{subject}, \text{relation}, \text{object} \rangle$, thus many papers have modelled this problem as one of tensor completion, where the modes of the tensor represent subjects, relations and objects. The general decompositions we reviewed in Section 3.2 are symmetric, in the sense that all modes are treated equally. However in relational data, while subjects and objects make often reference to the same kind of entities, relations have a very different meaning. Because of that, non symmetric decompositions have been proposed for this problem, where the mode relations receives a special treatment. The first model proposed in that direction was DEDICOM (DEcomposition into DIretional COMponents) [15]. In that, the p elements in the mode subjects and objects are assumed to be the same, having a unique set of latent variables. In particular, each second (relation) mode s-slice, $W_{\cdot,s,\cdot} \in \mathbb{R}^{p \times p}$, is decomposed as:

$$W_{\cdot,s,\cdot} = AD_sRD_sA^\top,$$

where $A \in \mathbb{R}^{n \times k}$ contains k latent variables representing each of the p elements, $R \in \mathbb{R}^{k \times k}$ is an asymmetric matrix shared by all relations, and $D_s \in \mathbb{R}^{k \times k}$ is a diagonal matrix that regulates the importance of the latent components for each relation s. This model has proved to be useful in cases where the relations are very homogeneous, such as in international trade predictions [16]. However, it could be too restrictive to model general relational data. In order to relax the strong assumptions on the relations of the model, the authors of [148, 147] propose RESCAL, a model in which each relation s is represented by a full and potentially asymmetric matrix $R_s \in \mathbb{R}^{k \times k}$, leading to the following decomposition:

$$W_{\cdot,s,\cdot} = AR_sA^\top.$$

This model is a constrained version of the Tucker decomposition of a 3-mode tensor, where two of the factor matrices are forced to be similar, and the remaining one is fixed

to be the identity matrix. The resultant method offers state of the art results in many relational data sets. Remarkably it achieves impressive results on the YAGO 2 ontology, a collection of millions of entities and relations among them, in a very efficient way [149]. In a recent paper, [90], the authors propose several changes over RESCAL. In this new model not only third-order interactions are modelled, but also first and second order between entities and relations; furthermore, matrices R_s are decomposed into further latent factors, thus reducing the number of learning parameters; finally they use logistic loss instead of square loss, modelling directly the probabilities of the events. These decisions are supported by empirical results.

3.5. Discussion

In this chapter we reviewed different tensor decompositions, their advantages and disadvantages. We paid special attention to the CP and the Tucker decomposition and concluded that in terms of tractability the latter was more convenient. We reviewed ways of controlling and minimizing the complexity of a tensor by constraining and regularizing its Tucker rank, and finally we reviewed some major areas within machine learning where tensors are applied.

Among the areas of application of tensors, the ones in Section 3.4.1 consider multi-aspect data directly. There, the focus is on unsupervised learning methods able to decouple the explanatory factors in each of the modes (aspects) of the data. Even though a few supervised learning methods were proposed, [194] (reviewed also in Section 3.4.1), these impose strong and often unrealistic assumptions about the data. One of the objectives of this thesis is to explore this gap. After this review, it has become clear that tensors are versatile structures that are useful to model situations involving multi-aspect data. Therefore, combination of ideas from multilinear algebra and multitask learning could be useful in the exploration of this problem.

Another interesting topic that emerged in this review is that of optimization methods for tensor recovery. While these have been deeply studied in matrices, many questions remain open for the more general case of tensors. One particularly interesting is that of finding convex envelopes, or more generally tight convex surrogates, to the notions of ranks of tensors. Finding tight convex surrogates of ranks are highly desirable, as they may lead to tractable optimization problems whose solutions are as similar as possible to the original problems.

4. Sparse Coding Multitask Learning

In this chapter we tackle the question of whether we can leverage the commonalities between tasks even in the case when their relation is unclear and there is no side information. By the time we started working on this question there were two main directions to account for negative transfer: dirty models (reviewed in Section 2.4.2) and task grouping (reviewed in Section 2.4.3). Whereas they are successful at the problems they are designed for, they are not general enough to avoid negative transfer in many cases. For example, dirty models cannot account for learning commonalities between several groups of tasks, whereas task grouping models cannot account for more general intra and inter-groups relationships and are very sensitive to the hyperparameters.

In this chapter we research an alternative method based on the assumption that tasks can be well approximated by a sparse linear combination of the atoms of a learned dictionary. This assumption naturally subsumes the ones in the previous frameworks, while being able to consider more general scenarios where groups of tasks are fuzzy or not well defined. While we developed the framework proposed in this chapter, a method based on similar ideas appeared [107]. We advance the research on this line, providing probabilistic analysis and studying its application to transfer learning problems, among others contributions.

4.1. Problem Statement

The last decade has witnessed many efforts in the machine learning community to exploit assumptions of sparsity in the design of algorithms. A central development in this respect is the Lasso [196], which estimates a linear predictor in a high dimensional space under a regularizing ℓ_1-penalty. Theoretical results guarantee a good performance of this method under the assumption that the vector corresponding to the underlying predictor is sparse, or at least has a small ℓ_1-norm, see e.g. [29] and references therein.

In this work we consider the case where the predictors are linear combinations of the atoms of a dictionary of linear functions on a high or infinite dimensional space, and we assume that we are free to choose the dictionary. We will show that a principled choice is possible if there are many learning problems, or "tasks", and there exists a dictionary allowing sparse representations of all or most of the underlying predictors. In such a case we can exploit the larger quantity of available data to estimate the "good" dictionary and still reap the benefits of the Lasso for the individual tasks. The main contribution of this chapter is to provide theoretical and experimental justification of this claim, both in the domain of multitask learning, where the new representation is applied to the tasks from which it was generated, and in the domain of transfer learning, where the dictionary is applied to new tasks of the same environment.

Our work combines ideas from sparse coding [153], multitask learning [6, 8, 10, 21, 35, 58, 128] and transfer learning [19, 195]. There is a vast literature on these subjects, with several related works reviewed in Chapter 2. Transfer learning was proposed in [19] and an error analysis is provided therein, showing that a common representation which performs well on the training tasks will also generalize to new tasks obtained from the same "environment". The precursors of the analysis presented here are [128] and [130]. The first paper provides a bound on the reconstruction error of sparse coding and may be seen as a special case of the ideas presented here when the sample size is infinite. The second paper provides a transfer learning analysis of the multitask feature learning method in [8].

A method similar to the one presented here was recently proposed within the multitask learning setting [107]. Our probabilistic analysis support the empirical results obtained in that work. Furthemore, here we highlight the connection between sparse coding and multitask learning and address the problem of transfer learning.

The paper is organized in the following manner. In Section 4.2, we set up our notation and introduce the learning problem. In Section 4.3, we present our learning bounds for multitask learning and transfer learning. In Section 4.4 we report on numerical experiments. Finally, we present concluding remarks in Section 4.5.

4.2. Method

In this section, we turn to a technical exposition of the proposed method, introducing some necessary notation on the way.

Let H be a finite or infinite dimensional Hilbert space with inner product $\langle \cdot, \cdot \rangle$, norm

$\|\cdot\|$, and fix an integer K. We study the problem

$$\min_{D\in\mathcal{D}_K} \frac{1}{T}\sum_{t=1}^{T} \min_{c\in\mathcal{C}_\alpha} \frac{1}{m}\sum_{i=1}^{m} \ell\left(\left\langle Dc, x_i^t\right\rangle, y_i^t\right), \tag{4.1}$$

where

- $\mathcal{D}_K$ is the set of K-dimensional dictionaries (or simply dictionaries), which means that every $D \in \mathcal{D}_K$ is a linear map $D : \mathbb{R}^K \to H$, such that $\|De_k\| \leq 1$ for every one of the canonical basis vectors e_k of $\mathbb{R}^K$. The number K can be regarded as one of the regularization parameters of our method.
- $\mathcal{C}_\alpha$ is the set of code vectors c in $\mathbb{R}^K$ satisfying $\|c\|_1 \leq \alpha$. The ℓ_1-norm constraint implements the assumption of sparsity and α is the other regularization parameter. Different sets $\mathcal{C}_\alpha$ could be readily used in our method, such as those associated with ℓ_p-norms.
- $\mathbf{Z} = ((x_i^t, y_i^t) : 1 \leq i \leq m, 1 \leq t \leq T)$ is a dataset on which our algorithm operates. Each $x_i^t \in H$ represents an input vector, and y_i^t is a corresponding real valued label. We also write $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) = \left(\mathbf{z}^1, \ldots, \mathbf{z}^T\right) = \left((X^1, y^1), \ldots, \left(X^T, y^T\right)\right)$ with $X^t = (x_1^t, \ldots, x_m^t)$ and $y^t = (y_1^t, \ldots, y_m^t)$. The index t identifies a learning task, and $\mathbf{z}^t$ are the corresponding training points, so the algorithm operates on T tasks, each of which is represented by m example pairs.
- ℓ is a loss function where $\ell(y, y')$ measures the loss incurred by predicting y when the true label is y'. We assume that ℓ has values in $[0, 1]$ and has Lipschitz constant L in the first argument for all values of the second argument.

The minimum in (4.1) is zero if the data are generated according to a noise-less model which postulates that there is a "true" dictionary $D^* \in \mathcal{D}_{K^*}$ with K^* atoms and vectors $c_1^*, \ldots, c_T^*$ satisfying $\|c_t^*\|_1 \leq \alpha^*$, such that an input $x \in H$ generates the label $y = \langle D^* c_t^*, x\rangle$ in the context of task t. If $K \geq K^*$ and $\alpha \geq \alpha^*$ then the minimum in (4.1) is zero. In Section 4.4, we will present experiments with such a generative model, when noise is added to the labels, that is $y = \langle D^* c_t^*, x\rangle + \xi$ with $\xi \sim \mathcal{N}(0, \sigma)$, the standard normal distribution.

The method (4.1) should output a minimizing $D(\mathbf{Z}) \in \mathcal{D}_K$ as well as a minimizing $c_1(\mathbf{Z}), \ldots, c_T(\mathbf{Z})$ corresponding to the different tasks. Our implementation, described in Section 4.4.1, does not guarantee exact minimization, because of the non-convexity of the problem. Below predictors are always linear, specified by a vector $w \in H$,

predicting the label $\langle w, x\rangle$ for an input $x \in H$, and a learning algorithm is a rule which assigns a predictor $A(\mathbf{z})$ to a given data set $\mathbf{z} = ((x_i, y_i) : 1 \leq i \leq m) \in (H \times \mathbb{R})^m$.

4.3. Learning bounds

In this section, we present learning bounds for method (4.1), both in the multitask and transfer learning settings, and discuss the special case of sparse coding.

4.3.1. Multitask learning

Let $\mu_1, \ldots, \mu_T$ be probability measures on $H \times \mathbb{R}$. We interpret $\mu_t(x, y)$ as the probability of observing the input/output pair (x, y) in the context of task t. For each of these tasks an i.i.d. training sample $\mathbf{z}^t = ((x_i^t, y_i^t) : 1 \leq i \leq m)$ is drawn from $(\mu_t)^m$ and the ensemble $\mathbf{Z} \sim \prod_{t=1}^T \mu_t^m$ is input to algorithm (4.1). Upon returning of a minimizing $D(\mathbf{Z})$ and $c_1(\mathbf{Z}), \ldots, c_T(\mathbf{Z})$, we will use the predictor $D(\mathbf{Z})\, c_t(\mathbf{Z})$ on the t-th task. The average over all tasks of the expected error incurred by these predictors is

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\left\langle D(\mathbf{Z})\, c_t(\mathbf{Z}), x\right\rangle, y\right)\right].$$

We compare this *task-average risk* to the minimal analogous risk obtainable by any dictionary $D \in \mathcal{D}_K$ and any set of vectors $c_1, \ldots, c_T \in \mathcal{C}_\alpha$. Our first result is a bound on the excess risk.

Theorem 4.3.1. *Let $\delta > 0$ and let $\mu_1, \ldots, \mu_T$ be probability measures on $H \times \mathbb{R}$. With probability at least $1 - \delta$ in the draw of $\mathbf{Z} \sim \prod_{t=1}^T \mu_t^m$ we have*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\left\langle D(\mathbf{Z})\, c_t(\mathbf{Z}), x\right\rangle, y\right)\right]$$

$$-\inf_{D\in\mathcal{D}_K} \frac{1}{T}\sum_{t=1}^{T} \inf_{c\in\mathcal{C}_\alpha} \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\left\langle Dc, x\right\rangle, y\right)\right]$$

$$\leq L\alpha\sqrt{\frac{2S_{\mathrm{Tr}}(\mathbf{X})(K+12)}{mT}} + L\alpha\sqrt{\frac{8S_{\mathrm{Sp}}(\mathbf{X})\ln(2K)}{m}} + \sqrt{\frac{8\ln 4/\delta}{mT}},$$

where $S_{\mathrm{Tr}}(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^T \operatorname{tr}\left(\hat{\Sigma}(X^t)\right)$ and $S_{\mathrm{Sp}}(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^T \lambda_{\max}\left(\hat{\Sigma}(X^t)\right)$. Here $\hat{\Sigma}(X^t)$ is the empirical covariance of the input data for the t-th task, $\operatorname{tr}(\cdot)$ denotes the trace and $\lambda_{\max}(\cdot)$ the largest eigenvalue.

We state several implications of this theorem.

1. The quantity $S_{\mathrm{Tr}}(\mathbf{X})$ appearing in the bound is just the average square norm of the input data points, while $S_{\mathrm{Sp}}(\mathbf{X})$ is roughly the average inverse of the observed dimension of the data for each task. Suppose that $H = \mathbb{R}^d$ and that the data-distribution is uniform on the surface of the Euclidean unit ball. Then $S_{\mathrm{Tr}}(\mathbf{X}) = 1$ and for $m \ll d$ it follows from Levy's isoperimetric inequality (see e.g. [111]) that $S_{\mathrm{Sp}}(\mathbf{X}) \approx 1/m$, so the corresponding term behaves like $\sqrt{\ln K}/m$. If the minimum in (4.1) is small and T is large enough for this term to become dominant then there is a significant advantage of the method over learning the tasks independently. If the data is essentially low dimensional, then $S_{\mathrm{Sp}}(\mathbf{X})$ will be large, and in the extreme case, if the data is one-dimensional for all tasks then $S_{\mathrm{Sp}}(\mathbf{X}) = S_{\mathrm{Tr}}(\mathbf{X})$ and our bound will always be worse by a factor of $\ln K$ than standard bounds for independent single task learning as in [18]. This makes sense, because for low dimensional data there can be little advantage to multitask learning.
2. In the regime $T < K$ the bound is dominated by the term of order $\sqrt{S_{\mathrm{Tr}}(\mathbf{X})\, K/mT}$ $> \sqrt{S_{\mathrm{Tr}}(\mathbf{X})/m}$. This is easy to understand, because the dictionary atoms De_k can be chosen independently, separately for each task, so we could at best recover the usual bound for linear models and there is no benefit from multitask learning.
3. Consider the noiseless generative model mentioned in Section 4.2. If $K \geq K^*$ and $\alpha \geq \alpha^*$ then the minimum in (4.1) is zero. In the bound the overestimation of K^* can be compensated by a proportional increase in the number of tasks considered and an only very minor increase of the sample size m, namely $m \to (\ln K^*/\ln K)\, m$.
4. Keeping in mind the phenomenon of negative transfer, suppose that we concatenate two sets of tasks so that tasks in the first group are not related to tasks in the second group. If the tasks are generated by the model described in Section 4.2 then the resulting set of tasks is also generated by such a model, obtained by concatenating the lists of atoms of the two true dictionaries D_1^* and D_2^* to obtain the new dictionary D^* of length $K^* = K_1^* + K_2^*$ and taking the union of the set of generating vectors $\{c_t^{*1}\}_{t=1}^T$ and $\{c_t^{*2}\}_{t=1}^T$, extending them to $\mathbb{R}^{K_1^*+K_2^*}$ so that the supports of the first group are disjoint from the supports of the second group. If $T_1 = T_2$, $K_1^* = K_2^*$ and we train with the correct parameters, then the excess risk for the total task set increases only by the order of $1/\sqrt{m}$, independent of K, despite the fact that the tasks in the second group are in no way related to those

in the first group. Our method has the property of finding the right clusters of mutually related tasks.

5. Consider the alternative method of subspace learning (SL) where $\mathcal{C}_\alpha$ is replaced by a Euclidean ball of radius α. With similar methods one can prove a bound for SL where, apart from slightly different constants, $\sqrt{\ln K}$ above is replaced by K. SL will be successful and will outperform the proposed method, whenever K can be chosen small, with $K < m$ and the vector c_t^* utilizes the entire span of the dictionary. For large values of K, a correspondingly large number of tasks and sparse c_t^* the proposed method will be superior.

The proof of Theorem 4.3.1, which is given in Section B.2.1 of the supplementary appendix, uses standard methods of empirical process theory, but also employs a concentration result related to Talagrand's convex distance inequality to obtain the crucial dependence on $S_{\mathrm{Sp}}(\mathbf{X})$. At the end of Section B.2.1 we sketch applications of the proof method to other regularization schemes, such as the one presented in [107], in which the Frobenius norm on the dictionary D is used in place of the $\ell_{2,\infty}$-norm employed here and the $\ell_{1,1}$ norm on the coefficient matrix $[c_1, \ldots, c_T]$ is used in place of the $\ell_{1,\infty}$.

4.3.2. Transfer learning

There is no absolute way to assess the quality of a learning algorithm. Algorithms may perform well on one kind of task, but poorly on another kind. It is important that an algorithm performs well on those tasks to which it is likely to be applied. To formalize this, [19] introduced the notion of an *environment*, which is a probability measure $\mathcal{E}$ on the set of tasks. Thus $\mathcal{E}(\tau)$ is the probability of encountering the task τ in the environment $\mathcal{E}$, and $\mu_\tau(x, y)$ is the probability of finding the pair (x, y) in the context of the task τ.

Given $\mathcal{E}$, the *transfer risk* (or simply risk) of a learning algorithm A is defined as follows. We draw a task from the environment, $\tau \sim \mathcal{E}$, which fixes a corresponding distribution μ_τ on $H \times \mathbb{R}$. Then we draw a training sample $\mathbf{z} \sim \mu_\tau^m$ and use the algorithm to compute the predictor $A(\mathbf{z})$. Finally we measure the performance of this predictor on test points $(x, y) \sim \mu_\tau$. The corresponding definition of the transfer risk of A reads as

$$R_{\mathcal{E}}(A) = \mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{\mathbf{z}\sim\mu_\tau^m}\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\langle A(\mathbf{z}), x\rangle, y\right)\right] \tag{4.2}$$

which is simply the expected loss incurred by the use of the algorithm A on tasks drawn from the environment $\mathcal{E}$.

For any given dictionary $D \in \mathcal{D}_K$ we consider the learning algorithm A_D, which for $\mathbf{z} \in \mathcal{Z}^m$ computes the predictor

$$A_D(\mathbf{z}) = D \arg\min_{c \in \mathcal{C}_\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell\left(\langle Dc, x_i \rangle, y_i\right). \tag{4.3}$$

Equivalently, we can regard A_D as the Lasso operating on data preprocessed by the linear map $D^\top$, the adjoint of D.

We can make a single observation of the environment $\mathcal{E}$ in the following way: one first draws a task $\tau \sim \mathcal{E}$. This task and the corresponding distribution μ_τ are then observed by drawing an i.i.d. sample $\mathbf{z}$ from μ_τ, that is $\mathbf{z} \sim \mu_\tau^m$. For simplicity the sample size m will be fixed. Such an observation corresponds to the draw of a sample $\mathbf{z}$ from a probability distribution $\rho_{\mathcal{E}}$ on $(H \times \mathbb{R})^m$ which is defined by

$$\rho_{\mathcal{E}}(\mathbf{z}) := \mathbb{E}_{\tau \sim \mathcal{E}}\left[(\mu_\tau)^m(\mathbf{z})\right]. \tag{4.4}$$

To estimate an environment a large number T of independent observations is needed, corresponding to a vector $\mathbf{Z} = \left(\mathbf{z}^1, \ldots, \mathbf{z}^T\right) \in \left((H \times \mathbb{R})^m\right)^T$ drawn i.i.d. from $\rho_{\mathcal{E}}$, that is $\mathbf{Z} \sim (\rho_{\mathcal{E}})^T$.

We now propose to solve the problem (4.1) with the data $\mathbf{Z}$, ignore the resulting $c_t(\mathbf{Z})$, $\forall t \in [T]$, but retain the dictionary $D(\mathbf{Z})$ and use the algorithm $A_{D(\mathbf{Z})}$ on future tasks drawn from the same environment. The performance of this method can be quantified as the transfer risk $R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right)$ as defined in equation (4.2) and again we are interested in comparing this to the risk of an ideal solution based on complete knowledge of the environment. For any fixed dictionary D and task τ the best we can do is to choose $c \in \mathcal{C}$ so as to minimize $\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\langle Dc, x\rangle, y\right)\right]$, so the best is to choose D so as to minimize the average of this over $\tau \sim \mathcal{E}$. The quantity

$$R_{\mathrm{opt}} = \min_{D \in \mathcal{D}_K} \mathbb{E}_{\tau \sim \mathcal{E}} \min_{c \in \mathcal{C}_\alpha} \mathbb{E}_{(x,y)\sim\mu_\tau} \ell\left[\left(\langle Dc, x\rangle, y\right)\right]$$

thus describes the optimal performance achievable under the given constraint. Our second result is

Theorem 4.3.2. *With probability at least $1-\delta$ in the multisample $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) \sim \rho_{\mathcal{E}}^T$ we have*

$$R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right) - R_{\mathrm{opt}} \leq L\alpha K \sqrt{\frac{2\pi S_{\mathrm{Tr}}(\mathbf{X})}{T}}$$

$$+4L\alpha\sqrt{\frac{S_{\mathrm{Sp}}\left(\mathcal{E}\right)\left(2+\ln K\right)}{m}}+\sqrt{\frac{8\ln 4/\delta}{T}},$$

where $S_{\mathrm{Tr}}\left(\mathbf{X}\right)$ *is as in Theorem 4.3.1 and* $S_{\mathrm{Sp}}\left(\mathcal{E}\right):=\mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{(X,y)\sim\mu_\tau^m}\lambda_{\max}\left(\hat{\Sigma}\left(X\right)\right)$.

We discuss some implications of the above theorem.

1. The interpretation of $S_{\mathrm{Sp}}\left(\mathcal{E}\right)$ is analogous to that of $S_{\mathrm{Sp}}\left(\mathbf{X}\right)$ in the bound for Theorem 4.3.1.
2. In the regime $T\leq K^2$ the result does not imply any useful behaviour. On the other hand, if $T\gg K^2$ the dominant term in the bound is of order $\sqrt{S_{\mathrm{Sp}}\left(\mathcal{E}\right)/m}$.
3. There is an important difference with the multitask learning bound, namely in Theorem 4.3.2 we have $\sqrt{T}$ in the denominator of the first term of the excess risk, and not $\sqrt{mT}$ as in Theorem 4.3.1. This is because in the setting of transfer learning there is always a possibility of being misled by the draw of the training tasks. This possibility can only decrease as T increases – increasing m does not help.

The proof of Theorem 4.3.2 is given in Section B.2.2 of the supplementary appendix and follows the method outlined in [128]: one first bounds the estimation error for the expected empirical risk on future tasks, and then combines this with a bound of the expected true risk by said expected empirical risk. The term $K/\sqrt{T}$ may be an artefact of our method of proof and the conjecture that it can be replaced by $\sqrt{K/T}$ seems plausible.

4.3.3. Connection to sparse coding

We discuss a special case of Theorem 4.3.2 in the limit $m\to\infty$, showing that it subsumes the sparse coding result in [130]. To this end, we assume the noiseless generative model $y_i^t=\langle w_t,x_i^t\rangle$ described in Section 4.2, that is $\mu(x,y)=p(x)\delta(y,\langle w,x\rangle)$, where p is the uniform distribution on the sphere in $\mathbb{R}^d$ (i.e. the Haar measure). In this case the environment of tasks is fully specified by a measure ρ on the unit ball in $\mathbb{R}^d$ from which a task $w\in\mathbb{R}^d$ is drawn and the measure μ is identified with the vector w. Note that we do not assume that these tasks are obtained as sparse combinations of some dictionary. Under the above assumptions and choosing ℓ to be the square loss, we have that $\mathbb{E}_{(x,y)\sim\mu_t}\ell(\langle w,x\rangle,y)=\|w_t-w\|^2$. Consequently, in the limit of $m\to\infty$ method

(4.1) reduces to a constrained version of sparse coding [153], namely

$$\min_{D\in\mathcal{D}_K}\frac{1}{T}\sum_{t=1}^{T}\min_{c\in\mathcal{C}_\alpha}\|Dc-w_t\|^2.$$

In turn, the transfer error of a dictionary D is given by the quantity $R(D):=\min_{c\in\mathcal{C}_\alpha}\|Dc-w\|^2$ and $R_{\mathrm{opt}}=\min_{D\in\mathcal{D}_K}\mathbb{E}_{w\sim\rho}\min_{c\in\mathcal{C}_\alpha}\|Dc-w\|^2$. Given the constraints $D\in\mathcal{D}_K$, $c\in\mathcal{C}_\alpha$ and $\|x\|\leq 1$, the square loss $\ell(y,y')=(y-y')^2$, evaluated at $y=\langle Dc,x\rangle$, can be restricted to the interval $y\in[-\alpha,\alpha]$, where it has the Lipschitz constant $2(1+\alpha)$ for any $y'\in[-1,1]$, as is easily verified. Since $S_{\mathrm{Tr}}(\mathbf{X})=1$ and $S_{\mathrm{Sp}}(\mathcal{E})<\infty$, the bound in Theorem 4.3.2 becomes

$$R(D)-R_{\mathrm{opt}}\leq 2\alpha(1+\alpha)K\sqrt{\frac{2\pi}{T}}+8\sqrt{\frac{\ln 4/\delta}{T}} \tag{4.5}$$

in the limit $m\to\infty$. The typical choice for α is $\alpha\leq 1$, which ensures that $\|Dc\|\leq 1$. In this case inequality (4.5) provides an improvement over the sparse coding bound in [130] (cf. Theorem 2 and Section 2.4 therein), which contains an additional term of the order of $\sqrt{(\ln T)/T}$ and the same leading term in K as in (4.5) but with slightly worse constant (14 instead of $4\sqrt{2\pi}$). The connection of our method to sparse coding is experimentally demonstrated in Section 4.4.4 and illustrated in Fig. 4.6.

4.4. Experiments

In this section, we present experiments on a synthetic and two real datasets. The aim of the experiments is to study the statistical performance of the proposed method, in both settings of multitask learning and transfer learning. We compare our method, denoted as Sparse Coding Multi Task Learning (SC-MTL), with independent ridge regression (RR) as a base line, multitask feature learning (MTFL) [8] as a standard MTL method which does not have mechanisms to avoid negative transfer, and GO-MTL [107], a method based on similar ideas as the ones we present but with different choices of regularizers. We also report on sensitivity analysis of the proposed method versus different number of parameters involved.

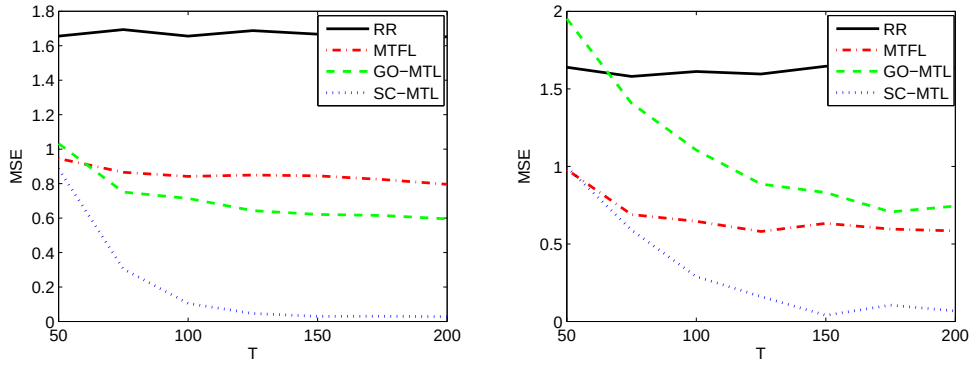

Figure 4.1.: Multitask error (left) and transfer error (right) vs. number of training tasks T.

4.4.1. Optimization algorithm

We solve problem (4.1) by alternating minimization over the dictionary matrix D and the code vectors $\mathbf{c}$. The techniques we use are very similar to standard methods for sparse coding and dictionary learning, see e.g. [91] and references therein for more information. Briefly, assuming that the loss function ℓ is convex and has Lipschitz continuous gradient, either minimization problem is convex and can be solved efficiently by proximal gradient methods, see e.g. [20, 45]. The key ingredient in each step is the computation of the proximity operator, which in either problem has a closed form expression.

4.4.2. Synthetic experiment

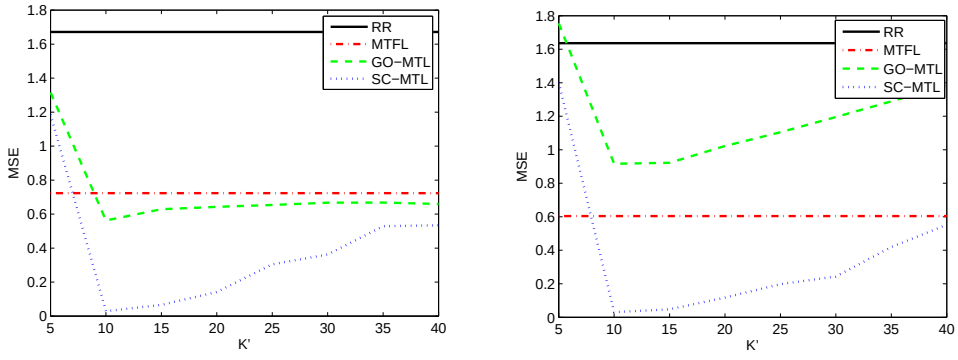

Figure 4.2.: Multitask error (left) and Transfer error (right) vs. number of atoms K' used by dictionary-based methods.

We generated a synthetic environment of tasks as follows. We choose a $d \times K$ matrix D

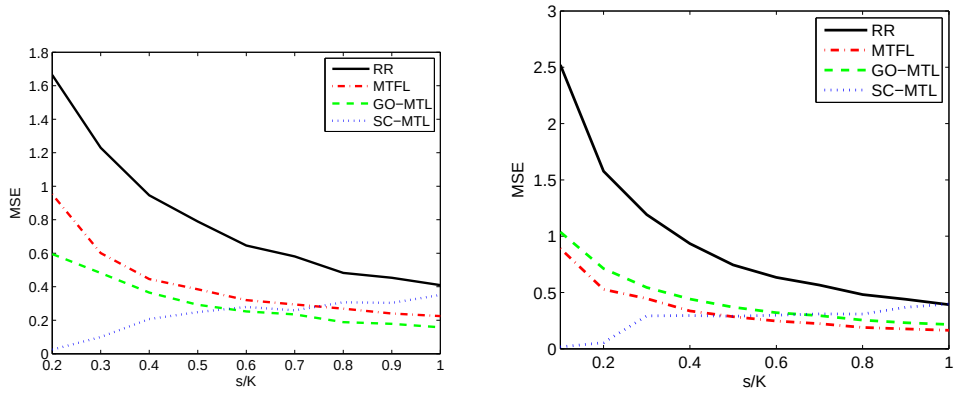

Figure 4.3.: Multitask error (left) and Transfer error (right) vs. sparsity ratio s/K.

by sampling its columns independently from the uniform distribution on the unit sphere in $\mathbb{R}^d$. Once D is created, a generic task in the environment is given by $w = Dc$, where c is an s-sparse vector obtained as follows. First, we generate a set $J \subseteq [K]$ of cardinality s, whose elements (indices) are sampled uniformly without replacement from the set $[K]$. We then set $c_j = 0$ if $j \notin J$ and otherwise sample $c_j \sim \mathcal{N}(0, 0.1)$. Finally, we normalize c so that it has ℓ_1-norm equal to some prescribed value α. Using the above procedure we generated T tasks $w_t = Dc_t$, $t = 1, \ldots, T$. Further, for each task t we generated a training set $\mathbf{z}^t = \{(x_i^t, y_i^t)\}_{i=1}^m$, sampling x_i^t i.i.d. from the uniform distribution on the unit sphere in $\mathbb{R}^d$. We then set $y_i^t = \langle w_t, x_i^t \rangle + \xi_{ti}$, with $\xi_{ti} \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is the variance of the noise. This procedure also defines the generation of new tasks in the transfer learning experiments below.

The above procedure depends on seven parameters: the number K and the dimension d of the atoms, the sparsity s and the ℓ_1-norm α of the codes, the noise level σ, the sample size per task m and the number of training tasks T. In all experiments we report both the multitask learning (MTL) and transfer learning (TL) performance of the methods. For MTL, we measure performance by the estimation error $1/T \sum_{t=1}^T \|w_t - \hat{w}_t\|^2$, where $\hat{w}_1, \ldots, \hat{w}_T$ are the estimated task vectors (in the case of SC-MTL, $\hat{w}_t = D(\mathbf{Z})c(\mathbf{Z})_t$ – see the discussion in Section 4.2). For TL, we use the same quantity but with a new set of tasks generated by the environment (in the experiment below we generate 100 new tasks). The regularization parameter of each method is chosen by cross validation. Finally, all experiments are repeated 50 times, and the average performance results are reported in the plots below.

In the first experiment, we fix $K = 10, d = 20, s = 2, \alpha = 10, m = 10, \sigma = 0.1$ and study the statistical performance of the methods as a function of the number of tasks.

The results, shown in Fig.4.1, clearly indicate that the proposed method outperforms the remaining approaches. In this experiment the number of atoms used by dictionary-based approaches, which here we denote by K' to avoid confusion with the number of atoms K of the target dictionary, was equal to $K = 10$. This gives an advantage to both GO-MTL and SC-MTL. We therefore also studied the performance of those methods with respect to the dependence on K'. Fig.4.2, reporting this result, is in qualitative agreement with our theoretical analysis: the performance of SC-MTL is not too sensitive to K' if $K' \geq K$, and the method still outperforms independent RR and MTFL if $K' = 4K$. On the other hand if $K' < K$ the performance of the method quickly degrades. In the last experiment we study performance vs. the sparsity ratio s/K. Intuitively we would expect our method to have greater advantage over MTL if $s \ll K$. The results, shown in Fig.4.3, confirm this fact, also indicating that SC-MTL is outperformed by both GO-MTL and MTFL as sparsity becomes less pronounced ($s/K > 0.6$).

4.4.3. Transfer learning for optical character recognition

We have conducted experiments on real data to study the performance of our method in a transfer learning setting. To this end, we employed the NIST dataset[1], which is composed of a set of 14×14 pixels images of handwritten characters (digits and lower and capital case letters, for a total of 52 characters).

We considered the following experimental protocol. First, a set of 20 characters is chosen randomly from the original pool of 52, as well as n instances for each character. These are used to learn all possibilities of 1-vs-1 train tasks, that is, we create one task for each possible pair of classes. That makes $T = 190$ tasks, each of which having $m = 2n$ instances. The knowledge learned in this stage is employed to learn another set of new tasks, which we will call target tasks hereafter. In our approach, the assumption that is made is that some of the components in the dictionary learned from the training tasks, can also be useful for representing the target tasks. In order to create the target tasks, another set of 10 characters are chosen among the remaining set of characters in the dataset, inducing a set of 45 1-vs-1 classification tasks. Since we are interested in the case where the training set size of the target tasks is small, we sample only 3 instances for each character, hence 6 examples per task.

In order to tune the hyperparameters of all compared approaches to be compared, we also created another set of 45 validation tasks by following the same process, simulating

[1]The NIST dataset is available at http://www.nist.gov/srd/nistsd19.cfm

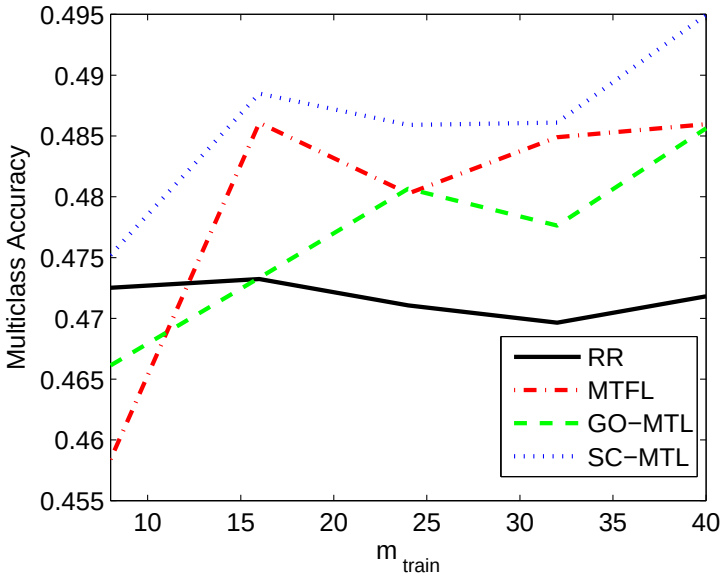

Figure 4.4.: Multiclassification accuracy (among 10 classes) of RR, MTFL GO-MTL and SC-MTL vs. the number of training instances in the transfer tasks, m.

the target set of tasks. Note that there is no overlapping between the digits associated to the train, target and validation tasks.

We ran 50 trials of the above process for different values of m and the average multiclass accuracy on the target tasks is reported in Fig. 4.4.

The results show, as expected, that all transfer learning methods obtain overall better performances as the number of training instances on the transfer tasks increases. Nevertheless, when this number is very low, transfer capabilities do not help, being in fact harmful in some cases. This may be because the low number of instances hinders the learned atoms to be meaningful enough for future tasks. In all cases, our approach consistently outperforms all competitors. The advantage over MTFL can be explained by the robustness of our approach against negative transfer. We hypothesize that our method outperforms GO-MTL because the latter has a regularizer that implies weaker constraints on the learning parameters, leading to poorer learning guarantees.

4.4.4. Sparse coding of images with missing pixels

In the last experiment we consider a sparse coding problem [153] of optical character images, with missing pixels. Note that the aim here is to perform image completion in a set of images of numbers with some missing pixels, whereas in the previous experiment the aim was classifying complete images in terms of the digit printed in them. We

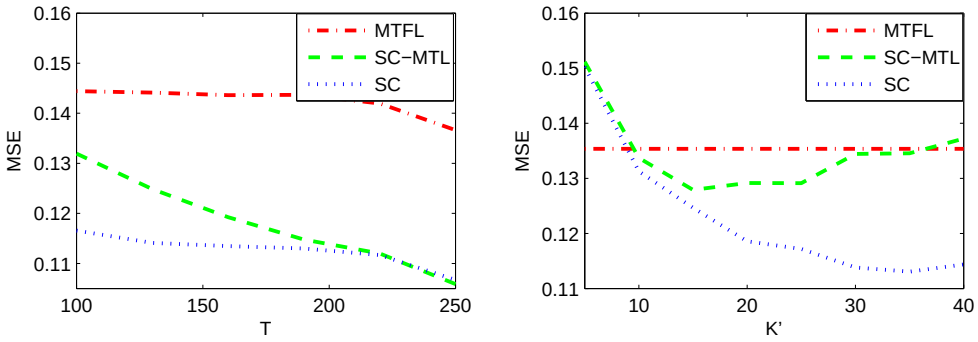

Figure 4.5.: Transfer error vs. number of tasks T (left) and vs. number of atoms K (right) on the Binary Alphadigits dataset.

employ the Binary Alphadigits dataset[2], which is composed of a set of binary 20×16 images of all digits and capital letters (39 images for each character). In the following experiment only the digits are used. We regard each image as a task, hence the input space is the set of 320 possible pixels indices, while the output space is the real interval $[0, 1]$, representing the gray level. We sample $T = 100, 130, 160, 190, 220, 250$ images, equally divided among the 10 possible digits. For each of these, a corresponding random set of $m = 160$ pixel values are sampled (so the set of sample pixels varies from one image to another).

We test the performance of the dictionary learned by using method (4.1) in a transfer learning setting, by choosing 100 new images. The regularization parameter for each approach is tuned using cross validation. The results, shown in Fig. 4.5, indicate some advantage of the proposed method over Multitask Feature Learning. Ridge regression performed significantly worse than the other approaches and is not shown in the figure. We also show as a reference the performance of sparse coding (SC) applied when all pixels are known.

With the aim of analyzing the atoms learned by the algorithm, we have carried out another experiment where we assume that there are 10 underlying atoms (one for each digit). We compare the resultant dictionary to that obtained by sparse coding, where all pixels are known. The results are shown in Fig. 4.6.

[2] Available at *http://www.cs.nyu.edu/~roweis/data.html*.

Figure 4.6.: Dictionaries found by SC-MTL using $m = 240$ pixels (missing 25% pixels) per image (left) and by Sparse Coding employing all pixels (right).

4.5. Discussion

In this chapter we have explored an application of sparse coding, which has been widely used in unsupervised learning and signal processing, to the domains of multitask learning and transfer learning. This was made with the aim of avoiding negative transfer when the relations among tasks are unspecified. Our learning bounds provide a justification of this method and offer insights into its advantage over independent task learning and learning dense representation of the tasks. The bounds depend on data dependent quantities which measure the intrinsic dimensionality of the data. Numerical simulations presented here indicate that sparse coding is a promising approach to multitask learning and can lead to significant improvements over competing methods.

While researching on the topics of this chapter, a method based on similar ideas appeared [107]. In that the authors empirically show its superiority to previous task grouping methods, reviewed in Section 2.4.3. In this chapter we presented a probabilistic analysis which complements well with the practical insights in the above work. We also addressed the different problem of transfer learning, demonstrating the utility of our approach in this setting by means of both learning bounds and numerical experiments. A further novelty of our approach is that it applies to a Hilbert spaces setting, thereby providing the possibility of learning nonlinear predictors using reproducing kernel Hilbert spaces.

In the future, it would be valuable to study extensions of our analysis to more general classes of code vectors. For example, we could use code sets $\mathcal{C}_\alpha$ which arise from structured sparsity norms, such as the group Lasso, see e.g. [89, 121] or other families of regularizers. A concrete example which comes to mind is to choose $K = Qr$, $Q, r \in \mathbb{N}$ and a partition $\mathcal{J} = \{\{(q-1)r + 1, \ldots, qr\} : q = 1, \ldots, Q\}$ of the index set $[K]$ into contiguous index sets of size r. Then using a norm of the type $\|c\| = \|c\|_1 + \sum_{J \in \mathcal{J}} \|c_J\|_2$ will encourage codes which are sparse and use only few of the groups in $\mathcal{J}$. Using the ball associated with this norm as our set of codes would allow to model sets of tasks which are divided into groups. A further natural extension of our method is nonlinear dictionary learning in which the dictionary columns correspond to

functions in a reproducing kernel Hilbert space and the tasks are expressed as sparse linear combinations of such functions.

5. Decoupling of Features

In the previous chapter we studied a general multitask learning scenario characterized by two properties. Firstly the relations between tasks need not be homogeneous, for example there may be groups of tasks such that the intra-group relationships are much stronger than inter-group relationships. Secondly, there is no side information regarding the relationships between tasks.

In this chapter we consider a multitask learning scenario in which side information is available, providing clues about how tasks are related. In particular we focus on a multi-aspect data scenario in which there are several learning tasks considered. Each task is known to belong to a group of tasks associated to an aspect of the data, so that tasks in different groups tend to use different features of the data. Applying traditional MTL methods on all tasks may lead to negative transfer due to the lack of common features. Hence, our objective is to leverage this situation to improve the performances of the learning tasks. Even though this need emerges from real life applications, such as emotion recognition filtering out personal idiosyncrasies, it has not been fully addressed in previous approaches.

5.1. Problem statement

The aim of this chapter is to consider the multitask learning scenario in which tasks are organized into two or more groups such that tasks belonging to one group aim at recognizing patterns regarding a particular aspect of the data. Our hypothesis is that tasks that belong to the same group tend to share the same set of features while tasks belonging to different groups tend not to share any features. One instance of the above scenario is the problem of identity/emotion recognition. Suppose that we have a data set of video clips of people expressing a set of emotions. We know from the literature, [30], that recognition of the identity of a person and recognition of the emotion expressed depend on different and uncorrelated features of the same image. Identity recognition is mainly based on features describing rigid characteristics of the face (e.g., face width,

hair color), whereas emotion recognition is based on features describing facial muscle configurations (e.g., eyes narrowed, corners of mouth raised).

We propose to take advantage of the prior knowledge that these tasks are unrelated to improve the learning accuracy on one of the groups of tasks. To simplify the presentation of the problem we will consider only two groups, where we call the group of interest *principal tasks* (e.g., emotion recognition) and the other group *auxiliary tasks* (e.g., identity recognition). We adopt this convention because usually the interest is on learning the tasks in one group. For example, in the identity/emotion application described above, the main interest is on learning a good classifier for detecting emotions in images. Nevertheless, our models treat similarly all groups of tasks, so nothing needs to be changed if the interest is in learning all tasks.

If the training sample per task is small enough with respect to the data dimensionality, a method which does not take into account the differentiation of groups can easily overfit, so that the facial features (idiosyncrasies) of a specific person can be mistaken as characteristics of a given emotion. To avoid this, our method exploits the identity labels of the instances at the training stage, but does not use them for prediction of emotion on the test instances.

The approach we propose builds on the multitask feature learning framework presented in [8], and described in Section 2.3.3. Let us recall that the aim of this approach is to learn a pool of linear features from the data that can be simultaneously useful to all tasks. We build on this because learning features from the data is on the core of our objective. However, the approach in [8] cannot account for unrelated tasks. Because of that, we add a regularization term which penalizes the inner product between the predictor functions of any two tasks belonging to two different groups. In this way, our formulation can discriminate those features important for each group of tasks and can lead to improvements in statistical performance.

Our methodology shares some aspects with some recent works in multitask learning, some of them already reviewed in Chapter 2. For example, [10] and [84] extended the multitask learning approach of [8] by assuming that there are a number of groups or clusters of tasks and that the weight vectors of the tasks belonging to the same group are similar to each other. In that case, the clusters are not known *a priori* (see Section 2.4.3). In addition, no constraint is imposed on tasks belonging to different clusters. A recent approach [227] partially shares the same motivation as our, in the sense that the dissimilarity between tasks is exploited. In that work the tasks weights are assumed to be sparse and to use exclusive attributes of the input data. The idea of exploiting unrelated factors

to improve learning has been also addressed in [194, 209, 213], see also Section 3.4.1. These studies rely on multilinear models to describe the relations between different factors (e.g., emotion and identity).

The remainder of this chapter is organized as follows. In Section 5.2 we review literature from psychology, psychophysics and neurology with the aim of providing some insights on the appropriateness of our assumption on emotion/identity separation. Particularly we want to know whether the human brain uses different neuronal areas to process identity and emotional information. In Section 5.3, we review previous work on multitask learning which is essential to understand our approach. In Section 5.4, we present our approach for incorporating unrelated auxiliary tasks in a multitask framework and an algorithm for solving the resulting optimization problem. In Section 5.5, we present our experiments with the proposed approach. Finally, in Section 5.6 we discuss our findings and future questions.

5.2. Survey on human perception of identity and emotion

As stated in the introduction, we assume that different sets of features are important for the recognition of different aspects of the data. In this section, we explore how human beings recognize different aspects of the data, and we do so by focusing on the problem of identity and emotion recognition from faces.

In [219] we can find the first empirical study that ponders whether facial expression and identity recognition follow independent processes in the brain of the perceiver. To do so, the authors carried out a set of experiments where they showed a pair of faces to the perceivers. These two images could belong to the same person or to different people and similarly they could express the same or different emotions. Furthermore, a subset of the identities were familiar to the subjects. They were then asked to judge if the pair of faces matched in terms of identity and/or in terms of expression. The authors collected the reaction times of the subjects and analyzed these measures to derive some conclusions. According to the results, identity matching times are shorter when the subject is presented with familiar faces; however, this condition does not affect expression matching times. The authors conclude that the processes for facial expression and identity recognition proceed independently.

Modern studies following non-intrusive approaches are based on aftereffects. This

methodology is grounded on the process of temporal brain adaptation when the subject is exposed to a set of stimulus for a period of time. Following the example of [87], a subject who is shown a sad face for three minutes (adaptation phase) tend to judge as happy a neutral face (aftereffect). In [62] the authors used this method to test whether facial identity aftereffect is invariant to changes in facial expression. The results obtained support an affirmative answer in experiments with both unknown and known faces.

Evidences of different neural processes involved in the recognition of identity and emotion come also from functional magnetic resonance imaging (fMRI) studies. These studies show that there is a specific area in the brain called superior temporal sulcus (STS) that becomes more active for identity recognition, whereas there is another separate region, the fusiform face area (FFA), responsive of the recognition process of emotions [72, 77, 95, 133]. Subsequent studies provide more support to this hypothesis. [7] and [216] conducted in parallel a set of experiments in which they obtained fMRI data from subjects who were presented a set of images of faces. The conclusions drawn from the experiments in both studies are similar. On the one hand, FFA activity decreases over time when the perceiver is presented the same identity. On the other hand, changes of expressions and viewpoints lead to activity in the STS region. Similar studies carried out on macaque monkeys lead to comparable conclusions [66].

The previous studies provide support to the hypothesis that there is a separation in the emotion and identity recogntion processes. Other studies have questioned the degree of separation of these two processes, or how early these two processes split. Contributions on this line come mostly from studies using prosopagnostic patients, that is, patients whose brain damage strongly hampers their facial recognition skills, yet their ability to identify objects remains intact [22].

In [50] the authors compare the identity recognition abilities of normal and prosopagnostic subjects. The authors report that non-neutral facial expressions have different effects on both kinds of subjects when it comes to recognizing identity. Whereas non-neutral facial expressions influence negatively normal perceivers, they have a positive effect on the identity recognition skills of prosopagnostic subjects. The authors conclude by stating that even though there seems to be a clear separation in the expression and identity recognition processes, the relationship between the correspondent brain regions may be closer than what it is currently believed. In [31] the authors question whether the two neural paths related to the two recognition processes are immediately separated. They argue that the immediate separation hypothesis is supported by weak evidence. The authors suggest as a plausible hypothesis that the brain route separation

between expression and identity processes occurs after a common representation of the inputs, a hypothesis also supported by [63, 81].

A different approach is explored in [30]. In that, the authors use principal component analysis (PCA) to express faces of different people and expressions as a combination of a few principal components. They conclude that there are different sets of components which account for expression and identity. With this experiment the authors claim that the different treatment within the brain of identity and expression is in part driven by the statistical properties of facial images, in opposition to explicit dedicated systems in the brain.

Through this survey we have found that there is dominant evidence that supports that different parts of the brain are in charge of identity and expression recognition. These findings encourage us to develop a machine learning model which takes this into account. Note that our objective is not trying to model how the brain works in terms of identity and expression recognition and we do not make any claim of that sort. Our aim is to exploit such knowledge to facilitate the modelling of the learning problem. Our reasoning is that the assumption we introduce in our model may improve the accuracy in both kinds of tasks. We use this emotion/identity recognition example as a test case for the method proposed in this chapter. However, the method is general to cases where it is known that tasks exploit different features.

5.3. Background on Multi-Task Learning

In this section, after formally introducing our problem, we revise the multitask learning method developed in [8], and reviewed in Section 2.3.3, focusing on the $\ell_{2,1}$-norm viewpoint. From this viewpoint the approach is decomposed into learning an optimal set of linear features, and learning a common sparse pattern of the utilization of those features by all tasks. This will be useful for the subsequent explanation of our approach, as it is based in the modification of the sparse patterns of the latter point.

5.3.1. Notation

As in previous chapters, all matrices are denoted by capital letters and all vectors are denoted by lower case letters. If a vector represents the i-th column from a matrix $A \in \mathbb{R}^{d_1 \times d_2}$, $i \in [d_2]$, it will be denoted as $a_{:,i} \in \mathbb{R}^{d_1}$ or just a_i. The element in the i-th column and the j-th row of A is denoted as $a_{j,i}$.

We are given a set of T principal supervised tasks. Each task $t \in [T]$ is identified by a function $f_t : \mathbb{R}^d \to \mathbb{R}$, which for simplicity we assume to be linear, that is $f_t(x) = w_t^\top x$. The vector of regression coefficients $w_t \in \mathbb{R}^d$ is unknown and we are provided with m data examples per task, $\{(x_i^t, y_i^t) : i \in [m]\} \subset \mathbb{R}^d \times \mathbb{R}$, such that $y_i^t = w_t^\top x_i^t + \xi_{ti}$, $i = 1, \ldots, m$, $t \in [T]$, where ξ_{ti} is some zero mean i.i.d. noise process[1]. We call these the *principal tasks* and the goal is to learn them jointly under the assumption that they are related. Whenever convenient we will arrange the inputs of task t in the matrix X^t, the output variables in the vector y^t. We will also use the set $\boldsymbol{X}$ and $\boldsymbol{Y}$ to designate the inputs and outputs for all tasks. In the same way, we will use matrix W to arrange the weight vectors of all tasks columnwise. Similarly, we assume there is a set of S auxiliary linear tasks which are described by the column vectors $v_1, \ldots, v_S$. We let V be the $d \times S$ matrix whose columns are given by the above vectors, in order. We also denote by $\{(x_i'^s, y_i'^s) : i \in [m]\} \subset \mathbb{R}^d \times \mathbb{R}$, $s \in [S]$ the examples for these additional tasks.

5.3.2. Multi-Task Feature Learning

As explained in Section 2.3.3, the approach based on trace norm regularization developed in [8] can be equivalently expressed using different viewpoints. In one of them, the matrix of tasks $W = [w_1, \ldots, w_T]$ can be factorized as the product of a $d \times d$ orthogonal matrix U and a $d \times T$ coefficient matrix A, which has only *few nonzero rows*. Note that the rows of A are associated with the features while the columns with the tasks. To learn such a factorization, we define the average empirical error

$$\mathcal{E}_{\text{pr}}(UA) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{m} \sum_{i=1}^{m} \ell(y_i^t, a_t^\top U^\top x_i^t) \tag{5.1}$$

and, following [8], we minimize the regularized error

$$\mathcal{E}_{\text{pr}}(UA) + \gamma \|A\|_{2,1}^2 \tag{5.2}$$

over all matrices $A \in \mathbb{R}^{d \times T}$ and orthogonal matrices U, that is, $U^\top U = I$. The norm appearing in the regularization term encourages matrices with many zero rows, under assumptions (e.g. Restricted Eigenvalue conditions) about the distribution of the data [120].

[1] In practice, the number of examples per task may vary but we have kept it constant for simplicity of notation.

In [8] it is proved that the above problem is equivalent to the convex problem

$$\begin{aligned} &\inf\ \mathbb{E}_{\mathrm{pr}}(W) + \gamma\,\mathrm{tr}(W^\top D^{-1} W) \\ &\text{s.t. } W \in \mathbb{R}^{d\times T},\ D \succ 0,\ \ \mathrm{tr}\,(D) \leq 1. \end{aligned} \tag{5.3}$$

If $(\hat{A}, \hat{U})$ is an optimal solution of (5.2), then $\hat{W} = \hat{U}\hat{A}$ is an optimal solution of (5.3), see [8, Thm. 1]. Moreover, for a fixed W the optimal D is given by

$$D(W) = \frac{(WW^\top)^{\frac{1}{2}}}{\mathrm{tr}\,(WW^\top)^{\frac{1}{2}}}.$$

5.4. Exploiting orthogonal tasks

We now present our method, which uses an auxiliary group of tasks, assumed to be unrelated to the principal group, to improve the learning process. Here we use the term unrelated to signify that the two groups of tasks are defined by orthogonal set of features. The intuition is that, by exploiting this orthogonality – that will be formalized shortly – we will improve the estimation of the principal group of tasks (and possibly the auxiliary ones as well).

We make the following assumption about the two group of tasks:

- a *low dimensional* representation is shared by the tasks within each group, and
- the principal tasks w_t *share no features* with the auxiliary tasks v_s.

To formalize these requirements, we write $V = UB$, where B is a $d \times S$ matrix of coefficients and let $C = [A, B]$ so that $[W, V] = UC$. We require that

- the matrix C has *few nonzero rows* and
- each of these rows has nonzero values in *only one* group of columns.

A schematic example of a matrix which our method should favor is

$$C = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & 0 & b_{31} & b_{32} \\ 0 & 0 & 0 & b_{41} & b_{42} \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In this example, there are three principal tasks and two auxiliary tasks. Furthermore, there are two important features for each group of tasks, but these features are not shared across the groups. Finally, there is a large number of features which are not relevant to any of the tasks.

We incorporate the above constraints into our method as follows. We let

$$\mathcal{E}_{\text{aux}}(UB) = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{m}\sum_{i=1}^{m}\ell(y_i'^s, b_s^\top U^\top x_i'^s)$$

and minimize the regularized error

$$\mathcal{E}_{\text{pr}}(UA) + \mathcal{E}_{\text{aux}}(UB) + \gamma\,\Phi(A,B) + \lambda\,\Psi(A,B) \tag{5.4}$$

over all matrices $A \in \mathbb{R}^{d\times T}$, $B \in \mathbb{R}^{d\times S}$ and orthogonal matrices $U \in \mathbb{R}^{d\times d}$. There are two regularization parameters $\gamma, \lambda > 0$ which may be tuned by cross validation. The first parameter controls the number of features shared by the tasks – the larger γ, the smaller the number of shared features will be; the second parameter controls the degree of orthogonality between the two groups of tasks – the larger λ, the less "correlated" the tasks within the two groups will be. In particular, in the limit $\lambda \to \infty$, the two groups of tasks will be orthogonal to each other.

The regularization term in (5.4) consists of two parts. The term $\Phi(A,B)$ favors few nonzero rows in the matrix $[A,B]$ and the term $\Psi(A,B)$ penalizes features shared by the different groups of tasks. Regarding the first term, we may choose $\Phi(A,B) = \|[A,B]\|_{2,1}^2$ as in standard multitask feature learning (Section 5.3.2). Regarding the second term, we want that $a_{jt}b_{js} = 0$, for every $t \in [T]$, $s \in [S]$ and $j \in [d]$. As in the previous section, a necessary condition for this to hold is that $A^\top B = 0$, where 0 denotes the $T \times S$ matrix of zeros. It turns out that this condition is also sufficient in this case. At first sight this condition does not seem to be the case, since $a_t^\top b_s = 0$ imposes orthogonality only on a_t and b_s. However, since this condition holds for every choice of t and s in their range *and* the matrix U is orthogonal, it implies that the subspace spanned by the principal tasks is orthogonal to the subspace spanned by the auxiliary tasks. Consequently, it must be the case that there is an orthogonal matrix U' and matrices A', B' such that $W = U'A'$, $V = U'B'$ and $[A', B']$ have the desired structure. Thus, we can use the square of the Frobenius norm of $A^\top B$ as the second regularization term, that is,

$$\Psi(A,B) = \|A^\top B\|_{\text{Fr}}^2\,. \tag{5.5}$$

Another valid choice would be the ℓ_1-norm of the vector formed by the entries of matrix $A^\top B$, see [225]. However, the Frobenius norm, besides being differentiable and easier to deal with, seems more appropriate in our context, since it drives all the inner products towards zero, whereas the ℓ_1-norm does not prevent some of the inner products from being large.

We now make the change of variable $[W, V] = U[A, B]$ in a way similar to Section 5.3.2 and derive the equivalent problem

$$\begin{aligned} &\inf\ \mathcal{E}(W,V) + \mathcal{R}_0(W,V,D) \\ &\text{s.t. } W \in \mathbb{R}^{d\times T},\ V \in \mathbb{R}^{d\times S},\ D \succ 0,\ \operatorname{tr}(D) \le 1, \end{aligned} \tag{5.6}$$

where $\mathcal{E}(W,V) = \mathcal{E}_{\mathrm{pr}}(W) + \mathcal{E}_{\mathrm{aux}}(V)$ and

$$\mathcal{R}_0(W,V,D) = \gamma \operatorname{tr}\left(D^{-1}(WW^\top + VV^\top)\right) + \lambda \|W^\top V\|_{\mathrm{Fr}}^2.$$

Note that unlike the standard multitask optimization problem (5.3), problem (5.6) is *nonconvex* due to the $\|W^\top V\|_{\mathrm{F}}^2$ term in the regularizer $\mathcal{R}_0$. To overcome this drawback, we add a strongly convex function to the regularizer. A natural choice, which we consider here, is to add a multiple of the squared Frobenius norm of the parameters. That is, we consider the optimization problem

$$\begin{aligned} &\inf\ \mathcal{E}(W,V) + \mathcal{R}(W,V,D) \\ &\text{s.t. } W \in \mathbb{R}^{d\times T},\ V \in \mathbb{R}^{d\times S},\ D \succ 0,\ \operatorname{tr}(D) \le 1, \end{aligned} \tag{5.7}$$

where

$$\mathcal{R} = \mathcal{R}_0(W,V,D) + \rho(\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2),$$

and ρ is a positive parameter. The following result, whose proof can be found in the appendix, establishes a condition under which problem (5.7) is convex.

Theorem 5.4.1. *If* $\rho > \sqrt{\frac{\mathcal{E}(0,0)\lambda}{2}}$ *then problem* (5.7) *is convex.*

We solve problem (5.7) by alternate minimization, see Algorithm 5.1. For fixed W, V the optimal D is given by

$$D(W,V) = \frac{(WW^\top + VV^\top)^{\frac{1}{2}}}{\operatorname{tr}(WW^\top + VV^\top)^{\frac{1}{2}}}. \tag{5.8}$$

We note, in passing, that if we substitute the right hand side of this expression in the

regularizer appearing in the objective function of problem (5.7), we obtain the following function of W and V

$$\gamma\|[W,V]\|_{\mathrm{Tr}}^2 + \rho(\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2) + \lambda\|W^\top V\|_{\mathrm{Fr}}^2. \tag{5.9}$$

The two first terms of the above expression are similar to a matrix version of the elastic net regularizer [229]. For this reason, we will refer to the learning method solving problem (5.7) as orthogonal multitask learning elastic-net (OrthoMTL-EN).

Returning to the algorithm we observe that, for fixed D, the regularizer separates across tasks. Indeed, using elementary properties of the trace of matrix products, it follows that

$$\begin{aligned}\mathcal{R}(W,V,D) &= \sum_{t=1}^{T} w_t^\top(\gamma D^{-1} + \rho I + \lambda VV^\top)w_t + \mathrm{tr}((\gamma D^{-1} + \rho I)VV^\top) \\ &= \sum_{s=1}^{S} v_s^\top(\gamma D^{-1} + \rho I + \lambda WW^\top)v_s + \mathrm{tr}((\gamma D^{-1} + \rho I)WW^\top).\end{aligned}$$

Thus, the minimization over W (resp. V) can be carried out independently across the tasks since the regularizer decouples when D and V (resp. W) are fixed.

We remark that the alternating process decreases the objective function in problem (5.7) and hence it is guaranteed to converge in objective value. One may modify the perturbation analysis in [8] to show that, under the hypothesis of Theorem 5.4.1, the iterates of the algorithm converge. Note also that we may still apply Algorithm 5.1 to approximately solve problem (5.7) for an arbitrary choice of the parameters γ, λ, ρ. In this case, however, the objective is not guaranteed to be convex and so the algorithm is only guaranteed to converge to a stationary point.

In practice our numerical experiments indicate that the algorithm converges in fewer than 20 iterations. Each W or V update can be executed very quickly by computing each column vector independently. For example, for the square loss this consists in solving a linear system of d equations. However if $d > m$, one may solve an equivalent dual problem, see e.g. [179]. Other loss functions, such as the hinge loss, can be handled similarly. Finally, the D step requires the computation of a matrix square root, which we solve by singular value decomposition.

Algorithm 5.1 Orthogonal Multi-Task Learning (OrthoMTL)

Input: training sets $\{(x_i^t, y_i^t)\}_{i=1}^m$, $\{(x_i'^s, y_i'^s)\}_{i=1}^m$, $t \in [T]$, $s \in [S]$.
Parameters: regularization parameters γ, λ, ρ, tolerance parameter tol
Output: regression matrices $W = [w_1, \ldots, w_T]$ and $V = [v_1, \ldots, v_S]$, $d \times d$ positive definite matrix D
Initialization: set $D = \frac{I}{d}$
while $\|W - W_{\text{prev}}\| > tol$ or $\|V - V_{\text{prev}}\| > tol$ **do**
  for $t = 1 \ldots T$ **do**
    compute the minimizer $w_t \in \mathbb{R}^d$ of the function $\sum_{i=1}^m \ell(y_i^t, w^\top x_i^t) + w^\top \left(\gamma D^{-1} + \rho I + \lambda V V^\top\right) w$
  end for
  for $s = 1 \ldots S$ **do**
    compute the minimizer $v_s \in \mathbb{R}^d$ of the function $\sum_{i=1}^m \ell(y_i'^s, v^\top x_i'^s) + v^\top \left(\gamma D^{-1} + \rho I + \lambda W W^\top\right) v$
  end for
  set $D = \frac{\left(WW^\top + VV^\top\right)^{\frac{1}{2}}}{\operatorname{tr}\left(WW^\top + VV^\top\right)^{\frac{1}{2}}}$
end while

5.5. Experiments

In this section we gradually test the performance of the methods developed by using increasingly complex datasets. First we present numerical experiments on one synthetic dataset, and then we use two real datasets on expression recognition. The first one is composed of posed expressions. The second dataset is composed of natural expressions. In all experiments we compare the following methods:

- OrthoMTL-EN: this is our method (cf. problem (5.7)).
- OrthoMTL-C: this is like OrthoMTL-EN but with parameter ρ set according to Theorem 5.4.1. In this way, problem (5.7) is guaranteed to be convex.
- OrthoMTL: this is like OrthoMTL-EN but with parameter $\rho = 0$, so that there is no convex relaxation.
- Ridge Regression (RR): this standard method corresponds to the choice $\lambda = \gamma = 0$ and can be interpreted as learning the tasks independently. This method is a baseline useful to test how much value MTL methods add.
- Multitask Feature Learning (MTFL): this is the multitask feature learning method of [8] and corresponds to the choice of $\rho = \lambda = 0$. It assumes homogeneous commonalities among tasks, regardless the group they belong to.
- MTFL-2G: this approach consists of applying the method of [8] to each group of

tasks separately.

In the figures below, for ease of visualization of the results, only the best five methods are reported. We use the same setting of parameters for all experiments and all algorithms: we perform 5-fold cross-validation to tune the value of the regularization parameters whenever those were treated as free parameters. We considered the values of $\gamma = 10^k$ with $k \in \{-4, \ldots, 2\}$, $\lambda = 10^k$, with $k \in \{4, \ldots, 7\}$ and $\rho = 10^k$ with $k \in \{-2, \ldots, 2\}$.

Finally, in all experiments we trained all learning methods using the square loss function $\ell(y, z) = (y - z)^2$, $y, z \in \mathbb{R}$.

5.5.1. Synthetic data

We use synthetic data to test whether Algorithm 5.1 finds the right solution on data that satisfy the prior orthogonality assumptions. To this end, we created a dataset consisting of 20 tasks, 10 of them belonging to the first subset and the remaining ones to the second subset ($T = S = 10$). The data are embedded in a $d = 100$ dimensional space. From these 100 dimensions, only the first 5 are useful for the first subset of tasks and the following 5 are useful for the second subset. Finally, the remaining 90 dimensions are not important at all. In this synthetic dataset, every task is represented as either $(w_{1t}, \ldots, w_{5t}, 0, \ldots, 0)$, $t = 1, \ldots, 10$ or $(0, \ldots, 0, w_{6s}, \ldots, w_{10s}, 0, \ldots, 0)$, $s = 1, \ldots, 10$, where each parameter w_{it} is chosen randomly from a uniform distribution, $\mathcal{U}(0, 0.1)$.

We build a set of $n = 1000$ instances, $Z \in \mathbb{R}^{d \times n}$, such that every element of matrix Z is sampled from the uniform distribution on the unit interval. The training set is composed of a random subset of m instances, for different values of the sample size $m = 10, 15, \ldots, 50$, and the test set is composed of the remaining instances. For every task t, we generate the output y^t as $y^t = Zw_t + \xi_t$, where $\xi_t \in \mathbb{R}^m$ and $\xi_{ti} \sim N(0, 1)$, $i = 1, \ldots m$. Finally we apply an orthogonal rotation to Z by sampling an orthogonal matrix U randomly from the Haar measure and set $X = UZ$.

We repeated the described experiment 750 times for each value of m. The results can be seen in Fig.5.1. MTFL-2G performed comparably to Ridge Regression and MTFL. It is interesting to see that when the training size is very small ($m = 10$), RR performs significantly better than MTFL. This may be due to negative transfer of knowledge between tasks in separate groups. All of our methods performed better than both Ridge Regression and MTFL. OrthoMTL-C gives the best results, followed by OrthoMTL-EN

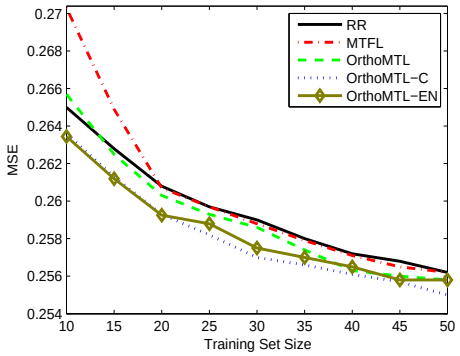

Figure 5.1.: Synthetic data: Comparison between Ridge Regression (RR), Multitask Feature Learning (MTFL) [8], OrthoMTL, OrthoMTL-C and OrthoMTL-EN.

and OrthoMTL. We have applied a paired t-test to check whether the difference between OrthoMTL-C and OrthoMTL-EN and either Ridge Regression or MTFL is equal to 0 and obtained a p-value below 10^{-7} for training set sizes below 45.

5.5.2. JAFFE dataset

The first experiment considered the Japanese Female Facial Expression (JAFFE) database [124]. It is composed of 213 images of 10 subjects displaying a range of expressions, as shown in Fig.5.2. There are 7, mutually exclusive, emotion classes that need to be detected. Given an unlabeled image, the objective is to predict the emotion expressed in it.

Figure 5.2.: Sample images taken from the JAFFE dataset.

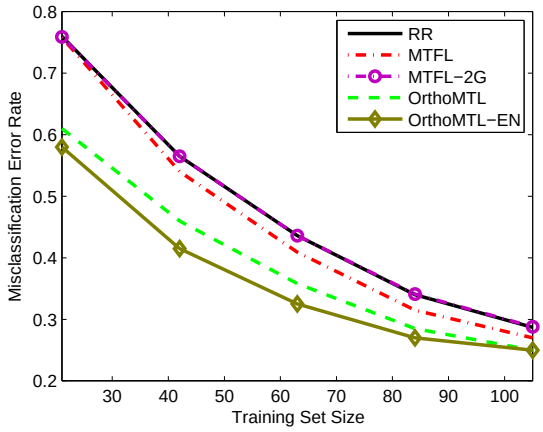

Figure 5.3.: JAFFE dataset: Comparison between Ridge Regression (RR), Multitask Feature Learning (MTFL), MTFL-2G, OrthoMTL and OrthoMTL-EN.

We represented an input image in the following manner. First, we extracted the face from the background. To this end, we used the OpenCV implementation of the Viola and Jones face detector [212] to detect the face and eyes in the image. After that, we rotated the face so that the eyes are horizontally aligned. Finally, we rescaled the face to a 200×200 size image. In order to obtain a descriptor of the textures of the image we used the Local Phase Quantization (LPQ) [152]. Specifically, we divided every image into 5×5 non overlapping regions. We computed the LPQ descriptor for each region and we created the image descriptor by concatenating all the LPQ descriptors. Finally we applied Principal Component Analysis to extract as many components as necessary to describe 99% of the data variance. After this process, we obtained a descriptor with 203 attributes for each image.

As discussed in Section 5.1, we assume that the features which are useful for recognizing the emotion are different from those which are useful for recognizing the identity of the subject. Therefore, it seems appropriate to apply our method when the principal tasks are those related to predicting the emotion and the auxiliary tasks are those related to the prediction of the identity. Each task discriminates one class from the others (one versus all), so that we have 7 tasks in the first group (one for each emotion) and 10 tasks in the second group (one for each actor).

In this experiment we randomly select m instances as a training set and use the remaining ones as a test set. We ran the experiments for different values of m and plot the learning curve. The experiments were executed 200 times and the results are shown in Fig. 5.3.

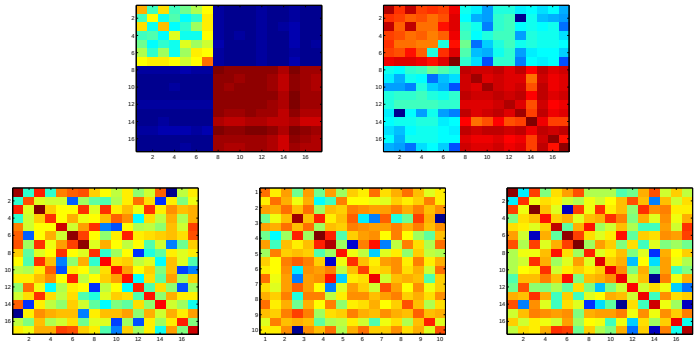

Figure 5.4.: Tasks correlation matrix learned by different methods: OrthoMTL-EN (top left), OrthoMTL (top right), MTFL-2G (bottom left), MTFL (bottom middle), and Ridge Regression (bottom right), Red (resp. blue) denotes high (resp. low) intensity values.

As we see, both OrthoMTL-EN and OrthoMTL outperform the other approaches, the improvement being more evident when the training set is small. This is reasonable since the prior information (the emotion tasks are unrelated to the identity tasks) is more useful when the data are scarce. On the other hand, when the training size increases, the prior information provided by the regularizer becomes comparatively less important with respect to the one provided by the training set. We have applied a paired t-test between our methods and either MTFL, MTFL-2G and Ridge Regression, obtaining always a p-value below 10^{-3} for any value of m. This result supports the hypothesis that the differences between both kinds of approaches are significant. In this experiment, OrthoMTL-C performed comparably to Ridge Regression. To try to explain this poor behaviour, let us recall that the term that makes the problem convex in Theorem 5.4.1 is data dependent. We hypothesize that this performance may be caused by the convexification carried out, which using this dataset may lead to a big perturbation of the original problem.

We also report in Fig. 5.4 the task correlation matrix $[W, V]^\top [W, V]$ learned by different methods. As we can see, the off-diagonal blocks of this matrix, which are formed by the inner products between tasks of different groups, are much smaller than the elements in the diagonal blocks, which correspond to inner products between tasks in the same group. This effect is more pronounced in the case of our methods, indicating that they can take advantage of the information contained in the auxiliary tasks.

In a separate experiment, we considered a transfer learning problem with the aim of comparing OrthoMTL-EN with the Bilinear Model proposed in [194]. A transfer learning problem requires test instances for identities which are not present in the training set.

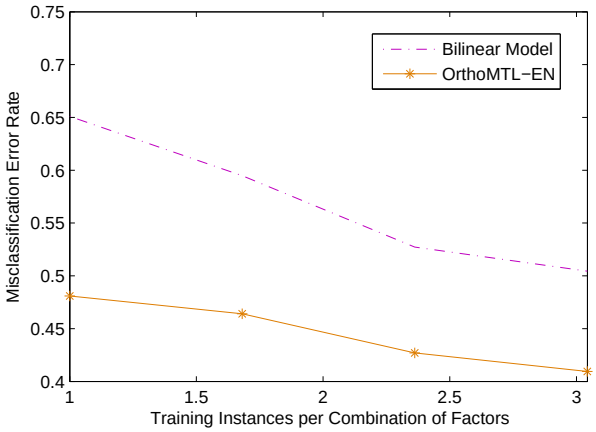

Figure 5.5.: JAFFE dataset: Comparison between Bilinear Model and OrthoMTL-EN in a transfer learning experiment – see text for description.

To do so, we used a leave-one-subject-out strategy. We consider this new setting here so that the algorithm in [194] can be applied. To tune the parameters of the Bilinear Model we also followed a cross-validation process. We ran the whole process 10 times (that is, each subject has been in the test set 10 times) and the results are shown in Fig. 5.5. According to our findings, our approach clearly outperforms the Bilinear Model for this dataset. The resulting p-value is below 0.01, supporting our claim.

5.5.3. Shoulder pain dataset

As a final test, we apply our methods to the UNBC-McMaster Shoulder Pain Expression Archive [123]. This data set contains video clips of the faces of people who suffer from shoulder pain while performing active and passive exercises. It presents higher variability than the stereotypical acted expressions of the dataset used in Section 5.5.2. The portion of this data set which is publicly available contains 200 video clips of facial expressions from 25 patients. The data set provides 66 tracked landmarks points of the face for each frame of each clip. Each frame is also coded in terms of the The Prkachin and Solomon pain intensity (PSPI) scale, which is the only standard metric which can define pain on a frame-by-frame basis [161]. Our task here is to recognize if a frame of a clip shows a pain expression (i.e., pain value bigger than 0) or not. Instead of texture features, in this experiment the attributes consist of distances of several not-crossing segments between provided landmarks points as shown in Fig. 5.6 (left). This set of attributes contains rich information about the deformation of local areas of the face.

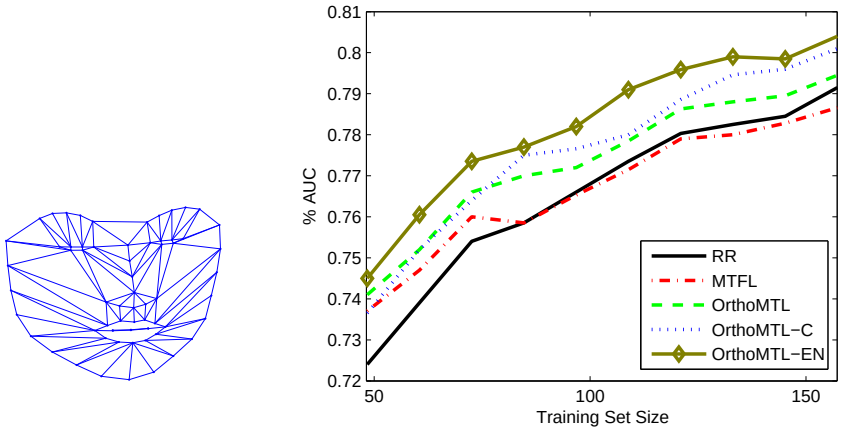

Figure 5.6.: Left: Landmark points and edges used to build the attributes for the UNBC-McMaster Shoulder Pain Expression Archive (selected according to Figure shown in [123]). Right: Comparison between Ridge Regression (RR), Multitask Feature Learning (MTFL), OrthoMTL-EN, OrthoMTL and OrthoMTL-C on the UNBC-McMaster Shoulder Pain Expression Archive Database.

Given that this is a naturalistic dataset, there are no guarantees about the distribution of expressions across subjects (for example, certain facial expressions may be more present in some subjects). Here we use again the assumption that affective state detection, and particularly pain detection, is unrelated to identity recognition. To test the algorithm, the experiments were carried out using a leave-one-subject-out protocol. At each step, the frames from one patient were used as test set and a percentage of $0.1\%, 0.125\%, \ldots, 0.325\%$ randomly selected frames from the remaining 24 patients were used as the training set. The process was repeated until all the subjects were used as the testing set once. The whole protocol was executed 30 times. The mean results (using Area Under the Curve as a measure of accuracy) are reported in Fig. 5.6 (right).

As it can be noted, all of OrthoMTL-EN, OrthoMTL-C and OrthoMTL perform significantly better than their competitors (MTFL and Ridge Regression). The advantage of our methods is particularly clear in the case of OrthoMTL-EN which performs the best. OrthoMTL also performs well, especially as the training set decreases: by applying a paired t-test, we observe that when the training set is small, $m = 48$ (corresponding to 0.1% of the number of available frames), the difference between each of our methods and both MTFL or Ridge Regression is significant ($p < 10^{-3}$) and it remains significant as the training set increases to $m = 140$ ($p < 0.025$).

5.6. Discussion

In this chapter we have focused on a scenario that commonly arises when working with multi-aspect datasets in a supervised learning context. We addressed the problem in which two (or more) groups of supervised learning tasks are unrelated in the sense that they involve different linear discriminative features of the same input data. At first sight it seems surprising that we can exploit one group of tasks to improve learning of the other group. However, the fact that the two groups of tasks use different features provides an implicit constraint about which features could be used by each group, thereby helping the learning process.

We proposed a regularization formulation based on orthogonality, which incorporates this information in the learning method. The regularizer encourages both a low dimensional representation and penalizes the inner product between any pair of weight vectors of tasks from different groups. The implication of this constraint is that we look for common sparse representations within each group of tasks and also that tasks from different groups share as few features as possible. The resulting regularizer is non-convex, which led us to explore a convex modification of our approach. The resulting regularizer depends on three parameters, as specified in eq. (5.9). For special choices of these parameters, the resulting method reduces to Multitask Feature Learning [8] and to Ridge Regression (independent tasks learning).

To validate the advantages of our approach, we presented experiments on synthetic and real datasets comparing our algorithms with corresponding competitors. The experimental results indicate that the proposed approaches consistently improve over the other methods, supporting our hypothesis that taking into account independence helps discriminate features for tasks in different groups.

Given the unrelatedness between tasks of different groups, we hypothesized that the application of traditional MTL methods on all tasks may lead to negative transfer. We found this phenomena in the synthetic experiment and in one real dataset experiment, in which MTFL was outperformed by RR (independent learning) in some of the regimes tried. Our framework is effective in avoiding this behaviour by exploiting the side information about the relations between tasks. In the remaining real dataset experiment, MTFL did consistently improved over RR. Hence, there is apparently positive transfer between tasks of different groups. Even in this case, our methods lead to significantly better performance.

It is however unclear whether the convexification of OrthoMTL leads to improvements:

we obtained good results on the synthetic dataset and one real dataset but no improvement was observed on the other real dataset. A possible explanation of those results can be found by observing that the convexification of the approach depends on the input data (see Theorem 5.4.1), and in some cases the input data may lead to a big perturbation of the original problem. This explanation is bolstered by the results obtained using a relaxed version of the previous approach, in which all regularization parameters are tuned by cross validation. This more general method (OrthoMTL-EN) obtains the best results in both real datasets tried.

Multilinear models could be also useful for managing multi-aspect data. As reviewed in Chapter 3, those methods have proved succesful for extracting meaningful information related to different aspects from the data in an unsupervised way. Although some efforts have been made in the supervised learning context [194], the resulting approach, based on rearranging the data instances in a tensor, presents a number of limitations that make it not always suitable to applications in which the training sets are not equally distributed among the elements of the aspects and when the variability between instances belonging to the same combination of elements is very high. The results of the experiment comparing that approach with OrthoMTL, shown in Fig. 5.5, make those limitations clear. Furthermore, that framework does not allow for addressing regression problems, while our approach is general to different learning problems.

The work presented in this section can be extended in different directions. On the theoretical side, it would be valuable to investigate whether the improved generalization performance of the method could be supported by a statistical analysis. When the weights of the auxiliary tasks are known *a priori* such a result would follow from the analysis in [128]. However when both the primary and auxiliary tasks need to be estimated from data, the above problem remains to be understood. On the practical side, it may be valuable to explore the application of our approach in the context of hierarchical classification where recent work has considered the incorporation of orthogonal constraints [225]. The ideas presented here could also be applied to matrix completion problems such as those arising in the context of collaborative filtering.

6. Multilinear Multitask Learning

In the previous chapter, we considered multi-aspect datasets with the objective of learning information about a particular aspect of interest. The underlying principle in the method developed in the previous chapter, OrthoMTL, was to filter out the information pertaining to the secondary aspects of the data. That is useful whenever we require a model to be robust to changes in those secondary aspects. In the example of emotion recognition from several subjects, our framework is able to predict emotions from people that do not appear in the training set. This is possible because the secondary aspect is not needed in the recognition stage. However, there are many cases where the secondary aspects of the data are known for the test instances, thus it is desirable to make use of this valuable additional information. In this chapter, we develop a framework that is motivated by this situation.

6.1. Problem statement

Many real world datasets can be organized into multi-modal structures. With such datasets, the tasks to be learned can be referenced by multiple indices. For example, consider a task of predicting restaurant ratings by a specific restaurant critic, given a restaurant as an input query. We can then extend the problem to predicting the ratings by N critics. This will lead to N tasks, each of them referenced by a single index $i \in [N]$. MTL methods attempt to learn the functions that model all N tasks together by exploiting the common trends among all of the critics as well as their individual preferences.

Traditional MTL methods do not consider any additional inherent structure in the dataset and therefore the referencing of the tasks is simplified to a single index i as in the previous case. However, it is clear that a loss of information would arise if these methods were applied to datasets that are defined by multiple indices. For example, if our restaurant critics rated M separate characteristics of each restaurant, this would give rise to a second index j ranging from 1 to M. This 2-dimensional indexing information would

be lost in a traditional MTL approach. We can find another example in Facial Action Coding System (FACS). It constitutes the standard codification system to precisely describe facial expressions. It is based on the decomposition of the facial expression in terms of a set of Action Units (AUs). Each AU makes reference to the activation of a set of facial muscle groups. Hence, one may consider learning how to recognize the degree of activation of each AU (index j) for each subject (index i).

In this chapter, we propose to extend the method developed in [8], which is based on learning a common set of linear features for all tasks, to consider the inherent structures in such kind of datasets by explicitly bringing into account the multi-index information which is associated with the tasks. In Chapter 3 we studied the concept of mode as a set of indices by which data might be classified. For this reason it is natural to use multilinear models to represent structural information defining the relations between tasks. We will refer to our proposed framework as Multilinear Multitask Learning (MLMTL).

Multilinear models have been shown to be effective in determining separate underlying factors in data (see Section 3.4.1). In this chapter we form a multilinear model by structuring the weight parameters of all tasks into a tensor (see Fig. 6.1). This is a departure from previous studies, where the multilinear decomposition was applied directly to the input data, obtaining unsupervised learning models which can be seen as higher-order generalizations of principal component analysis. The tensor representation allows us to account for the multimodal interactions between the tasks. In addition, our approach provides a means to *make predictions even in absence of training data for one or more of the tasks,* which hereafter we will refer to as *zero-shot transfer learning* [156]. As an illustration, an estimate for the (i, j)-task can be obtained provided that there exists at least one task (i, k), $k \neq j$ and one other task (ℓ, j), $\ell \neq i$ such that both tasks have available training data.

In order to formulate MLMTL, we follow a complexity regularization approach which encourages low rank matricizations [100] of the weight tensor. This regularizer favours simple solutions, where simplicity is measured by the degrees of freedom of the tensor. The resulting regularization term gives rise to a non convex minimization problem. This leads us to explore two learning approaches based on convex and non-convex problems respectively.

The first of our learning approaches involves a convex relaxation of the original minimization problem. This solution is based on several recent studies which have shown that the use of the trace norm of tensors provides close convex approximations of similar minimization problems [64, 117, 183, 186, 198].

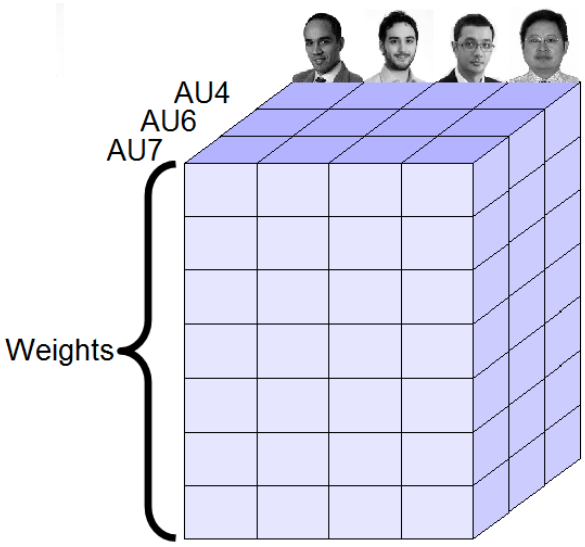

Figure 6.1.: Weight tensor modelling the relation between learning tasks to recognize several AUs from different subjects.

For our second approach we investigate a different strategy that makes use of an alternating minimization scheme for the Tucker decomposed components of the original weight tensor.

In summary, the main contributions of this chapter are:

- The extension of multitask learning to account for multi-modal relationships between tasks using multilinear models;
- Introducing a framework capable of zero-shot transfer learning with multilinear models.
- The introduction of an alternating minimization algorithm for MLMTL which implements the Tucker decomposition;

The remainder of this chapter is organized as follows. In Section 6.2, we describe the proposed learning problem, and provide insights for its interpretation. In the following two sections, we describe two alternative ways to obtain solutions to the proposed problem: Section 6.3 describes a convex relaxation of the learning problem, and Section 6.4 presents an alternating minimization algorithm for the Tucker decomposition. In Section 6.5 we compare the proposed tensor based methods with respect to their matrix based MTL counterparts, in addition to non MTL baseline models. Finally, in Section 6.6 we conclude with a discussion of the results obtained and propose potential applications of this framework, emphasizing its zero-shot transfer learning capabilities.

Definition	Notation
The set of natural numbers from 1 to N	$[N]$
Vectors	Lower case letters, e.g: w
Matrices	Upper case letters, e.g: W
Higher order tensor	Boldface Euler scripts, e.g: $\boldsymbol{\mathcal{W}}$
Inner product	$\langle \cdot, \cdot \rangle$
n-th matricization of a tensor $\boldsymbol{\mathcal{W}}$	$W_{(n)}$
n-mode product of a tensor $\boldsymbol{\mathcal{W}}$ and a matrix U	$\boldsymbol{\mathcal{W}} \times_n U$
Kronecker product	$\otimes$
n-rank of a tensor $\boldsymbol{\mathcal{W}}$	$\operatorname{rank}_n(\boldsymbol{\mathcal{W}})$
Number of dimensions of the data	d
Number of instances for task t	m_t
i-th labeled instance for task t	$(x_i^t, y_i^t) \in \mathbb{R}^d \times \mathbb{R}$
Weight vector for task t	w_t
Frobenius norm	$\|\cdot\|_{\mathrm{Fr}}$
Trace norm	$\|\cdot\|_{\mathrm{Tr}}$
Loss function	$\ell(z, y)$, e.g: $\ell(z, y) = (z - y)^2$

Table 6.1.: Index of the notation employed in this chapter.

6.2. MLMTL framework

In Tab. 6.1 we summarize the notation used in this chapter. We recall from Chapter 3 that a matricization of a tensor is a rearrangement of all the elements of the tensor to form a matrix. In particular, the mode-n matricization is obtained by concatenating the mode-n fibers of a tensor. The n-rank of a tensor is the rank of the mode-n matricization of the tensor, and the n-mode product of a tensor $\boldsymbol{\mathcal{A}}$ with a matrix U, $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_n U$, is such that $B_{(n)} = UA_{(n)}$. See Section 3.2.1 for a more detailed description of those concepts.

We consider a set of T linear regression tasks, where each of them is associated with two or more indices so that their weight vectors can be arranged in a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N}$. We regard the $d \times T$ matrix $[w_1, \ldots, w_T]$ as the mode-1 matricization, $W_{(1)}$, of the tensor $\boldsymbol{\mathcal{W}}$. Thus $p_1 = d$, $T = \prod_{n=2}^{N} p_n$ and the index t can be identified by the multi-index $(i_2, \ldots, i_N) \in [p_2] \times \cdots \times [p_N]$. We also use the shorthand notation for the data term

$$F(\boldsymbol{\mathcal{W}}) = \sum_{t=1}^{T} \sum_{i=1}^{m_t} \ell(\langle x_i^t, w_t \rangle, y_i^t) \tag{6.1}$$

We estimate the regression vectors as the solution of the joint optimization problem

$$\min \{F(\boldsymbol{\mathcal{W}}) + \gamma R(\boldsymbol{\mathcal{W}})\} \tag{6.2}$$

where γ is a positive parameter which may be chosen by cross validation. The regularizer R encourages common structure between the tasks. In particular, our goal is to encourage tensors which involve a small number of "degrees of freedom". To this end, a natural choice is to consider the average of the ranks of the matricizations of the tensors. Specifically, we define

$$R(\boldsymbol{\mathcal{W}}) = \frac{1}{N} \sum_{n=1}^{N} \text{rank}_n(\boldsymbol{\mathcal{W}}). \tag{6.3}$$

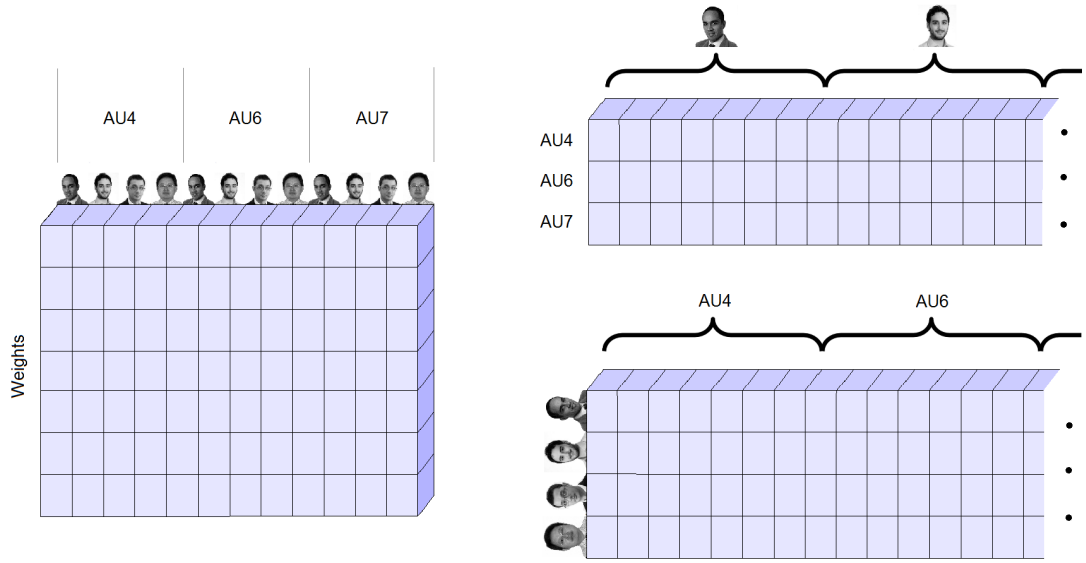

Figure 6.2.: The matricizations of the 3-mode tensor shown in Fig. 6.1.

An immediate advantage of this regularizer is that the ranks of all the matricizations are considered *simultaneously*. In order to understand the effect of this, let us analyze the implications of constraining the rank of each matricization separately. For simplicity, we will consider an $N = 3$-mode tensor composed of S subjects and R elements of a different mode of interest, such as AUs. Hence the total number of tasks is $T = RS$. We note that this example can be easily extended to bigger N.

If we only constrain the rank of the first matricization, $W_{(1)}$, which corresponds to the weights (Fig. 6.2, left), then we are considering exactly the same problem as in multitask

feature learning in eq. (2.16). Thus, in this case the weights of all tasks are assumed to lie on a common low-dimensional subspace.

The second matricization, $W_{(2)}$ (Fig.6.2, top, right) corresponds to the AUs, thus by regularizing its rank, we limit the information regarding the R AUs to have as low dimensionality as possible. To see this more clearly we can reformulate this scenario as a regular multitask learning problem with R tasks, one for each AU. To do so, we group the inputs and the outputs related to the same AU r in the following way:

$$\widetilde{X}^r = \begin{bmatrix} X_{(r,1)} & 0 & \cdots & 0 \\ 0 & X_{(r,2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & X_{(r,S)} \end{bmatrix}, \quad \widetilde{y}^r = \begin{bmatrix} y_{(r,1)} \\ y_{(r,2)} \\ \vdots \\ y_{(r,S)} \end{bmatrix}, \tag{6.4}$$

for each $r \in [R]$, where $X_{(r,s)}$ and $y_{(r,s)}$ are the inputs and outputs of all training instances associated simultaneously to AU r and subject s. Thus, $\widetilde{X}^r \in \mathbb{R}^{dS \times m_r}$, $\widetilde{y}^r \in \mathbb{R}^{m_r}$, where m_r is the number of training instances from AU r. Then we consider the problem:

$$W_{(2)} = \underset{W}{\operatorname{argmin}} \sum_{r=1}^{R} \sum_{i=1}^{m_r} \ell(\langle \widetilde{x}_i^r, w_r^R \rangle, \widetilde{y}_i^r) + \gamma \operatorname{rank}(W), \tag{6.5}$$

where $w_r^R \in \mathbb{R}^{dS}$ is the r-th row of $W_{(2)}$. The regularizer based on the rank makes this problem equivalent to learning a solution, $W_{(2)}$, that can be decomposed such that $w_r^R = B^R a_r^R$, where $B^R \in \mathbb{R}^{dS \times K_R}$ and $a_r^R \in \mathbb{R}^{K_R}$. The matrix B can be expressed as $B^R = \left[B^{R,1^\top}, B^{R,2^\top}, \dots B^{R,S^\top}\right]$, where each matrix $B^{R,s}$, $\forall s \in [S]$, is the learned embeddings for mapping the input data belonging to subject s to a common representation invariant across subjects, in a K_R dimensional space. This invariant representation will be useful for learning the R AUs recognition tasks. This is remarkably powerful, for example let us assume that several subjects smile in different ways. Then matrix B^R has the capacity to learn the mapping from the pieces of the input data that encode information about the smile of different subjects, say s_1 and s_2, to the same higher level feature through $b_k^{R,1}$ and $b_k^{R,2}$, for some $k \in K_R$. The K_R higher level features learned from the data will be available for any task r by means of $a_r^R \in \mathbb{R}^{K_R}$, which specifies the way these learned features are linearly combined to obtain the weight vector w_r for AU r.

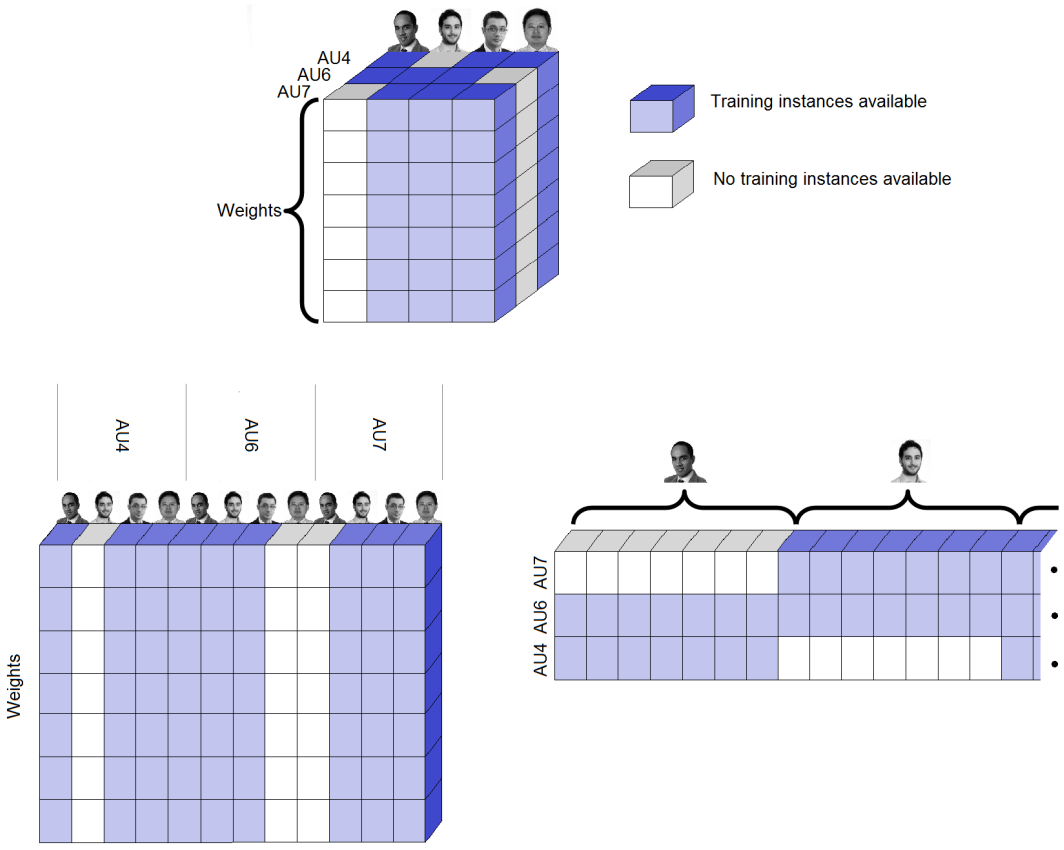

Figure 6.3.: Tensor of weight vectors and two of its matricizations, illustrating the scenario when some tasks receive no training instances.

A similar argument can be elaborated for the remaining mode(s) of the tensor. In our example, $W_{(3)}$ (Fig.6.2, bottom, right), corresponds to the matricization associated to the subjects. Thus constraining the rank of the third matricization leads to modelling all information learned about the subjects in a low dimensional space. This is equivalent to learning a $B^S = \left[B^{S,1^\top}, B^{S,2^\top}, \dots B^{S,R^\top}\right]$, in which each $B^{S,r} \in \mathbb{R}^{d \times K_S}$ is composed of projections that are useful for detection of AU r. Hence, the resultant K_S-dimensional space conveys high-level features for AUs detection, invariant to the AUs themselves. For example, $b_k^{S,1}$ and $b_k^{S,2}$ could refer to the amount of wrinkles caused when AUs 1 and 2 are active. The importance of those higher level features depends on each particular subject s, and will be encoded in a_s^S. For example, if subject s corresponds to an elder person, then $a_{s,k}^S$ may be near 0, as wrinkles are not that discriminative for AUs on a person that normally has wrinkles.

With this understanding about the effect of regularizing the different matricizations in isolation, we can expect that constraining the rank of the matricizations jointly, as con-

sidered in problem (6.2-6.3), combines all these effects.

As outlined in Section 6.1, MLMTL is capable of zero-shot transfer learning, estimating the weights of a task even when there are no training instances available for it, as shown in Fig. 6.3 (top). To see this, let us assume that task t has no training instances. If one constrains only the rank of the weights matricization, $W_{(1)}$ (see Fig. 6.3, bottom-left), then its t-th row cannot be estimated as there is no according information available. On the other hand, if one constrains any other matricization, for example $W_{(2)}$ (see Fig. 6.3, bottom-right), all rows and all columns contain information (coloured in blue in the image) and thus, the whole matrix can be estimated.

Another way to view the regularization of all matricizations simultaneously is by relating it to the Tucker decomposition. Recall from Section 3.2.2.2 that the Tucker decomposition of a tensor establishes that any tensor $\mathcal{W} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_N}$ can be decomposed as follows:

$$\mathcal{W} = \mathcal{G} \times_1 A^{(1)} \cdots \times_N A^{(N)}. \tag{6.6}$$

where $A^{(n)} \in \mathbb{R}^{p_n \times k_n}$, $n \in [N]$, are called the factor matrices, and $\mathcal{G} \in \mathbb{R}^{k_1 \times \ldots \times k_N}$ is the core tensor and models the interaction between factors.

It can be proved [51] that the rank of the n-th matricization of a tensor $\mathcal{W}$ is equal to the dimension of the n-th mode of its core tensor $\mathcal{G}$. Hence, by minimizing simultaneously the ranks of the matricizations of $\mathcal{W}$ as in problem (6.2-6.3), we are decreasing the degrees of freedom by constraining the dimensions of the core tensor.

6.3. Convex relaxation

Problem (6.2-6.3) is non-convex because of the rank function operating on each matricization. Finding a convex relaxation of $R(\cdot)$ has been the objective of recent works [64, 117, 186]. All of them suggest to use the trace norm for tensors as a good convex proxy. This is defined as the average of the trace norm of each matricization of $\mathcal{W}$,

$$\|\mathcal{W}\|_{\mathrm{Tr}} = \frac{1}{N} \sum_{n=1}^{N} \|W_{(n)}\|_{\mathrm{Tr}} \tag{6.7}$$

We highlight that the regularizer in this problem is a convex approximation of the average of the ranks of the matricizations, but unlike the matrix, this approximation may not be the best one, as noted in [183].

Note that in the particular case of a 2-order tensor, eq. (6.7) coincides with the usual notion of the trace norm of a matrix. This observation motivates us to consider the convex problem

$$\min_{\boldsymbol{\mathcal{W}}} \left\{ F\left(\boldsymbol{\mathcal{W}}\right) + \gamma \left\|\boldsymbol{\mathcal{W}}\right\|_{\mathrm{Tr}} \right\}. \tag{6.8}$$

When $N = 2$ the problem (6.8) is equivalent to the one proposed in [8]. However, if $N > 2$, problem (6.8) is more difficult to solve due to the *composite* nature of the regularizer (6.7). To explain this observation, we introduce N auxiliary tensors $\boldsymbol{\mathcal{B}}_n \in \mathbb{R}^{p_1 \times \cdots \times p_N}$, $n \in [N]$, each of which represents a version of the original tensor $\boldsymbol{\mathcal{W}}$. With this notation, problem (6.8) can be reformulated as[1]

$$\min_{\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{B}}_1, \ldots, \boldsymbol{\mathcal{B}}_N} \left\{ F\left(\boldsymbol{\mathcal{W}}\right) + \gamma \sum_{n=1}^{N} \left\| (B_n)_{(n)} \right\|_{\mathrm{Tr}} \quad \text{s.t} : \boldsymbol{\mathcal{B}}_n = \boldsymbol{\mathcal{W}},\ n \in [N] \right\} \tag{6.9}$$

where all the trace norm regularizers on the auxiliary matrices are related through the equality constraints.

As noted by [64] and [183] problem (6.9) can be solved by the alternating direction method of multipliers (ADMM) [see e.g. 23]. This optimization strategy allows problem (6.9) to be decoupled into independent subproblems which no longer have interdependent trace norm constraints. This decoupling is achieved by introducing a set of Lagrange multipliers $\boldsymbol{\mathcal{C}}_n$, $\forall n \in [N]$. The resultant augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{B}}\right) = F\left(\boldsymbol{\mathcal{W}}\right) + \sum_{n=1}^{N} \Big(& \gamma \left\| (B_n)_{(n)} \right\|_{\mathrm{Tr}} \\ & - \left\langle \boldsymbol{\mathcal{C}}_n, \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n \right\rangle + \frac{\beta}{2} \left\| \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n \right\|_{\mathrm{Fr}}^2 \Big) \end{aligned} \tag{6.10}$$

for some $\beta > 0$, where the inner product between tensors is defined as the regular inner product between the vectorized form of the tensors. We will describe in detail an algorithm to solve problem (6.10) in the next chapter, as it is key for the approach described there.

The main advantage of this approach is that it always obtains the global solution of problem (6.7). However the fact that the outputs of the algorithm are the weight vectors themselves leads to two important drawbacks. First, transfer learning is not possible directly from the model. This is because the factors (see equation (6.6) below) are

[1] The somewhat cumbersome notation $B_{n(n)}$ denotes the mode-n matricization of tensor $\boldsymbol{\mathcal{B}}_n$, that is, $B_{n(n)} = (\boldsymbol{\mathcal{B}}_n)_{(n)}$.

learned implicitly but one does not have access to them under this approach. Therefore, if we want to add a new entity (e.g. a new restaurant in the example described in the introduction), the whole algorithm needs to be run again from scratch. The second drawback is related to memory requirements. In some problems, dealing with the whole weight tensor $\mathcal{W}$ can be problematic since this can be very large. Furthermore, this approach needs to keep $N+1$ versions of the tensor in memory so the total memory needed to run the algorithm is $\mathrm{O}\Big((N+1)\prod_{n=1}^{N} p_n\Big)$, which can be unfeasible in many cases. Also, note that this approach does not optimize the original problem but a convex approximation of it. To overcome these shortcomings, we propose a new method in the following section.

6.4. Approach based on the Tucker Decomposition

In this section, we describe an alternative method which encourages low rank representations of the tensor using the Tucker decomposition, [see e.g. 100]. To do so, we minimize the error term $F(\mathcal{W})$, expressing the learning tensor using its Tucker decomposition as in eq. (6.6). Note that the Tucker decomposition is invariant under multiplication and division of different factors by the same scalar. With the aim of avoiding this issue and reducing overfitting, we add Frobenius norm regularization terms to the components. The resultant problem is

$$\min_{\mathcal{G},\mathrm{A}^{(1)},\ldots,\mathrm{A}^{(N)}} H(\mathcal{G}, A^{(1)}, \ldots, A^{(N)})$$

where we defined

$$\begin{aligned} H(\mathcal{G}, A^{(1)}, \ldots, A^{(N)}) := F(\mathcal{G} \times_1 A^{(1)} \cdots \times_N A^{(N)}) \\ +\alpha \left(\|\mathcal{G}\|_{\mathrm{Fr}}^2 + \sum_{n=1}^{N} \left\|A^{(n)}\right\|_{\mathrm{Fr}}^2 \right) \end{aligned} \tag{6.11}$$

and α is a regularization parameter. Although the regularization term is heuristic in nature, we argue in Section 6.5 that it helps avoiding overfitting.

We attempt to solve problem (6.11) by alternate minimization, where in each step we fix all components but one and solve the resultant convex problem. We distinguish three different cases: minimizing over $\mathcal{G}$, over $A^{(1)}$ (the set of components for the input data), and over $A^{(n)}$ for any $n \in \{2, \ldots, N\}$.

Minimizing over $\mathcal{G}$. Equation (6.11) can be minimized over $\mathcal{G}$ by noticing that

$$w_t = A^{(1)} G_{(1)} \left(A^{(N)} \otimes \cdots \otimes A^{(2)} \right)^\top e^t \tag{6.12}$$

where $e^t \in \mathbb{R}^T$ is a vector such that $e^t_t = 1$ and $e^t_s = 0$ for $s \neq t$. Here, we express the weight vector estimators in terms of the product of the first matricization of $\mathcal{G}$ with the other factor matrices. This leads to the convex problem

$$\min_{\mathcal{G}} \sum_{t=1}^{T} \sum_{i=1}^{m_t} \ell \left(x_i^{t\top} A^{(1)} G_{(1)} \left(A^{(N)} \otimes \cdots \otimes A^{(2)} \right)^\top e^t, y_i^t \right) + \alpha \left\| \mathcal{G} \right\|_{\mathrm{Fr}}^2$$

which we can solve by gradient descent if ℓ is differentiable. The gradient of H w.r.t $G_{(1)}$ is given by

$$\sum_{t=1}^{T} \sum_{i=1}^{m_t} \ell'_{i,t} A^{(1)\top} x_i^t e^{t\top} \left(A^{(N)} \otimes \cdots \otimes A^{(2)} \right) + 2\alpha G_{(1)}$$

where $\ell'_{i,t}$ is the derivative of ℓ with respect to its first argument evaluated at

$x_i^{t\top} A^{(1)} G_{(1)} \left(A^{(N)} \otimes \cdots \otimes A^{(2)} \right)^\top e^t$.

Finally, in order to obtain the tensor $\mathcal{G}$, we only need to invert the matricization operation.

Minimizing over $A^{(1)}$. In this case, we can reuse the equality (6.12) to minimize over $A^{(1)}$. This can be solved by gradient descent, where the gradient of H w.r.t. to $A^{(1)}$ is given by

$$\sum_{t=1}^{T} \sum_{i=1}^{m_t} \ell'_{i,t} x_i^t e^{t\top} \left(A^{(N)} \otimes \cdots \otimes A^{(2)} \right) G_{(1)}^\top + 2\alpha A^{(1)}.$$

Minimizing over $A^{(n)},\ n \in \{2, \ldots, N\}$. This set of cases is more difficult to describe. In order to simplify the presentation we assume that $N = 3$ and $n = 2$, such that $\mathcal{W} \in \mathbb{R}^{d \times R \times S}$, so that modes 2 and 3 are composed of R and S elements respectively and the number of tasks is $T = RS$. The generalization to larger values is straightforward.

First of all, it is useful to note that the 2-mode splits all tasks into R sets, each of which has S tasks. Then, we rearrange the input data belonging to each of those groups of tasks such that for every $r \in [R]$ we consider $\widetilde{X}^r$ and $\widetilde{y}^r$ as in eq. (6.4). Recall that $\widetilde{X}^r$ and $\widetilde{y}^r$ contain all the m_r training instances associated to the r-th element of the second mode, such that $\widetilde{X}^r \in \mathbb{R}^{dS \times m_r}$, $\widetilde{y}^r \in \mathbb{R}^{m_r}$.

Then, we can write

$$\sum_{t=1}^{T}\sum_{i=1}^{m_t}\ell\left(x_i^{t\top}W_{(1)}e^t,\, y_i^t\right) = \sum_{r=1}^{R}\sum_{i=1}^{m_r}\ell\left(\widetilde{x}_i^{r\top}W_{(2)}e^r,\, \widetilde{y}_i^r\right)$$
$$= \sum_{r=1}^{R}\sum_{i=1}^{m_r}\ell\left(\widetilde{x}_i^{r\top}\left(A^{(3)}\otimes A^{(1)}\right)G_{(2)}^{\top}A^{(2)\top}e^r,\, \widetilde{y}_i^r\right).$$

Notice that, unlike the previous cases, the columns of $A^{(2)\top}$ are decoupled, so we can solve instead R simpler problems. The corresponding gradient of H with respect to $\left(A^{(2)\top}\right)_r = A^{(2)\top}e^r$ is given by:

$$\sum_{i=1}^{m_r}\ell'_{i,r}G_{(2)}\left(A^{(3)}\otimes A^{(1)}\right)^{\top}\widetilde{x}_i^r + 2\alpha\left(A^{(2)\top}\right)_r.$$

The local approach has a number of advantages derived from the explicit calculation of the factors. First, it allows for adding new factors without the necessity of relearning the previous factors, thereby allowing for transfer learning in a natural way. Second, the memory needed is $\mathrm{O}\left(\sum_{n=1}^{N}p_nk_n + \prod_{n=1}^{N}k_n\right)$ which can be much smaller than that of the convex approach, particularly if $k_n \ll p_n$ for some $n \in [N]$.

The main drawback of this approach is that the solution obtained is a local optimum, and depends on the initialization of the algorithm. There is no guarantee about how far this is from the global optimum, thus the solutions obtained may be poor. In this study we initialize the parameters at random, sampling each of them from the standard normal distribution, $\mathcal{N}(0,1)$. A more elaborated initialization may lead to better performance.

6.5. Experiments

In this section we empirically test the performances of the proposed methods on multi-aspect data with the objective of checking whether this new methods provide some advantages over traditional MTL methods. With that aim, we conducted a set of experiments on one synthetic dataset and two real world datasets. For each experiment we explain the experimental procedure, and analyze the results. We compare the predictive performance of the following five methods:

- Ridge Regression (RR): this model, chosen as a baseline, makes no assumption regarding the relationships among the tasks.
- Multitask Feature Learning (MTFL): a convex MTL approach developed in [8], and previously described in Section 2.3.3, which assumes that all tasks share a common low dimensional representation of the data.

- Matrix Factorization (MTL-NC): a non convex MTL approach consisting of applying the matrix based counterpart to the method proposed in Section 6.4.
- Convex Multilinear Multitask Learning (MLMTL-C): this approach, based on tensor trace norm regularization, is described in Section 6.3 and corresponds to an extension of MTFL to multilinear algebra.
- Non-convex Multilinear Multitask Learning (MLMTL-NC): this is the approach proposed in Section 6.4.

The last two methods were implemented using the Tensor Toolbox [17]. The non-convex methods (MTL-NC and MLMTL-NC) require the (Tucker) rank as a hyper-parameter. The way this is chosen is described in each experiment. Additionally, all methods have one further hyper-parameter which needs to be tuned. This is always done by means of a validation set. The range of values explored for the hyperparameter is 10^s for $s \in \{-3, -2, -1, \ldots, 4, 5, 6\}$. Preliminary experiments show that this range empirically contains the best solution for all approaches.

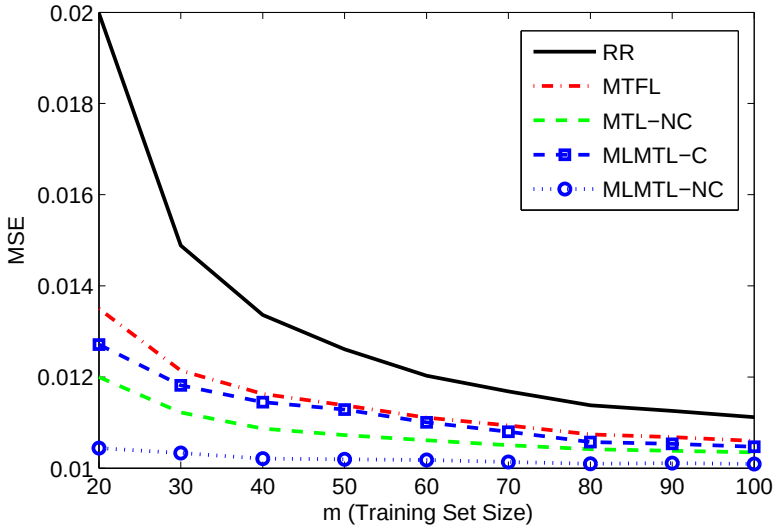

Figure 6.4.: Synthetic dataset: Mean Square Error (MSE) comparison between Ridge Regression (RR), Multitask Feature Learning [8] (MTFL), Matrix Factorization MTL (MTL-NC), Convex Multilinear Multitask Learning (MLMTL-C) and Non-convex Multilinear Multitask Learning (MLMTL-NC).

6.5.1. Synthetic data

In order to test the implementation of the algorithms and to check whether the methods lead to improvements, we created a synthetic dataset where the weight tensor is decom-

posable as described in eq. (6.6). The dataset is generated as follows: we created a set of $T = 100$ tasks, organized in an $p_2 \times p_3$ grid where $p_2 = p_3 = 10$ and the input data have dimensionality $p_1 = 10$. The tasks' weight vectors could consequently be organized in an $N = 3$ mode tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$. Furthermore, this tensor was generated so that $\text{rank}_n(\boldsymbol{\mathcal{W}}) = 3, \forall n \in [N]$. In particular, every element in the tensor was generated as $w_{h_1,h_2,h_3} = \sum_{k_1,k_2,k_3=1}^{3} g_{k_1,k_2,k_3} a^{(1)}_{h_1,k_1} a^{(2)}_{h_2,k_2} a^{(3)}_{h_3,k_3}$, $\forall h_1 \in [p_1]$, $\forall h_2 \in [p_2]$, $\forall h_3 \in [p_3]$, where all elements $a^{(1)}_{h_1,k_1}$, $a^{(2)}_{h_2,k_2}$, $a^{(3)}_{h_3,k_3}$, g_{k_1,k_2,k_3} are generated by randomly sampling from a standard Gaussian distribution $\mathcal{N}(0,1)$. For each task $t = (i,j)$, a set of m training instances $x^t_1, \ldots, x^t_m \in \mathbb{R}^d$ were sampled from $\mathcal{N}(0, I)$ and the labels were generated as the linear regression $y^t_i = w_t^\top x^t_i + \xi^t_i$, where $w_t = \boldsymbol{\mathcal{W}}_{i,j}$ and ξ^t_i are sampled i.i.d. from $\mathcal{N}(0, 0.1)$. Similarly, a set of m_{val} and m_{test} instances and their corresponding labels were generated for each task for validation and testing purposes. The validation set was used to tune the regularization parameter for all approaches. Additionally for the factorization techniques (MTL-NC and MLMTL-NC), the number of factors for each mode was fixed to the known values of the ranks.

In order to investigate the effect of the number of training samples available, we repeated the experiment 20 times, each has been done for several values of m in the range $[20, 100]$. The average results are shown in Fig.6.4, where we see that all MTL approaches perform better than ridge regression as expected. Furthermore, we see that among the convex approaches, MLMTL-C is slightly better than its matrix counterpart MTFL although these differences are only significant for $m < 60$. Regarding the non-convex approaches, we see that MLMTL-NC obtains the best performance with a clear improvement with respect to all remaining approaches. Nevertheless, in the current setting, the non-convex approaches have advantage in that the ground truth ranks of the tensor are known for the synthetic dataset.

To test the sensitivity of the MLMTL-NC approach with respect to incorrect values of the ranks, we carried out a similar experiment where we compared MLMTL-C and several versions of MLMTL-NC, taking different values for the ranks. The results are shown in Fig.6.5. The MLMTL-NC approaches with ranks $= (1,1,1)$ and ranks $= (2,2,2)$, which have ranks smaller than the true values, are not shown due to very poor performance[2]. As expected, the best approach is the one where the n-rank $= (3,3,3)$ coincides with the actual ranks of the tensor. However, we see that MLMTL-NC approaches with higher values of ranks perform quite similarly and in all of these cases there is an improvement with respect to MLMTL-C approach. This supports the

[2]MLMTL-NC $(2,2,2)$ approach obtains an error around 0.08 whereas the error of MLMTL-NC $(1,1,1)$ approach is above 0.16

hypothesis that MLMTL-NC approach is quite insensitive to the values of ranks, as long as they are an overestimation of the actual values.

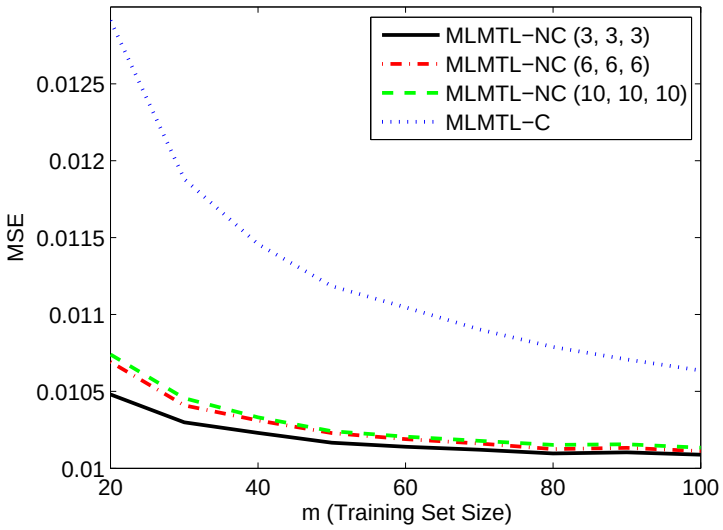

Figure 6.5.: Synthetic dataset: Mean Square Error (MSE) comparison between Convex Multilinear Multitask Learning (MLMTL-C) and three versions of Non-convex Multilinear Multitask Learning (MLMTL-NC) (having different values for the ranks).

6.5.2. Real data

In this section, we test the methods on two real world datasets. The two datasets are composed of several regression functions to be learned for several subjects. In order to test the generality of our framework, the datasets were chosen so that they differ on the topic and kind of data. For both datasets we want to infer the weight tensor $\mathcal{W} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$, where p_1 is the number of attributes, p_2 is the number of tasks we want to learn for each subject, and p_3 is the number of subjects involved in the data.

In these experiments, in order to appraise the effect of discarding the information about the mode related to subjects, we also compared with versions of the non-MLMTL approaches that ignore the subject identifier index and groups all of the instances. This leads to a single generic impersonal predictor for each of the p_2 tasks. The resulting approaches are denoted as GRR, GMTFL and GMTL-NC, where *G* stands for "grouped".

For the multilinear approaches, the value of each n-rank for MLMTL-NC was set to $\min(10, \mathrm{p_n})$ for both experiments. This value was chosen as a safe overestimate of the true rank on these data. The results of the previous experiments, presented in Fig.6.5, show that overestimates have a minimal effect on the final performance.

6.5.2.1. Restaurant & Consumer Dataset

The Restaurant & Consumer Dataset [206] contains data to build a restaurant recommender system where the objective is to predict consumer ratings given to different restaurants. Each of the $p_3 = 138$ consumers gave $p_2 = 3$ scores for food quality, service quality and overall quality. The dataset also contains $p_1 = 44$ various descriptive attributes of the restaurants (such as geographical position, cuisine type and price band). We consider this to be a regression problem where the objective is to predict the scores given the attributes of a restaurant as an input query. Since there are 138 consumers and 3 scores to predict, this leads to a multitask problem composed of 138×3 regression tasks.

This experiment was conducted in a similar way to the synthetic dataset, so that the training, validation and test sets were randomly selected for each task. The process was repeated 20 times for each value of $m \in \{250, 500, \ldots, 1500\}$ and the average results are shown in Fig. 6.6.

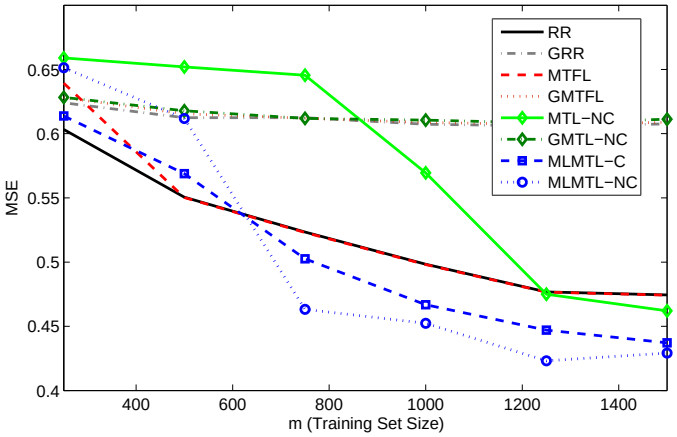

Figure 6.6.: Restaurant & Consumer Dataset: Mean Square Error (MSE) comparison between Ridge Regression (RR), Grouped Ridge Regression (GRR), Multitask Feature Learning (MTFL) Grouped Multitask Feature Learning (GMTFL), Matrix Factorization MTL (MTL-NC), Grouped Matrix Factorization (GMTL-NC), MTL Convex Multilinear Multitask Learning (MLMTL-C) and Non-convex Multilinear Multitask Learning (MLMTL-NC).

Analysing the results we can distinguish two regimes with respect to the training set size. In the first one, where $m < 750$, MTL methods do not seem to lead to any advantage. We observe that when $m = 250$ the best method is RR (independent task learning), whereas the worst convex method is MTFL. The poor performance of the latter can be

explained by the fact that MTFL assumes homogeneous relationships between tasks. This assumption does not hold here, and as a result negative transfer arises. MLMTL-C performs better than MTFL, but still worse than RR, which suggests that a low number of training instances does not provide sufficient information regarding all three modalities of the tensor. The poor performance of non-convex methods in this regime may be caused by local minima.

In the second regime, when $m \geq 750$, we observe that both MLMTL approaches outperform the remaining methods. A set of paired t-tests conducted between each pair of MLMTL and non-MLMTL shows that this improvement in the performance is significant obtaining p-values below 0.01 in each case. This fact supports our hypothesis about the multimodal relation among tasks and confirms that MLMTL can better take advantage of this over conventional MTL methods. We also checked the significance of the improvement observed between both MLMTL methods for $m \geq 750$, obtaining p-values< 0.025.

6.5.2.2. Shoulder pain dataset

In the second real world experiment we used the Shoulder Pain dataset [123], which was also used in Section 5.5 and contains video clips of the faces of people experiencing shoulder pain while performing active and passive exercises. In this case our objective is to recognise the degree of intensity of several FACS Action Units (AU) [57] for each of the patients. The importance of recognizing AUs activation is that the psychology literature provides AU-based mathematical formulae for the recognition of emotion type and intensity.

One common problem on this kind of data is that some subjects may not have shown any intensity for specific AUs in the training set. For such AU/patient tasks, traditional supervised learning approaches will not be effective. In contrast, MLMTL methods can naturally handle this zero-shot transfer learning scenario. Therefore, in this dataset we focus on assessing the performance of the methods in situations where no instances are provided to learn some of the tasks. The performance of the approaches is measured only on those tasks with no training instances, which we refer to as target tasks hereafter.

Let us recall that for each frame of the video, the facial expression is described by a set of 132 attributes (2D positions of 66 anatomical points). Furthermore, each frame has been coded in terms of AUs related to expressions of pain. For the purpose of this experiment we use the first five AUs, which are among the ones involved in pain expression [123]. These are AU4 (brow-lowering), AU6 (cheek-raising), AU7 (eyelid tightening), AU9

(nose wrinkling), and AU10 (upper-lip raising). Each AU was coded on a scale of 5 levels of intensity. The objective is to learn how to recognize the intensity of each of the $p_2 = 5$ enumerated AUs for each of the $p_3 = 5$ subjects by using the training instances provided, which have a dimensionality of $p_1 = 132$ attributes.

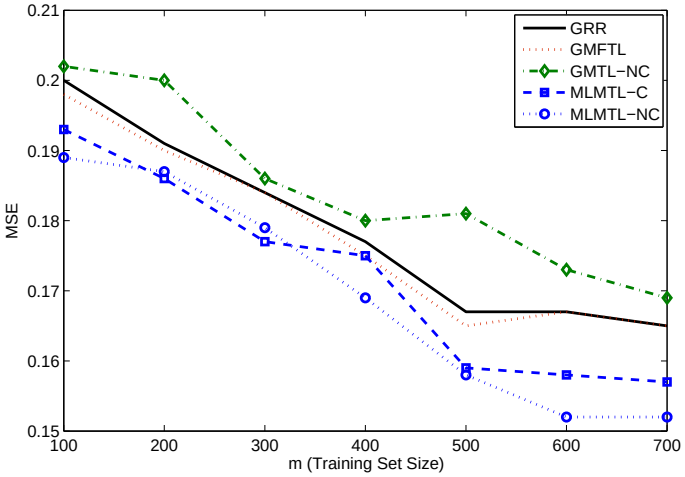

Figure 6.7.: Shoulder Pain database: Mean Square Error (MSE) comparison between Grouped Ridge Regression (GRR), Grouped Multitask Feature Learning (GMTFL), Matrix Trace Norm Regularization (GMTL-NC), Convex Multilinear Multitask Learning (MLMTL-C) and Non-Convex Multilinear Multitask Learning (MLMTL-NC).

We randomly selected $T_{target} = 2$ tasks and exclude their data from the training set. Similarly, another set of tasks $T_{validation} = 2$ were selected randomly for tuning the hyperparameters so that at the training stage, no instances from these tasks were used. Finally, the models used $m \in \{100, 200, \ldots, 700\}$ instances to estimate the remaining tasks. Note that classic supervised learning approaches cannot learn predictors for tasks where there are no training instances. Therefore, we only compare with the grouped approaches (GRR, GMTFL and GMTL-NC). We ran 30 trials for each value of m, and the averaged results are shown in Fig.6.7.

The results show that the tensor approaches outperform their matrix counterparts. A paired t-test shows that the improvement between MLMTL-NC and any other matrix approach is significant ($p < 0.01$) in all cases. Also we see that MLMTL-NC generally outperforms MLMTL-C.

6.6. Discussion

In this chapter, we investigated two approaches for multilinear multitask learning. The first is an adaptation of the low-rank tensor recovery strategy which employs a convex relaxation of the tensor decomposition problem. The second is based on an alternate minimization algorithm which optimizes the original non-convex problem together with a set of Frobenius norm regularizers to avoid overfitting. The experiments carried out on both synthetic and real data support the hypothesis that employing multilinear methods in the described MTL scenarios is advantageous.

These approaches are useful in a multitask learning scenario where there is a priori information about how tasks are related, and these relations are expressed as combinations of prescribed aspects of data. This is the case for many real world datasets that contain multiple aspects. Even though such datasets are now commonplace, often inter-task relationships are only exploited in one modality such as in classical MTL. Furthermore, we have seen that multilinear models can obtain predictors for tasks which have no training instances, so long as there are enough training instances for other related tasks. This could potentially be of significant value in scenarios where specific instances in the data are missing or more difficult to gather.

MLMTL can open up a new way of designing affect recognitions experiments which require natural behaviour data gathering. As mentioned in the introduction, collecting data of this nature requires long recordings and/or inducing some stimulus on the subjects in order to capture the desired affective states. This is particularly inconvenient for strongly negative affective expressions, such as anxiety and pain, which raise ethical concerns in inducing them. Let us for example consider the case of building a system able to recognize when a person is feeling different degrees of pain by using facial expressions as input data. This requires that we collect data from a set of patients performing different exercises which make them feel pain. These data will be labelled by a group of specialists, procedure which is usually very costly. Furthermore, we may ask these patients to perform additional simple tasks with their face, such as raising the eyebrows within an interval of time, smiling in the next interval and so on. These simple tasks have two important properties. First, we do not need any specialist to label these data; in fact, the labeling can be done automatically by inducing the subject to perform particular gestures at predefined ranges of time. Second, these tasks are somehow related to the ones of interest to us, as they provide clues about the muscular movements of the face of patients.

MLMTL is capable of extrapolating the information learned from auxiliary tasks in

order to build the principal tasks for a new subject. The auxiliary tasks are required to be both related to the principal tasks and very easy to label. In this way, one can make use of the labelled instances of these auxiliary tasks to allow the model to learn the information representing new users, and leverage the information from the training set subjects to learn the task for the new subject.

The training process is illustrated in Fig. 6.8, left, and we call it *model construction*. When the system needs to be applied to a new subject, he/she is asked to perform the simple exercises which comprise the auxiliary tasks. By doing this kind of calibration process, the system obtains labelled data from the new patient about the auxiliary tasks. This is useful for the model to acquire information regarding this new subject, so that it can learn estimators for the principal tasks that are tailored to her/him. We call this stage *user calibration* and it is illustrated in Fig. 6.8, center. Finally the principal tasks (pain recognition) can be tested on the new subject, as shown in Fig. 6.8, right.

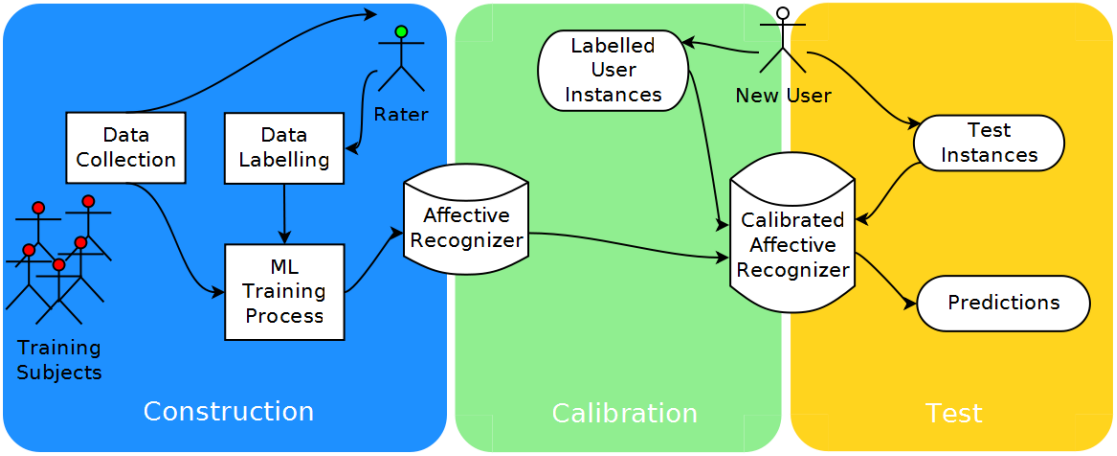

Figure 6.8.: Affect recognition models adapt themselves to operate on new subjects by means of MLMTL.

The results in the experimental section, particularly those in Section 6.5.2.2 which show that MLMTL methods are advantageous in zero-shot transfer learning settings, suggest that the described data gathering process can be succesfully applied to the scenario outlined above.

The framework explained in this chapter has recently been used and extended by other researchers in [185], where the authors use reproducing kernel spaces to consider non Euclidean features, such as graphs or probability distributions, in any mode of the tensor. There have been other works that have independently considered problems that share some points with the one explored here. For example, in [215] the authors con-

sider a machine learning model tailored to the recognition of markers for Alzheimer's disease which involves learning several tasks across time. The problem can be seen as learning a tensor of weights whose dimensions are: number of dimensions of the input data, number of tasks, and time points. In their model the authors impose trace norm regularization on the matricization related to the attributes, and a weighted sum of trace norm and $\ell_{2,1}$-norm regularization on the matricization related to the tasks, having no constraints on the remaining matricization. Zero-shot transfer learning cannot be exploited in this case as all tasks receive the same instances. Another more recent work is presented in [43], where the authors model how several annotators judge the output of different machine translation systems. They model the problem using multitask Gaussian Processes, where the task covariance matrix is expressed as the Kronecker product of smaller kernels, one for each aspect of the data.

To conclude this chapter we would like to highlight two limitations of our methods that may lead to further research. The first one is regarding the use of only one hyperparameter to control the regularization over all matricizations of the tensor (γ in MLMTL-C and α in MLMTL-NC). An avenue for further study would be to assign a hyperparameter to each matricization regularizer, in order to trade-off the regularizing effect on each matricization. An interesting goal would be to find a way to tune these hyperparameters without any significant increase in computational expense. A second limitation is that of scalability of the proposed algorithms to big tensors where the number of elements, $\prod_{n=1}^{N} p_n$, is large. The datasets used in this chapter imply the use of small tensors of weights, as otherwise the proposed algorithms would not be computationally feasible. For example, a standard desktop computer is unable to run our algorithms with tensors in $\mathbb{R}^{200\times200\times200\times200}$. The study of scalable approaches based on concurrent algorithms and perhaps different models is thus very appealing.

7. A New Convex Relaxation for Tensor Completion

In the previous chapter, we proposed and studied Multilinear Multitask Learning, a framework that by means of multilinear algebra, can leverage the relation between tasks when they can be referred by multiple indices. This framework can be cast as a more general problem of learning a tensor from a set of linear measurements. This problem, called tensor recovery, has attracted the attention of many researchers, [64, 117, 185, 186, 198, 199, 200], due to the wide range of applications it finds, such as collaborative filtering [97], to computer vision [117], and medical imaging [64], among others. In this chapter, we first describe a weakness of the most used convex method to tensor completion. Then we propose a new convex method that avoids that weakness, and we develop an algorithm for solving the associated regularization problem.

7.1. Problem statement

The most widely used convex approach to tensor recovery is based upon the extension of trace norm regularization [190] to tensors. As we explained in Section 6.3, this involves computing the average of the trace norm of each matricization of the tensor [100]. A key insight behind using trace norm regularization for matrix completion is that this norm provides a tight convex relaxation of the rank of a matrix defined on the spectral unit ball [61]. Unfortunately, the extension of this methodology to the more general tensor setting is not straightforward, as it imposes simultaneous constraints on the same tensor. In particular, we shall prove that the tensor trace norm is not a tight convex relaxation of the tensor rank.

This downside stems from the fact that the spectral norm, used to compute the convex relaxation for the trace norm, is not an invariant property of the matricization of a tensor. This observation leads us to take a different route and study afresh the convex relaxation of tensor rank on the Euclidean ball. We show that this relaxation is tighter than

the tensor trace norm, and we describe a technique to solve the associated regularization problem. This method builds upon the alternating direction method of multipliers and a subgradient method to compute the proximity operator of the proposed regularizer. Furthermore, we present numerical experiments on one synthetic dataset and three real-world datasets, which suggest that the proposed method improves significantly over tensor trace norm regularization in terms of estimation error, while remaining computationally tractable.

The chapter is organized in the following manner. In Section 7.2, we describe the tensor completion framework. In Section 7.3, we highlight some limitations of the tensor trace norm regularizer and present an alternative convex relaxation for the tensor rank. In Section 7.4, we describe a method to solve the associated regularization problem. In Section 7.5, we report on our numerical experiments with the proposed method. Finally, in Section 7.6, we summarize the main contributions of the chapter and discuss future directions of research.

7.2. Tensor trace norm

In Tab.7.1 we summaryze the notation used in this chapter. We refer to Section 3.2.1 for more details. If $x \in \mathbb{R}^d$ then for every $r \leq s \leq d$, we define $x_{r:s} := (x_i : r \leq i \leq s)$. We also use the notation $p_{\min} = \min\{p_1, \ldots, p_N\}$ and $p_{\max} = \max\{p_1, \ldots, p_N\}$.

We are now ready to describe the learning problem. We choose a linear operator $\mathcal{I} : \mathbb{R}^{p_1 \times \cdots \times p_N} \to \mathbb{R}^m$, representing a set of linear measurements obtained from a target tensor $\boldsymbol{\mathcal{W}}^0$ as $y = \mathcal{I}(\boldsymbol{\mathcal{W}}^0) + \xi$, where ξ is some noise. Tensor completion is an important

Definition	Notation
The set of natural numbers from 1 to N	$[N]$
Vectors	Lower case letters, e.g: w
Matrices	Upper case letters, e.g: W
Higher order tensor	Boldface Euler scripts, e.g: $\boldsymbol{\mathcal{W}}$
Inner product	$\langle \cdot, \cdot \rangle$
n-th matricization of a tensor $\boldsymbol{\mathcal{W}}$	$W_{(n)}$
Vector of singular values of matrix W	$\sigma(W)$
Frobenius norm	$\|\cdot\|_{\mathrm{Fr}}$
Trace norm	$\|\cdot\|_{\mathrm{Tr}}$
Spectral norm	$\|\cdot\|_{\mathrm{Sp}}$

Table 7.1.: Index of the notation employed in this chapter.

example of this setting; in this case the operator $\mathcal{I}$ returns the known elements of the tensor. That is, we have $\mathcal{I}(\boldsymbol{\mathcal{W}}^0) = (\boldsymbol{\mathcal{W}}^0_{i_1(j),\dots,i_N(j)} : j \in [m])$, where for every $j \in [m]$ and $n \in [N]$, the index $i_n(j)$ is a prescribed integer in the set $[p_n]$. Our aim is to recover the tensor $\boldsymbol{\mathcal{W}}^0$ from the data $(\mathcal{I}, y)$. To this end, we solve the regularization problem

$$\min \left\{ \|y - \mathcal{I}(\boldsymbol{\mathcal{W}})\|_2^2 + \gamma R(\boldsymbol{\mathcal{W}}) : \boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N} \right\} \tag{7.1}$$

where γ is a positive parameter which may be chosen by cross validation. The role of the regularizer R is to encourage solutions $\boldsymbol{\mathcal{W}}$ which have a simple structure in the sense that they involve a small number of "degrees of freedom". A natural choice [64, 117, 185, 186, 198, 199, 200], which we also made in the previous chapter, is to consider the average of the rank of the tensor's matricizations.

$$R(\boldsymbol{\mathcal{W}}) = \frac{1}{N} \sum_{n=1}^{N} \operatorname{rank}(W_{(n)}). \tag{7.2}$$

Finding a convex relaxation of this combinatorial regularizer has been the subject of recent works [64, 117, 186]. They all propose to use the average of the trace norm of each matricization of $\boldsymbol{\mathcal{W}}$, that is,

$$\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}} = \frac{1}{N} \sum_{n=1}^{N} \|W_{(n)}\|_{\mathrm{Tr}} \tag{7.3}$$

where $\|W_{(n)}\|_{\mathrm{Tr}}$ is the trace (or nuclear) norm of matrix $W_{(n)}$, namely the ℓ_1-norm of the vector of singular values of matrix $W_{(n)}$ (see, e.g. [78]).

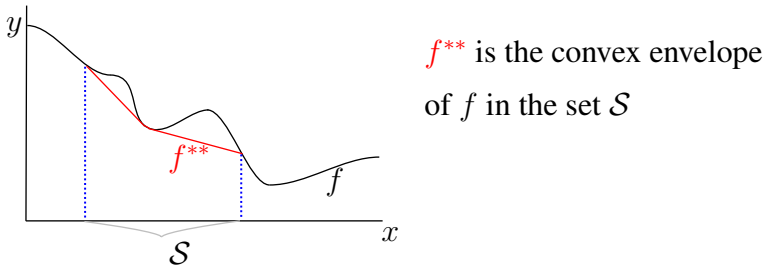

Figure 7.1.: Illustration of the convex envelope of a function f on a given set $\mathcal{S}$.

In order to find a rationale behind the regularizer (7.3) let us recall the concept of convex envelope. We say that the convex envelope of a function f on a set $\mathcal{S}$ is the largest convex function f^{**} which is upper-bounded by f for all points in $\mathcal{S}$ (see Fig.7.1). The trace norm is the convex envelope of the rank of a matrix on the spectral unit ball, see [61, Thm. 1]. A way to proceed is by defining a function which behaves as the rank for all

points in $\mathcal{S}$, but becomes infinity for all points out of $\mathcal{S}$: $\Psi : \mathbb{R}^{p_1 \times p_2} \to \mathbb{R} \cup \{\infty\}$

$$\Psi(W) = \begin{cases} \operatorname{rank}(W), & \text{if } \|W\|_{\mathrm{Sp}} \leq 1 \\ +\infty, & \text{otherwise} \end{cases} \tag{7.4}$$

where $\|\cdot\|_{\mathrm{Sp}}$ is the spectral norm, that is, the largest singular value of W. The convex envelope can be derived by computing the double convex conjugate of Ψ. This is defined as

$$\Psi^{**}(W) = \sup\left\{\langle W, S\rangle - \Psi^*(S) : S \in \mathbb{R}^{p_1 \times p_2}\right\} \tag{7.5}$$

where Ψ^* is the conjugate of Ψ, that is $\Psi^*(S) = \sup\{\langle W, S\rangle - \Psi(W) : W \in \mathbb{R}^{p_1 \times p_2}\}$. Note that Ψ is a spectral function, that is, $\Psi(W) = \psi(\sigma(W))$ where $\psi : \mathbb{R}^d_+ \to \mathbb{R}$ denotes the associated symmetric function on the singular values. Using von Neumann's trace theorem (see e.g. [78]) it is easily seen that $\Psi^*(S)$ is also a spectral function. That is, $\Psi^*(S) = \psi^*(\sigma(S))$, where

$$\psi^*(\sigma) = \sup\left\{\langle \sigma, w\rangle - \psi(w) : w \in \mathbb{R}^d\right\}, \quad \text{with } d := \min(p_1, p_2).$$

We refer to [61] for a detailed discussion of these ideas. We will use this equivalence between spectral and singular values functions repeatedly in this chapter.

7.3. Alternative convex relaxation

In this section, we show that the tensor trace norm is not a tight convex relaxation of the tensor rank R in equation (7.2). We then propose an alternative convex relaxation for this function.

Note that due to the composite nature of the function R, computing its convex envelope is a challenging task and one needs to resort to approximations. In [185], the authors note that the tensor trace norm $\|\cdot\|_{\mathrm{Tr}}$ in equation (7.3) is a convex lower bound to R on the set

$$\mathcal{S}_{\mathrm{Sp}} := \left\{\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N} : \left\|W_{(n)}\right\|_{\mathrm{Sp}} \leq 1, \ \forall n \in [N]\right\}.$$

The key insight behind this observation is summarized in the following lemma.

Lemma 7.3.1. *Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_N$ be convex subsets of a Euclidean space and let $\mathcal{D} = \bigcap_{n=1}^N \mathcal{Q}_n \neq \emptyset$. Let $g : \prod_{n=1}^N \mathcal{Q}_n \to \mathbb{R}$ and let $h : \mathcal{D} \to \mathbb{R}$ be the function defined, for*

every $x \in \mathcal{D}$, as $h(x) = g(x, \dots, x)$. Then, for every $x \in \mathcal{D}$, it holds that

$$h^{**}(x) \geq g^{**}(x_1, \dots, x_N) \Big|_{x_n = x,\ \forall n \in [N]} .$$

Proof. Since the restriction of g on $\mathcal{D}^N \subseteq \prod_{n=1}^N \mathcal{Q}_n$ is equivalent to h, the convex envelope of g when evaluated on the smaller set $\mathcal{D}^N$ cannot be larger than the convex envelope of h on $\mathcal{D}$. □

Using this result it is immediately possible to derive a convex lower bound for the function R in equation (7.2). Since the convex envelope of the rank function on the unit ball of the spectral norm is the trace norm, using Lemma 7.3.1 with $\mathcal{Q}_n = \{\boldsymbol{\mathcal{W}} : \|W_{(n)}\|_{\mathrm{Sp}} \leq 1\}$ and

$$g(\boldsymbol{\mathcal{W}}_1, \dots, \boldsymbol{\mathcal{W}}_N) = \frac{1}{N} \sum_{n=1}^{N} \operatorname{rank}((W_n)_{(n)}),$$

we can conclude that the convex envelope of the function R on the set $\mathcal{S}_{\mathrm{Sp}}$ is bounded from below by $\frac{1}{N} \sum_{n=1}^N \|W_{(n)}\|_{\mathrm{Tr}}$.

However, the authors of [185] leave open the question of whether the tensor trace norm is the convex envelope of R on the set $\mathcal{S}_{\mathrm{Sp}}$. In the following, we prove that this question has a negative answer by showing that there exists a convex function $\Omega \neq \|\cdot\|_{\mathrm{Tr}}$ which minorizes the function R on $\mathcal{S}_{\mathrm{Sp}}$ and such that for some tensor $\boldsymbol{\mathcal{W}} \in \mathcal{S}_{\mathrm{Sp}}$ it holds that $\Omega(\boldsymbol{\mathcal{W}}) > \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}}$.

To describe our observation we introduce the set

$$\mathcal{S}_{\mathrm{Fr}} := \left\{ \boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \dots \times p_N} : \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Fr}} \leq 1 \right\}$$

where $\|\cdot\|_{\mathrm{Fr}}$ is the Frobenius norm for tensors, that is,

$$\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Fr}}^2 := \sum_{i_1=1}^{p_1} \cdots \sum_{i_N=1}^{p_N} w_{i_1, \dots, i_N}^2 .$$

We will choose

$$\Omega(\boldsymbol{\mathcal{W}}) = \Omega_\alpha(\boldsymbol{\mathcal{W}}) := \frac{1}{N} \sum_{n=1}^{N} \omega_\alpha^{**} \left(\sigma \left(W_{(n)} \right) \right) \tag{7.6}$$

where ω_α^{**} is the convex envelope of the cardinality of a vector on the ℓ_2-ball of radius α and we will choose $\alpha = \sqrt{p_{\min}}$. Note, by Lemma 7.3.1, that for every $\alpha > 0$, function

Ω_α is a convex lower bound of function R on the set $\alpha\mathcal{S}_{\mathrm{Fr}}$.

Below, for every vector $s \in \mathbb{R}^d$ we denote by $s^\downarrow$ the vector obtained by reordering the components of s so that they are non increasing in absolute value, that is $|s_1^\downarrow| \geq \cdots \geq |s_d^\downarrow|$.

Lemma 7.3.2. *Let ω_α^{**} be the convex envelope of the cardinality on the ℓ_2-ball of radius α. Then, for every $x \in \mathbb{R}^d$ such that $\|x\|_2 = \alpha$, it holds that $\omega_\alpha^{**}(x) = \mathrm{card}\,(x)$.*

Proof. First, we note that the conjugate of the function card on the ℓ_2 ball of radius α is given by the formula

$$\omega_\alpha^*(s) = \sup_{\|y\|_2 \leq \alpha} \{\langle s, y\rangle - \mathrm{card}\,(y)\} = \max_{r \in \{0,\ldots,d\}} \{\alpha \|s_{1:r}^\downarrow\|_2 - r\}. \tag{7.7}$$

Hence, by the definition of the double conjugate, we have, for every $s \in \mathbb{R}^d$ that

$$\omega_\alpha^{**}(x) \geq \langle s, x\rangle - \max_{r \in \{0,\ldots,d\}} \{\alpha \|s_{1:r}^\downarrow\|_2 - r\}.$$

In particular, if $s = kx$ for some $k > 0$ this inequality becomes

$$\omega_\alpha^{**}(x) \geq k\|x\|_2^2 - \max_{r \in \{0,\ldots,d\}} (\alpha k \|x_{1:r}^\downarrow\|_2 - r).$$

If k is large enough, the maximum is attained at $r = \mathrm{card}(x)$. Consequently,

$$\omega_\alpha^{**}(x) \geq k\alpha^2 - k\alpha^2 + \mathrm{card}(x) = \mathrm{card}(x).$$

By the definition of the convex envelope, it also holds that $\omega_\alpha^{**}(x) \leq \mathrm{card}(x)$. The result follows. □

The function ω_α^{**} resembles the norm developed in [9], which corresponds to the convex envelope of the indicator function of the cardinality of a vector in the ℓ_2 ball. The extension of its application to tensors is not straightforward however, as it is necessary to specify beforehand the rank of each matricization.

The next lemma together with Lemma 7.3.2 provide a sufficient condition for the existence of a tensor $\boldsymbol{\mathcal{W}} \in \mathcal{S}_{\mathrm{Sp}}$ at which the regularizer in equation (7.6) is strictly larger than the tensor trace norm.

Lemma 7.3.3. *If $N \geq 3$ and $p_1, \ldots, p_N$ are not all equal to each other, then there exists $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_N}$ such that:* (a) $\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Fr}} = \sqrt{p_{\min}}$, (b) $\boldsymbol{\mathcal{W}} \in \mathcal{S}_{\mathrm{Sp}}$, (c) $\min_{n \in [N]} \mathrm{rank}(W_{(n)}) <$

$\max_{n\in[N]} \operatorname{rank}(W_{(n)})$.

Proof. Without loss of generality we assume that $p_1 \leq \cdots \leq p_N$. By hypothesis $p_1 < p_N$. First we consider the special case

$$p_1 = \cdots = p_{N-1}, \text{ and } p_N = p_1 + 1. \tag{7.8}$$

We define a class of tensors $\boldsymbol{\mathcal{W}}$ by choosing a singular value decomposition for their mode-N matricization,

$$w_{i_1,i_2,\ldots,i_N} = \sum_{k=1}^{p_N} \sigma_k u^k_{i_N} v^k_{i_1,\ldots,i_{N-1}} \tag{7.9}$$

where $\sigma_1 = \cdots = \sigma_{p_N} = \sqrt{p_1/(p_1+1)}$, the vectors $u^k \in \mathbb{R}^{p_N}, \forall k \in [p_N]$ are orthonormal and the vectors $v^k \in \mathbb{R}^{p_1 p_2 \cdots p_{N-1}}, \forall k \in [p_N]$ are orthonormal as well. Moreover, we choose v^k as

$$v^k_{i_1,\ldots,i_{N-1}} = \begin{cases} 1 & \text{if } i_1 = \cdots = i_{N-1} = k, \quad k < p_N \\ \frac{1}{\sqrt{p_1}} & \text{if } i_2 = \cdots = i_{N-1} = \operatorname{mod}(i_1, p_1) + 1, \quad k = p_N \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

By construction the matrix $W_{(N)}$ has rank equal to p_N and Frobenius norm equal to $\sqrt{p_1}$. Thus properties (a) and (c) hold true. It remains to show that $\boldsymbol{\mathcal{W}}$ satisfies property (b). To this end, we will show, for every $n \in [N]$ and every $x \in \mathbb{R}^{p_n}$, that

$$\|W_{(n)}^\top x\|_2 \leq \|x\|_2.$$

The case $n = N$ is immediate. If $n = 1$ we have

$$\begin{aligned} \|W_{(1)}^\top x\|_2^2 &= \sum_{i_2,\ldots,i_N} \left(\sum_k \sigma_k \sum_{i_1} u^k_{i_N} v^k_{i_1,\ldots,i_{N-1}} x_{i_1} \right)^2 \\ &= \sum_{i_2,\ldots,i_N} \sum_{k,\ell} \sum_{i_1,j_1} x_{i_1} x_{j_1} \sigma_k \sigma_\ell u^k_{i_N} u^\ell_{i_N} v^k_{i_1,i_2,\ldots,i_{N-1}} v^\ell_{j_1,i_2,\ldots,i_{N-1}} \\ &= \sum_k \sigma_k^2 \sum_{i_1,j_1} x_{i_1} x_{j_1} \left(\sum_{i_2,\ldots,i_{N-1}} v^k_{i_1,i_2,\ldots,i_{N-1}} v^k_{j_1,i_2,\ldots,i_{N-1}} \right) \\ &= \sum_k \sigma_k^2 x_k^2 + \frac{\sigma_{p_N}^2}{p_1} \sum_k x_k^2 = \|x\|_2^2 \end{aligned}$$

where we used $\sum_{i_N} u^k_{i_N} u^\ell_{i_N} = \delta_{k,\ell}$ in the third equality, equation (7.10) and a direct

computation in the fourth equality, and the definition of σ_k in the last equality.
All other cases, that is $n = 2, \dots, N-1$, are conceptually identical, so we only discuss the case $n = 2$. We have

$$\begin{aligned}
\|W_{(2)}^\top x\|_2^2 &= \sum_{i_1,i_3,\dots,i_N} \left(\sum_k \sigma_k \sum_{i_2} u_{i_N}^k v_{i_1,\dots,i_{N-1}}^k x_{i_2} \right)^2 \\
&= \sum_{i_1,i_3,\dots,i_N} \sum_{k,\ell} \sum_{i_2,j_2} x_{i_2} x_{j_2} \sigma_k \sigma_\ell u_{i_N}^k u_{i_N}^\ell v_{i_1,i_2,\dots,i_{N-1}}^k v_{i_1,j_2,\dots,i_{N-1}}^\ell \\
&= \sum_k \sigma_k^2 \sum_{i_2,j_2} \left(x_{i_2} x_{j_2} \sum_{i_1,i_3,\dots,i_{N=1}} v_{i_1,i_2,\dots,i_{N-1}}^k v_{i_1,j_2,\dots,i_{N-1}}^k \right) \\
&= \sum_k \sigma_k^2 x_k^2 + \frac{\sigma_{p_N}^2}{p_1} \sum_k x_k^2 = \|x\|_2^2
\end{aligned}$$

where again we used $\sum_{i_N} u_{i_N}^k u_{i_N}^\ell = \delta_{k,\ell}$ in the third equality, equation (7.10) and a direct computation in the fourth equality, and the definition of σ_k in the last equality.
Finally, if assumption (7.8) is not true we set $w_{i_1,\dots,i_N} = 0$ if $i_n \geq p_1 + 1$, for some $n \leq N-1$ or $i_N > p_1 + 1$. We then proceed as in the case $p_1 = \cdots = p_{N-1}$ and $p_N = p_1 + 1$.
Note that one can build infinitely many tensors following this process, since the left singular vectors can be arbitrarily chosen in equation (7.9). □

We are now ready to formulate the main result of this section.

Proposition 7.3.1. *Let $p_1, \dots, p_N \in \mathbb{N}$, let $\|\cdot\|_{\mathrm{Tr}}$ be the tensor trace norm in equation* (7.3) *and let Ω_α be the function in equation* (7.6) *for $\alpha = \sqrt{p_{\min}}$. If $p_{\min} < p_{\max}$, then there are infinitely many tensors $\boldsymbol{\mathcal{W}} \in \mathcal{S}_{\mathrm{Sp}}$ such that $R(\boldsymbol{\mathcal{W}}) \geq \Omega_\alpha(\boldsymbol{\mathcal{W}}) > \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}}$. Moreover, for every $\boldsymbol{\mathcal{W}} \in \mathcal{S}_{\mathrm{Fr}}$, it holds that $\Omega_1(\boldsymbol{\mathcal{W}}) \geq \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}}$.*

Proof. By construction $\Omega_\alpha(\boldsymbol{\mathcal{W}}) \leq R(\boldsymbol{\mathcal{W}})$ for every $\boldsymbol{\mathcal{W}} \in \alpha\mathcal{S}_{\mathrm{Fr}}$. Since $\mathcal{S}_{\mathrm{Sp}} \subset \alpha\mathcal{S}_{\mathrm{Fr}}$ then Ω_α is a convex lower bound for the tensor rank R on the set $\mathcal{S}_{\mathrm{Sp}}$ as well. The first claim now follows by Lemmas 7.3.2 and 7.3.3. Indeed, all tensors obtained following the process described in the proof of Lemma 7.3.3 have the property that

$$\begin{aligned}
\|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}} &= \frac{1}{N} \sum_{n=1}^N \|\sigma(W_{(n)})\|_1 = \frac{1}{N} \left(p_{\min}(N-1) + \sqrt{p_{\min}^2 + p_{\min}} \right) \\
&< \frac{1}{N} \left(p_{\min}(N-1) + p_{\min} + 1 \right) = \Omega_\alpha(\boldsymbol{\mathcal{W}}) = R(\boldsymbol{\mathcal{W}}).
\end{aligned}$$

Furthermore, as implied by Lemma 7.3.3, there are infinitely many such tensors which satisfy this claim. With respect to the second claim, given that ω_1^{**} is the convex enve-

lope of the cardinality function on the Euclidean unit ball in $\mathbb{R}^d$, then $\omega_1^{**}(\sigma) \geq \|\sigma\|_1$ for every vector σ such that $\|\sigma\|_2 \leq 1$. Consequently,

$$\Omega_1(\boldsymbol{\mathcal{W}}) = \frac{1}{N}\sum_{n=1}^{N} \omega_1^{**}\left(\sigma\left(W_{(n)}\right)\right) \geq \frac{1}{N}\sum_{n=1}^{N} \|\sigma(W_{(n)})\|_1 = \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Tr}}.$$

□

This result can be explained by noticing that the spectral norm is not an invariant property of the matricization of a tensor, whereas the Frobenius (Euclidean) norm is. A visual example illustrating this is shown in Fig. 7.2. Here we consider a 4-mode tensor of dimensions $2 \times 2 \times 2 \times 2$. The singular values of each matricization of that tensor are shown in black. The smallest ℓ_∞ and ℓ_2 balls that contain these vectors are shown in different colours. We notice that the same ℓ_2 ball tightly contains all vectors of singular values, whereas there are several distinct ℓ_∞ balls. This observation leads us to further study the function Ω_α.

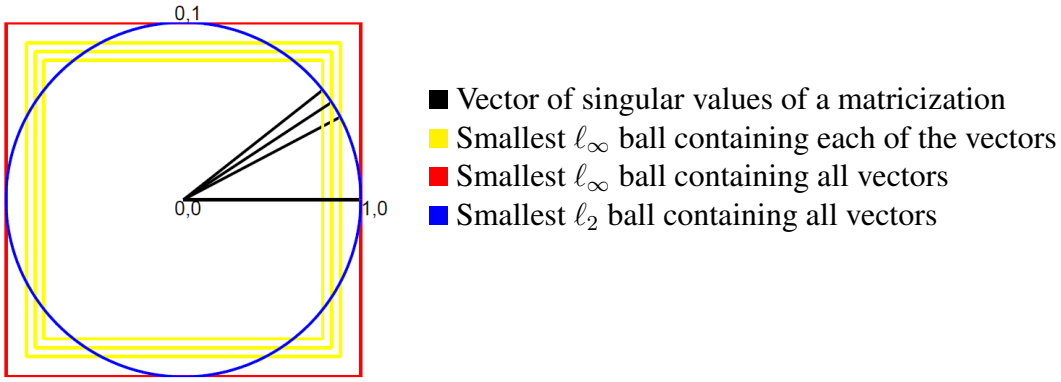

Figure 7.2.: Example, using a $2 \times 2 \times 2 \times 2$ tensor, illustrating that the spectral norm is not an invariant property across matricizations of a tensor, in contrast to the Frobenius norm.

7.4. Optimization method

In this section, we explain how to solve the regularization problem associated with the regularizer (7.6). For this purpose, we first recall the alternating direction method of multipliers (ADMM) [23], which was applied to tensor trace norm regularization in Section 6.3, and in other works such as [64, 185].

7.4.1. Alternating Direction Method of Multipliers (ADMM)

In Section 6.3 we introduced ADMM as an optimization algorithm that allows to decouple the regularization term appearing in problem (7.3). In this section we explain the details of ADMM considering a more general problem comprising both tensor trace norm regularization and the regularizer we propose,

$$\min_{\boldsymbol{\mathcal{W}}} \left\{ F\left(\boldsymbol{\mathcal{W}}\right) + \gamma \sum_{n=1}^{N} \Psi\left(W_{(n)}\right) \right\} \tag{7.11}$$

where $F(\boldsymbol{\mathcal{W}})$ is an error term such as $\|y - \mathcal{I}(\boldsymbol{\mathcal{W}})\|_2^2$ and Ψ is a convex spectral function. It is defined, for every matrix A, as

$$\Psi(A) = \psi(\sigma(A))$$

where ψ is a symmetric convex function invariant under permutations. In particular, if ψ is the ℓ_1 norm then problem (7.11) corresponds to tensor trace norm regularization, whereas if $\psi = \omega_\alpha^{**}$ it implements the proposed regularizer.

As we studied in Chapter 6, problem (7.11) poses some difficulties because the terms under the summation are interdependent, due to the different matricizations of $\boldsymbol{\mathcal{W}}$ having the same elements rearranged in a different way. The way ADMM overcomes this problem is by introducing N auxiliary tensors, $\boldsymbol{\mathcal{B}}_1, \ldots, \boldsymbol{\mathcal{B}}_N \in \mathbb{R}^{p_1 \times \cdots \times p_N}$, so that problem (7.11) can be reformulated as

$$\min_{\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{B}}_1, \ldots, \boldsymbol{\mathcal{B}}_N} \left\{ \frac{1}{\gamma} F\left(\boldsymbol{\mathcal{W}}\right) + \sum_{n=1}^{N} \Psi\left(B_{n(n)}\right) \;:\; \boldsymbol{\mathcal{B}}_n = \boldsymbol{\mathcal{W}},\, n \in [N] \right\} \tag{7.12}$$

The corresponding augmented Lagrangian (see e.g. [23, 25]) is given by

$$\mathcal{L}\left(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}\right) = \frac{1}{\gamma} F\left(\boldsymbol{\mathcal{W}}\right) + \sum_{n=1}^{N} \left(\Psi\left(B_{n(n)}\right) - \langle \boldsymbol{\mathcal{C}}_n, \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n \rangle + \frac{\beta}{2} \left\| \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n \right\|_{\mathrm{Fr}}^2 \right), \tag{7.13}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product between tensors, β is a positive parameter and $\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_N \in \mathbb{R}^{p_1 \times \cdots \times p_N}$ are the set of Lagrange multipliers associated with the constraints in problem (7.12).

ADMM is based on the following iterative scheme

$$\boldsymbol{\mathcal{W}}^{[i+1]} \leftarrow \underset{\boldsymbol{\mathcal{W}}}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{B}}^{[i]}, \boldsymbol{\mathcal{C}}^{[i]}\right) \tag{7.14}$$

$$\boldsymbol{\mathcal{B}}_n^{[i+1]} \leftarrow \underset{\boldsymbol{\mathcal{B}}_n}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{\mathcal{W}}^{[i+1]}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}^{[i]}\right) \tag{7.15}$$

$$\boldsymbol{\mathcal{C}}_n^{[i+1]} \leftarrow \boldsymbol{\mathcal{C}}_n^{[i]} - \left(\beta \boldsymbol{\mathcal{W}}^{[i+1]} - \boldsymbol{\mathcal{B}}_n^{[i+1]}\right). \tag{7.16}$$

Step (7.16) is straightforward, whereas step (7.14) is described in Section D.1 (also see [64]). Here we focus on the step (7.15) since this is the only problem which involves function Ψ. We restate it with more explanatory notations as

$$\underset{B_{n(n)}}{\operatorname{argmin}} \left\{ \Psi\left(B_{n(n)}\right) - \left\langle C_{n(n)}, W_{(n)} - B_{n(n)} \right\rangle + \frac{\beta}{2} \left\| W_{(n)} - B_{n(n)} \right\|_{\mathrm{Fr}}^2 \right\}.$$

By completing the square in the right hand side, the solution of this problem is given by

$$\hat{B}_{n(n)} = \operatorname{prox}_{\frac{1}{\beta}\Psi}(Z) := \underset{B_{n(n)}}{\operatorname{argmin}} \left\{ \frac{1}{\beta} \Psi\left(B_{n(n)}\right) + \frac{1}{2} \left\| B_{n(n)} - Z \right\|_{\mathrm{Fr}}^2 \right\},$$

where $Z = W_{(n)} - \frac{1}{\beta} C_{n(n)}$. By using properties of proximity operators (see e.g. [12, Prop. 3.1]) we know that

$$\operatorname{prox}_{\frac{1}{\beta}\Psi}(Z) = U_Z \operatorname{diag}\left(\operatorname{prox}_{\frac{1}{\beta}\psi}(\sigma(\mathrm{Z}))\right) V_Z^{\top},$$

where U_Z and V_Z are the orthogonal matrices formed by the left and right singular vectors of Z, respectively. If we choose $\psi = \|\cdot\|_1$ the associated proximity operator is the well-known soft thresholding operator, that is, $\operatorname{prox}_{\frac{1}{\beta}\|\cdot\|_1}(\sigma) = v$, where the vector v has components

$$v_i = \operatorname{sign}(\sigma_i) \max\left(|\sigma_i| - \frac{1}{\beta}, 0\right).$$

On the other hand, if we choose $\psi = \omega_\alpha^{**}$, we need to compute $\operatorname{prox}_{\frac{1}{\beta}\omega_\alpha^{**}}$. In the next section, we describe a method to accomplish this task.

7.4.2. Computation of the proximity operator

To compute the proximity operator of the function $\frac{1}{\beta}\omega_\alpha^{**}$ we use several properties of calculus of proximity operators. First, we use the formula (see e.g. [44]) $\operatorname{prox}_{g^*}(x) = x - \operatorname{prox}_g(x)$ for $g^* = \frac{1}{\beta}\omega_\alpha^{**}$. Next we use a property of conjugate functions from [181,

74], which states that $g(\cdot) = \frac{1}{\beta}\omega_\alpha^*(\beta\cdot)$. Finally, by the scaling property of proximity operators [44], we have that $\mathrm{prox}_g(x) = \frac{1}{\beta}\mathrm{prox}_{\beta\omega_\alpha^*}(\beta x)$.

It remains to compute the proximity operator of a multiple of the function ω_α^* in equation (7.7), that is, for any $\beta > 0$, $y \in \mathcal{S}$, we wish to compute

$$\mathrm{prox}_{\beta\omega_\alpha^*}(y) = \operatorname*{argmin}_w \{h(w) : w \in \mathcal{S}\}$$

where we define $\mathcal{S} := \{w \in \mathbb{R}^d : w_1 \geq \cdots \geq w_d \geq 0\}$ and

$$h(w) = \frac{1}{2}\|w - y\|_2^2 + \beta \max_{r=0}^{d} \{\alpha \|w_{1:r}\|_2 - r\}.$$

In order to solve this problem we employ the projected subgradient method, see e.g. [26]. It consists in applying two steps at each iteration. First, it advances along a negative subgradient of the current solution; second, it projects the resultant point onto the feasible set $\mathcal{S}$. In fact, according to [26], it is sufficient to compute an approximate projection, a step which we describe in Section D.2. To compute a subgradient of h at w, we first find any integer k such that $k \in \operatorname*{argmax}_{r=0}^{d} \{\alpha \|w_{1:r}\|_2 - r\}$. Then, we calculate a subgradient g of the function h at w by the formula

$$g_i = \begin{cases} \left(1 + \frac{\alpha\beta}{\|w_{1:k}\|_2}\right) w_i - y_i, & \text{if } i \leq k, \\ w_i - y_i, & \text{otherwise.} \end{cases}$$

Now we have all the ingredients to apply the projected subgradient method, which is summarized in Algorithm 7.1. In our implementation we stop the algorithm when an update of $\hat{w}$ is not made for more than 10^2 iterations.

7.5. Experiments

We conducted a set of experiments to assess whether there is any advantage in using the proposed regularizer over the tensor trace norm for tensor completion. First, we designed a synthetic experiment to evaluate the performance of both approaches under controlled conditions. Then, we tried both methods on two tensor completion real data problems. Finally, we repeated one experiment from Chapter 6 with the aim of testing whether the proposed method leads to improvements in the MLMTL framework.

In all cases, we have used a validation procedure to tune the hyper-parameter γ, present in both approaches, among the values $\{10^j : j = -7, -6, \ldots, 1\}$. In our proposed

Algorithm 7.1 Computation of $\mathrm{prox}_{\beta\omega_\alpha^*}(y)$

Input: $y \in \mathbb{R}^d$, $\alpha, \beta > 0$.
Output: $\hat{w} \in \mathbb{R}^d$.
Initialization: initial step $\tau_0 = \frac{1}{2}$, initial and best found solution $w^0 = \hat{w} = P_{\mathcal{S}}(y) \in \mathbb{R}^d$.
for $t = 1, 2, \ldots$ **do**
  $\tau \leftarrow \frac{\tau_0}{\sqrt{t}}$
  Find k such that $k \in \mathrm{argmax}\left\{\alpha\|w_{1:r}^{t-1}\|_2 - r : 0 \le r \le d\right\}$
  $\tilde{w}_{1:k} \leftarrow w_{1:k}^{t-1} - \tau\left(w_{1:k}^{t-1}\left(1 + \frac{\alpha\beta}{\|w_{1:k}^{t-1}\|_2}\right) - y_{1:k}\right)$
  $\tilde{w}_{k+1:d} \leftarrow w_{k+1:d}^{t-1} - \tau\left(w_{k+1:d}^{t-1} - y_{k+1:d}\right)$
  $w^t \leftarrow \tilde{P}_{\mathcal{S}}\left(\tilde{w}\right)$
  If $h(w^t) < h(\hat{w})$ then $\hat{w} \leftarrow w^t$
  If "Stopping Condition = True" then terminate.
end for

approach there is one further hyper-parameter, α, to be specified. According to Lemma 7.3.2, it should take a value as close as possible to the Euclidean norm of the underlying tensor. Since this is unknown, we propose to use the estimate

$$\hat{\alpha} = \sqrt{\|w\|_2^2 + (\mathrm{mean}(w)^2 + \mathrm{var}(w))\left(\prod_{i=1}^{N} p_i - m\right)},$$

where m is the number of known entries (training set size) and $w \in \mathbb{R}^m$ contains their values. This estimator assumes that each value in the tensor is sampled from $\mathcal{N}(\mathrm{mean}(w), \mathrm{var}(w))$, where $\mathrm{mean}(w)$ and $\mathrm{var}(w)$ are the average and the variance of the elements in w.

7.5.1. Synthetic data

We have generated a 3 mode tensor $\boldsymbol{\mathcal{W}}^0 \in \mathbb{R}^{40\times 20\times 10}$ by the following procedure. First we built a tensor $\boldsymbol{\mathcal{W}}$ with ranks $(12, 6, 3)$ using the Tucker decomposition (see e.g. [100])

$$w_{i_1,i_2,i_3} = \sum_{j_1=1}^{12}\sum_{j_2=1}^{6}\sum_{j_3=1}^{3} g_{j_1,j_2,j_3} a_{i_1,j_1}^{(1)} a_{i_2,j_2}^{(2)} a_{i_3,j_3}^{(3)}, \quad (i_1, i_2, i_3) \in [40] \times [20] \times [10]$$

where each entry of the Tucker decomposition components is sampled from the standard Gaussian distribution $\mathcal{N}(0, 1)$. We then created the ground truth tensor $\boldsymbol{\mathcal{W}}^0$ by using

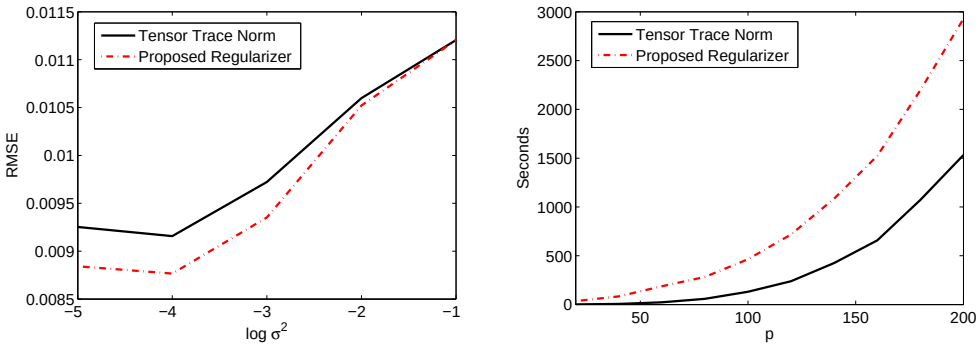

Figure 7.3.: Synthetic dataset: Root Mean Squared Error (RMSE) of tensor trace norm and the proposed regularizer (left). Running time execution for different sizes of the tensor (right).

the equation

$$w^0_{i_1,i_2,i_3} = \frac{w_{i_1,i_2,i_3} - \text{mean}(\boldsymbol{\mathcal{W}})}{\sqrt{M}\text{std}(\boldsymbol{\mathcal{W}})} + \xi_{i_1,i_2,i_3}$$

where $\text{mean}(\boldsymbol{\mathcal{W}})$ and $\text{std}(\boldsymbol{\mathcal{W}})$ are the mean and standard deviation of the elements of $\boldsymbol{\mathcal{W}}$, M is the total number of elements of $\boldsymbol{\mathcal{W}}$, and the ξ_{i_1,i_2,i_3} are i.i.d. Gaussian random variables with zero mean and variance σ^2. We have randomly sampled 10% of the elements of the tensor to compose the training set, 45% for the validation set, and the remaining 45% for the test set. After repeating this process 20 times, we report the average results in Fig. 7.3 (left). Having conducted a paired t-test for each value of σ^2, we conclude that the visible differences in the performances are highly significant, obtaining p-values less than 0.01 for $\sigma^2 \leq 10^{-2}$ in all cases.

In addition, we conducted an experiment to test the running time of both approaches. We generated tensors $\boldsymbol{\mathcal{W}}^0 \in \mathbb{R}^{p\times p\times p}$ for different values of $p \in \{20\ ,\ 40,\ \ldots, 200\}$, following the same procedure outlined above. The results are reported in Fig. 7.3 (right). For small values of p, the ratio between the running times of our method and the trace norm regularization method is quite high. For example in the lowest value tried for p in this experiment, $p = 20$, this ratio is 22.66. However, as the volume of the tensor increases, the ratio quickly decreases. For example, for $p = 200$, the running time ratio is 1.91. These outcomes are expected because when p is low, the most demanding routine in our method is the one described in Algorithm 7.1, where each iteration is of order $O(p)$ and $O(p^2)$ in the best and worst case, respectively. However, as p increases the singular value decomposition routine, which is common to both methods, becomes the most demanding because it has a time complexity $O(p^3)$ [65]. Therefore, we can conclude that even though our approach is slower than the trace norm based method,

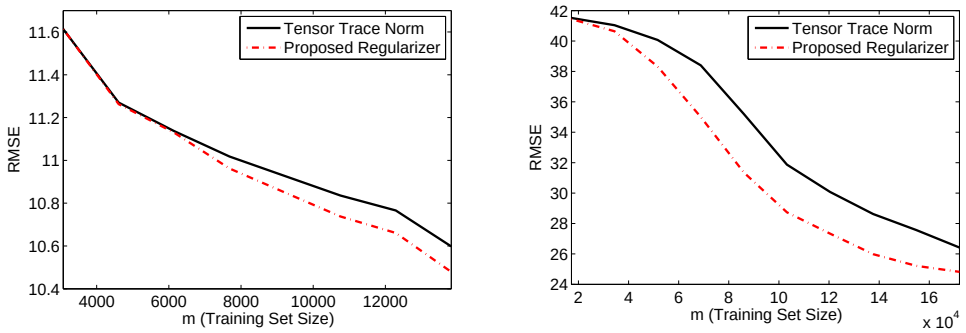

Figure 7.4.: Root Mean Squared Error (RMSE) of tensor trace norm and the proposed regularizer for ILEA dataset (left) and Ocean video (right).

this difference of time becomes much smaller as the size of the tensor increases.

7.5.2. School dataset

In the first real experiment we employ tensor completion to perform regression when the attributes of the instances are given by categorical variables. To do so, we have used the Inner London Education Authority (ILEA) dataset. It is composed of examination marks ranging from 0 to 70, of 15362 students who are described by a set of attributes such as school and ethnic group. Most of these attributes are categorical, thereby we can think of exam mark prediction as a tensor completion problem where each of the modes corresponds to a categorical attribute. In particular, we have used the following attributes: school (139), gender (2), VR-band (3), ethnicity (11), and year (3), leading to a 5-order tensor $\mathcal{W} \in \mathbb{R}^{139\times 2\times 3\times 11\times 3}$.

We randomly selected 5% of the instances to make the test set and another 5% of the instances for the validation set. From the remaining instances, we randomly chose m of them for several values of m accounting for $20\%, 30\%, \dots 90\%$ of the total pool of instances. This procedure was repeated 20 times and the average performance is presented in Fig. 7.4 (left).

The results show that there is a distinguishable improvement of our approach with respect to tensor trace norm regularization for values of $m > 7000$. To check whether this gap is significant, we conducted a set of paired t-tests in this regime. In all these cases we obtained a p-value below 0.01.

7.5.3. Video completion

In the second real-data experiment we have performed a video completion test. Any video can be treated as a 4-order tensor: "width" $\times$ "height" $\times$ "RGB" $\times$ "video length", so we can use tensor completion algorithms to rebuild a video from a small number of inputs, a procedure that can be useful for compression purposes. In our case, we have used the Ocean video, available at [117]. This video sequence can be treated as a tensor $\mathcal{W} \in \mathbb{R}^{160\times112\times3\times32}$. We have randomly sampled m tensors elements comprising $1\%, 2\%, \ldots 10\%$ of the whole tensor as training data, 5% of the tensor as validation data, and the remaining elements composed the test set. After repeating this procedure 10 times, we present the average results in Fig. 7.4 (right).

The proposed approach is noticeably better than the tensor trace norm in this experiment. This apparent outcome is strongly supported by the paired t-tests which we computed for each value of m, obtaining always p-values below 0.01, and for the cases $m > 5 \times 10^4$, we obtained p-values below 10^{-6}.

7.5.4. MLMTL experiment

Let us recall that the framework developed in Chapter 6 can be cast as a tensor completion problem. Therefore, we can apply the proposed convex approach in that framework. In this final experimental section, we test the performance of the proposed approach in this context. To do so, we have repeated the Restaurant & Consumer experiment, conducted in Section 6.5.2.1. In that, the aim is to predict $p_2 = 3$ kinds of ratings given by $p_3 = 138$ critics to different restaurants, each of them described by $p_1 = 44$ attributes.

The results of those experiments, using exactly the same settings, are shown in Fig. 7.5. For clarity, we only show the performances of the MLMTL methods, as the superiority of these over other approaches was already shown in Chapter 6.

Analyzing the results we can see that the proposed method consistently improves over the other convex approach in all training sizes considered. In comparison with the non-convex approach, we see that the proposed method is clearly better for small training sets, and it behaves comparably when the training set increases.

7.6. Discussion

In this chapter, we proposed a convex relaxation for the average of the rank of the matricizations of a tensor. We compared this relaxation to a commonly used convex relax-

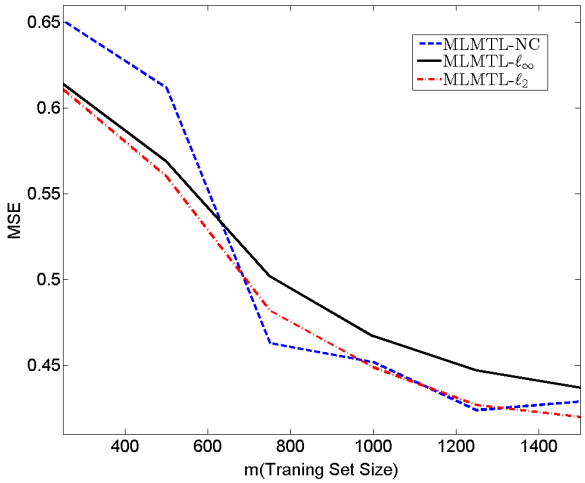

Figure 7.5.: Restaurant & Consumer Dataset: Mean Square Error (MSE) comparison between the three MLMTL methods described in Chapter 6 and 7: non-convex approach, convex approach based on trace norm, and approach based on the convex relaxation developed in Chapter 7.

ation in the context of tensor completion, which is based on the trace norm. We proved that this second relaxation is not tight and argued that the proposed convex regularizer may be advantageous.

One challenge of our approach is that the objective function is not in closed form. We addressed this problem by applying projected subgradient methods to calculate its proximal operator. Our numerical experiments indicate that our method consistently improves in terms of estimation error over tensor trace norm regularization, while being computationally comparable on the range of problems we considered.

After the publication of the content of this chapter [174], another related work appeared [141], focused on the analysis of convex methods for tensor completion. Despite the analysis of the authors is based on recovery bounds instead of convex envelopes, they lead to similar conclusions regarding the suboptimality of the average of the trace norm for tensors.

Finally, in the same line as in the conclusion of the previous chapter, it should be noted that scalability is a problem for this kind of approaches. In the future it would be interesting to study methods to speed up the computation of the proximity operator of our regularizer, and more generally, parallel implementations of ADMM that allow for faster computations.

8. Conclusion

In this thesis we explored ways of controlling negative transfer in multitask and transfer learning methods, focusing in particular on multi-aspect data scenarios. In the first section we summarize the contributions we made, together with their strengths, implications and potential limitations. In the second section we describe several research directions that stem out from this thesis.

8.1. Contributions

Our contributions can be summarized in the following four points.

Sparse Coding Multitask Learning Firstly, we tackled the negative transfer problem in the most general case, where there is no side information about the tasks, which in turn may have different degrees of relatedness. Thus, the challenge was to set mild assumptions on the learning model in order to avoid negative transfer, but strong enough that allow positive transfer. In the literature there were two main strategies proposed to address this: dirty models and task grouping, reviewed in Section 2.4.2 and Section 2.4.3 respectively. However they were limited to specific situations, for example dirty models cannot account for learning commonalities between several groups of tasks, whereas task grouping models assume that groups of tasks are completely independent. We proposed a model that allows subsets of tasks to be completely or partially unrelated by assuming that each of them use a sparse combination of a set of common features learned from the data. Note that the induced sparsity is the key point to account for mildly related tasks. While studying this method, another similar approach appeared in [107], where the authors empirically show that this approach outperforms the previous strategies. We further studied this approach considering both multitask and transfer learning settings. We proposed a different set of constraints to model the described assumptions, being those choices justified by our learning bounds. These bounds show, among other properties, that our approach is robust against negative transfer, being able

to learn disjoint features for unrelated groups. These theoretical bounds are supported by the experiments carried out. The synthetic experiments were done such that the parameters of the process generating the artificial data were controlled (number of atoms of the dictionary, and number of tasks among others). The results obtained strongly support the learning bounds presented. Experimental results on real data, one in the context of sparse coding and another in transfer learning, show improvement over the methods presented in [8] and [107] when the predictor functions can be represented sparsely.

Decoupling of Features Secondly, we studied the case where each task is known to belong to an aspect so that tasks in different aspects or groups tend to use different features of the data. We presented and analyzed a method which imposes orthogonality between the features belonging to tasks in different groups. Orthogonal constraints have been recently studied in hierarchical classification, with the aim of encouraging mutually exclusive features between parent and child nodes [82, 225]. We studied in parallel the use of this regularizer in our scenario, where previous MTL approaches (see Chapter 2) may fail because of their inability to leverage prescribed groups of unrelated tasks. The justification of that assumption is based on how human beings process different aspects of the same data. The experiments results show a clear improvement over general multitask learning approaches. This is particularly significant when the training set size is small compared to the number of attributes describing the data. A limitation of the proposed approach is that the associated optimization problem is non-convex. This led us to develop a convex modification of our approach. The experiments made with the convex approach are not conclusive. They show it outperforms the original and other baseline methods on the synthetic and one real dataset, but its performance is poor on another real dataset. A possible explanation is that the convexification used in the method depends on the input data, and in some cases the input data may lead to a big perturbation of the original problem. This hypothesis is supported by the results obtained using a relaxed version of the previous approach, which achieved the best results in both real datasets.

Multilinear Multitask Learning Thirdly, we considered the multi-aspect scenario in which there is available information about how tasks are related, and this information takes the form of linking each task with a combination of elements of aspects. In Chapter 2 we reviewed some MTL approaches that considered side information regarding relations between tasks. However, they are not appropriate to deal with this scenario, as these MTL approaches considered different relational structures between tasks. In

order to reflect the desired structure of relations between tasks, we employed a combination of concepts coming from both multitask learning and multilinear models literature, particularly those reviewed in Section 3.3. The resulting framework, which we call Multilinear Multitask Learning (MLMTL), allows us to perform zero-shot transfer learning, that is, the capacity of learning a task even when there are no training instances available for it. This can be made by leveraging the knowledge obtained from other related tasks across different modes of the model. This opens a new way to tackle scenarios where specific domains in the data are either missing or more difficult to gather, such as in naturalistic behaviour recognition. Two strategies are developed within this framework. The first one is an adaptation from a novel convex multilinear approach [64, 117, 184, 200] to this problem. The second one, suggested in this thesis to alleviate the memory requirements of the previous strategy, is based on alternate minimization of the Tucker components of a tensor. Results obtained from experiments run on synthetic and real datasets in different domains show that both MLMTL strategies perform better than general multitask learning approaches.

A New Convex Relaxation for Tensor Completion Finally, the previous scenario led us to examine convex approximations for tensor completion. The most extended convex approach for tensor learning has been the one based on regularizing a generalization of the trace norm for tensors [64, 117, 184, 200]. The authors of [183], proved that this regularizer is indeed a convex lower bound of the average of the Tucker ranks, but left as an open question whether this convex function was tight or not. We investigated this question and we found that this approach is not the tightest among the convex functions to the average of the Tucker ranks. We also proposed an alternative convex function that is tighter. One challenge that arose with this proposed function was that it is not in closed form. We approached this problem by applying projected subgradient methods to calculate its proximal operator. We justified the appropriateness of our approach with respect to the generalization of the trace norm both theoretically and empirically with synthetic and real data experiments. Recently, similar conclusions regarding the suboptimality of the average of the trace norm for tensors were reached in [141], where the authors use a different strategy based on recovery bounds. A key general message of these results is that no matter how intuitive or widely used an approach is, it must be analyzed and justified. Furthermore, this process may lead to the discovery of better approaches.

The novelties introduced in this thesis can be of interest both to machine learning theorists and to practitioners. Theorists can extend the proposed frameworks to consider

even more general settings. We find a recent example of that in [185], where the authors extend multilinear multitask learning (in Chapter 6) by using reproducing kernel spaces to consider non Euclidean features such as graphs or probability distributions. Practitioners can make use of the frameworks developed here to treat multi-aspect datasets, particularly those related to personalization. One example which has not been covered in this thesis but has potential interest is personalized item search, in which users (aspect 1) want to retrieve items (aspect 2) by introducing queries (input). The retrieved items are those whose related tasks output the highest values.

8.2. Future research directions

The advances made in this thesis lead to potential new lines of research. Here we highlight those related to exploiting properties of multi-aspect data. As we have shown, multilinear models are a natural framework for modeling interrelations between several aspects. Although there have been some advances on the application of multilinear models to machine learning (see Chapter 3), several questions remain unanswered. These include those related to complexity issues, where algorithmic challenges arise if they are to be applied to real problems involving large amounts of data.

8.2.1. Study of different constraints on tensors and their implications

As we have reviewed in Section 3.2, there exist several notions of rank for a tensor. In Chapters 6 and 7 we opted to use the Tucker rank and we provided some reasoning about this choice in Section 6.2. However, using other notions of rank, such as the CP, may also be a valid option, particularly in Chapter 7. Therefore, it would be desirable to provide understanding and theoretical justifications of when a notion is preferable to the other.

One difficulty that arises when dealing with optimization on tensor rank functions is its non-convexity. This has the consequence that the obtained solution to the problem is not optimal and is dependent on the initialization of the solver. There are two paths to be explored in this regard. One is the study of convex relaxation methods that consider tractable problems that are as similar as possible to the original one. Chapter 7 advanced the state of the art in this regard, but there is arguably a research gap to be filled with respect to this topic, both in terms of developing convex relaxations for other notions

of ranks and regularizers, and analyzing and comparing their performances with respect to suboptimal optimizers over the original problems. A second path is to study guarantees over the solutions of the non-convex problem. This approach is appealing for two reasons. First, this approach would not alter the problem of interest, and second, non-convex approaches based on learning explicitly hidden latent variables usually lead to memory efficient algorithms.

Another aspect of interest is the introduction of prior knowledge into the model. In optimization problems this is done by means of regularizers and constraints imposing properties such as sparsity [88, 126, 127, 153, 190, 191], non-negativity [54, 79, 112], and orthogonality [82, 168, 225], on the learning variables. We carried out some analysis in this sense on matrices and multitask learning in Chapter 4, where we studied sparsity constraints in one modality. In Bayesian frameworks this is done by specifying explicitly the prior distributions on the learning variables [49, 115, 142, 197]. In this context it is interesting to study variational approaches, [36, 197, 178], which allow performing inference efficiently.

8.2.2. Developing multilinear optimization methods for large scale data

In many real applications, data are available in huge amounts. This scenario, which is often described as "Big Data", leads to myriad possibilities, but also entails further constraints and difficulties which require new methodologies. Among those constraints, there are two of special relevance: firstly, the impossibility of keeping all the data in memory; secondly, the need to parallelize the machine learning algorithms in order to obtain a model in a reasonable amount of time. The way to overcome both issues involves the development of distributed optimization methods. Due to the high complexity that tensors pose, the implementation of solvers that can operate in a distribute fashion becomes a challenging problem.

In this thesis, as well as in the vast majority of papers which use multilinear algebra in machine learning models [64, 117, 140, 186, 184, 200, 209, 207, 210], experiments are carried out on small datasets that can easily fit in memory. One reason for it is that, due to the complexity of tensors, even fast sequential optimization methods are expensive. In order to overcome this issue, some algorithmic frameworks such as parallel stochastic gradient descent [228], and parallel coordinate descent methods [27] could be explored. These approaches parallelize over samples and over coordinates respectively, and have

been proved to be highly scalable optimizers for some specific problems. Another approach of interest is that of alternating direction method of multipliers (ADMM) [23], covered in Chapters 6 and 7, which can be adapted for distributed convex optimization by means of coordinating the solutions of small local sub-problems. The analysis and use of optimization approaches for our problems must be done jointly with the study of frameworks for parallel computation that allows for their implementation. That may include both the use of graphic processing units for general purpose computing, and distributed computing environments.

A. Multitask Learning Literature Review: Appendix

A.1. MAP derivation of Inverse-Wishart prior on the covariance of the task weight vectors

The likelihood of the model in eq. (2.22) is:

$$\prod_{t=1}^{T} p\left(Y_t|X_t, w_t, D, \Psi, \nu, \tau\right) \propto p\left(D|\Psi, \nu\right) \prod_{t=1}^{T} p\left(w_t|0, D\right) \prod_{t=1}^{T} p\left(Y_t|X_t, w_t, \tau\right)$$

$$\propto |D|^{-\frac{\nu+d+1}{2}} \exp\left(-\tfrac{1}{2}\mathrm{trace}\left(\Psi D^{-1}\right)\right)$$

$$\prod_{t=1}^{T} |D|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2} w_t^T D^{-1} w_t\right) \prod_{t=1}^{T} \prod_{i=1}^{m_t} \exp\left(-\tfrac{\tau}{2}\left(y_{ti} - x_{ti}^T w_t\right)^2\right).$$

The negative log-likelihood of that is:

$$-\log p\left(Y|X_t, w_t, D, \Psi, \nu, \tau\right) = -\tfrac{\nu+d+T+1}{2}\log\left(\det\left(D^{-1}\right)\right)$$

$$+\frac{1}{2}\mathrm{trace}\left(\left(\Psi + WW^T\right) D^{-1}\right) + \frac{\tau}{2}\sum_{t=1}^{T}\left\|X_t^T w_t - Y_t\right\|_2^2 + C. \qquad \text{(A.1)}$$

Dividing it over $\frac{\tau}{2}$ and minimizing over both D and W leads to the problem in eq. (2.21).

B. Sparse Coding Multitask Learning: Appendix

In this appendix, we present the proof of Theorem 4.3.1 and Theorem 4.3.2. We begin by introducing some more notation and auxiliary results. Andreas Maurer helped develop the content of this appendix.

B.1. Notation and tools

Issues of measurability will be ignored throughout, in particular, if $\mathcal{F}$ is a class of real valued functions on a domain $\mathcal{X}$ and X a random variable with values in $\mathcal{X}$ then we will always write $\mathbb{E}\sup_{f\in\mathcal{F}} f(X)$ to mean $\sup\{\mathbb{E}\max_{f\in\mathcal{F}_0} f(X) : \mathcal{F}_0 \subseteq \mathcal{F}, \mathcal{F}_0 \text{ finite}\}$.

In the sequel H denotes a finite or infinite dimensional Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\|$. If T is a bounded linear operator on H its operator norm is written $\|T\|_{\mathrm{Sp}} = \sup\{\|Tx\| : \|x\| = 1\}$.

Members of H are denoted with lower case italics such as x, v, w. Let B be the unit ball in H. An *example* is a pair $z = (x, y) \in B \times \mathbb{R} =: \mathcal{Z}$, a sample is a vector of such pairs $\mathbf{z} = (z_1, \ldots, z_m) = ((x_1, y_1), \ldots, (x_m, y_m))$. Here we also write $\mathbf{z} = (X, y)$, with $X = (x_1, \ldots, x_m) \in H^m$ and $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$.

A multisample is a vector $\mathbf{Z} = \left(\mathbf{z}^1, \ldots, \mathbf{z}^T\right)$ composed of samples. We also write $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$ with $\mathbf{X} = \left(X^1, \ldots, X^T\right)$.

Depending on context the inner product and euclidean norm on $\mathbb{R}^K$ will also be denoted with $\langle\cdot,\cdot\rangle$ and $\|.\|$. The ℓ_1-norm $\|\cdot\|_1$ on $\mathbb{R}^K$ is defined by $\|c\|_1 = \sum_{k=1}^K |c_k|$.

In the sequel we denote with $\mathcal{C}_\alpha$ the set $\left\{c \in \mathbb{R}^K : \|c\|_1 \leq \alpha\right\}$, and abbreviate $\mathcal{C}$ for the ℓ_1-unit ball $\mathcal{C}_1$. The canonical basis of $\mathbb{R}^K$ is denoted $e_1, \ldots, e_K$. Unless otherwise specified the summation over the index i will always run from 1 to m, t will run from 1 to T, and k will run from 1 to K.

B.1.1. Covariances

For $X \in H^m$ the empirical covariance operator $\hat{\Sigma}(X)$ is specified by

$$\left\langle \hat{\Sigma}(X) v, w \right\rangle = \frac{1}{m} \sum_i \langle v, x_i \rangle \langle x_i, w \rangle , \ v, w \in H.$$

The definition implies the inequality

$$\sum_i \langle v, x_i \rangle^2 = m \left\langle \hat{\Sigma}(X) v, v \right\rangle \leq m \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}} \|v\|^2 . \tag{B.1}$$

It also follows that $\mathrm{tr}\left(\hat{\Sigma}(X)\right) = (1/m) \sum_i \|x_i\|^2$.

For a multisample $\mathbf{X} \in H^{mT}$ we will consider two quantities defined in terms of the empirical covariances.

$$\begin{aligned} S_{\mathrm{Tr}}(\mathbf{X}) &= \frac{1}{T} \sum_t \left\| \hat{\Sigma}\left(X^t\right) \right\|_{\mathrm{Tr}} := \frac{1}{T} \sum_t \mathrm{tr}\left(\hat{\Sigma}\left(X^t\right)\right) \\ S_{\mathrm{Sp}}(\mathbf{X}) &= \frac{1}{T} \sum_t \left\| \hat{\Sigma}\left(X^t\right) \right\|_{\mathrm{Sp}} := \frac{1}{T} \sum_t \lambda_{\max}\left(\hat{\Sigma}\left(X^t\right)\right) \end{aligned}$$

where $\lambda_{\max}$ is the largest eigenvalue. If all data points x_{ti} lie in the unit ball of H then $S_{\mathrm{Tr}}(\mathbf{X}) \leq 1$. Of course $S_{\mathrm{Tr}}(\mathbf{X})$ can also be written as the trace of the total covariance $(1/T) \sum_t \hat{\Sigma}(X_t)$, while $S_{\mathrm{Sp}}(\mathbf{X})$ will always be at least as large as the largest eigenvalue of the total covariance. We always have $S_{\mathrm{Sp}}(\mathbf{X}) \leq S_{\mathrm{Tr}}(\mathbf{X})$, with equality only if the data is one-dimensional for all tasks. The quotient $S_{\mathrm{Tr}}(\mathbf{X}) / S_{\mathrm{Sp}}(\mathbf{X})$ can be regarded as a crude measure of the effective dimensionality of the data. If the data have a high dimensional distribution for each task then $S_{\mathrm{Sp}}(\mathbf{X})$ can be considerably smaller than $S_{\mathrm{Tr}}(\mathbf{X})$.

B.1.2. Concentration inequalities

Let $\mathcal{H}$ be any space. For $\mathbf{h} \in \mathcal{H}^n$, $1 \leq k \leq n$ and $j \in \mathcal{H}$ we use $\mathbf{h}_{k \leftarrow j}$ to denote the object obtained from $\mathbf{h}$ by replacing the k-th coordinate of $\mathbf{h}$ with j. That is

$$\mathbf{h}_{k \leftarrow j} = (h_1, \ldots, h_{k-1}, j, h_{k+1}, \ldots, h_n) .$$

The concentration inequality in part (i) of the following theorem, known as the bounded difference inequality is given in [134]. A proof of inequality (ii) is given in [127].

Theorem B.1.1. *Let $F : \mathcal{H}^n \to \mathbb{R}$ and define A and B by*

$$\begin{aligned} A^2 &= \sup_{\mathbf{h}\in\mathcal{H}^n} \sum_{k=1}^{n} \sup_{j_1,j_2\in\mathcal{H}} \left(F\left(\mathbf{h}_{k\leftarrow j_1}\right) - F\left(\mathbf{h}_{k\leftarrow j_2}\right)\right)^2 \\ B^2 &= \sup_{\mathbf{h}\in\mathcal{H}^n} \sum_{k=1}^{n} \left(F\left(\mathbf{h}\right) - \inf_{j\in\mathcal{H}} F\left(\mathbf{h}_{k\leftarrow j}\right)\right)^2. \end{aligned}$$

Let $\mathbf{H} = (H_1, \ldots, H_n)$ be a vector of independent random variables with values in $\mathcal{H}$, and let $\mathbf{H}'$ be i.i.d. to $\mathbf{H}$. Then for any $s > 0$
(i) $\Pr\left\{F\left(\mathbf{H}\right) > \mathbb{E}F\left(\mathbf{H}'\right) + s\right\} \leq e^{-2s^2/A^2}$;
(ii) $\Pr\left\{F\left(\mathbf{H}\right) > \mathbb{E}F\left(\mathbf{H}'\right) + s\right\} \leq e^{-s^2/\left(2B^2\right)}$.

B.1.3. Rademacher and Gaussian averages

We will use the term *Rademacher variables* for any set of independent random variables, uniformly distributed on $\{-1, 1\}$, and reserve the symbol σ for Rademacher variables. A set of random variables is called *orthogaussian* if the members are independent $\mathcal{N}(0, 1)$-distributed (standard normal) variables and reserve the letter ζ for standard normal variables. Thus $\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_{11}, \ldots, \sigma_{ij}$ etc. will always be independent Rademacher variables and $\zeta_1, \zeta_2, \ldots, \zeta_i, \ldots, \zeta_{11}, \ldots, \zeta_{ij}$ will always be orthogaussian. For $A \subseteq \mathbb{R}^n$ we define the Rademacher and Gaussian averages of A [111, 18] as

$$\begin{aligned} \mathcal{R}\left(A\right) &= \mathbb{E}_\sigma \sup_{(x_1,\ldots,x_n)\in A} \frac{2}{n} \sum_{i=1}^{n} \sigma_i x_i, \\ \mathcal{G}\left(A\right) &= \mathbb{E}_\zeta \sup_{(x_1,\ldots,x_n)\in A} \frac{2}{n} \sum_{i=1}^{n} \zeta_i x_i. \end{aligned}$$

If $\mathcal{F}$ is a class of real valued functions on a space $\mathcal{X}$ and $X = (x_1, \ldots, x_n) \in \mathcal{X}^n$ we write

$$\mathcal{F}\left(X\right) = \mathcal{F}\left(x_1, \ldots, x_n\right) = \left\{\left(f\left(x_1\right), \ldots, f\left(x_n\right)\right) : f \in \mathcal{F}\right\} \subseteq \mathbb{R}^n.$$

The empirical Rademacher and Gaussian complexities of $\mathcal{F}$ on X are respectively $\mathcal{R}\left(\mathcal{F}\left(X\right)\right)$ and $\mathcal{G}\left(\mathcal{F}\left(X\right)\right)$.

The utility of these concepts for learning theory comes from the following key-result (see [18, 101]), stated here in two portions for convenience in the sequel.

Theorem B.1.2. *Let $\mathcal{F}$ be a real-valued function class on a space $\mathcal{X}$ and let $\mu_1, \ldots, \mu_m$*

be probability measures on $\mathcal{X}$ with product measure $\boldsymbol{\mu} = \prod_i \mu_i$ on $\mathcal{X}^m$. For $X \in \mathcal{X}^m$ define

$$\Phi(X) = \sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \left(\mathbb{E}_{x \sim \mu_i} [f(x)] - f(x_i) \right).$$

Then $\mathbb{E}_{X \sim \boldsymbol{\mu}} [\Phi(X)] \leq \mathbb{E}_{X \sim \boldsymbol{\mu}} \mathcal{R}(\mathcal{F}(X))$.

Proof. For any realization $\sigma = \sigma_1, \ldots, \sigma_m$ of the Rademacher variables

$$\begin{aligned} &\mathbb{E}_{X \sim \boldsymbol{\mu}} [\Phi(X)] \\ &= \mathbb{E}_{X \sim \boldsymbol{\mu}} \sup_{f \in \mathcal{F}} \frac{1}{m} \mathbb{E}_{X' \sim \boldsymbol{\mu}} \sum_{i=1}^{m} (f(x_i') - f(x_i)) \\ &\leq \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \boldsymbol{\mu} \times \boldsymbol{\mu}} \sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i (f(x_i') - f(x_i)), \end{aligned}$$

because of the symmetry of the measure $\boldsymbol{\mu} \times \boldsymbol{\mu}(X, X') = \prod_i \mu_i \times \prod_i \mu_i (X, X')$under the interchange $x_i \leftrightarrow x_i'$. Taking the expectation in σ and applying the triangle inequality gives the result. □

Theorem B.1.3. *Let $\mathcal{F}$ be a $[0,1]$-valued function class on a space $\mathcal{X}$, and $\boldsymbol{\mu}$ as above. For $\delta > 0$ we have with probability greater than $1 - \delta$ in the sample $X \sim \boldsymbol{\mu}$ that for all $f \in \mathcal{F}$*

$$\mathbb{E}_{x \sim \boldsymbol{\mu}} [f(x)] \leq \frac{1}{m} \sum_{i=1}^{m} f(x_i) + \mathbb{E}_{X \sim \boldsymbol{\mu}} \mathcal{R}(\mathcal{F}(X)) + \sqrt{\frac{\ln(1/\delta)}{2m}}.$$

To prove this we apply the bounded-difference inequality (part (i) of Theorem B.1.1) to the function Φ of the previous theorem (see e.g. [18]). Under the conditions of this result, changing one of the x_i will not change $\mathcal{R}(\mathcal{F}(X))$ by more than 2, so again by the bounded difference inequality applied to $\mathcal{R}(\mathcal{F}(X))$ and a union bound we obtain the data dependent version

Corollary B.1.1. *Let $\mathcal{F}$ and $\boldsymbol{\mu}$ be as above. For $\delta > 0$ we have with probability greater than $1 - \delta$ in the sample $X \sim \boldsymbol{\mu}$ that for all $f \in \mathcal{F}$*

$$\mathbb{E}_{x \sim \boldsymbol{\mu}} [f(x)] \leq \frac{1}{m} \sum_{i=1}^{m} f(x_i) + \mathcal{R}(\mathcal{F}(X)) + \sqrt{\frac{9 \ln(2/\delta)}{2m}}.$$

To bound Rademacher averages the following result is very useful [18, 6, 111]

Lemma B.1.1. *Let $A \subseteq \mathbb{R}^n$, and let $\psi_1, \ldots, \psi_n$ be real functions such that*
$\psi_i(s) - \psi_i(t) \leq L|s-t|$, *$\forall i$, and $s, t \in \mathbb{R}$.*
Define $\boldsymbol{\psi}(A) = \{\psi_1(x_1), \ldots, \psi_n(x_n) : (x_1, \ldots, x_n) \in A\}$.
Then

$$\mathcal{R}(\boldsymbol{\psi}(A)) \leq L\mathcal{R}(A).$$

Sometimes it is more convenient to work with Gaussian averages which can be used instead, by virtue of the next lemma. For a proof see e.g. [111]

Lemma B.1.2. *For $A \subseteq \mathbb{R}^k$ we have $\mathcal{R}(A) \leq \sqrt{\pi/2}\,\mathcal{G}(A)$.*

The next result is known as Slepian's lemma ([188], [111]).

Theorem B.1.4. *Let Ω and Ξ be mean zero, separable Gaussian processes indexed by a common set $\mathcal{S}$, such that*

$$\mathbb{E}(\Omega_{s_1} - \Omega_{s_2})^2 \leq \mathbb{E}(\Xi_{s_1} - \Xi_{s_2})^2 \text{ for all } s_1, s_2 \in \mathcal{S}.$$

Then

$$\mathbb{E}\sup_{s\in\mathcal{S}} \Omega_s \leq \mathbb{E}\sup_{s\in\mathcal{S}} \Xi_s.$$

B.2. Proofs

B.2.1. Multitask learning

In this section we prove Theorem 4.3.1. It is an immediate consequence of Hoeffding's inequality and the following uniform bound on the estimation error.

Theorem B.2.1. *Let $\delta > 0$, fix K and let $\mu_1, \ldots, \mu_T$ be probability measures on $H \times \mathbb{R}$. With probability at least $1-\delta$ in the draw of $\mathbf{Z} \sim \prod_{t=1}^T \mu_t$ we have for all $D \in \mathcal{D}_K$ and all $\mathbf{c} \in \mathcal{C}_\alpha^T$ that*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\langle Dc_t, x\rangle, y\right)\right] - \frac{1}{mT}\sum_{t=1}^T\sum_{i=1}^m \ell\left(\left\langle Dc_t, x_i^t\right\rangle, y_i^t\right)$$
$$\leq L\alpha\sqrt{\frac{2S_{\mathrm{Tr}}(\mathbf{X})(K+12)}{mT}} + L\alpha\sqrt{\frac{8S_{\mathrm{Sp}}(\mathbf{X})\ln(2K)}{m}} + \sqrt{\frac{9\ln 2/\delta}{2mT}}.$$

The proof of this theorem requires auxiliary results. Fix $\mathbf{X} \in H^{mT}$ and for $\mathbf{c} = (c_1, \ldots, c_T) \in \left(\mathbb{R}^K\right)^T$ define the random variable

$$F_{\mathbf{c}} = F_{\mathbf{c}}(\boldsymbol{\sigma}) = \sup_{D \in \mathcal{D}_K} \sum_{t,i} \sigma_{ti} \left\langle D c_t, x_i^t \right\rangle. \tag{B.2}$$

Lemma B.2.1. *(i) If* $\mathbf{c} = (c_1, \ldots, c_T)$ *satisfies* $\|c_t\| \leq 1$ *for all* t*, then*

$$\mathbb{E} F_{\mathbf{c}} \leq \sqrt{mTK\ S_{\mathrm{Tr}}(\mathbf{X})}.$$

(ii) If $\mathbf{c}$ *satisfies* $\|c_t\|_1 \leq 1$ *for all* t*, then for any* $s \geq 0$

$$\Pr\left\{F_{\mathbf{c}} \geq \mathbb{E}\left[F_{\mathbf{c}}\right] + s\right\} \leq \exp\left(\frac{-s^2}{8mT\ S_{\mathrm{Sp}}(\mathbf{X})}\right).$$

Proof. (i) We observe that

$$\begin{aligned}
\mathbb{E} F_{\mathbf{c}} &= \mathbb{E} \sup_D \sum_k \left\langle D e_k, \sum_{t,i} \sigma_{ti} c_{tk} x_i^t \right\rangle \\
&\leq \sup_D \left(\sum_k \|D e_k\|^2\right)^{1/2} \mathbb{E}\left(\sum_k \left\|\sum_{t,i} \sigma_{ti} c_{tk} x_i^t\right\|^2\right)^{1/2} \\
&\leq \sqrt{K}\left(\sum_k \mathbb{E}\left\|\sum_{t,i} \sigma_{ti} c_{tk} x_i^t\right\|^2\right)^{1/2} \\
&= \sqrt{K}\left(\sum_{k,t,i} |c_{tk}|^2 \left\|x_i^t\right\|^2\right)^{1/2} \\
&= \sqrt{K}\left(\sum_t \left(\sum_k |c_{tk}|^2\right) \sum_i \left\|x_i^t\right\|^2\right)^{1/2} \\
&\leq \sqrt{K \sum_{t,i} \|x_i^t\|^2} = \sqrt{mTK\ S_{\mathrm{Tr}}(\mathbf{X})}.
\end{aligned}$$

(ii) For any configuration $\boldsymbol{\sigma}$ of the Rademacher variables let $D(\boldsymbol{\sigma})$ be the maximizer in the definition of $F_{\mathbf{c}}(\boldsymbol{\sigma})$. Then for any $s \in [T]$, $j \in [m]$ and any $\sigma' \in \{-1, 1\}$ to replace σ_{sj} we have

$$F_{\mathbf{c}}(\boldsymbol{\sigma}) - F_{\mathbf{c}}\left(\boldsymbol{\sigma}_{(sj) \leftarrow \sigma'}\right) \leq 2\left|\left\langle D(\boldsymbol{\sigma})\, c_s, x_j^s \right\rangle\right|.$$

Using the inequality (B.1) we then obtain

$$\sum_{sj}\left(F_c\left(\boldsymbol{\sigma}\right)-\inf_{\sigma'\in\{-1,1\}}F_{\mathbf{c}}\left(\boldsymbol{\sigma}_{(sj)\leftarrow\sigma'}\right)\right)^2$$

$$\begin{aligned}&\leq 4\sum_{t,i}\left\langle D\left(\boldsymbol{\sigma}\right)c_t,x_i^t\right\rangle^2\\&\leq 4m\sum_t\left\|\hat{\Sigma}\left(X^t\right)\right\|_{\mathrm{Sp}}\left\|D\left(\boldsymbol{\sigma}\right)c_t\right\|^2\\&\leq 4m\sum_t\left\|\hat{\Sigma}\left(X^t\right)\right\|_{\mathrm{Sp}}.\end{aligned}$$

In the last inequality we used the fact that for any $D\in\mathcal{D}_K$ we have $\|Dc_t\|\leq$

$\sum_k|c_{tk}|\,\|De_k\|\leq\|c_t\|_1\leq 1$. The conclusion now follows from part (ii) of Theorem B.1.1. □

Proposition B.2.1. *For every fixed* $\mathbf{Z}=(\mathbf{X},\mathbf{Y})\in(H\times\mathbb{R})^{mT}$ *we have*

$$\mathbb{E}_\sigma\sup_{D\in D,\mathbf{c}\in(\mathcal{C}_\alpha)^T}\sum_{t,i}\sigma_{it}\ell\left(\left\langle Dc_t,x_i^t\right\rangle,y_i^t\right)$$

$$\leq L\alpha\sqrt{2mTS_{\mathrm{Tr}}\left(\mathbf{X}\right)\left(K+12\right)}+L\alpha T\sqrt{8mS_{\mathrm{Sp}}\left(\mathbf{X}\right)\ln\left(2K\right)}.$$

Proof. It suffices to prove the result for $\alpha=1$, the general result being a consequence of rescaling. By Lemma B.1.1 and the Lipschitz properties of the loss function ℓ we have

$$\mathbb{E}_\sigma\sup_{D\in\mathcal{D}_K,\mathbf{c}\in(\mathcal{C})^T,}\sum_{t,i}\sigma_{it}\ell\left(\left\langle Dc_t,x_i^t\right\rangle,y_i^t\right)$$

$$\leq L\mathbb{E}_\sigma\sup_{D\in\mathcal{D}_K,\mathbf{c}\in(\mathcal{C})^T,}\sum_{t,i}\sigma_{it}\left\langle Dc_t,x_i^t\right\rangle. \tag{B.3}$$

For ease of notation, hereafter we will omit the random variables on $\mathbb{E}$ whenever they are clear from the context. Since linear functions on a compact convex set attain their maxima at the extreme points, we have

$$\mathbb{E}\sup_{D\in\mathcal{D}_K,\mathbf{c}\in(\mathcal{C})^T,}\sum_{t=1}^T\sum_{i=1}^m\sigma_{it}\left\langle Dc_t,x_i^t\right\rangle=\mathbb{E}\max_{\mathbf{c}\in\mathrm{ext}(\mathcal{C})^T}F_{\mathbf{c}}, \tag{B.4}$$

where $F_{\mathbf{c}}$ is defined as in (B.2). Let $c=\sqrt{mKTS_{\mathrm{Tr}}\left(\mathbf{X}\right)}$. Now for any $\delta\geq 0$ we have, since $F_{\mathbf{c}}\geq 0$,

$$\mathbb{E}\max_{\mathbf{c}\in\mathrm{ext}(\mathcal{C})^T}F_{\mathbf{c}}=\int_0^\infty\Pr\left\{\max_{\mathbf{c}\in\mathrm{ext}(\mathcal{C})^T}F_{\mathbf{c}}>s\right\}ds$$

$$
\begin{aligned}
&\leq c+\delta+\sum_{\mathbf{c}\in(\mathrm{ext}(\mathcal{C}))^T}\int_{\sqrt{mKTS_{\mathrm{Tr}}(\mathbf{X})}+\delta}^{\infty}\Pr\left\{F_{\mathbf{c}}>s\right\}ds\\
&\leq c+\delta+\sum_{\mathbf{c}\in(\mathrm{ext}(\mathcal{C}))^T}\int_{\delta}^{\infty}\Pr\left\{F_{\mathbf{c}}>\mathbb{E}F_{\mathbf{c}}+s\right\}ds\\
&\leq c+\delta+(2K)^T\int_{\delta}^{\infty}\exp\left(\frac{-s^2}{8mTS_{\mathrm{Sp}}(\mathbf{X})}\right)ds\\
&\leq c+\delta+\frac{4mTS_{\mathrm{Sp}}(\mathbf{X})(2K)^T}{\delta}\exp\left(\frac{-\delta^2}{8mTS_{\mathrm{Sp}}(\mathbf{X})}\right).
\end{aligned}
$$

Here the first inequality follows from the fact that probabilities never exceed 1 and a union bound. The second inequality follows from Lemma B.2.1, part (i), since $\mathbb{E}F_{\mathbf{k}}\leq\sqrt{mKTS_{\mathrm{Tr}}(\mathbf{X})}$. The third inequality follows from Lemma B.2.1, part (ii), and the fact that the cardinality of $\mathrm{ext}(\mathcal{C})$ is $2K$, and the last inequality follows from a well known estimate on Gaussian random variables. Setting $\delta=\sqrt{8mTS_{\mathrm{Sp}}(\mathbf{X})\ln\left(e(2K)^T\right)}$ we obtain with some easy simplifying estimates

$$
\mathbb{E}\max_{\mathbf{c}\in\mathrm{ext}(\mathcal{C})^T}F_{\mathbf{c}}\leq\sqrt{2mT(K+12)S_{\mathrm{Tr}}(\mathbf{X})}+T\sqrt{8mS_{\mathrm{Sp}}(\mathbf{X})\ln(2K)},
$$

which together with (B.3) and (B.4) gives the result. □

Theorem B.2.1 now follows from Corollary B.1.1.

If the set $\mathcal{C}_\alpha$ is replaced by any other subset $\mathcal{C}'$ of the ℓ_2-ball of radius α, a similar proof strategy can be employed. The denominator in the exponent of Lemma B.2.1-(ii) then obtains another factor of $\sqrt{K}$. The union bound over the extreme points in $\mathrm{ext}(\mathcal{C})$ in the previous proposition can be replaced by a union bound over a cover $\mathcal{C}'$. This leads to the alternative result mentioned in Remark 5 following the statement of Theorem 4.3.1.

Another modification leads to a bound for the method presented in [107], where the constraint $\|De_k\|\leq 1$ is replaced by $\|D\|_{\mathrm{Fr}}\leq\sqrt{K}$ (here $\|\cdot\|_{\mathrm{Fr}}$ is the Frobenius or Hilbert Schmidt norm) and the constraint $\|c_t\|_1\leq\alpha,\forall t$ is replaced by $\sum\|c_t\|_1\leq\alpha T$. To explain the modification we set $\alpha=1$. Part (i) of Lemma B.2.1 is easily verified. The union bound over $(\mathrm{ext}(\mathcal{C}))^T$ in the previous proposition is replaced by a union bound over the $2TK$ extreme points of the ℓ_1-Ball of radius T in $\mathbb{R}^{TK}$. For part (ii) we use the fact that the concentration result is only needed for $\mathbf{c}$ being an extreme point (so that it involves only a single task) and obtain the bound $\sum_t\left\|\hat{\Sigma}(X^t)\right\|_{\mathrm{Sp}}\|Dc_t\|^2\leq TKS'_{\mathrm{Sp}}(\mathbf{X})$,

leading to

$$\Pr\left\{F_{\mathbf{c}} \geq E\left[F_{\mathbf{c}}\right] + s\right\} \leq \exp\left(\frac{-s^2}{8mTK\ S'_{\mathrm{Sp}}\left(\mathbf{X}\right)}\right).$$

Proceeding as above we obtain the excess risk bound

$$L\alpha\sqrt{\frac{2S_{\mathrm{Tr}}\left(\mathbf{X}\right)\left(K+12\right)}{mT}} + L\alpha\sqrt{\frac{8KS'_{\mathrm{Sp}}\left(\mathbf{X}\right)\ln\left(2KT\right)}{m}} + \sqrt{\frac{8\ln 4/\delta}{mT}},$$

to replace the bound in Theorem 4.3.1. The factor $\sqrt{K}$ in the second term seems quite weak, but it must be borne in mind that the constraint $\|D\|_{\mathrm{Fr}} \leq \sqrt{K}$ is much weaker than $\|De_k\| \leq 1$, and allows for a smaller approximation error. If we retain $\|De_k\| \leq 1$ and only modify the c-constraint to $\sum \|c_t\|_1 \leq \alpha T$ the $\sqrt{K}$ in the second term disappears and by comparison to Theorem 4.3.1 there is only and additional $\ln T$ and the switch from $S_{\mathrm{Sp}}\left(\mathbf{X}\right)$ to $S'_{\mathrm{Sp}}\left(\mathbf{X}\right)$, reflecting the fact that $\sum \|c_t\|_1 \leq \alpha T$ is a much weaker constraint than $\|c_t\|_1 \leq \alpha, \forall t$, so that, again, a smaller minimum in (4.1) is possible for the modified method.

B.2.2. Transfer learning

In this section we prove Theorem 4.3.2. The basic strategy is as follows. Recall the definition (4.4) of the measure $\rho_{\mathcal{E}}$, which governs the generation of a training sample in the environment $\mathcal{E}$. On a given training sample $\mathbf{z} \sim \rho_{\mathcal{E}}$ the algorithm A_D as defined in (4.3) incurs the empirical risk

$$\hat{R}_D\left(\mathbf{z}\right) = \min_{c \in \mathcal{C}_\alpha} \frac{1}{m}\sum_{i=1}^{m} \ell\left(\left\langle Dc, x_i\right\rangle, y_i\right).$$

The algorithm A_D, essentially being the Lasso, has very good estimation properties, so $\hat{R}_D\left(\mathbf{z}\right)$ will be close to the true risk of A_D in the corresponding task. This means that we only really need to estimate the expected empirical risk $\mathbb{E}_{\mathbf{z}\sim\rho_{\mathcal{E}}}\hat{R}_D\left(\mathbf{z}\right)$ of A_D on future tasks. On the other hand the minimization problem (4.1) can be written as

$$\min_{D \in \mathcal{D}_K} \frac{1}{T}\sum_{t=1}^{T} \hat{R}_D\left(\mathbf{z}^t\right) \text{ with } \mathbf{Z} = \left(\mathbf{z}^1, \ldots, \mathbf{z}^T\right) \sim \left(\rho_{\mathcal{E}}\right)^T,$$

with dictionary $D\left(\mathbf{Z}\right)$ being the minimizer. If $\mathcal{D}_K$ is not too large this should be similar to $\mathbb{E}_{\mathbf{z}\sim\rho_{\mathcal{E}}}\hat{R}_{D(\mathbf{Z})}\left(\mathbf{z}\right)$. In the sequel we make this precise.

Lemma B.2.2. *For $v \in H$ with $\|v\| \leq 1$ and $X \in H^m$ let F be the random variable*

$$F = \left| \left\langle v, \sum_i \sigma_i x_i \right\rangle \right|.$$

Then (i) $\mathbb{E}F \leq \sqrt{m} \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}}^{1/2}$ and (ii) for $s \geq 0$

$$\Pr\{F > \mathbb{E}F + s\} \leq \exp\left(\frac{-s^2}{2m \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}}} \right).$$

Proof. (i). Using Jensen's inequality and (B.1) we get

$$\begin{aligned} \mathbb{E}F &\leq \left(\mathbb{E} \left\langle v, \sum_i \sigma_i x_i \right\rangle^2 \right)^{1/2} \\ &= \left(\sum_i \langle v, x_i \rangle^2 \right)^{1/2} \leq \sqrt{m \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}}}. \end{aligned}$$

(ii) Let $\boldsymbol{\sigma}$ be any configuration of the Rademacher variables. For any $\sigma', \sigma'' \in \{-1, 1\}$ to replace σ_{sj} we have

$$F\left(\boldsymbol{\sigma}_{(sj)\leftarrow\sigma'}\right) - F\left(\boldsymbol{\sigma}_{(sj)\leftarrow\sigma''}\right) \leq 2\left|\langle v, x_j \rangle\right|,$$

so the conclusion follows from the bounded difference inequality, Theorem B.1.1 (i). □

Lemma B.2.3. *For $v_1, \ldots, v_K \in H$ satisfying $\|v_k\| \leq 1$, $X \in H^m$ we have*

$$\mathbb{E} \max_k \left| \left\langle v_k, \sum_i \sigma_i x_i \right\rangle \right| \leq \sqrt{2m \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}}} \left(2 + \sqrt{\ln K}\right).$$

Proof. Let $F_k = |\langle v_k, \sum_i \sigma_i x_i \rangle|$. Setting $c = \sqrt{m \left\| \hat{\Sigma}(X) \right\|_{\mathrm{Sp}}}$ and using integration by parts we have for $\delta \geq 0$

$$\begin{aligned}
\mathbb{E}\max_k F_k &\leq c+\delta+\int_{\sqrt{m\left\|\hat{\Sigma}(X)\right\|_{\mathrm{Sp}}}+\delta}^{\infty}\max_k \Pr\left\{F_k \geq s\right\} ds \\
&\leq c+\delta+\sum_k \int_{\delta}^{\infty}\Pr\left\{F_k \geq \mathbb{E}F_k + s\right\} ds \\
&\leq c+\delta+\sum_k \int_{\delta}^{\infty}\exp\left(\frac{-s^2}{2m\left\|\hat{\Sigma}\left(X\right)\right\|_{\mathrm{Sp}}}\right) ds \\
&\leq c+\delta+\frac{mK\left\|\hat{\Sigma}\left(X\right)\right\|_{\mathrm{Sp}}}{\delta}\exp\left(\frac{-\delta^2}{2m\left\|\hat{\Sigma}\left(X\right)\right\|_{\mathrm{Sp}}}\right).
\end{aligned}$$

Above the first inequality is trivial, the second follows from Lemma B.2.2 (i) and a union bound, the third inequality follows from Lemma B.2.2 (ii) and the last from a well known approximation. The conclusion follows from substitution of

$$\delta = \sqrt{2m\left\|\hat{\Sigma}\left(X\right)\right\|_{\mathrm{Sp}}\ln\left(eK\right)}.$$ □

Proposition B.2.2. *Let* $S_{\mathrm{Sp}}(\mathcal{E}) := \mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{(\mathbf{z})\sim\mu_\tau^m}\left\|\hat{\Sigma}\left(X\right)\right\|_{\mathrm{Sp}}$. *With probability at least* $1-\delta$ *in the multisample* $\mathbf{Z}\sim\rho_{\mathcal{E}}^T$

$$\begin{aligned}
&\sup_{D\in\mathcal{D}_K} R_{\mathcal{E}}\left(A_D\right) - \frac{1}{T}\sum_{t=1}^{T}\hat{R}_D\left(\mathbf{z}^t\right) \qquad \text{(B.5)}\\
\leq\ & L\alpha K\sqrt{\frac{2\pi S_{\mathrm{Tr}}\left(\mathbf{X}\right)}{T}} + 4L\alpha\sqrt{\frac{S_{\mathrm{Sp}}\left(\mathcal{E}\right)\left(2+\ln K\right)}{m}} + \sqrt{\frac{9\ln 2/\delta}{2T}}.
\end{aligned}$$

Proof. Following our strategy we write (abbreviating $\rho = \rho_{\mathcal{E}}$)

$$\begin{aligned}
&\sup_{D\in\mathcal{D}_K} R_{\mathcal{E}}\left(A_D\right) - \frac{1}{T}\sum_{t=1}^{T}\hat{R}_D\left(\mathbf{z}^t\right) \\
\leq\ &\sup_{D\in\mathcal{D}_K}\mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{\mathbf{z}\sim\mu_\tau^m} \qquad \text{(B.6)}\\
&\left[\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\left\langle A_D\left(\mathbf{z}\right), x\right\rangle, y\right)\right] - \hat{R}_D\left(\mathbf{z}\right)\right] \\
+\ &\sup_{D\in\mathcal{D}_K}\mathbb{E}_{\mathbf{z}\sim\rho}\left[\hat{R}_D\left(\mathbf{z}\right)\right] - \frac{1}{T}\sum_{t=1}^{T}\hat{R}_D\left(\mathbf{z}^t\right)
\end{aligned}$$

and proceed by bounding each of the two terms in turn.

For any fixed dictionary D and any measure μ on $\mathcal{Z}$ we have

$$\begin{aligned}
& \mathbb{E}_{\mathbf{z}\sim\mu^m}\left[\mathbb{E}_{(x,y)\sim\mu}\left[\ell\left(\langle A_D\left(\mathbf{z}\right),x\rangle,y\right)\right]-\hat{R}_D\left(\mathbf{z}\right)\right] \\
\leq\ & \mathbb{E}_{\mathbf{z}\sim\mu^m}\sup_{c\in\mathcal{C}_\alpha}\left[\mathbb{E}_{(x,y)\sim\mu}\left[\ell\left(\langle Dc,x\rangle,y\right)\right]\right. \\
& \left.-\frac{1}{m}\sum_{i=1}^{m}\ell\left(\langle Dc,x_i\rangle,y_i\right)\right] \\
\leq\ & \frac{2}{m}\mathbb{E}_{\mathbf{z}\sim\mu^m}\mathbb{E}_\sigma\sup_{c\in\mathcal{C}_\alpha}\sum_{i=1}^{m}\sigma_i\ell\left(\langle Dc,x_i\rangle,y_i\right) \quad \text{[Theorem B.1.2]} \\
\leq\ & \frac{2L}{m}\mathbb{E}_{\mathbf{z}\sim\mu^m}\mathbb{E}_\sigma\sup_{c\in\mathcal{C}_\alpha}\sum_{k}c_k\left\langle De_k,\sum_{i=1}^{m}\sigma_i x_i\right\rangle \quad \text{[Lemma B.1.1]} \\
\leq\ & \frac{2L\alpha}{m}\mathbb{E}_{\mathbf{z}\sim\mu^m}\mathbb{E}_\sigma\max_{k}\left|\left\langle De_k,\sum_{i=1}^{m}\sigma_i x_i\right\rangle\right| \quad \text{[Hölder's ineq.]} \\
\leq\ & \frac{2L\alpha}{m}\mathbb{E}_{\mathbf{z}\sim\mu^m}\sqrt{2m\lambda_{\max}\left(\hat{\Sigma}\left(X\right)\right)}\left(2+\sqrt{\ln K}\right) \quad \text{[Lemma B.2.3 (i)]} \\
\leq\ & 2L\alpha\sqrt{\frac{4\mathbb{E}_{\mathbf{z}\sim\mu^m}\lambda_{\max}\left(\hat{\Sigma}\left(X\right)\right)\left(2+\ln K\right)}{m}} \quad \text{[Jensen's ineq.]}.
\end{aligned}$$

This gives the bound

$$\begin{aligned}
& \mathbb{E}_{\mathbf{z}\sim\mu^m}\left[\mathbb{E}_{(x,y)\sim\mu}\left[\ell\left(\langle A_D\left(\mathbf{z}\right),x\rangle,y\right)\right]-\hat{R}_D\left(\mathbf{z}\right)\right] \\
& \leq 4L\alpha\sqrt{\frac{\mathbb{E}_{\mathbf{z}\sim\mu^m}\lambda_{\max}\left(\hat{\Sigma}\left(X\right)\right)\left(2+\ln K\right)}{m}} \qquad \text{(B.7)}
\end{aligned}$$

valid for every measure μ on $H\times\mathbb{R}$ and every $D\in\mathcal{D}_K$. Replacing μ by μ_τ, taking the expectation as $\tau\sim\mathcal{E}$ and using Jensen's inequality bounds the first term on the right hand side of (B.6) by the second term on the right hand side of (B.5).

We proceed to bound the second term. From Corollary B.1.1 and Lemma B.1.2 we get that with probability at least $1-\delta$ in $\mathbf{Z}\sim\left(\rho_{\mathcal{E}}\right)^T$

$$\begin{aligned}
& \sup_{D\in\mathcal{D}_K}\mathbb{E}_{\mathbf{z}\sim\rho}\left[\hat{R}_D\left(\mathbf{z}\right)\right]-\tfrac{1}{T}\textstyle\sum_{t=1}^{T}\hat{R}_D\left(\mathbf{z}^t\right) \\
& \leq \frac{\sqrt{2\pi}}{T}\mathbb{E}_\zeta\sup_{D\in\mathcal{D}_K}\sum_{t=1}^{T}\zeta_t\hat{R}_D\left(\mathbf{z}_t\right)+\sqrt{\frac{9\ln 2/\delta}{2T}},
\end{aligned}$$

where ζ_t is an orthogaussian sequence. Define two Gaussian processes Ω and Ξ indexed by $\mathcal{D}_K$ as

$$\Omega_D=\textstyle\sum_{t=1}^{T}\zeta_t\hat{R}_D\left(\mathbf{z}^t\right)$$

and

$\Xi_D = \frac{L\alpha}{\sqrt{m}} \sum_{t=1}^{T} \sum_{i=1}^{m} \sum_{k=1}^{K} \zeta_{kij} \langle De_k, x_i^t \rangle$,

where the ζ_{ijk} are also orthogaussian. Then for $D_1, D_2 \in \mathcal{D}_K$

$$
\begin{aligned}
& \mathbb{E}\left(\Omega_{D_1} - \Omega_{D_2}\right)^2 = \\
&= \sum_{t=1}^{T} \left(\hat{R}_{D_1}\left(\mathbf{z}^t\right) - \hat{R}_{D_2}\left(\mathbf{z}^t\right)\right)^2 \\
&\leq \sum_{t=1}^{T} \left(\sup_{c \in \mathcal{C}_\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell\left(\left\langle D_1 c, x_i^t \right\rangle, y_i^t\right) - \ell\left(\left\langle D_2 c, x_i^t \right\rangle, y_i^t\right) \right)^2 \\
&\leq L^2 \sum_{t=1}^{T} \sup_{c \in \mathcal{C}_\alpha} \left(\frac{1}{m} \sum_{i=1}^{m} \left\langle c, \left(D_1^\top - D_2^\top\right) x_i^t \right\rangle \right)^2 \text{ Lipschitz} \\
&\leq \frac{L^2}{m} \sum_{t=1}^{T} \sup_{c \in \mathcal{C}_\alpha} \sum_{i=1}^{m} \left\langle c, \left(D_1^\top - D_2^\top\right) x_i^t \right\rangle^2 \text{ Jensen} \\
&\leq \frac{L^2 \alpha^2}{m} \sum_{t=1}^{T} \sum_{i=1}^{m} \left\| \left(D_1^\top - D_2^\top\right) x_i^t \right\|^2 \text{ (Cauchy-Schwarz)} \\
&= \frac{L^2 \alpha^2}{m} \sum_{t=1}^{T} \sum_{i=1}^{m} \sum_{k=1}^{K} \left(\left\langle D_1 e_k, x_i^t \right\rangle - \left\langle D_2 e_k, x_i^t \right\rangle \right)^2 \\
&= \mathbb{E}\left(\Xi_{D_1} - \Xi_{D_2}\right)^2 .
\end{aligned}
$$

So by Slepian's Lemma

$\mathbb{E} \sup_{D \in \mathcal{D}_K} \sum_{t=1}^{T} \zeta_j \hat{R}_D\left(\mathbf{z}_t\right)$

$$
\begin{aligned}
&= \mathbb{E} \sup_{D \in \mathcal{D}_K} \Omega_D \leq \mathbb{E} \sup_{D \in \mathcal{D}} \Xi_D \\
&= \frac{L\alpha}{\sqrt{m}} \mathbb{E} \sup_{D \in \mathcal{D}_K} \sum_{t=1}^{T} \sum_{i=1}^{m} \sum_{k=1}^{K} \zeta_{kij} \langle De_k, x_{ti} \rangle \\
&= \frac{L\alpha}{\sqrt{m}} \mathbb{E} \sup_{D \in \mathcal{D}_K} \sum_{k=1}^{K} \left\langle De_k, \sum_{t=1}^{T} \sum_{i=1}^{m} \zeta_{kij} x_i^t \right\rangle \\
&\leq \frac{L\alpha}{\sqrt{m}} \sup_{D \in \mathcal{D}_K} \left(\sum_k \|De_k\|^2 \right)^{1/2} \mathbb{E} \left(\sum_k \left\| \sum_{t,i} \zeta_{tki} x_i^t \right\|^2 \right)^{1/2} \\
&\leq \frac{L\alpha\sqrt{K}}{\sqrt{m}} \left(\sum_k \mathbb{E} \left\| \sum_{t,i} \zeta_{tki} x_i^t \right\|^2 \right)^{1/2} \\
&\leq \frac{L\alpha\sqrt{K}}{\sqrt{m}} \left(\sum_k \sum_{t,i} \left\| x_i^t \right\|^2 \right)^{1/2} \leq L\alpha K \sqrt{T S_{\mathrm{Tr}}\left(\mathbf{X}\right)}.
\end{aligned}
$$

We therefore have that with probability at least $1-\delta$ in the draw of the multi sample $\mathbf{Z} \sim \rho^T$

$$\sup_{D \in \mathcal{D}_K} \mathbb{E}_{\mathbf{z} \sim \rho}\left[\hat{R}_D\left(\mathbf{z}\right)\right] - \frac{1}{T}\sum_{i=1}^{T} \hat{R}_D\left(\mathbf{Z}^t\right)$$

$$\leq L\alpha K \sqrt{\frac{2\pi S_{\mathrm{Tr}}\left(\mathbf{X}\right)}{T}} + \sqrt{\frac{9\ln 2/\delta}{2T}}. \tag{B.8}$$

which in (B.6) combines with (B.7) to give the conclusion. □

Proof of Theorem 4.3.2. Let D_{opt} and c_τ be the minimizers in the definition of R_{opt}, so that

$$R_{\mathrm{opt}} = \mathbb{E}_{\tau \sim \mathcal{E}} \mathbb{E}_{(x,y)\sim \mu_\tau} \ell\left[\left(\left\langle D_{\mathrm{opt}} c_\tau, x\right\rangle, y\right)\right].$$

$R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right) - R_{\mathrm{opt}}$ can be decomposed as the sum of four terms,

$$\left(R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right) - \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{D(\mathbf{Z})}\left(\mathbf{z}^t\right)\right) \tag{B.9}$$

$$+\left(\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{D(\mathbf{Z})}\left(\mathbf{z}^t\right) - \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{D_{\mathrm{opt}}}\left(\mathbf{z}^t\right)\right) \tag{B.10}$$

$$+\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{D_{\mathrm{opt}}}\left(\mathbf{z}^t\right) - \mathbb{E}_{\mathbf{z}\sim\boldsymbol{\rho}_{\mathcal{E}}} \hat{R}_{D_{\mathrm{opt}}}\left(\mathbf{z}\right) \tag{B.11}$$

$$+\mathbb{E}_{\tau\sim\mathcal{E}}\Bigg[\mathbb{E}_{\mathbf{z}\sim\mu_\tau^m} \hat{R}_{D_{\mathrm{opt}}}\left(\mathbf{z}\right)$$

$$-\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\left\langle D_{\mathrm{opt}} c_\tau, x\right\rangle, y\right)\right]\Bigg]. \tag{B.12}$$

By definition of $\hat{R}$ we have for every τ that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\mu_\tau^m} \hat{R}_{D_{\mathrm{opt}}}\left(\mathbf{z}\right) \\
&= \mathbb{E}_{\mathbf{z}\sim\mu_\tau^m} \min_{c\in\mathcal{C}_\alpha} \frac{1}{m}\sum_{i=1}^{m} \ell\left[\left(\left\langle D_{\mathrm{opt}} c, x_i\right\rangle, y_i\right)\right] \\
&\leq \mathbb{E}_{\mathbf{z}\sim\mu_\tau^m} \frac{1}{m}\sum_{i=1}^{m} \ell\left[\left(\left\langle D_{\mathrm{opt}} c_\tau, x_i\right\rangle, y_i\right)\right] \\
&= \mathbb{E}_{(x,y)\sim\mu_\tau} \ell\left[\left(\left\langle D_{\mathrm{opt}} c_\tau, x\right\rangle, y\right)\right].
\end{aligned}$$

The term (B.12) above is therefore non-positive. By Hoeffding's inequality the term (B.11) is less than $\sqrt{\ln\left(2/\delta\right)/2T}$ with probability at least $1-\delta/2$. The term (B.10) is non-positive by the definition of $D\left(\mathbf{Z}\right)$. Finally we use Proposition B.2.2 to obtain with

probability at least $1 - \delta/2$ that
$R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right) - \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{D(\mathbf{Z})}\left(\mathbf{z}_t\right)$

$$\begin{aligned} &\leq \sup_{D \in \mathcal{D}_K} R_{\mathcal{E}}\left(A_D\right) - \frac{1}{T}\sum_{t=1}^{T} \hat{R}_D\left(\mathbf{z}^t\right) \\ &\leq L\alpha K\sqrt{\frac{2\pi S_{\mathrm{Tr}}\left(\mathbf{X}\right)}{T}} \\ &+ 4L\alpha\sqrt{\frac{S_{\mathrm{Sp}}\left(\mathcal{E}\right)\left(2 + \ln K\right)}{m}} + \sqrt{\frac{9\ln 4/\delta}{2T}}. \end{aligned}$$

Combining these estimates on (B.9), (B.10), (B.11) and (B.12) in a union bound gives the conclusion. □

C. Decoupling of Features: Appendix

C.1. Proof of Theorem 5.4.1

We define the function

$$\Omega(W,V) = \frac{1}{2}\left(\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2 + \alpha\|W^\top V\|_{\mathrm{Fr}}^2\right).$$

The proof is based on the following lemma[1].

Lemma C.1.1. *Assume that* $\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2 < R^2$. *Then the function* Ω *is convex on this domain provided that* $\alpha < \frac{2}{R^2}$.

Proof. We will compute the Hessian matrix H of function Ω and establish that it is positive semidefinite in the domain of interest, whenever $\alpha \leq \frac{2}{R^2}$. From calculus we find that

$$H(W,V) = \begin{bmatrix} A(W,V) & C(W,V) \\ C(W,V)^\top & B(W,V) \end{bmatrix}$$

where

$$A_{ti,\hat{t}j}(W,V) = \frac{\partial^2\Omega(W,V)}{\partial w_{ti}\partial w_{\hat{t}j}} = (\delta_{ij} + \alpha\sum_s v_{si}v_{sj})\delta_{t\hat{t}}$$

$$B_{si,\hat{s}j}(W,V) = \frac{\partial^2\Omega(W,V)}{\partial v_{si}\partial v_{\hat{s}i}} = (\delta_{ij} + \alpha\sum_t w_{ti}w_{tj})\delta_{s\hat{s}}$$

[1] We also refer to [225] for a similar result for the regularizer $\Omega(W,V) = \|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2 + \alpha\|W^\top V\|_1$. See also our remarks preceding equation (5.5).

$$C_{ti,sj}(W,V) = \frac{\partial^2 \Omega(W,V)}{\partial w_{ti} \partial v_{sj}} = \alpha(\langle w_t, v_s \rangle \delta_{ij} + v_{si} w_{tj}).$$

The matrix H is positive semidefinite if, for every $X \in \mathbb{R}^{d \times T}$ and $Z \in \mathbb{R}^{d \times S}$ it holds that

$$\sum_{tij} X_{ti} A_{ti,tj} X_{tj} + \sum_{sij} Z_{si} B_{si,sj} Z_{sj} + 2 \sum_{stij} X_{ti} C_{tisj} Z_{sj} \geq 0$$

where $t \in [T]$, $s \in [S]$ and $i, j \in [d]$. In matrix notation we obtain

$$\|X\|_{\mathrm{Fr}}^2 + \|Z\|_{\mathrm{Fr}}^2 + \alpha \|X^\top V + W^\top Z\|_{\mathrm{Fr}}^2 + 2\alpha \langle W^\top V, X^\top Z \rangle_{\mathrm{Fr}}.$$

Discarding the third term and using Cauchy-Schwarz inequality, we bound from below the above quantity by

$$\|X\|_{\mathrm{Fr}}^2 + \|Z\|_{\mathrm{Fr}}^2 - 2\alpha \|W^\top V\|_{\mathrm{Fr}} \|X^\top Z\|_{\mathrm{Fr}}.$$

Next, using the inequality $2\|X^\top Z\|_{\mathrm{Fr}} \leq \|X\|_{\mathrm{Fr}}^2 + \|Z\|_{\mathrm{Fr}}^2$, we have the lower bound

$$(\|X\|_{\mathrm{Fr}}^2 + \|Z\|_{\mathrm{Fr}}^2)(1 - \alpha \|W^\top V\|_{\mathrm{Fr}}).$$

The result follows.

Proof of Theorem 5.4.1. We first use equation (5.8) and rewrite problem (5.7) as an optimization problem in W and V only. Specifically, we obtain the objective function

$$f(W,V) = \mathcal{E}(W,V) + \gamma \|[W,V]\|_{\mathrm{Tr}}^2 + \lambda \|W^\top V\|_{\mathrm{Fr}}^2 + \rho(\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2)$$

where recall that $\|\cdot\|_{\mathrm{Tr}}$ denotes the trace norm.

Since the function f is continuous and grows at infinity, it has a minimum. Moreover, if the pair $(\hat{W}, \hat{V})$ is a minimizer then $f(\hat{W}, \hat{V}) \leq f(0,0)$, which readily implies that $\|W\|_{\mathrm{Fr}}^2 + \|V\|_{\mathrm{Fr}}^2 \leq \mathcal{E}(0,0)/\rho$. The result now follows by applying Lemma C.1.1 with $R^2 = \mathcal{E}(0,0)/\rho$ and $\alpha = \lambda/\rho$. □

D. A New Convex Relaxation for Tensor Completion: Appendix

D.1. Minimizing over $\mathcal{W}$

In order to solve Step (a), we need to solve the problem

$$\min_{\boldsymbol{\mathcal{W}}}\left\{F(\boldsymbol{\mathcal{W}}) - \sum_{n=1}^{N}\left(\langle \boldsymbol{\mathcal{C}}_n, \boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n\rangle + \tfrac{\beta}{2}\left\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}_n\right\|_{\mathrm{Fr}}^2\right)\right\}$$

which is equal to

$$\min_{\boldsymbol{\mathcal{W}}}\left\{F(\boldsymbol{\mathcal{W}}) - \left\langle \sum_{n=1}^{N}\boldsymbol{\mathcal{C}}_n + \beta\boldsymbol{\mathcal{B}}_n, \boldsymbol{\mathcal{W}}\right\rangle + \frac{N\beta}{2}\left\|\boldsymbol{\mathcal{W}}\right\|_{\mathrm{Fr}}^2 + c\right\},$$

for some constant c whose value is independent of $\boldsymbol{\mathcal{W}}$.

Notice that the terms where the whole tensor $\boldsymbol{\mathcal{W}}$ appears are both the square of its Frobenius norm and inner products with other tensors. By using the definition of the tensor inner products, it is easy to see that in both cases we can decouple the whole tensor $\boldsymbol{\mathcal{W}}$ in terms of the fibers of its mode-1 unfolding, that is the original tasks weight vectors: $\langle \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{W}}\rangle = \sum_{t=1}^{T}\langle \boldsymbol{\mathcal{Z}}_{:,t}, \boldsymbol{\mathcal{W}}_{:t}\rangle\,,\ \forall \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1\times\cdots\times p_N}$. Consequently, solving the above optimization problem is equivalent to solving the following $T = p_2 p_3 \ldots p_N$ minimization problems

$$\min_{w}\ \textstyle\sum_{i=1}^{m_t}\ell\left(\langle x_i^t, w_t\rangle, y_i^t\right) - \left\langle \left(\sum_{n=1}^{N}\boldsymbol{\mathcal{C}}_n + \beta\boldsymbol{\mathcal{B}}_n\right)_{(1),t}, w\right\rangle + \tfrac{N\beta}{2}\left\|w_t\right\|_{\mathrm{Fr}}^2, \tag{D.1}$$

for all $t \in [T]$, where we use the notation $w_t = \boldsymbol{\mathcal{W}}_{(1),t}$. In particular, if we consider one half of the square loss function, then the solution to problem (D.1) has the closed form

$$w_t = \left(X^t X^{t\top} + N\beta I\right)^{-1} \left[X^t y^t + \left(\sum_{n=1}^{N} \boldsymbol{\mathcal{C}}_n + \beta \boldsymbol{\mathcal{B}}_n\right)_{(1),t}\right]$$

where $X^t \in \mathbb{R}^{d \times m_t}$ is the data matrix for task t, that is, the columns of X^t are the inputs x_i^t, $i = 1, \ldots, m_t$, and $y^t = (y_1^t, \ldots, y_{m_t}^t)^\top$.

D.2. Computation of an Approximated Projection

Here, we address the issue of computing an approximate Euclidean projection onto the set

$$\mathcal{S} = \{v \in \mathbb{R}^d : v_1 \geq \cdots \geq v_d \geq 0\}.$$

That is, for every v, we shall find a point $\tilde{P}_{\mathcal{S}}(v) \in \mathcal{S}$ such that

$$\left\|\tilde{P}_{\mathcal{S}}(v) - z\right\|_2 \leq \|v - z\|_2, \forall z \in \mathcal{S}. \tag{D.2}$$

As noted in [26], in order to build $\tilde{P}_{\mathcal{S}}$ such that this property holds true, it is useful to express the set of interest as the smallest one in a series of nested sets. In our problem, we can express $\mathcal{S}$ as

$$\mathcal{S} = \mathcal{S}_d \subseteq \mathcal{S}_{d-1} \subseteq \ldots \subseteq \mathcal{S}_1,$$

where $\mathcal{S}_i := \left\{v \in \mathbb{R}^d : v_1 \geq v_2 \geq \ldots \geq v_i, v \geq 0\right\}$. This property allows us to sequentially compute an approximate projection on the set $\mathcal{S}$ using the formula

$$\tilde{P}_{\mathcal{S}}(v) = P_{\mathcal{S}_d}\left(P_{\mathcal{S}_{d-1}} \cdots (P_{\mathcal{S}_1}(v))\right) \tag{D.3}$$

where, for every close convex set $\mathcal{C}$, we let $P_{\mathcal{C}}$ be the associated projection operator. Indeed, following [26], we can argue by induction on i that $\tilde{P}_{\mathcal{S}}(v)$ verifies condition (D.2). The base case is $\|P_{\mathcal{S}_1}(v) - z\|_2 = \|v - z\|_2$, which is obvious. Now, if for a given $1 \leq i \leq d-1$ it holds that

$$\|P_{\mathcal{S}_i}(\cdots P_{\mathcal{S}_1}(v)) - z\|_2 \leq \|v - z\|_2$$

then

$$\left\|P_{\mathcal{S}_{i+1}}(P_{\mathcal{S}_i}(\cdots P_{\mathcal{S}_1}(v))) - z\right\|_2 \leq \|P_{\mathcal{S}_i}(\cdots P_{\mathcal{S}_1}(v)) - z\|_2 \leq \|v - z\|_2,$$

since z is also contained in $\mathcal{S}_{i+1}$.

Algorithm D.1 Computing an approximated projection onto the set $\mathcal{S} = \{v \in \mathbb{R}^d : v_1 \geq \cdots \geq v_d \geq 0\}$.

Input: $y \in \mathbb{R}^d_+$.
Output: $v \in \mathcal{S}$.
Initialization: $v \leftarrow y$.
for $i = 1, 2, \ldots, d$ **do**
  while $v_i < v_{i+1}$ **do**
    $j \leftarrow \operatorname{argmax}\{\ell : \ell \in [i], v_i = v_{i-\ell+1}\}$
    if $v_{i-j} \geq v_i + \frac{v_{i+1}-v_i}{j+1}$ **then**
      $v_{1:i+1} \leftarrow \left[v_{1:i-j},\ \left(v_i + \frac{v_{i+1}-v_i}{j+1}\right)\mathbf{1}^{j+1}\right]$
    else
      $v_{1:i+1} \leftarrow \left[v_{1:i-j},\ v_{i-j}\mathbf{1}^j,\ v_{i+1} - (v_{i-j} - v_i)\, j\right]$
    end if
  end while
end for

Note that to evaluate the right hand side of equation (D.3) we do not require full knowledge of $P_{\mathcal{S}_i}$, we only need to compute $P_{\mathcal{S}_{i+1}}(v)$ for $v \in \mathcal{S}_i$. The next proposition describes a recursive formula to achieve this step.

Proposition D.2.1. *For any $v \in \mathcal{S}_i$, we express its first i elements as $v_{1:i} = \left[v_{1:i-j},\ v_i\mathbf{1}^j\right]$, where the last $j \in [i]$ is the largest integer such that $v_{i-j+1} = v_{i-j+2} = \cdots = v_i$, and $\mathbf{1}^d \in \mathbb{R}^d$ denotes the vector containing 1 in all its elements. It holds that*

$$P_{\mathcal{S}_{i+1}}(v) = \begin{cases} v & \text{if } v_i \geq v_{i+1} \\ \left[v_{1:i-j},\ \left(v_i + \frac{v_{i+1}-v_i}{j+1}\right)\mathbf{1}^{j+1},\ v_{i+2:d}\right] & \text{if } v_i < v_{i+1} \text{ and} \\ & v_{i-j} \geq v_i + \frac{v_{i+1}-v_i}{j+1} \\ P_{\mathcal{S}_{i+1}}\left(\left[v_{1:i-j},\ v_{i-j}\mathbf{1}^j,\ v_{i+1} - (v_{i-j} - v_i)\, j,\ v_{i+2:d}\right]\right) & \text{otherwise.} \end{cases}$$

Proof. The first case is straightforward. In the following we prove the remaining two. In both cases it will be useful to recall that the projection operator $P_{\mathcal{C}}$ on any convex set $\mathcal{C}$ is characterized as

$$x = P_{\mathcal{C}}(y) \iff \langle y - x,\ z - x\rangle \leq 0,\ \forall z \in \mathcal{C}. \tag{D.4}$$

To prove the second case, we use property (D.4) and apply simple algebraic transforma-

tions to obtain, for all $z \in \mathcal{S}_{i+1}$, that

$$\left\langle v - P_{\mathcal{S}_{i+1}}(v),\, z - P_{\mathcal{S}_{i+1}}(v) \right\rangle = \frac{v_{i+1} - v_i}{j+1} \left(j z_{i+1} - \|z_{i-j+1:i}\|_1 \right) \leq 0.$$

Finally we prove the third case. We want to show that if $x = P_{\mathcal{S}_{i+1}}(v)$ then

$$x = P_{\mathcal{S}_{i+1}} \left(\left[v_{1:i-j},\, v_{i-j}\mathbf{1}^j,\, v_{i+1} - (v_{i-j} - v_i)\, j,\, v_{i+2:d} \right] \right).$$

By using property (D.4), the last equation is equivalent to the statement that if

$$\langle v - x,\, z - x \rangle \leq 0,\ \forall z \in \mathcal{S}_{i+1},\ \text{ then} \tag{D.5}$$

$$\left\langle \left[v_{1:i-j},\, v_{i-j}\mathbf{1}^j,\, v_{i+1} - (v_{i-j} - v_i)\, j,\, v_{i+2:d} \right] - x,\, z - x \right\rangle \leq 0,\ \forall z \in \mathcal{S}_{i+1}. \tag{D.6}$$

A way to show that it holds true is to prove that the term in the left hand side of (D.6) is upper bounded by the corresponding term in (D.5). That is, for every $z \in \mathcal{S}_{i+1}$, we want to show that

$$\left\langle \left[v_{1:i-j},\, v_{i-j}\mathbf{1}^j,\, v_{i+1} - (v_{i-j} - v_i)\, j,\, v_{i+2:d} \right] - v, z - x \right\rangle \leq 0.$$

A direct computation yields the equivalent inequality

$$(v_{i-j} - v_i) \left(j x_{i+1} - \|x_{i-j+1:i}\|_1 + \|z_{i-j+1:i}\|_1 - j z_{i+1} \right) \leq 0. \tag{D.7}$$

Since $x = P_{\mathcal{S}_{i+1}}(v)$, $v_{i-j+1} = v_{i-j+2} = \cdots = v_i$ and $v_{i+1} > v_i$, then $x_{i-j+1} = x_{i-j+2} = \cdots = x_{i+1}$. Consequently, the left hand side of inequality (D.7) is equivalent to

$$(v_{i-j} - v_i) \left(\|z_{i-j+1:i}\|_1 - j z_{i+1} \right) \leq 0.$$

Note that the first factor is negative and the second is positive because z and v are in $\mathcal{S}_{i+1}$. The result follows. □

Algorithm D.1 summarizes our method to compute the approximated projection operator onto the set $\mathcal{S}$, based on Proposition D.2.1.

Bibliography

[1] J.A. Adams. Historical Review and Appraisal of Research on the Learning, Retention, and Transfer of Human Motor Skills. *Psychological Bulletin*, 101(1):41, 1987.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145, January 2005.

[3] M. Aharon, M. Elad, and A. Bruckstein. K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Technology and Society Magazine Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[4] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering Shared Structures in Multiclass Classification. *International Conference on Machine Learning (ICML)*, 24:17–24, 2007.

[5] A. Anandkumar, R. Ge, D. Hsu, S.M. Kakade, and M. Telgarsky. Tensor Decompositions for Learning Latent Variable Models. *arXiv preprint arXiv:1210.7559*, pages 1–55, 2012.

[6] R.K. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[7] T.J. Andrews and M.P. Ewbank. Distinct Representations for Facial Identity and Changeable Aspects of Faces in the Human Temporal Lobe. *NeuroImage*, 23(3):905–13, November 2004.

[8] A. Argyriou, T. Evgeniou, and M. Pontil. Convex Multi-Task Feature Learning. *Machine Learning*, 73(3):243–272, January 2008.

[9] A. Argyriou, R. Foygel, N. Srebro, et al. Sparse Prediction with the k-Support Norm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1466–1474, 2012.

[10] A. Argyriou, A. Maurer, and M. Pontil. An Algorithm for Transfer Learning in a Heterogeneous Environment. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 71–85, 2008.

[11] A. Argyriou, C.A. Micchelli, and M. Pontil. On Spectral Learning. *Journal of Machine Learning Research*, pages 935–953, 2010.

[12] A. Argyriou, C.A. Micchelli, M. Pontil, L. Shen, and Y. Xu. Efficient First Order Methods for Linear Composite Regularizers. *arXiv preprint arXiv:1104.1436*, 2011.

[13] M.S.H. Aung, B. Romera-Paredes, A. Singh, S. Lim, N. Kanakam, A.C.D.C Williams, and N. Bianchi-Berthouze. Getting rid of pain-related behaviour to improve social and self perception: a technology-based perspective. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on*, pages 1–4. IEEE, 2013.

[14] T. Aviv. Reduced-Rank Regression for the Multivariate Linear Model. *Journal of Multivariate Analysis 5*, 264:248–264, 1975.

[15] B.W. Bader, R.A. Harshman, and T.G. Kolda. Temporal Analysis of Social Networks Using Three-Way Dedicom. *Sandia National Laboratories TR SAND2006-2161*, 119, 2006.

[16] B.W. Bader, R.A. Harshman, and T.G. Kolda. Temporal Analysis of Semantic Graphs Using ASALSAN. *IEEE International Conference on Data Mining (ICDM)*, 17:33–42, October 2007.

[17] B.W. Bader and T.G. Kolda. Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):635–653, 2006.

[18] P.L. Bartlett and S. Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[19] J. Baxter. A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

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

[21] S. Ben-David and R. Schuller. Exploiting Task Relatedness for Multiple Task

Learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

[22] A.L. Benton and M.W. Van Allen. Impairment in Facial Recognition in Patients with Cerebral Disease. *Cortex*, 4(4):344–IN1, 1968.

[23] D.P. Bertsekas and J.N. Tsitsiklis. Parallel and Distributed Computation: Numerical Methods, 1989.

[24] B.D. Blume, J.K. Ford, T.T. Baldwin, and J.L. Huang. Transfer of Training: A Meta-Analytic Review. *Journal of Management*, 36(4):1065–1105, December 2009.

[25] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[26] S. Boyd, L. Xiao, and A. Mutapcic. Subgradient Methods. *Lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004, 2003.

[27] J.K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel Coordinate Descent for L1-Regularized Loss Minimization. *CoRR*, abs/1105.5379, 2011.

[28] L. Breiman and J.H. Friedman. Predicting Multivariate Responses in Multiple Linear Regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):3–54, 1997.

[29] P. Bühlmann and S. Van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011.

[30] A.J. Calder, A.M. Burton, P. Miller, A.W. Young, and S. Akamatsu. A Principal Component Analysis of Facial Expressions. *Vision Research*, 41(9):1179–1208, 2001.

[31] A.J. Calder and A.W. Young. Understanding the Recognition of Facial Identity and Facial Expression. *Nature reviews. Neuroscience*, 6(8):641–51, August 2005.

[32] E.J. Candès and B. Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[33] R. Caruana. Algorithms and Applications for Multitask Learning. *International Conference on Machine Learning (ICML)*, 13:87–95, 1996.

[34] R. Caruana. Multitask Learning*. *Machine Learning*, 75(28):41–75, 1997.

[35] R. Caruana. *Multitask Learning*. PhD thesis, 1997.

[36] E. Challis and D. Barber. Affine Independent Variational Inference. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2195–2203, 2012.

[37] V. Chandrasekaran, B. Recht, P.A. Parrilo, and A.S. Willsky. The Convex Geometry of Linear Inverse Problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[38] J. Chen, J. Liu, and J. Ye. Learning Incoherent Sparse and Low-Rank Patterns from Multiple Tasks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 16:1179, 2010.

[39] J. Chen, L. Tang, J. Liu, and J. Ye. A Convex Formulation for Learning Shared Structures from Multiple Tasks. *International Conference on Machine Learning (ICML)*, 26:1–8, 2009.

[40] J. Chen, J. Zhou, and J. Ye. Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning. *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, 19:42–50, 2011.

[41] X. Chen, J. He, R. Lawrence, and J.G. Carbonell. Adaptive Multi-task Sparse Learning with an Application to fMRI Study. *SIAM International Conference on Data Mining (SDM)*, 2012.

[42] A. Cichocki, D. Mandic, A.H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer. Tensor Decompositions for Signal Processing Applications. *IEEE Signal Processing Magazine*, 2014.

[43] T. Cohn and L. Specia. Modelling Annotator Bias with Multi-Task Gaussian Processes: An Application to Machine Translation Quality Estimation. In *Annual Meeting of the Association for Computational Linguistics*, volume 51. Citeseer, 2013.

[44] P.L. Combettes and J.C. Pesquet. Proximal Splitting Methods in Signal Processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[45] P.L. Combettes and V.R. Wajs. Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, January 2005.

[46] P. Comon. Tensors: A brief survey. *IEEE Signal Processing Magazine*, 2014.

[47] T. Croonenborghs, K. Driessens, and M. Bruynooghe. Learning Relational Options for Inductive Transfer in Relational Reinforcement Learning. *Conference on Inductive Logic Programming (ILP)*, 17:88–97, 2007.

[48] S. Dasgupta and A. Gupta. An Elementary Proof of a Theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, January 2003.

[49] H. Daumé III. Bayesian Multitask Learning with Latent Hierarchies. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 25:135–142, 2009.

[50] B. de Gelder, I. Frissen, J. Barton, and N. Hadjikhani. A Modulatory Role for Facial Expressions in Prosopagnosia. *National Academy of Sciences of the United States of America*, 100(22):13105–10, October 2003.

[51] L. De Lathauwer, B. De Moor, and J. Vandewalle. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253, 2000.

[52] V. De Silva and L.H. Lim. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[53] C.B. Do and A.Y. Ng. Transfer Learning for Text Classification. *Advances in Neural Information Processing Systems (NIPS)*, 18(3):299–306, 2006.

[54] D.L. Donoho and V. Stodden. When Does Non-Negative Matrix Factorization Give a Correct Decomposition into arts? In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, 2003.

[55] J. Douglas and H.H. Rachford. On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables. *Transactions of the American Mathematical Society*, 82(2):pp. 421–439, 1956.

[56] J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[57] P. Ekman and W.V. Friesen. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Palo Alto, 1978.

[58] T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning Multiple Tasks with Kernel Methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[59] T. Evgeniou and M. Pontil. Regularized Multi-Task Learning. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 109–117, 2004.

[60] T. Evgeniou, M. Pontil, and O. Toubia. A Convex Optimization Approach to Modeling Consumer Heterogeneity in Conjoint Estimation. *Marketing Science*, pages 805–818, 2007.

[61] M. Fazel, H. Hindi, and S.P. Boyd. A Rank Minimization Heuristic with Applica-

tion to Minimum Order System Approximation. *American Control Conference*, 6(2):4734–4739, 2001.

[62] C.J. Fox and J.J.S. Barton. It Doesn't Matter How You Feel. The Facial Identity Aftereffect Is Invariant to Changes in Facial Expression. *Journal of Vision*, 8:1–13, 2008.

[63] C.J. Fox, S.Y. Moon, G. Iaria, and J.J.S. Barton. The Correlates of Subjective Perception of Identity and Expression in the Face Network: An fMRI Adaptation Study. *NeuroImage*, 44(2):569–80, January 2009.

[64] S. Gandy, B. Recht, and I. Yamada. Tensor Completion and low-n-rank Tensor Recovery Via Convex Optimization. *Inverse Problems*, 27(2):025010, 2011.

[65] G.H. Golub and C.F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[66] K.M. Gothard, F.P. Battaglia, C.A. Erickson, K.M. Spitler, and D.G. Amaral. Neural Responses to Facial Expression and Face Identity in the Monkey Amygdala. *Journal of Neurophysiology*, 97(2):1671–83, February 2007.

[67] L. Grasedyck. Hierarchical Singular Value Decomposition of Tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[68] L. Grasedyck, D. Kressner, and C. Tobler. A Literature Survey of Low-Rank Tensor Approximation Techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[69] H.J. Griffin, M.S.H. Aung, B. Romera-Paredes, C. McLoughlin, G. McKeown, W. Curran, and N. Bianchi-Berthouze. Laughter type recognition from whole body motion. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference ongriffin2013laughter*, pages 349–355. IEEE, 2013.

[70] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*, volume 42. Springer, 2012.

[71] W. Hackbusch and S. Kühn. A New Scheme for the Tensor Representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, October 2009.

[72] J.V. Haxby, E.A. Hoffman, and M.I. Gobbini. The Distributed Human Neural System for Face Perception. *Trends in Cognitive Sciences*, 4(6):223–233, June 2000.

[73] C.J. Hillar and L.H. Lim. Most Tensor Problems Are NP-Hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[74] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms: Part 2*, volume 305. Springer, 1993.

[75] F.L. Hitchcock. Multiple Invariants and Generalized Rank of a p-way Matrix or Tensor. *Journal of Mathematical Physics*, 7(1):39–79, 1927.

[76] F.L. Hitchcock. *The Expression of a Tensor or a Polyadic as a Sum of Products*. Institute of Technology, 1927.

[77] E.A. Hoffman and J.V. Haxby. Distinct Representations of Eye Gaze and Identity in the Distributed Human Neural System for Face Perception. *Nature Neuroscience*, 3(1):80–4, January 2000.

[78] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 2005.

[79] P.O. Hoyer. Non-Negative Matrix Factorization with Sparseness Constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[80] R. Hübener, V. Nebendahl, and W. Dür. Concatenated Tensor Network States. *New Journal of Physics*, 12(2):025004, 2010.

[81] K. Humphreys, G. Avidan, and M. Behrmann. A Detailed Investigation of Facial Expression Processing in Congenital Prosopagnosia as Compared to Acquired Prosopagnosia. *Experimental Brain Research*, 176(2):356–73, January 2007.

[82] S.J. Hwang, K. Grauman, and F. Sha. Learning a Tree of Metrics with Disjoint Visual Features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 621–629, 2011.

[83] A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural networks: the Official Journal of the International Neural Network Society*, 13(4-5):411–30, 2000.

[84] L. Jacob, F. Bach, J.P. Vert, et al. Clustered Multi-Task Learning: A Convex Formulation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 745–752, 2008.

[85] P. Jain, R. Meka, and I.S. Dhillon. Guaranteed Rank Minimization via Singular Value Projection. In *Advances in Neural Information Processing Systems (NIPS)*, volume 23, pages 937–945, 2010.

[86] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A Dirty Model for Multi-task Learning. *Advances in Neural Information Processing Systems (NIPS)*, 23:964–972, 2010.

[87] E. Jaquet and G. Rhodes. Face Aftereffects Indicate Dissociable, But Not Dis-

tinct, Coding of Male and Female Faces. *Journal of Experimental Psychology. Human Perception and Performance*, 34(1):101–12, February 2008.

[88] T. Jebara. Multitask Sparsity via Maximum Entropy Discrimination. *Journal of Machine Learning Research*, 12:75–110, 2011.

[89] R. Jenatton, J.Y. Audibert, and F. Bach. Structured Variable Selection with Sparsity-Inducing Norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

[90] R. Jenatton, N. Le Roux, A. Bordes, G. Obozinski, et al. A Latent Factor Model for Highly Multi-Relational Data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3176–3184, 2012.

[91] R. Jenatton, G. Obozinski, F. Bach, I. Fr, and I. Willow Project-team. Proximal Methods for Hierarchical Sparse Coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

[92] S. Ji and J. Ye. An Accelerated Gradient Method for Trace Norm Minimization. *International Conference on Machine Learning (ICML)*, 26:457–464, 2009.

[93] F. Jin and S. Sun. Neural Network Multitask Learning for Traffic Flow Forecasting. *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1897–1901, June 2008.

[94] Z. Kang, K. Grauman, and F. Sha. Learning with Whom to Share in Multi-task Feature Learning. *International Conference on Machine Learning (ICML)*, 28:521–528, 2011.

[95] N. Kanwisher, J. McDermott, and M.M. Chun. The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face Perception. *The Journal of Neuroscience*, 17(11):4302–4311, 1997.

[96] A. Kapteyn, H. Neudecker, and T. Wansbeek. An Approach to n-mode Components Analysis. *Psychometrika*, 51(2):269–275, 1986.

[97] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: n-dimensional Tensor Factorization for Context-Aware Collaborative Filtering. In *ACM Recommender Systems (RecSys)*, volume 4, pages 79–86. ACM, 2010.

[98] J. Khan, J.S. Wei, M. Ringnér, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P.S. Meltzer. Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks. *Nature Medicine*, 7(6):673–9, June 2001.

[99] S. Kim and E.P. Xing. Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity. In *International Conference on Machine Learning (ICML)*, volume 27, pages 543–550, 2010.

[100] T.G. Kolda and B.W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455, 2009.

[101] V. Koltchinskii and D. Panchenko. Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers. *The Annals of Statistics*, 30:1–50, 2002.

[102] Y. Koren. The BellKor Solution to the Netflix Grand Prize. *Netflix prize documentation*, (August):1–10, 2009.

[103] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-Rank Tensor Completion by Riemannian Optimization. *BIT Numerical Mathematics*, pages 1–22, 2013.

[104] P.M. Kroonenberg. *Applied Multiway Data Analysis*, volume 702. John Wiley & Sons, 2008.

[105] P.M. Kroonenberg and J. de Leeuw. Principal Component Analysis of Three-mode Data by Means of Alternating Least Squares Algorithms. *Psychometrika*, 45(1), 1980.

[106] J.B. Kruskal. Three-Way Arrays: Rank and Uniqueness of Trilinear Decompositions, with Application to Arithmetic Complexity and Statistics. *Linear Algebra and Its applications*, 18(2):95–138, 1977.

[107] A. Kumar and H. Daumé III. Learning Task Grouping and Overlap in Multi-Task Learning. *International Conference on Machine Learning (ICML)*, 29, 2012.

[108] J.M. Landsberg. *Tensors: Geometry and Applications*, volume 128. American Mathematical Society, 2012.

[109] D. Lathauwer and D. Moor. On the Best Rank-1 and Rank-(R1 ,R2 ,. . .,RN) Approximation of Higher-Order Tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[110] N.D. Lawrence and J.C. Platt. Learning to Learn with the Informative Vector Machine. *International Conference on Machine Learning (ICML)*, 21:65–72, 2004.

[111] M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer, 1991.

[112] D.D. Lee and H.S. Seung. Learning the Parts of Objects by Non-Negative Matrix Factorization. *Nature*, 401(6755):788–791, 1999.

[113] Y. Li, Y. Du, and X. Lin. Kernel-Based Multifactor Analysis for Image Synthesis and Recognition. *IEEE International Conference on Computer Vision (ICCV)*, 1:114–119, 2005.

[114] J.J. Lim, R. Salakhutdinov, and A. Torralba. Transfer Learning by Borrowing Examples for Multiclass Object Detection. *Advances in Neural Information Processing Systems (NIPS)*, 24:118–126, 2011.

[115] Y.J. Lim and Y.W. Teh. Variational Bayesian Approach to Movie Rating Prediction. In *KDD Cup and Workshop*, volume 7, pages 15–21. Citeseer, 2007.

[116] H. Liu, M. Palatucci, and J. Zhang. Blockwise Coordinate Descent Procedures for the Multi-Task Lasso, with Applications to Neural Semantic Basis Discovery. In *International Conference on Machine Learning (ICML)*, volume 26, pages 649–656. ACM, 2009.

[117] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor Completion for Estimating Missing Values in Visual Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[118] Y. Liu and F. Shang. An Efficient Matrix Factorization Method for Tensor Completion. *IEEE Signal Processing Letters*, 20(4):307–310, 2013.

[119] C.F.V. Loan. The Ubiquitous Kronecker Product. *Journal of Computational and Applied Mathematics*, 123(1):85–100, 2000.

[120] K. Lounici, M. Pontil, A.B. Tsybakov, and S. van de Geer. Taking Advantage of Sparsity in Multi-Task Learning. In *Conference on Learning Theory (COLT)*, volume 22, pages 73–82, 2009.

[121] K. Lounici, M. Pontil, S. Van de Geer, and A.B. Tsybakov. Oracle Inequalities and Optimal Inference Under Group Sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

[122] H. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. A Survey of Multilinear Subspace Learning for Tensor Data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[123] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, and I. Matthews. PAINFUL DATA: The UNBC-McMaster Shoulder Pain Expression Archive Database. In *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG)*, pages 57–64. IEEE, 2011.

[124] M. Lyons and S. Akamatsu. Coding Facial Expressions with Gabor Wavelets. *Computer*, pages 200–205, 1998.

[125] J. Mairal, F. Bach, and J. Ponce. Task-Driven Dictionary Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:791–804, 2012.

[126] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, et al. Supervised Dictionary Learning. *arXiv preprint arXiv:0809.3083*, 2008.

[127] A. Maurer. Bounds for Linear Multi-Task Learning. *The Journal of Machine Learning Research*, 7:117–139, 2006.

[128] A. Maurer. Transfer Bounds for Linear Feature Learning. *Machine Learning*, 75(3):327–350, 2009.

[129] A. Maurer and M. Pontil. A Uniform Lower Error Bound for Half-Space Learning. In *Algorithmic Learning Theory*, pages 70–78. Springer, 2008.

[130] A. Maurer and M. Pontil. K-Dimensional Coding Schemes in Hilbert Spaces. *IEEE Transactions on Information Theory*, 56:5839–5846, 2010.

[131] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*, pages 343–351, 2013.

[132] A. Maurer, M. Pontil, and B. Romera-Paredes. An inequality with applications to structured sparsity and multitask dictionary learning. *Conference on Learning Theory (COLT)*, 2014.

[133] G. McCarthy, A. Puce, J.C. Gore, and T. Allison. Face-Specific Processing in the Human Fusiform Gyrus. *Journal of Cognitive Neuroscience*, 9(5):605–610, 1997.

[134] C. McDiarmid. *Probabilistic Methods of Algorithmic Discrete Mathematics*, chapter Concentration, pages 195–248. Springer, 1998.

[135] A.M. McDonald, M. Pontil, and D. Stamos. New Perspectives on k-support and Cluster Norms. *arXiv preprint arXiv:1403.1481*, 2014.

[136] H. Meng, B. Romera-Paredes, and N. Bianchi-Berthouze. Emotion recognition by two view svm_2k classifier on dynamic facial expression features. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 854–859. IEEE, 2011.

[137] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. Fixed-Rank Matrix Factorizations and Riemannian Low-Rank Optimization. *Computational Statistics*, pages 1–31, 2014.

[138] K. Mohan and M. Fazel. Iterative Reweighted Least Squares for Matrix

Rank Minimization. *Conference on Communication, Control and Computing*, 1(X):653–661, 2010.

[139] M. Mørup. Applications of Tensor (Multiway Array) Factorizations and Decompositions in Data Mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):24–40, 2011.

[140] I. Mpiperis, S. Malassiotis, and M.G. Strintzis. Bilinear Models for 3-D Face and Facial Expression Recognition. *IEEE Transactions on Information Forensics and Security*, 3(3):498–511, September 2008.

[141] C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. *arXiv preprint arXiv:1307.5870*, pages 1–22, 2013.

[142] S. Nakajima and M. Sugiyama. Theoretical Analysis of Bayesian Matrix Factorization. *The Journal of Machine Learning Research*, 12:2583–2648, 2011.

[143] S. Negahban and M.J. Wainwright. Joint Support Recovery Under High-Dimensional Scaling: Benefits and Perils of l1,°-regularization. *Advances in Neural Information Processing Systems (NIPS)*, 21:1161–1168, 2008.

[144] Y. Nesterov. Smooth Minimization of Non-Smooth Functions. *Journal Mathematical Programming: Series A and B*, 103(1):127–152, 2005.

[145] A.Y. Ng. Feature Selection, L 1 vs. L 2 Regularization, and Rotational Invariance. *International Conference on Machine Learning (ICML)*, 21:78, 2004.

[146] T.T. Ngo and Y. Saad. Scaled Gradients on Grassmann Manifolds for Matrix Completion. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1421–1429, 2012.

[147] M. Nickel and V. Tresp. Tensor Factorization for Multi-relational Learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 617–621. Springer, 2013.

[148] M. Nickel, V. Tresp, and H.P. Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In *International Conference on Machine Learning (ICML)*, pages 809–816, 2011.

[149] M. Nickel, V. Tresp, and H.P. Kriegel. Factorizing YAGO: Scalable Machine Learning for Linked Data. In *Conference on World Wide Web*, volume 21, pages 271–280. ACM, 2012.

[150] G. Obozinski, B. Taskar, and M. Jordan. Multi-Task Feature Selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[151] G. Obozinski, B. Taskar, and M.I. Jordan. Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems. *Statistics and Computing*, 20(2):231–252, January 2009.

[152] V. Ojansivu and J. Heikkilä. A Method for Blur and Affine Invariant Object Recognition Using Phase-Only Bispectrum. In *ICIAR*, pages 527–536, 2008.

[153] B.A. Olshausen and D.J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[154] I.V. Oseledets. A New Tensor Decomposition. *Doklady Mathematics*, 80(1):495–496, August 2009.

[155] I.V. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317, 2011.

[156] M. Palatucci, D. Pomerleau, G.E. Hinton, and T.M. Mitchell. Zero-Shot Learning with Semantic Output Codes. In *Advances in Neural Information Processing Systems (NIPS)*, volume 3, pages 5–2, 2009.

[157] S.J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Technology and Society Magazine Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[158] E. Papalexakis, U. Kang, C. Faloutsos, N. Sidiropoulos, and A. Harpale. Large Scale Tensor Decompositions: Algorithmic Developments and Applications. *IEEE Data Engineering Bulletin*, pages 59–66, 2013.

[159] E.E. Papalexakis, C. Faloutsos, and N.D. Sidiropoulos. Parcube: Sparse Parallelizable Tensor Decompositions. In *Machine Learning and Knowledge Discovery in Databases*, pages 521–536. Springer, 2012.

[160] K. Pearson. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[161] K.M. Prkachin and P.E. Solomon. The Structure, Reliability and Validity of Pain Expression: Evidence from Patients with Shoulder Pain. *Pain*, 139(2):267–274, 2008.

[162] A. Quattoni, M. Collins, and T. Darrell. Transfer Learning for Image Classification with Sparse Prototype Representations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[163] H. Rauhut, R. Schneider, and Z. Stojanac. Low Rank Tensor Recovery via Iter-

ative Hard Thresholding. In *Conference on Sampling Theory and Applications*, volume 10, 2013.

[164] V.C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, and R.B. Rao. Bayesian Multiple Instance Learning: Automatic Feature Selection and Inductive Transfer. In *International Conference on Machine learning (ICML)*, volume 25, pages 808–815. ACM, 2008.

[165] B. Recht. A Simpler Approach to Matrix Completion. *Matrix*, 12(October):13, 2009.

[166] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(3):471–501, 2010.

[167] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast Context-Aware Recommendations with Factorization Machines. *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 34:635, 2011.

[168] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting Unrelated Tasks in Multi-Task Learning. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, XX:951–959, 2012.

[169] B. Romera-Paredes, A. Argyriou, A.C.D.C. Williams, N. Bianchi-Berthouze, and M. Pontil. Automatic recognition of facial expressions. *14th World Congress on Pain (IASP)*, 2012.

[170] B. Romera-Paredes, M.S.H. Aung, and N. Bianchi-Berthouze. A one-vs-one classifier ensemble with majority voting for activity recognition. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, (ESANN)*, 2013.

[171] B. Romera-Paredes, M.S.H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*, pages 1444–1452, 2013.

[172] B. Romera-Paredes, M.S.H. Aung, M. Pontil, N. Bianchi-Berthouze, A.C.D.C. de Williams, and P. Watson. Transfer learning to account for idiosyncrasy in face and body expressions. In *10th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2013.

[173] B. Romera-Paredes, N. Bianchi-Berthouze, and M. Pontil. Leveraging different transfer learning assumptions: Shared features, hierarchical and semi-supervised. *Challenges in Learning Hierarchical Models, NIPS Workshop*, 2011.

[174] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2967–2975, 2013.

[175] B. Romera-Paredes, C. Zhang, and Z. Zhang. Facial expression tracking from head-mounted, partially observing cameras. *IEEE International Conference on Multimedia & Expo, IEEE ICME*, 2014.

[176] U. Rückert and S. Kramer. Kernel-Based Inductive Transfer. In *Machine Learning and Knowledge Discovery in Databases*, pages 220–233. Springer, 2008.

[177] J. Schneider and Y. Zhang. Learning Multiple Tasks with a Sparse Matrix-Normal Penalty. *Advances in Neural Information Processing Systems (NIPS)*, 23:2550–2558, 2010.

[178] M. Seeger and G. Bouchard. Fast Variational Bayesian Inference for Non-Conjugate Matrix Factorization Models. In *International Conference on Artificial Intelligence and Statistics*, pages 1012–1018, 2012.

[179] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[180] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. TFMAP: Optimizing MAP for top-n Context-Aware Recommendation. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 155–164. ACM, 2012.

[181] N.Z. Shor, K.C. Kiwiel, and A. Ruszcaynski. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag New York, Inc., 1985.

[182] N.D. Sidiropoulos and R. Bro. On the Uniqueness of Multilinear Decomposition of N-way Arrays. *Journal of Chemometrics*, 14(3):229–239, 2000.

[183] M. Signoretto, L. De Lathauwer, and J.A.K. Suykens. Nuclear Norms for Tensors and Their Use for Convex Multilinear Estimation. *Submitted to Linear Algebra and Its Applications*, 43, 2010.

[184] M. Signoretto, Q.T. Dinh, L. De Lathauwer, and J.A.K. Suykens. Learning with Tensors: A Framework Based on Convex Optimization and Spectral Regularization. *Machine Learning*, pages 1–49, 2013.

[185] M. Signoretto and J.A.K. Suykens. Multilinear Spectral Regularization for Kernel-based Multitask Learning. *Data Engineering*, 22(10):1345–1359, 2013.

[186] M. Signoretto, R. Van de Plas, B. De Moor, and J.A.K. Suykens. Tensor Versus

Matrix Completion: A Comparison With Application to Spectral Data. *IEEE Signal Processing Letters*, 18(7):403–406, July 2011.

[187] D.L. Silver. The Parallel Transfer of Task Knowledge Using Dynamic Learning Rates Based on a Measure of Relatedness. *Connection Science*, 8(2):277–294, 1996.

[188] D. Slepian. The One-Sided Barrier Problem for Gaussian Noise. Technical report, Bell System Tech. J 41:463–501, 1962.

[189] E. Spyropoulou, T. De Bie, and Mario. Boley. Mining interesting patterns in multi-relational data with n-ary relationships. In *Discovery Science*, pages 217–232. Springer, 2013.

[190] N. Srebro, J.D.M. Rennie, and T. Jaakkola. Maximum-Margin Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 1329–1336, 2004.

[191] N. Srebro and A. Shraibman. Rank, Trace-Norm and Max-Norm. In *Learning Theory*, pages 545–560. Springer, 2005.

[192] A. Stegeman and P. Comon. Subtracting a Best Rank-1 Approximation May Increase Tensor Rank. *Linear Algebra and Its Applications*, 433(7):1276–1300, December 2010.

[193] S. Sun. Multitask Learning for EEG-Based Biometrics. *Conference on Pattern Recognition*, 19:1–4, December 2008.

[194] J.B. Tenenbaum and W.T. Freeman. Separating Style and Content with Bilinear Models. *Neural computation*, 12(6):1247–83, 2000.

[195] S. Thrun and L. Pratt. *Learning to Learn*. Springer, 1998.

[196] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[197] M.K. Titsias and M. Lázaro-Gredilla. Spike and Slab Variational Inference for Multi-Task and Multiple Kernel Learning. *Advances in Neural Information Processing Systems (NIPS)*, 24:2339–2347, 2011.

[198] R. Tomioka, K. Hayashi, H. Kashima, and J.S.T. Presto. Estimation of Low-Rank Tensors Via Convex Optimization. *arXiv preprint arXiv:1010.0789*, pages 1–19, 2011.

[199] R. Tomioka and T. Suzuki. Convex Tensor Decomposition Via Structured Schatten Norm Regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1331–1339, 2013.

[200] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical Performance of Convex Tensor Decomposition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 972–980, 2011.

[201] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2(v):762–769, 2004.

[202] L.R. Tucker. Some Mathematical Notes on Three-Mode Factor Analysis. *Psychometrika*, 31(3), 1966.

[203] G. Tur. Multitask Learning for Spoken Language Understanding. In *Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–I. IEEE, 2006.

[204] B.A. Turlach, W.N. Venables, and S.J. Wright. Simultaneous Variable Selection. *Technometrics*, 47(3):349–363, August 2005.

[205] B. Vandereycken. Low-Rank Matrix Completion by Riemannian Optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[206] B. Vargas-Govea, G. González-Serna, and R. Ponce-Medellın. Effects of Relevant Contextual Features in the Performance of a Restaurant Recommender System. *ACM Recommender Systems (RecSys)*, 11, 2011.

[207] M.A.O. Vasilescu. Human Motion Signatures: Analysis, Synthesis, Recognition. *Object Recognition Supported by User Interaction for Service Robots*, 3:456–460, 2002.

[208] M.A.O. Vasilescu and D. Terzopoulos. Multilinear Analysis of Image Ensembles: TensorFaces. *New York*, 2350(2-3):447–460, 2002.

[209] M.A.O. Vasilescu and D. Terzopoulos. Multilinear Image Analysis for Facial Recognition. *Conference on Pattern Recognition (ICPR)*, pages 511–514, 2002.

[210] M.A.O. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear Image-Based Rendering. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 336–342. ACM, 2004.

[211] M.A.O. Vasilescu and D. Terzopoulos. Multilinear Independent Components Analysis. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, (June):547–553, 2005.

[212] P. Viola and M.J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[213] H. Wang and N. Ahuja. Facial Expression Decomposition. *IEEE International Conference on Computer Vision (ICCV)*, 9:958–965 vol.2, 2003.

[214] H. Wang and N. Ahuja. Compact Representation of Multidimensional Data Using Tensor Rank-One Decomposition. *Vectors*, 1:5, 2004.

[215] H. Wang, F. Nie, H. Huang, J. Yan, S. Kim, S.L. Risacher, A.J. Saykin, and L. Shen. High-Order Multi-Task Feature Learning to Identify Longitudinal Phenotypic Markers for Alzheimer's Disease Progression Prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1286–1294, 2012.

[216] J.S. Winston, R.N.A. Henson, M.R. Fine-Goulden, and R.J. Dolan. fMRI-Adaptation Reveals Dissociable Neural Representations of Identity and Expression in Face Perception. *Journal of Neurophysiology*, 92(3):1830–9, September 2004.

[217] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel Matrix Factorization for Low-Rank Tensor Completion. *arXiv preprint arXiv:1312.1254*, 2013.

[218] X. Yang, S. Kim, and E.P. Xing. Heterogeneous Multitask Learning with Joint Sparsity Constraints. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2151–2159, 2009.

[219] A.W. Young, K.H. Mcweeny, D.C. Hay, and A.W. Ellis. Matching Familiar and Unfamiliar Faces on Identity and Expression. *Psychological Research*, 48(2):63–68, 1986.

[220] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. *International Conference on Machine Learning (ICML)*, 22:1012–1019, 2005.

[221] S. Yu, J. Bi, and J. Ye. Probabilistic Interpretations and Extensions for a Family of 2D PCA-Style Algorithms. In *KDD Workshop on Data Mining Using Matrices and Tensors (DMMT)*, pages 1–7, 2008.

[222] M. Yuan and Y. Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[223] X.T. Yuan, X. Liu, and S. Yan. Visual Classification with Multitask Joint Sparse Representation. *IEEE Transactions on Image Processing*, 21(10):4349–4360, 2012.

[224] D. Zagrodny. The Cancellation Law for Inf-Convolution of Convex Functions. *Studia Mathematika*, 110(3):271–282, 1994.

[225] D. Zhou, L. Xiao, and M. Wu. Hierarchical Classification via Orthogonal Transfer. *International Conference on Machine Learning (ICML)*, 28:801–808, 2011.

[226] J. Zhou, J. Chen, and J. Ye. Clustered Multi-Task Learning Via Alternating Structure Optimization. *Advances in Neural Information Processing Systems (NIPS)*, 24:702–710, 2011.

[227] Y. Zhou, R. Jin, and S.C.H. Hoi. Exclusive Lasso for Multi-task Feature Selection. *Group*, 9(11817):988–995, 2010.

[228] M. Zinkevich, M. Weimer, A.J. Smola, and L. Li. Parallelized Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems (NIPS)*, volume 4, page 4, 2010.

[229] H. Zou and T. Hastie. Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.