Structured Sparse Learning on Graphs in High-Dimensional Data with Applications to NeuroImaging

Eugene Belilovsky

université PARIS-SACLAY

KU LEUVEN

CentraleSupélec

Thèse de
Doctorat de l'Université Paris-Saclay
& Doctor of Engineering Science, KU Leuven

École doctorale N°580
Sciences et Technologies de l'Information et de la Communication
Specialité: Traitement du signal et des images

Arenberg Doctoral School
Faculty of Engineering Science: Electrical Engineering

par Eugene Belilovsky

Structured Sparse Learning on Graphs in High-Dimensional Data with Applications to Neuroimaging

Thèse présentée et soutenue à Gif-sur-Yvette, le 2 mars 2018.

Composition du Jury:

M.	**Nikos Paragios**	Professeur Université Paris-Saclay & Inria	(Président du Jury)
M.	**Gabriel Peyré**	Directeur de Recherche ENS and CNRS	(Rapporteur)
M.	**Dimitri Van De Ville**	Professeur associé EPFL	(Rapporteur)
M.	**Gael Varoquaux**	Professeur INRIA Saclay	(Examinateur)
M.	**Jesse Davis**	Professeur KU Leuven	(Examinateur)
Mme.	**Tinne Tuytelaars**	Professor KU Leuven	(Examinateur)
M.	**Matthew Blaschko**	Professeur KU Leuven	(Directeur de thèse)

Acknowledgements

I would first like to express my gratitude towards my thesis advisor Prof. Matthew Blaschko for his continuous support and advice during my PhD. Your passion and expertise have constantly renewed my spirits and allowed me to grow and successfully complete this challenging endeavour.

I am very grateful to my reviewers Prof. Gabriel Peyré and Prof. Dimitri Van De Ville for their efforts in the reviews and useful comments and suggestions. I would also like to thank Gael Varoquaux, Prof. Tinne Tuytelaars, Prof. Jesse Davis and Prof. Nikos Paragios for their roles as examiners.

I would like to acknowledge the important contribution of my research collaborators in the works in this thesis. Particularly Gael Varoquaux who has taken a big role in co-advising all the work in this thesis. Andreas Argyriou for his great patience in introducing me to the topic of structured sparsity. I am very lucky to have been able to do several fellow PhD students including Kyle Kastner, Edouard Oyallon, Wacha Bounliphone, and Jamie Kiros.

I would also like to thank all the great people at CVN who made the years in France very memorable. Thank you Erwan, Dimitris, Grigorios, Puneet, Enzo, Evgenios, Siddhartha, Hariprasad , Mihir, Maxim, Alp, Stefan, Marie-Caroline, Stavros and Khue. And a special thanks to Natalia Leclercq for going above and beyond with our endless administrative paperwork. And equally my colleagues at Leuven: Xuanli, Yu-Hui, José, Xu, Amal, Maxim (again!), Rahaf, Bert, Klaas, and Jay.

I would also like to thank Richard Zemel and Raquel Urtasun for giving me the chance to spend a summer at the University of Toronto. I was lucky to get to know some of the great folks who made me feel very welcome in the lab including Eleni, Namdar, Kaustauv, and Min.

A special thank you to Maxim Berman, Amal Rannen, Edouard Oyallon and Bernard Fino for their help in translating the abstract to French.

I would like to thank Marie-Laure for her endless support. Finally I want to thank my parents whose choices opened so many doors in life for me.

Abstracts

Abstract — This dissertation presents novel structured sparse learning methods on graphs that address commonly found problems in the analysis of neuroimaging data as well as other high dimensional and few sample data.

The first part of the thesis focuses on developing and utilizing convex relaxations of discrete and combinatorial penalties. These are developed with the aim of learning an interpretable predictive linear model satisfying sparse and graph based constraints. In the context of neuroimaging these models can leverage implicit structured sparsity properties to learn predictive and interpretable models that can inform analysis of neuro-imaging data. In particular we study the problem of statistical estimation with a signal known to be sparse, spatially contiguous, and containing many highly correlated variables. We take inspiration from the k-support norm, which has been successfully applied to sparse prediction problems with correlated features, but lacks any explicit structural constraints commonly found in machine learning and image processing. We address this problem by incorporating a total variation penalty in the k-support framework. We introduce the (k, s) *support total variation norm* as the *tightest* convex relaxation of the intersection of a set of discrete sparsity and total variation penalties. We show that this norm leads to an intractable combinatorial graph optimization problem, which we prove to be NP-hard. We then introduce a tractable relaxation with approximation guarantees. We demonstrate the effectiveness of this penalty on classification in the low-sample regime with M/EEG neuroimaging data and fMRI data, as well as image recovery with synthetic and real data background subtracted image recovery tasks. We show that our method is particularly useful compared to existing methods in terms of accuracy, interpretability, and stability.

We consider structure discovery of undirected graphical models from observational data. We then consider the problem of learning the structure of graphical models with structured sparse constraints. Functional brain networks are well described and estimated from data with Gaussian Graphical Models (GGMs), e.g. using sparse inverse covariance estimators. In this thesis we make two contributions for estimating Gaussian Graphical Models under various onstraints. Our first contribution is to identify differences in GGMs known to have similar structure. We characterize the uncertainty of differences with confidence intervals obtained using a parametric distribution on parameters of a sparse estimator. Sparse penalties enable statistical guarantees and interpretable models even in high-dimensional and low-sample settings. Characterizing the distributions of sparse models is inherently challenging as the penalties produce a biased estimator. Recent work invokes the sparsity assumptions to effectively remove the bias from a sparse estimator such as the lasso. These distributions can be used to give confidence intervals on edges in GGMs, and by extension their differences. However, in the case of comparing GGMs, these estimators do not make use of any assumed joint structure among the GGMs. Inspired by priors from brain functional connectivity we derive the distribution of parameter differences under a joint penalty when parameters are

known to be sparse in the difference. This leads us to introduce the debiased multi-task fused lasso, whose distribution can be characterized in an efficient manner. We show how the debiased lasso and multi-task fused lasso can be used to obtain confidence intervals on edge differences in GGMs. We validate the techniques proposed on a set of synthetic examples as well as neuro-imaging dataset created for the study of autism.

Finally, we consider a novel approach to the structure discovery of undirected graphical models from observational data. Although, popular methods rely on estimating a penalized maximum likelihood of the precision matrix, in these approaches structure recovery is an indirect consequence of the data-fit term, the penalty can be difficult to adapt for domain-specific knowledge, and the inference is computationally demanding. By contrast, it may be easier to generate training samples of data that arise from graphs with the desired structure properties. We propose to leverage this latter source of information as training data to learn a function, parametrized by a neural network that maps empirical covariance matrices to estimated graph structures. Learning this function brings two benefits: it implicitly models the desired structure or sparsity properties to form suitable priors, and it can be tailored to the specific problem of edge structure discovery, rather than maximizing data likelihood. Applying this framework, we find our learnable graph-discovery method trained on synthetic data generalizes well: identifying relevant edges in both synthetic and real data, completely unknown at training time. We find that on genetics, brain imaging, and simulation data we obtain performance generally superior to analytical methods.

Keywords: structured sparsity, Gaussian Graphical Models, neuroimaging, deep learning.

Titre — Apprentissage de graphes structuré et parcimonieux dans des données de haute dimension avec applications à l'imagerie cerebrale

Résumé — Cette thèse présente de nouvelles méthodes d'apprentissage parcimonieux structuré sur des graphes qui abordent les problèmes couramment rencontrés dans l'analyse des neuroimages ainsi que ceux relatifs à des échantillons de petite taille dans des espaces de grande dimension.

La première partie de la thèse se concentre sur le développement et l'utilisation de relaxations convexes bien fondées de fonctions de coût discrètes et combinatoires. Ces relaxations sont développées dans le but d'apprendre un modèle linéaire prédictif interprétable avec des contraintes basées sur des graphes. Dans le contexte de la neuro-imagerie, ces modèles peuvent exploiter des propriétés implicites de parcimonie structurée pour apprendre des modèles prédictifs et interprétables pouvant faciliter l'analyse des images. En particulier, nous étudions le problème de l'estimation statistique d'un signal supposé creux, spatialement contigu et contenant de nombreuses variables fortement corrélées.

Nous nous inspirons de la norme à k-support introduite récemment, ayant été appliquée avec succès à des problèmes de prédiction parcimonieuse avec des caractéristiques fortement corrélées, mais qui ne couvre pas les contraintes structurelles explicites communément rencontrées dans l'apprentissage machine et le traitement d'image. Nous résolvons ce problème en intégrant une pénalité de variation totale dans le cadre de la norme à k-support. Nous introduisons la (k, s) *norme de totale variation à k-support* comme la relaxation convexe minimale de l'intersection d'un ensemble de parcimonie discrète et sous pénalités de totale variation. Nous montrons que cette norme conduit à un problème d'optimisation de graphe combinatoire intractable, que nous prouvons être d'ordre de complexité NP. Nous introduisons ensuite une relaxation tractable de ce problème avec des garanties d'approximation. Nous élaborons plusieurs stratégies d'optimisation de premier ordre pour l'estimation des paramètres statistiques avec la pénalité décrite. Nous démontrons l'efficacité de cette pénalité sur la classification dans le régime de faible données, la classification avec des données de neuroimagerie M / EEG, et la récupération d'image avec des tâches synthétiques et réelles de récupération d'image sans arrière-plan. Nous analysons exhaustivement l'application de notre pénalité à des tâches complexes d'identification de régions prédictives à partir de données cérébrales fMRI de à faible échantillons et haute dimension. Nous montrons que notre méthode est particulièrement efficace comparée aux méthodes existentes en terme de performance, d'interprétabilité, et de stabilité. montrons que notre méthode est particulièrement utile par rapport aux méthodes existantes en termes de précision, d'interprétabilité et de stabilité.

Dans la seconde partie de cette thèse, nous nous intéressons à la découverte de structure de modèles graphiques non dirigés à partir de données d'observation. Nous considérons en particulier le problème de l'apprentissage de la structure de modèles graphiques sous contraintes parcimonieuses. Les réseaux fonctionnels cérébraux sont bien décrits et estimés à partir de données avec des modèles graphiques gaussiens (MGG), e.g. avec des estimateurs de covariance inverse. Comparer la connectivité fonctionnelle entre des sujets de deux pop-

ulations appelle à comparer ces MGG estimés. Notre objectif est d'identifier les différences dans des MGG ayant une structure similaire. Nous caractérisons l'incertitude sur les différences avec des intervalles de confiance obtenus au moyen de distributions paramétriques sur les paramètres d'un estimateur parcimonieux. Les contraintes parcimonieuses mènent à des garanties statistiques et des modèles interprétables même dans le cas d'une dimension haute et d'un faible nombre de données. La caractérisation des distributions de modèles parcimonieux est rendue délicate par le biais introduit par les contraintes parcimonieuses. De récents travaux mettent en jeu des à-prioris parcimonieux pour débiaiser un estimateur tel que le lasso. Ces distributions peuvent être utilisées pour assigner des intervalles de confiance à des arêtes de MGG, et par extension à leurs différences. Néanmoins, dans le cas de la comparaison de MGGs, ces estimateurs ne font pas appel à un à-priori sur la structure des MGGs. Inspirés des à-prioris de connectivité fonctionnelle cérébrale, nous développons la distribution des différences entre les paramètres sous une pénalité jointe quand les paramètres sont parcimonieux en cette différence. Ceci nous mène à introduire un lasso débiaisé multitâche, dont la distribution peut être efficacement caractérisée. Nous montrons comment le lasso débiaisé et le lasso joint multi-tâche peuvent être utilisés pour obtenir des intervalles de confiance sur les différence entre arêtes dans les MGGs. Nous validons les techniques proposées sur un ensemble d'exemples synthétiques ainsi qu'un ensemble de données d'imagerie neuronale créé pour l'étude de l'autisme.

Enfin, nous considérons une approche nouvelle de découverte de structure de modèles graphiques non-dirigés à partir de données observées. Inférer des structures probables à partir d'un faible nombre d'exemples est une tâche complexe qui demande souvent l'introduction d'à-priors et de procédures d'inférence sophistiquées. Des méthodes populaires reposent sur une estimation par maximization de probabilité pénalisée de la matrice de précision. Cependant, dans ces approches, la détermination de la structure est une conséquence indirecte du terme d'adaptation à la donnée, la pénalité peut être difficile à adapter dans le cas d'une connaissance experte du domaine, et l'inférence est computationnellement exigeante. Par opposition, il peut être plus facile de générer des exemples d'entraînement qui pourraient découler de graphes avec les structures désirées. Nous proposons ici d'utiliser cette source additionnelle d'information comme donnée d'apprentissage pour apprendre une fonction, paramétrisée par un réseau de neurones qui associe des structures de graphes estimées à des matrices de covariance empiriques. L'apprentissage de cette fonction présente deux avantages: la structure désirée ou les propriétés parcimonieuses formant des à-prioris adaptés sont modélisés implicitement; et la fonction peut être adaptée au problème spécifique de découverte de structure, plutôt que de maximiser la probabilité des données. Appliquant ce principe, nous constatons que notre méthode de découverte de structure appris sur des données synthétiques se généralise bien, identifiant des arêtes appropriées dans des données synthétiques et des données réelles, non accessibles à l'entraînement. Nous notons une performance obtenue par des méthodes généralement supérieure à des méthodes analytiques classiques sur les données génétiques, d'imagerie cérébrales, ainsi que sur des données d'entraînement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decades the explosion of data collected has revolutionized many fields and created a host of new disciplines or sub-disciplines devoted to its analysis. Combined with the continually improved computing power this has permitted a host of new applications for obtaining insights and predictions using this data. In many disciplines the number of data samples collected has lead to revolutionary performance, with notably examples in speech recognition, image recognition, and machine translation. At the same time for other disciplines and domains the data explosion has seen the number of measurements for a given sample collected far outpace the number of samples being collected. For example in the fields of biology applications, neuro-imaging, finance, astronomy, text mining, and climate research [Bühlmann et al., 2014]. Indeed when considering many medical and biological applications, on each individual, more and more features are measured to a point that it usually far exceeds the number of observations [Rigollet, 2015]. This has led to a renewed thinking on classical statistical methods. Furthermore, in biological and medical applications the inference tasks that we might want to perform can often be more subtle than prediction problems in large scale classification tasks such as natural image or speech recognition. For example we might want to do variable selection or obtain reliable confidence intervals in cases with no available ground truth. Here the inference procedure and underlying model must be well understood in order for the results to be trusted.

We take as an example the field of fMRI analysis. A technology which has become heavily used since the 1990s that allows to "see" into the brain in order to understand its function. This involves a process of scanning the user that is rather costly and time consuming for a specific task of interest. At the same time the number of individually discern-able brain regions can be large, with millions of voxels resulting from modern devices and pre-processing techniques. Furthermore, there is a great deal of noise in the acquistion and pre-processing phases of the fMRI analysis pipeline. All this adds up to the need for reliable statistical tools that can take into account the multivariate nature of the data.

In this setting of greater samples assumptions on the underlying data are often required to effectively reduce the degrees of freedom. Consider the simple least squares estimator $\hat{f}_n(x) = x^T \hat{w}_n$, where x is a sample and $w_n \in R^p$ the least squares estimate from n samples. It can be shown that for a constant C,

$$\mathbb{E}\|\hat{f}_n - f\| \leq C\frac{p}{n}. \tag{1.0.1}$$

This cannot be improved upon without adding an assumption [Rigollet, 2015] on the true underlying model. This poses a problem for the case $p \gg n$. A common approach assumes that the true model is in fact sparse, more formally that

$$\|w\|_0 \leq k,$$

where k is small. For many natural data, the underlying sparsity assumption is indeed true. In other cases it is a reasonable approximation that permits to defeat the statistical intractability posed by high dimensional data. Consider fMRI where brain activations are generally believed to be sparse or natural images which exhibit sparsity in a wavelet or fourier basis.

The application of sparsity inducing assumptions, the efficient computation of such models (note the combinatorial nature), and how to perform statistical inference has been studied widely in the past decade [Bühlmann et al., 2014]. More recently interest has arisen in further structure combined with sparsity, for example many datasets might contain inherent graph structured which can be leveraged. In this thesis we focus on several problems in the case of data exhibiting structure and often graphical structure.

High-dimensional inference tasks require reasoning in settings with few samples and a large number of variables. Inference in high dimensional data is a common task for a wide variety of applications. The areas of high dimensional statistics we focus on in this thesis is graphical modeling, variable selection, classification, and regression. We make use of graph structured properties both for the case of predictive models with known underlying graphical structure in the input and then subsequently for tasks where determining the unknown graphical structure of an undirected graphical model is the primary interest. We consider cases that have both sparsity as well as additional structured constraints.

We motivate our work from problems arising from the analysis of neuroimaging data, specifically functional magnetic resonance imaging (fMRI). We first consider the use for predictive and structured linear models. A recent paradigm for analyzing fMRI data attempts to predict cognitive variables with brain activation. By predicting some cognitive variables related to brain activation maps, this approach aims at decoding brain activity. Several examples of tasks which can be addressed are shown in Figure 1.2. Unlike some classic approaches this technique, sometimes referred to as "reverse inference" or brain decoding, takes into account the multivariate information between voxels and is one of the only ways to assess how precisely some cognitive information is encoded by the activity of neural populations within the whole brain. However, fMRI is very high dimensional and has few samples and thus we must control the prediction function with appropriate regularization. In the first chapter of this work we construct a principled penalty for linear models that can be applied for fMRI data [Jenatton et al., 2012].

The second problem of interest we take motivation from in fMRI is the determination of functional brain networks. Correlations in brain activity reveal brain interactions between distant regions, a process know as *functional connectivity*. Functional connectivity can provide interesting insights into brain mechanisms as it persists in the absence of tasks (the so-called "resting-state") and is thus applicable to study populations of impaired subjects, as

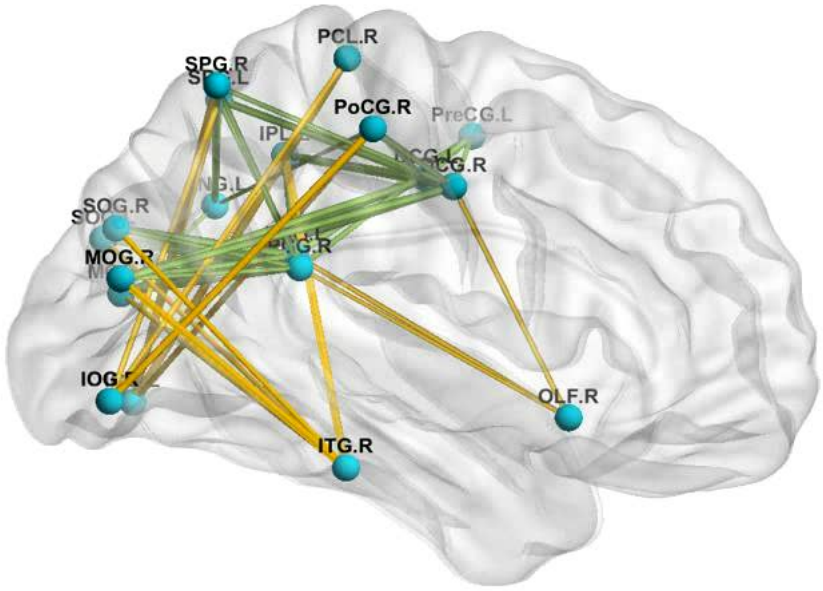


Figure 1.1 – Example of a brain networks implicated in one recent study on Autism Spectrium Disorder from [Chen et al., 2017]. The connectivity is determined between 90 anatomic brain regions described in [Chen et al., 2017].

in neurologic or psychiatric diseases [Castellanos et al, 2013]. Thus determining the brain networks, which in this work we interpret as undirected graphical models, from a limited number of samples is an increasingly important task of interest in the analysis of fMRI data. An example of a brain network is shown in Figure 1.1.

Contents

1.1 Overview

Having introduced these two general motivations from the neuro-imaging point of view we know delve into the specific problems we pose in the context of machine learning and statistics.

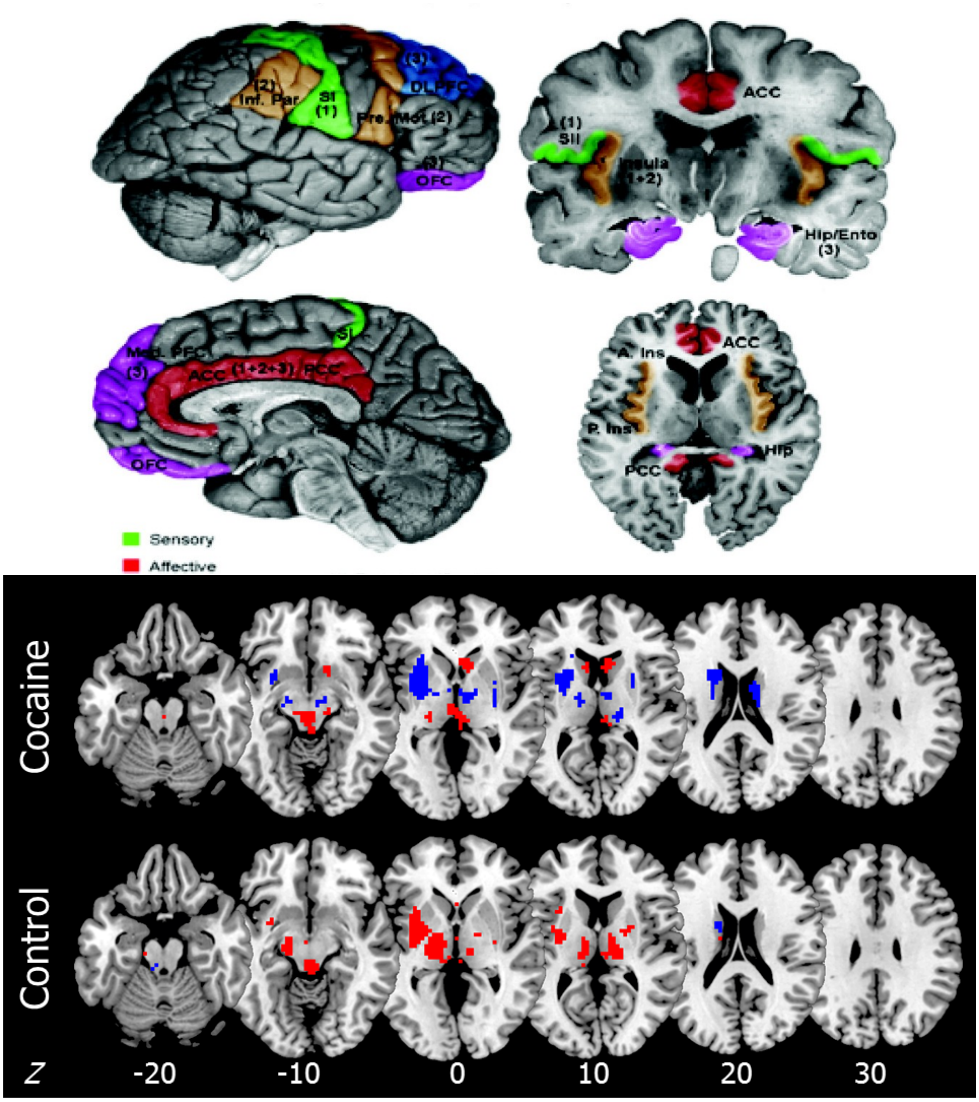


Figure 1.2 – Examples of brain areas determined by fMRI studies to be active for specific conditions or stimulus [Borsook et al., 2007]. The top diagram gives an example of cortical areas involved with pain processing as determined by an fMRI study. This image is taken from [Borsook et al., 2007]. The bottom diagram shows the result of a study on cocaine addiction using sparse linear modeling. The study is described in Appendix A and in [Belilovsky et al., 2015b].

1.1.1 Convex Relaxations of Graph Structured Penalties

With the motivating examples for "brain decoding" presented we consider the task of constructing a predictive linear model that captures a graph constraint between the predictive variables. Indeed the natural graph to use in the context of neuroimaging is based on the spatial adjacency. A common recent approach for this problem is to apply a total variation penalty. However, this can be inefficient alone as brain activations occur in a sparse manner, thus a penalty which encourages sparsity on the number of active nodes in the graph can be of relevance. Finally, permitting correlated variables has been shown to be critical in models which encourage sparsity, both for predictive performance and for permitting low variance and interpretable models. Thus in the first part of this thesis we focus on the problem of formulating a constraint that captures these three common and relatively generic properties and showing how it can be used to construct predictive linear models in a tractable manner.

To this end we study the problem of statistical estimation with a signal known to be sparse, spatially contiguous, and containing many highly correlated variables. We take inspiration from the recently introduced k-support norm, which has been successfully applied to sparse prediction problems with correlated features, but lacks any explicit structural constraints commonly found in machine learning and image processing. We address this problem by incorporating a total variation penalty in the k-support framework. We introduce the (k, s) *support total variation norm* as the *tightest* convex relaxation of the intersection of a set of discrete sparsity and total variation penalties. We show that this norm leads to an intractable combinatorial graph optimization problem, which we prove to be NP-hard. We then introduce a tractable relaxation with approximation guarantees that scale well for grid structured graphs. We devise several first-order optimization strategies for statistical parameter estimation with the described penalty. We demonstrate the effectiveness of this penalty on classification in the low-sample regime, classification with M/EEG neuroimaging data, and image recovery with synthetic and real data background subtracted image recovery tasks. We extensively analyse the application of our penalty on the complex task of identifying predictive regions from low-sample high-dimensional fMRI brain data, we show that our method is particularly useful compared to existing methods in terms of accuracy, interpretability, and stability.

We next turn to consider the question of how to find unknown graphical model structure, particularly in the context of brain connectivity, and how to determine differences between two graphical models.

1.1.2 Graphical Model Structure Discovery and Difference Testing

Functional brain networks are well described and estimated from data with Gaussian Graphical Models (GGMs), e.g. using sparse inverse covariance estimators. Comparing functional connectivity of subjects in two populations calls for comparing these estimated GGMs. Our goal is to identify differences in GGMs known to have similar structure. We characterize the uncertainty of differences with confidence intervals obtained using a parametric distri-

bution on parameters of a sparse estimator. Sparse penalties enable statistical guarantees and interpretable models even in high-dimensional and low-sample settings. Characterizing the distributions of sparse models is inherently challenging as the penalties produce a biased estimator. Recent work invokes the sparsity assumptions to effectively remove the bias from a sparse estimator such as the lasso. These distributions can be used to give confidence intervals on edges in GGMs, and by extension their differences. However, in the case of comparing GGMs, these estimators do not make use of any assumed joint structure among the GGMs. Inspired by priors from brain functional connectivity we derive the distribution of parameter differences under a joint penalty when parameters are known to be sparse in the difference. This leads us to introduce the debiased multi-task fused lasso, whose distribution can be characterized in an efficient manner. We then show how the debiased lasso and multi-task fused lasso can be used to obtain confidence intervals on edge differences in GGMs. We validate the techniques proposed on a set of synthetic examples as well as neuro-imaging dataset created for the study of autism.

1.1.3 Learning to Discover Sparse Graphical Models

Although penalized maximum likelihood procedures and pseudolikelihood as used in our difference estimation are effective for the case of Gaussian Graphical Models. The construction of inference procedures for determining networks can be cumbersome and the inference procedures themselves time consuming, ultimately the modeling assumptions are possible to simulate. We ask whether we can thus learn such an inference procedure, gaining an advantage in both model construction, speed, and accuracy.

We consider structure discovery of undirected graphical models from observational data. Inferring likely structures from few examples is a complex task often requiring the formulation of priors and sophisticated inference procedures. Popular methods rely on estimating a penalized maximum likelihood of the precision matrix. However, in these approaches structure recovery is an indirect consequence of the data-fit term, the penalty can be difficult to adapt for domain-specific knowledge, and the inference is computationally demanding. By contrast, it may be easier to generate training samples of data that arise from graphs with the desired structure properties. We propose here to leverage this latter source of information as training data to learn a function, parametrized by a neural network that maps empirical covariance matrices to estimated graph structures. Learning this function brings two benefits: it implicitly models the desired structure or sparsity properties to form suitable priors, and it can be tailored to the specific problem of edge structure discovery, rather than maximizing data likelihood. Applying this framework, we find our learnable graph-discovery method trained on synthetic data generalizes well, identifying relevant edges in both synthetic and real data, completely unknown at training time. We find that on genetics, brain imaging, and simulation data we obtain performance generally superior to analytical methods.

1.2 Structure and Contributions of the Thesis

The line of work presented in this thesis focuses structured reasoning on high dimensional data. Particularly we leverage formulations where data arises as a sparse graph. In the first part of the thesis we construct penalties that attempt to satisfy both a graph constraint and a sparsity constraint. In the next phase we focus on data which has an unknown sparse graph associated with it, and the goal is not to predict an external variable but to determine the undirected graphical model. The work covered in this thesis has been collected in several papers. The content and contributions of these papers are presented in Chapters 3, 4 and 5. As a whole, the contents of these papers address a range of problems in high dimensional few sample problems and the use of structured sparsity priors to make efficient prediction, variable selection, graphical model structure selection.

In Chapter 2, we present fundamental principles and tools used in the different methods presented in the thesis. We also present many of the related works. The objective of this chapter is to lay the foundations for the rest of the thesis.

Chapter 3 presents a novel structured sparsity penality and analyzes tractable approximations to this penalty. A variety of experiments are presented including simulations, image recovery, and neuroimaging data from MEEG and fMRI data. Additional work that serves as background to this work is presented in A. The content of this work is based on the following publication

- E. Belilovsky, A. Argyriou, G. Varoquaux, and M. Blaschko. Convex relaxations of penalties for sparse correlated variables with bounded total variation. *Machine Learning*, 100(2-3):533–553, 2015a.

Chapter 4 of this thesis presents a novel approach to the problem of determining difference in the edge structure of Gaussian Graphical Models (GGM). We propose a hypothesis test, that under inductive hypothesis of difference sparsity, permits to test whether an edge is different between two related GGMs. The content of this work is based on the following publication,

- E. Belilovsky, G. Varoquaux, and M. B. Blaschko. Testing for differences in gaussian graphical models: Applications to brain connectivity. *Advances in Neural Information Processing Systems*, 2016c.

Chapter 5 presents a novel approach to using advances in deep learning to create learned estimators for GGM model structure, and permit to obtain fast and accurate estimators by specifying a generative model of the underlying process. The content of this work is based on the following publication,

- E. Belilovsky, K. Kastner, G. Varoquaux, and M. Blaschko. Learning to discover graphical model structures. *International Conference on Machine Learning*, 2017b.

Chapter 6 concludes the thesis and proposes future work.

The work during this PhD additionally addresses several other related problems in Machine Learning and Computer Vision. These span the gauntlet of model selection in deep generative models, representation learning in multimodal data, and structured deep learning models. Below is the full publication list

- E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. *International Conference on Computer Vision*, 2017
- E. Belilovsky, K. Kastner, G. Varoquaux, and M. Blaschko. Learning to discover graphical model structures. *International Conference on Machine Learning*, 2017b
- E. Belilovsky, M. Blaschko, J. R. Kiros, R. Urtasun, and R. Zemel. Joint embeddings of scene graphs and images. *International Conference on Representation Learning (ICLR) Workshop Track*, 2017a
- E. Belilovsky, W. Bounliphone, M. B. Blaschko, I. Antonoglou, and A. Gretton. A test of relative similarity for model selection in generative models. *International Conference on Representation Learning*, 2016a
- E. Belilovsky, G. Varoquaux, and M. B. Blaschko. Testing for differences in gaussian graphical models: Applications to brain connectivity. *Advances in Neural Information Processing Systems*, 2016c
- E. Belilovsky, A. Argyriou, G. Varoquaux, and M. Blaschko. Convex relaxations of penalties for sparse correlated variables with bounded total variation. *Machine Learning*, 100(2-3):533–553, 2015a
- E. Belilovsky, K. Gkirtzou, M. Misyrlis, A. B. Konova, J. Honorio, N. Alia-Klein, R. Z. Goldstein, D. Samaras, and M. B. Blaschko. Predictive sparse modeling of fmri data for improved classification, regression, and visualization using the k-support norm. *Computerized Medical Imaging and Graphics*, 46:40–46, 2015b

Chapter 2

Foundations and State-of-the-Art

In this chapter we discuss background materials which permit to frame this thesis in the context of the state of the art research in high dimensional statistics, machine learning, and the analysis of fMRI and other high dimensional data.

We begin by discussing details of the fMRI data and typical problems found in its analysis. We also discuss similar problems from genomics. We then discuss the field of structured sparsity. We review MRFs and structured discovery in Markov Random Fields. Finally we discuss deep learning methods with a particular focus on recent work in approximating combinatorial optimization problems and in particular learned sparse recovery algorithms

Contents

2.1 fMRI analysis

Since we use fMRI analysis as a motivating example throughout this thesis we will briefly review several key concepts from the fMRI literature. For a more detailed discussion of these topics please refer to Lindquist et al. [2008], Poldrack et al. [2011].

2.1.1 BOLD signal and Noise in fMRI anaylsis

Human brain activity is based on electrical signals derived from the firing of neurons throughout the brain. In order to better understand the function of the human brain neuroscience and medical researchers aim to have a detailed data on the neural activity under various tasks as well as in the so called resting state.

The Blood-oxygen-level dependent contrast imaging (BOLD) signal is derived from the functional Magnetic Resonance Imaging device and is the primary signal used in the analysis of fMRI data. The Magnetic Resonance Device is able to measure the flow of blood in the brain. Important to the proper interpretation of the fMRI data is that the BOLD signal is only a proxy of the electrical activity of the brain that one ultimately aims to measure.

Analysis of the brains electrical activity is based on the observation that neuron firing causes a need for energy to be brought quickly to the affected area. This results in what is called a hemodynamic response where blood releases oxygen to areas with firing neurons at a greater rate than inactive neurons. Using an appropriate model of the hemodynamic response one can approximately infer the neuron firing. However, it is note worthy that the time scales involved and use of a model for the hemodynamic response mean that the exact signal of interest (the neuronal firing) is sometimes obfuscated. This is in contrast to the electroencephalography (EEG) and magnetoencephalography (MEG) signals which have much shorter temporal resolutions but uncertain spatial localization and severe noise [Lindquist et al., 2008, Poldrack et al., 2011].

One must thus take care in interpreting the BOLD signal due to a number of noise factors that arise in the acquisition. This includes among others potential inaccuracies in the modeling of the hemodynamic response, the movement of the patient, and noise inherent in the MR signal acquisition. A large body of literature as well as multiple software toolkits exist that permit to process fMRI data into a form digestable for statistical analysis. Standard preprocessing often includes slice timing, motion correction, first level analysis with a general linear model, and inter-subject spatial normalization [Lindquist et al., 2008].

2.1.2 Statistical Analysis of fMRI data

Several typical question of interest in fMRI analysis are which brain activity is associated with certain stimuli and conditions. More recently the interactions between potentially distant regions has led to great interest in determining functional brain networks.

Although standard practice in any machine learning method, within fMRI analysis, many traditional methods do not assume any relationships between different brain voxels. The traditional approach, often dubbed univariate analysis, considers each voxel independently and determines if it is relevant with respect to a stimulus or response variable. Methods from statistical machine learning and multivariate statistics are often referred to as Mutli-Voxel Pattern Analysis (MVPA) as they do consider the brain voxels jointly. This is in contrast to

univariate methods which consider each voxel or region independently. For example univariate analysis of one condition versus another might determine if each voxel or region is significantly different across two conditions, with a statistical hypothesis test, e.g. t-test. Notably many MVPA pipelines will often have a univariate pre-screening phase where certain voxels are discarded.

In this thesis and in particular the applications presented in Chapter 3 and in Appendix A consider the MVPA paradigm. Moreover we consider that predictive models from machine learning can be extremely valuable for the analysis of fMRI data by demonstrating whether an output variable is distinguishable using the fMRI data. For stimulus based tasks this demonstrates that the brain has performed a specific encoding of the stimulus and the noisy signal derived from fMRI has captured sufficient information to make predicitons. Secondly the model which gives the prediction can shed light on which areas are critical for this prediction. As an example consider Appendix A where cocaine patients are compared to control subjects. The predictive models show that the brain activity can be used to distinguish addicts and control groups, and the subsequent implicated regions correspond to existing brain models of addiction.

There exist two distinct types of fMRI studies, task based and resting state fMRI. In task based fMRI a subject is stimulated while inside the scanning device. Resting state fMRI considers subject who are not stimulated inside the scanner. This permits to better explore the brain's functional organization and determine if it is affected by pathologies. Many studies have identified the so called default mode network, which is a large scale brain network activated at rest. It is hypothesized by some researchers to be implicated in diseases such as Alzheimer's and autism spectrum disorder.

Brain connectivity attempts to understand what are the interactions between different brain areas. This can be modeled as a graph where the nodes are brain regions and edges represent correlations or partial correlations. There are various types of connectivity often discussed in the fMRI literature. The three primary distinctions are often made between anatomical, functional, and effective connectivity. Anatomical connectivity deals with describing how different brain regions are physically connected and can be tackled using diffusion tensor imaging (DTI). Functional connectivity is defined as the undirected association between two or more fMRI time series, while effective connectivity is the directed influence of one brain region on others [Lindquist et al., 2008]. The distinction between functional and effective connectivity can often be ambigous, indeed some argue that it is effectively difficult to assess and often the consequences of the directed connection (whether it is causal in nature) are not well understood. In this work we concentrate on the functional connectivity, which itself can refer to different types of association. In the simplest case functional connectivity can refer to correlations between brain regions, while others have considered partial correlation.

2.2 Sparse Regularizers and Structured Sparsity

A basis of statistical inference is the application of regularized risk, in which a loss function is evaluated over a sample of data and is linearly combined with a regularizer that penalizes some norm of the prediction function as in Eq. (2.2.1), where the first term is the loss function and the second is the penalty term:

$$\min_{w} f(w, X, y) + \lambda J(w). \tag{2.2.1}$$

Here we denote by $X \in \mathbb{R}^{n \times d}$ the design matrix of n samples each with d dimensions; we denote by $y \in \mathbb{R}^n$ the vector of targets. In the sequel, we assume that we have a sample of labeled training data $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in \left(\mathbb{R}^d \times \mathbb{R}\right)^n$ where x_i is the output of a fMRI scan, and y_i is a ground truth label that we would like to be able to predict. The scalar parameter $\lambda > 0$ controls the degree of regularization and J is a scalar valued function monotonic in a norm of $w \in \mathbb{R}^n$. Sparsity regularization is a key family of priors over linear functions that prevents overfitting and aids interpretability of the resulting models [Argyriou et al., 2012a, Tibshirani, 1996b].

One of the most important sparsity regularizers is the LASSO [Tibshirani, 1996b], where $J(w) = \|w\|_1$ and f corresponds to squared loss. In many learning problems of interest, LASSO has been observed to shrink too many of the w variables to zero. In the presence of a group of highly correlated variables, LASSO may prefer a sparse solution. However including all correlated variables in the model could potentially lead to higher predictive accuracy [Argyriou et al., 2012a] and more stable support recovery. The elastic net [Zou and Hastie, 2005] and more recently the k-support norm address this problem by providing a way of calibrating the cardinality of the regression vector w so as to include more variables.

For example in order to create a model which can make inferences about labels for new fMRI samples and provide a map of the key voxels we must specify an appropriate loss function. This specifies the prediction properties we are interested in obtaining. We must then specify an appropriate sparse regularizer which captures the *a priori* structure of the data. Finally, we must optimize the objective specified in Equation (2.2.1). In the case of fMRI this permits to obtain a brain map which can predict labels for new samples and provide insights on which brain regions are most associated with this prediction.

It is often the case we desire penalties which are inherently discrete and lead to non-convex constraints. A very popular example of this is the standard sparsity penalty (ℓ_0) which selects for k variables. We now discuss in more detail different approaches to relaxing such penalties.

2.2.1 Convex Relaxations of Discrete Penalties and the k-support norm

Key to the mathematical understanding of sparsity regularizers is their interpretation as convex relaxations to quantities involving the ℓ_0 norm, which simply counts the number of non-

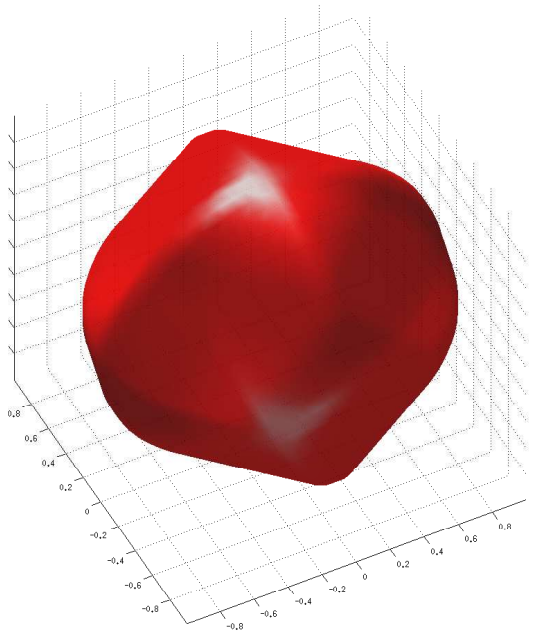


Figure 2.1 – The k-support unit ball with $k = 2$ and $d = 3$.

zero elements of a vector. Consider the sparsity inducing problem framed as a combinatorial optimization:

$$\min_{w} f(w, X, y)$$
$$s.t. \|w\|_0 \leq \lambda.$$

For example ℓ_0 can also be replaced by more complex permitted structures. Two primary approaches for approximating a solution under sparse or structured sparse constraints exist in the literature. In one case we can consider approximating the problem directly using combinatorial methods. A popular approach in this vain is the class of hard-thresholding methods . Alternatively, the approach we focus on in this thesis is based on taking a convex relaxation of the set of discrete constraints, permitting to solve exactly for the approximate solution.

The class of hard thresholding methods can in some cases yield more efficient custom solutions with guarantees, but can also often produce unstable solutions and relies on assumptions on the data matrix for its approximation guarantees, whereas obtaining a convex relaxation of the sparse penalty can permit to easily incorporate the constraint into standard first order optimization methods and combine it more easily with typical machine learning loss functions.

The ℓ_1 norm, which is the sum of the absolute values of the vector, is the convex relaxation of the ℓ_0 norm, meaning it is the tightest sparsity norm that retains convexity, which is key for computational tractability. The elastic net Zou and Hastie [2005] attempts to improve on

this penalty by considering $\lambda_1\|w\|_1 + \lambda_2\|w\|_2^2$. The ℓ_2 penalty prevents correlated values from being dropped as in the case of just the sparse penalty. Argyriou et al. [2012a] reformulate this from first principal, indicating that the desired discrete penalty is the intersection of sparsity and the ℓ_2 norm ball, the desired penalty is thus,

$$\{w \in \mathbb{R}^d : \|w\|_0 \leq k, \|w\|_2 \leq 1\} \tag{2.2.2}$$

While the ℓ_1 norm can therefore be interpreted as employing the convex hull of the ℓ_0 sparsity regularizer, the elastic net is looser than the convex hull of (2.2.2). Argyriou et al. [2012a] then show that the tightest convex relaxation of this is given by the k-support norm.

However, one may employ the k-support norm, which is exactly the convex hull of that hybrid norm. A visualization of the k-support norm unit ball is given in Figure 2.1. We see that there is a non-differentiability of the norm, which restricts the set of optimization strategies that we may employ (cf. Section 3.2.4).

The k-support norm can be computed as

$$\|w\|_k^{sp} = \left(\sum_{i=1}^{k-r-1} (|w|_i^{\downarrow})^2 + \frac{1}{r+1} \left(\sum_{i=k-r}^{d} |w|_i^{\downarrow} \right)^2 \right)^{\frac{1}{2}} \tag{2.2.3}$$

where $|w|_i^{\downarrow}$ is the ith largest element of the vector and r is the unique integer in $\{0, \ldots, k-1\}$ satisfying

$$|w|_{k-r-1}^{\downarrow} > \frac{1}{r+1} \sum_{i=k-r}^{d} |w|_i^{\downarrow} \geq |w|_{k-r}^{\downarrow}. \tag{2.2.4}$$

The k-support norm is closely related to the elastic net, in that it can be bounded to within a constant factor of the elastic net, but it leads to different sparsity patterns. One can see from Equation (3.2.33) that the norm trades off a squared ℓ_2 penalty for the largest components with an ℓ_1 penalty for the smallest components.

A difficulty in using sparse regularizers is that they tend to lead to non-smooth functions which can cause difficulties when using gradient based convex optimization procedures. For this class of functions proximal methods are a very popular way to quickly find optimal solutions with the bottleneck generally being the computation of the proximal mapping. Among many advantages of the k-support norm, it has an easy to compute proximal operator given in Argyriou et al. [2012a].

While initial experiments have shown promising results with the k-support norm for a range of machine learning problems [Argyriou et al., 2012a], to the best of our knowledge the studies discussed in Appendix A and Chapter 3 are the first applications to fMRI. The k-support norm forms the basis of our work in Chapter 3. We now briefly described related literature on structured sparsity.

2.2.2 Related Work in Structured Sparsity

A number of works have proposed techniques which assume a structure on the sparse support set. We briefly review several relevant recent works.

Group sparsity is a common form of structured sparsity that has been considered by many authors [Yuan and Lin, 2006]. The general framework often partitions variables into groups where only a sparse subset of the groups can be present in the solution. Early works considered the case of disjoint groups [Yuan and Lin, 2006]. More recent work in Jacob et al. [2009] propose a variant of the group lasso that has an efficient convex penalty for the case where data can be described by overlapping groups which jointly go to zero. This penalty has some interesting relations to the k-support norm described in Argyriou et al. [2012a] in that the k-support an be seen as the group lasso where the groups are not known. A large class of related penalties which enforce groupings and hierarchies among variables are described in Bach et al. [2012b].

Another popular structured penalty is the total variation. For the case of images or graph structured data we consider two common forms of the TV penalties corresponding to an-sitropic, TV_a and isotropic TV

$$TV_I(w) = \sum \nabla(w) = \sum \sqrt{\nabla_x w + \nabla_y w + \nabla_z w} \tag{2.2.5}$$

$$TV_A(w) = \|Dw\|_1. \tag{2.2.6}$$

Here D represents an incidence matrix and can generalizes to more generic graph structured data. This penalty assumes a smoothness between adjacent nodes on the graph. It has been successful in a number of applications in image processing Rudin et al. [1992]. Combining this with sparsity priors has shown to be very effective in fMRI analysis [Michel et al., 2011] where the signal is known to be sparse in addition to locally coherent. This however can lead to convex optimization problems which are difficult to optimize effectively [Dohmatob et al., 2014].

2.2.3 Proximal algorithms

Sparse regularizers often lead to non-smooth convex optimization problems. A very common tool for solving a convex objective of the form (2.2.1), where $J(w)$ is a non-smooth term is by the use of proximal operators.

The proximal operator of $J(w)$ is defined as

$$\text{prox}_J(v) = \arg\min_w J(w) + (1/2)\|w - v\|^2. \tag{2.2.7}$$

Note that the indicator function leads to a proximal operator which is the euclidean projection. Thus one view of proximal operators is as a generalized projection [Parikh et al., 2014]. The

proximal operator can also be seen as analogous to a gradient step for non-smooth functions. This leads to it being an important tool in minimizing non-smooth functions. Obtaining an expression for the proximal operator of a non-smooth function will permit us to use it in different accelerated proximal methods described in Parikh et al. [2014].

2.3 Estimation of the Structure of Undirected Graphical Models

Probabilistic graphical models express conditional dependence structure between random variables. They have become a common tool in many applications of machine learning. Probabilistic models generally represent nodes as random variables and use edges to represent statistical dependencies.

These models can form an intuitive way to understand relationships of different variables to each other. This can permit to both more naturally specify the model and in the case of structure learning covered in this thesis to visualize the relationships. Additionally probabilistic inference can often be efficiently implemented in these models.

A property that is found commonly in many real distributions is that a given variable tends to interact directly only with very few others (interacting with many others indirectly) [Koller and Friedman, 2009]. In other words many graphical models of interest are sparse in the edge set. This permits distributions to be encoded compactly using a graphical model. We will further discuss and exploit this common property in the sequel.

We now formalize the probabilistic graphical models we will work with. A *graph* is a set of vertices $\mathrm{V} = \{1, \ldots, p\}$ and a set of edges $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$. An *undirected graphical model* is a joint probability distribution, $\mathbb{P}(\mathbf{x})$, defined on an undirected graph G, where the vertices V in the graph index a collection of random variables $\mathbf{x} = \{\mathbf{x}_1, \ldots \mathbf{x}_p\}$ and the edges encode conditional independence relationships among random variables

In relation to graphical models and machine learning, four common procedures are of interest

- **Inference** Involves evaluating the probability distribution over some set of variables given the values of another set of variables. This permits to query the graphical model for answering questions using the underlying distribution.
- **Learning** Refers to learning the probability distribution especially when the probability distribution is described by complex parametrizations.
- **Sampling** We may want to sample a set of variables given another set.
- **Structure Discovery or Learning** This refers to problems where the graphical model structure is not given and we want to determine the structure.

In this dissertation we will focus on the structure learning or discovery problem in graphical models. We will also largely restrict ourselves to models with continuous variables.

In general there are two popular classes of graphical models,

Directed Acyclic Graphical Models Directed acyclic graphical models are a rich family of graphical models that specify directed graph between nodes. Particularly directed acyclic graphs, sometimes referred to as Bayesian Networks, are commonly considered.

The underlying structure in the graphical model consist of a directed acyclic graph, G. Describing the set of parent nodes of X as Pa_X we can state

Definition 2.1. *Koller and Friedman [2009] Let G be a Bayesian network over the variables $X_1, .., X_n$. The distribution factorizes according to G if we can write*

$$P(X_1, .., X_n) = \Pi_{i=1}^{n} P(X_i | Pa_{Xi}) \tag{2.3.1}$$

Thus we can see this as permitting an efficient factorization of the distribution. The Bayesian network can also be seen as prescribing the conditional independence structure of the data. Denoting $NonDescendants_X$ the variables which are not descendants of X

Definition 2.2. *Koller and Friedman [2009] Given a Bayesian network with structure G over random variables $X_1, ..., X_n$. Then G encodes the following set of conditional independence assumptions:*

For each variable X_i, we have that $X_i \perp NonDescendants_{X_i} | Pa_{X_i}$

These models can be expressive. However, learning the structure of a DAG is difficult. Two common approaches are score based algorithms which optimize the global Bayesian information criterion (BIC) or Bayesian marginal likelihood often using greedy approaches or relying on sampling in Markov Chain Monte Carlo (MCMC) procedures. Another technique that has been applied relies on performing a series of conditional independence tests to narrow the family of DAGs which describe the graph [Murphy, 2012].

In general there are few efficient deterministic way to find the best network, and you might need randomized algorithms like Markov chain Monte Carlo to find a good network. We may also add some restrictions to the Bayesian network (e.g. requiring it to be a tree) to make the problem more tractable.

Undirected Graphical Models Undirected graphical models, sometimes referred to as Markov random fields, specify the probability distribution using an undirected graph. This class of models is very popular as it can permit efficient inference and learning procedures while still being very expressive. Similar to the Bayesian network, the Markov random field specifies a compact set of conditional independence assumptions on the underlying graph.

Definition 2.3. *Koller and Friedman [2009] Let G represent the undirected graph. Then for each node $x \in X$, the Markov blanket of x denoted $N_h(X)$ is the set of neighbors of x in the graph. The local Markov independence associated with G gives that*

$$x \perp X - x - N_h(X) | N_h(X)$$

Thus each node is conditionally independent of the rest given its neighbors.

Similar to the Bayesian network the undirected graphical model can be shown to factorize. Let us associate a potential function with each maximal clique in the graph, $\Psi_c(x_c)$. The potential function can be any non-negative function of its arguments. We can now describe the joint distribution as proportional to the product of clique potentials. This leads to a famous result, the Hammersley-Clifford theorem.

Theorem 2.1. *Murphy [2012] A positive distribution $P(x) > 0$ satisfies the conditional independence properties of an undirected graph, G, iff P can be represented as a product of factors, one per maximal clique,*

$$P(x|\theta) = \frac{1}{Z(\theta)} \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{x}_c),$$

where $\mathcal{C}$ is the set of maximal cliques in the graph and $Z(\theta)$ is the partition function that assures the distribution sums to 1.

In this thesis we focus on the class of Undirected graphical models with continuous variables. Particularly we often analyze the Gaussian graphical model which corresponds to a multivariate Gaussian.

2.3.1 Gaussian Graphical Models and Conditional Independence

In this section, we give an overview of the estimation of precision matrices and conditional independence on undirected graphical models. More details on this topic can be found in Dawid [1979], Dempster [1972], Lauritzen [1996] and Whittaker [2009].

The importance of estimating covariance matrices and their inverses, called precision matrices, is fundamental in modern multivariate analysis and in a wide array of scientific applications. The covariance matrix reveals marginal correlations between variables, while the precision matrix encodes conditional correlations between pairs of variables given the remaining variables.

A Gaussian is fully described by it's mean and covariance matrix. If we take the undirected graphical model view of the multivariate Gaussian, several interesting properties arise. It is well known that recovering the structure of an undirected Gaussian graph is equivalent to the recovery of the support of the precision matrix (Figure 2.2). Formally, suppose we have a sample $\mathbf{X}_p = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ of dimension p and size n with the mean of each $\mathbf{x}_i$ equal to zero,

$$\mathbf{\Sigma^{-1}} = \begin{pmatrix} 1 & 0 & 0.4 & 0.3 \\ 0 & 1 & 0 & -0.3 \\ 0.4 & 0 & 1 & 0 \\ 0.3 & -0.3 & 0 & 1 \end{pmatrix} \longrightarrow$$

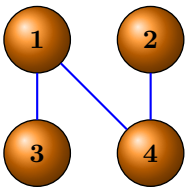


Figure 2.2 – Example of Gaussian graphical model defined by the inverse of the covariance matrix.

and a covariance matrix of size $p \times p$ is $\Sigma_{ij} = \mathbb{E}(\mathbf{x}_i \mathbf{x}_j^T)$ such that $\mathbf{x} \sim \mathcal{N}_p(0, \mathbf{\Sigma})$ then

$$(i,j) \notin \mathrm{E} \iff \mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j | V \setminus \{i,j\} \iff \Sigma_{ij}^{-1} = 0. \tag{2.3.2}$$

We can see that the conditional independence properties of a Markov random field. Thus the support of the precision matrix gives us the undirected graphical model structure.

Testing conditional independence is an important concept in statistics, artificial intelligence, and related fields [Dawid, 1979]. A common measure for the testing of independence of two variables conditioned on a set of variables is the *partial correlation* $\rho_{\mathbf{x}_1\mathbf{x}_2 \cdot \mathbf{x}_3}$. With the assumption that all variables are multivariate Gaussian, the partial correlation is zero if and only if $\mathbf{x}_1$ is conditionally independent from $\mathbf{x}_2$ given a set of variables, $\mathbf{x}_3$:

$$\mathcal{H}_0 : \rho_{\mathbf{x}_1\mathbf{x}_2 \cdot \mathbf{x}_3} = 0 \qquad \text{vs} \qquad \mathcal{H}_1 : \rho_{\mathbf{x}_1\mathbf{x}_2 \cdot \mathbf{x}_3} \neq 0. \tag{2.3.3}$$

The distribution of the sample partial correlation for a Gaussian distribution was described by Fisher [1924] and we would reject $\mathbf{H}_0$ if the absolute value of a transformed test statistic exceeded the critical value from the Student table evaluated at $\delta/2$. The computational complexity of the partial correlation is $\mathcal{O}(np^2 + p^3)$ which simplifies to $\mathcal{O}(np^2)$ as $n \geq p$.

Two dominant approaches for fitting a high dimensional sparse Gaussian graphical model where introduced in [Friedman et al., 2008, Meinshausen and Bühlmann, 2006]. The former fits an ℓ_1-penalized nodewise regression and the latter penalizes the likelihood of the precision matrix referred to as the graphical lasso. We can look at the graphical lasso as fitting a precision matrix Θ given a set of samples from a high-dimensional GMRF which is known (or expected) to be sparse. The objective in graphical lasso is:

$$\min_{\Theta \geq 0} \log \|\Theta\| - \mathrm{Tr}(\hat{\mathbf{\Sigma}}\Theta) - \lambda \|\Theta\|_1. \tag{2.3.4}$$

Recent work in Loh and Wainwright [2013] further expands the property of precision matrices of indicating independence to several classes of non-Gaussian Markov random fields.

Meinshausen and Bühlmann [2010] introduce stability selection, sometimes called random lasso. This can be applied to graphs as well as the traditional lasso regularization setting. The algorithm essentially consists of retraining subsets of the data and measuring stability. It additionally provides finite sample bounds on the false discovery rate. We will illustrate some of the key results from this work in a description of the work in Ryali et al. [2012]. This method

also introduces an analysis tool called the stability path, analogous to the regularization path, wherein we can view the probability of selecting a variable.

Ryali et al. [2012] extend the graphical lasso framework with the use of an elastic net style penalty and stability selection for learning functional connectivity in a reliable manner. In this work they also discuss the critical issue of regularization parameters selection as this can lead to different solutions for the precision matrix. First they note that the Akaike information criterion (AIC) and BIC, in the high dimensional low sample brain data tended to result in overly sparse solutions. They note that it often reduced the elastic net penalty to lasso type solutions, which they deemed overly sparse. Here they utilize the stability selection theory from Meinshausen and Bühlmann [2010] to select the edges using the entire regularization path.

We define an estimated stability for a given edge (m, n) and a given regularization setting (in elastic net we use two parameters λ, α) as $\hat{\pi}^{m,n}_{\lambda,\alpha}$ .To estimate $\hat{\pi}^{m,n}_{\lambda,\alpha}$ Meinshausen and Bühlmann [2010] specify subsampling N/2 observations randomly 100 times for each regularization setting. The estimate is now simply the average over these. In this work it is also noted in fMRI we should take care to make sure the subsamples are from different sessions.

The set of stable variables S^{stable} consists of all variables for which $\hat{\pi}^{m,n}_{\lambda,\alpha}$ is above a threshold π_{thr}. Stability selection now states that the expected number of falsely selected edges ,V, is bounded by

$$E(V) \leq \frac{1}{2\pi_{thr}} \frac{q^2}{p}, \tag{2.3.5}$$

where p is the number of model variables and q is the average number of selected edges for a given range of regularization parameter setting. To say it another way q is p minus the number of variables that are not selected anywhere on the regularization path for the given data. We can now control the error in our stability selection. Given a desired per-comparison error rate, $E(V)/p$, (e.g. we specify 0.05), we can calculate the π_{thr} using Equation (2.3.5). This procedure was described in Meinshausen and Bühlmann [2010] for graph lasso and applied in Ryali et al. [2012] for graph elastic net and brain connectivity. Finally in this work the use of simulated data in addition to real data is emphasized as a key evaluation.

Other works have considered alternatives on the sparsity regularization to be used with the estimation of the precision matrix. Some of these penalties are analogous to ones used in structured sparsity posed in the context of empirical risk minimization in prediction or in recovery problems.

Honorio et al. [2009] introduces a smoothing penalty into the regularizer, somewhat analogous to total variation, this has been applied for functional connectivity in brain imaging.

Another recent line of work in graphical structure learning is learning related graphical models jointly. For example we may know that brain connectivity is similar in many ways but different in others across subjects. Thus we learn a separate connectivity per subject. This line of work has a vast body of literature in recent years.

Honorio and Samaras [2010] show another structure learning approach with a penalty based on the multi-task learning literature. Here "tasks" are relevant related groups. In the specific example used, brain connectivity, each task refers to a subject or a session. The multi-task structure learning problem is defined as

$$\max_{\mathbf{\Theta}^1,\ldots,\mathbf{\Theta}^k} \sum_k l(\hat{\mathbf{\Sigma}}^k, \Theta^k) - \lambda \|\mathbf{\Theta}\|_{1,\infty},$$

where l refers to the maximum likelihood and Θ_i to each individual precision matrix within a group. A related penalty was proposed by Varoquaux et al. [2010].

Danaher et al. [2014] discusses two extensions in the joint model learning case called the fused graphical lasso, where differences in precision matrices are penalized with an ℓ_1 penalty. Similarly a group graphical lasso is discussed where groups of precision matrices are penalized.

In Mohan et al. [2012] a similar objective to the fused graph lasso is discussed but here node perturbations are emphasized versus the edge perturbations. This is dubbed the node-perturbed joint graph lasso.

Another recent work [Goncalves et al., 2014] presents a class of algorithms for jointly learning both the multi-task structure and the parameters of a sparse model. The authors call this Multi-task Sparse Structure Learning (MSSL). Given W, the task matrix Ω represents the relationship between different tasks in terms of a graphical model. Specifically we can think of the task relationships as a GGM. For example when $\Omega_{i,j} = 0$, the parameters for task i, w_i and task j, w_j have no influence on each other.

The objective of MSSL in this work is as follows:

$$\min_{\Omega, W>0} \sum_{k=1}^{K} \mathcal{L}(y_k, X_k, w_k) + \mathcal{B}(W, \Omega) + \mathcal{R}_1(W) + \mathcal{R}_2(\Omega) \tag{2.3.6}$$

Here $\mathcal{L}$ is the loss function $\mathcal{R}_1$ and $\mathcal{R}_2$ are sparsity regularization terms on both the graph structure and the task matrix. $\mathcal{B}$ is the inductive bias term which captures the interaction between the task variables w_i and w_j. The paper proposes several objectives in this framework and optimization algorithms for them. As an example one objective considers squared loss functions ℓ_1 regularization for $\mathcal{R}_1$ and $\mathcal{R}_2$. For the inductive bias term $Tr(W\Omega W^T)$ is used. If we assume Gaussianity for the rows of W then Ω becomes the precision matrix for a GMRF.

Another approach to structure discovery in GGMs is given by the Bayesian structure learning. Many approaches based on Bayesian inference techniques have been proposed [Mohammadi and Wit, 2015, Roverato, 2002]. These typically rely on Monte Carlo sampling techniques.

Many recent techniques rely on the G-Wishart distribution Letac et al. [2007], Roverato [2002] which is given by

$$P(\Theta|G) = I_G(b, D)^{-1} \|\Theta\|^{(b-2)/2} \exp\left(\frac{1}{2} \operatorname{Tr}(D\Theta)\right) 1_{\Theta \in M^+(G)}. \tag{2.3.7}$$

Here D is a positive definite matrix and $b > 2$ is the degree of freedom of the parameters, while I_G is a normalizing constant. $M^+(G)$ represents the positive definite cone of matrices with the graph structure G.

Multiple recent works propose efficient sampling procedures based on Monte-Carlo sampling that permit to include this distribution in hierarchical sampling procedures. These include Lenkoski [2013] as well as Wang et al. [2012].

2.4 Deep Learning

Deep learning is a branch of machine learning which relies on the use of neural networks as a flexible function class. The use of flexible compositional functions and automatic differentiation schemes, combined with hardware acceleration from graphics cards and large datasets has yielded state of the art performance in many fields.

The body of literature on neural networks is vast and we refer the reader to [Bengio et al., 2015, Schmidhuber, 2015] for comprehensive overviews. Here we review several recent advances that are relevant for our purposes. Neural networks with a sufficiently large number of hidden units can be seen as universal function approximators. Motivated by this result many works recently consider using neural networks as approximators for combinatorial problems. We discuss this and the application of neural networks on graph structured inputs in the sequel.

2.4.1 Approximating Combinatorial Optimization problems

Neural network based techniques have been proposed for providing approximate solutions to combinatorial optimization problems. For example Hopfield and Tank [1985] proposed a technique for approximating traveling salesman problems. A review of the early work in this field in the 90s can be found in [Smith, 1999].

In recent years there has been some resurgence of this line of research. Vinyals et al. [2015] proposed recurrent architectures that are particularly suitable for ingesting an unordered set of elements (e.g. points) and sequentially outputing a result which can directly refer to the input elements instead of a predefined vocabulary. This was applied to the classic traveling salesman problem. However, the results are limited to relatively small sizes. Vinyals et al. [2016] follow up on this work by adding several additional elements to deal with the unordered properties of various problems

Bello et al. [2017] proposed a reinforcement learning based approach for approximating the combinatorial optimization problems. They argue that in the case of many combinatorial optimization problems, especially, NP-hard problems where ground truth is an inherently intractable but relative value of solutions can be evaluated easily, justifies the reinforcement learning framework. They adapt the pointer network architecture but apply policy gradient methods to train the model.

Another class of combinatorial optimization problems recently addressed by deep learning and closely related to the problem we consider in Chapter 5 is that of sparse coding, which generally involve finding a solution to an objective given a sparsity constraint.

Gregor and LeCun [2010], Xin et al. [2016] use deep networks to approximate steps of a known sparse recovery algorithm thus producing faster solutions. They consider approximations of the solution of the following

$$\min_{w} \|y - \boldsymbol{X}w\|_2^2 \quad s.t. \quad \|w\|_0 \leq k. \tag{2.4.1}$$

A standard approach for this problem in the case where sparsity is approximated with convex relaxation to the ℓ_1 norm is Iterative Soft-Thresholding (ISTA) and modern variants such as FISTA. These alternate gradient steps on the objective with soft thresholding operations, which correspond to a proximal operator of the ℓ_1 norm. ISTA corresponds to a class of proximal splitting methods from convex optimization [Beck and Teboulle, 2009, Moreau and Bruna, 2017]. These more generally alternate between gradient steps on the differentiable part of an objective and proximal steps on the non-differentiable. Gregor and LeCun [2010] show that one can unfold the ISTA algorithm as a recurrent neural network. Parametrizing the steps of this and permitting adaptation in each layer now permits to learn the operation in few steps, from training data, giving the LISTA algorithm. In Moreau and Bruna [2017] conditions are described for X when the solution can indeed be improved drastically through adaptation.

Similarly Xin et al. [2016] make an analogy to the class of Iterative Hard Thresholding (IHT) techniques, which alternative between gradient steps on the typically differentiable objective and thresholding steps corresponding to the sparsity constraints. These can be shown to converge under certain conditions on $\boldsymbol{X}$, but can be more unstable than convex relaxation approaches. Xin et al. [2016] similarly show that they can parameterize the procedure of solving the IHT and perform it as a feedforward pass through a neural networks

Moreau and Bruna [2017] shed light on when such techniques can be effective based on the properties of the dictionary matrix. They highlight that when the dictionary matrix does not fulfill the properties described, adaptive methods do not lead to acceleration.

Neural Message Passing Message passing and particularly sum-product and max-product message passing is a classic algorithm for performing inference on graphical models. Given an edge set on a graph. A typical message passing algorithm such as sum-product can be used to find the marginal distribution of a tree structured graphical model. Consider the undirected tree graph with edge set, E, and vertices, V. The joint distribution is given by

$$p(x) = \frac{1}{Z}\Pi_{i\in V}\phi_u(x_i)\Pi_{(i,j)\in E}\phi_p(x_i, x_j). \tag{2.4.2}$$

A sum-product message passing permits us to compute the marginal probabilities at every

tree node in $O(n)$. We can define a message passing algorithm to compute the marginals at each node as follows:

$$M_{ij}(x_j) = \sum_{x'_i \in X_i} \phi_u(x'_i)\phi_p(x'_i, x_j)\Pi_{k \in Nbr(i) \setminus s} M_{ki}(x_i) \tag{2.4.3}$$

If the message passing update is repeated at each node the algorithm will converge for tree graphs to a unique solution. For graphs with cycles message passing can nonetheless yield an approximate solution. Message passing can be seen as an instance of dynamic programming.

Inspired by such algorithms Duvenaud et al [2015], Li et al. [2016] and others propose graph processing techniques which propagate information to each node in the graph in the same fashion, by iterative message passing. These can be generalized under the moniker of neural message passing algorithms. These generally consists of assigning each node a state vector and performing an update, parametrized by a neural network, of each nodes state vector based on it's neighbors [Gilmer et al., 2017]:

$$m_v^{t+1} = \sum_{w \in Nbr(v)} M_t(h_v^t, h_w^t) \tag{2.4.4}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}). \tag{2.4.5}$$

Here h is the state vector at each node v and the update steps M_t and U_t are parametrized by a neural network.

An advantage of such algorithms is that information can be propagated to distant parts of the graph in an efficient manner, the neural network only needs to learn the message passing rule. Finally we note the connection made by many authors to message passing algorithms and convolutional networks, which we use as inspiration in Chapter 5

Chapter 3

Convex Relaxations of Penalties for Sparse Correlated Variables With Bounded Total Variation

In this Chapter we discuss a method for statistical estimation with a signal known to be sparse, spatially contiguous, and containing many highly correlated variables. We follow the approach of constructing a tight convex relaxation and introduce the (k, s) *support total variation norm* as the *tightest* convex relaxation of the intersection of a set of discrete sparsity and total variation penalties. We show that this norm leads to an intractable combinatorial graph optimization problem, which we prove to be NP-hard. We then introduce a tractable relaxation. We devise several first-order optimization strategies for statistical parameter estimation with the described penalty. We demonstrate the effectiveness of this penalty on classification in the low-sample regime, classification with M/EEG neuroimaging data, and image recovery with synthetic and real data background subtracted image recovery tasks. We extensively analyse the application of our penalty on the complex task of identifying predictive regions from low-sample high-dimensional fMRI brain data, we show that our method is particularly useful compared to existing methods in terms of accuracy, interpretability, and stability.

Contents

3.1 Introduction

Regularization methods utilizing the ℓ_1 norm such as Lasso [Tibshirani, 1996a] have been used widely for feature selection. They have been particularly successful at learning problems in which very sparse models are required. However, in many problems a better approach is to balance sparsity against an ℓ_2 constraint. One reason is that very often features are correlated and it may be better to combine several correlated features than to select fewer of them, in order to obtain a lower variance estimator and better interpretability. This has led to the method of *elastic net* in statistics [Zou and Hastie, 2005], which regularizes with a weighted sum of ℓ_1 and ℓ_2 penalties. More recently, it has been shown that the elastic net is not in fact the tightest convex penalty that approximates sparsity (ℓ_0) and ℓ_2 constraints at the same time [Argyriou et al., 2012b]. The tightest convex penalty is given by the k-support norm, which is parametrized by an integer k, and can be computed efficiently. This norm has been successfully applied to a variety of sparse vector prediction problems [Gkirtzou et al., 2013b, McDonald et al., 2014, Misyrlis et al., 2014a].

We study the problem of introducing structural constraints to sparsity and ℓ_2, from first principles. In particular, we seek to introduce a total variation smoothness prior in addition to sparsity and ℓ_2 constraints. Total variation is a popular regularizer used to enforce local smoothness in a signal [Michel et al., 2011, Rudin et al., 1992, Tibshirani et al., 2005]. It has successfully been applied in image de-noising and has recently become of particular interest in the neural imaging community where it can be used to reconstruct sparse but locally smooth brain activation [Baldassarre et al., 2012b, Michel et al., 2011]. Two kinds of total variation are commonly considered in the literature, isotropic $TV_I(w) = \|\nabla w\|_{2,1}$ and anisotropic $TV_A(w) = \|\nabla w\|_1$ [Beck and Teboulle, 2009]. In our theoretical analysis we focus on the anisotropic penalty.

To derive a penalty incorporating these constraints we follow the approach of [Argyriou et al., 2012b] by taking the convex hull of the intersection of our desired penalties and then recovering a norm by applying the gauge function. We then derive a formulation for the dual norm which leads us to a combinatorial optimization problem, which we prove to be NP-hard. We find an approximation to this penalty and prove a bound on the approximation error. Since the k-support norm is the tightest relaxation of sparsity and ℓ_2 constraints, we propose to use the intersection of the TV norm ball and the k-support norm ball. This leads to a convex optimization problem in which (sub)gradient computation can be achieved with a computational complexity no worse than that of the total variation. Furthermore, our approximation can be computed for variation on an arbitrary graph structure.

We discuss and utilize several first order optimization schemes including stochastic subgradient descent, iterative Nesterov-smoothing methods, and FISTA with an estimated proximal operator. We demonstrate the tractability and utility of the norm through applications of classification on MNIST with few samples, M/EEG classification, and background-subtracted image recovery. For the problem of identifying predictive regions in fMRI we show that we can get improved accuracy, stability, and interpretability along with providing the user with several potential tools and heuristics to visualize the resulting predictive models. This includes

several interesting properties that apply to the special case of k-support norm optimization as well.

3.2 Convex Relaxation of Sparsity, ℓ_2 and Total Variation

In this section we formulate the (k, s) *support total variation* norm, a tight convex relaxation of sparsity, ℓ_2, and total variation (TV) constraints. We derive its dual norm which results in an intractable optimization problem. Finally we describe a looser convex relaxation of these penalties which leads to a tractable optimization problem.

3.2.1 Derivation of the Norm

We start by defining the set of points corresponding to simultaneous sparsity, ℓ_2 and total variation (TV) constraints:

$$Q^2_{k,s} := \{w \in \mathbb{R}^d : \|w\|_0 \leq k, \|w\|_2 \leq 1, \|Dw\|_0 \leq s\} \tag{3.2.1}$$

where $k \in \{1, \ldots, d\}, s \in \{1, \ldots, m\}$ and $D \in \mathbb{R}^{m \times d}$ is a prescribed matrix. The bound of one on the ℓ_2 term is used for convenience since the cardinality constraints are invariant under scaling. D generally take the form of a discrete difference operator, but the discussion in the following sections is more general than that. It is easy to see that the set $Q^2_{k,s}$ is not convex due to the presence of the $\|\cdot\|_0$ terms. Hence using $Q^2_{k,s}$ in a regularization method is impractical. Thus we consider instead the convex hull of $Q^2_{k,s}$:

$$C^2_{k,s} := \operatorname{conv}(Q^2_{k,s}) = \{w : w = \sum_{i=1}^{r} c_i z_i, \sum_{i=1}^{r} c_i = 1, c_i \geq 0, z_i \in \mathbb{R}^d, \tag{3.2.2}$$

$$\|z_i\|_0 \leq k, \|z_i\|_2 \leq 1, \|Dz_i\|_0 \leq s, r \in \mathbb{N}\} . \tag{3.2.3}$$

For some values of D, k and s, this convex set may not span the entire $\mathbb{R}^d$, that is, it may be contained within a smaller subspace. In Section 3.2.2 we show a condition for which the set will span $\mathbb{R}^d$ (see Proposition 1). For a matrix D that is the transpose of an incidence matrix representing a graph with a maximum degree of l_{deg}, the value of s should be greater than or equal to l_{deg}.

Assuming some mild technical conditions on D,[1] the convex set $C^2_{k,s}$ is the unit ball of a certain norm. We call this norm the (k, s) *support total variation* norm. It equals the gauge function of $C^2_{k,s}$, that is,

$$\|x\|^{sptv}_{k,s} := \inf\Big\{\lambda \in \mathbb{R}_+ : x = \lambda \sum_{i=1}^{r} c_i z_i, \sum_{i=1}^{r} c_i = 1, \tag{3.2.4}$$

$$c_i \geq 0, z_i \in \mathbb{R}^d, \|z_i\|_0 \leq k, \|z_i\|_2 \leq 1, \|Dz_i\|_0 \leq s, r \in \mathbb{N}\Big\}. \tag{3.2.5}$$

[1] The conditions are given in Proposition 1.

Performing a variable substitution we define a set of components of x, $v_i = \lambda c_i z_i \Rightarrow \lambda = \frac{\sum_{i=1}^{r} \|v_i\|_2}{\sum_{i=1}^{r} c_i \|z_i\|_2}$. To maximize the denominator for fixed v_i, we note that $\sum_{i=1}^{r} c_i \|z_i\|_2 \leq \left(\sum_{i=1}^{r} c_i\right) \max_{i=1}^{r} \|z_i\|_2 = 1$. The equality can be attained by applying the constraints in Equation (3.2.4). Substituting for λ and removing the constraints already applied above our norm now becomes

$$\|x\|_{k,s}^{sptv} = \inf\Big\{\sum_{i=1}^{r} \|v_i\|_2 : \sum_{i=1}^{r} v_i = x, \|v_i\|_0 \leq k, \|Dv_i\|_0 \leq s, r \in \mathbb{N}\Big\}. \tag{3.2.6}$$

The special case $s = m$ is simply the k-support norm [Argyriou et al., 2012b], which trades off between the ℓ_1 norm ($k = 1, s = m$) and the ℓ_2 norm ($k = d, s = m$). Formula 3.2.6 is combinatorial in nature and hence is difficult to directly include in an optimization problem.

3.2.2 Derivation of the Dual Norm

A standard approach for analyzing structured norms is through analysis of the dual norm [Argyriou et al., 2012b, Bach et al., 2012a, Mairal and Yu, 2013]. As such, it will be useful to derive an expression for the dual norm of $\|\cdot\|_{k,s}^{sptv}$. This will allow us to connect the norm with an optimization problem on a graph, use this to show the norm is NP-hard, and show an approximation bound (Proposition 2).

To obtain the dual of (k, s) support TV norm we first consider a more general class of norms. Each norm in this class is associated with a set of subspaces $S_1, \ldots, S_n$ and a set of norms $\|\cdot\|_{(1)}, \ldots, \|\cdot\|_{(n)}$. We assume that these subspaces span $\mathbb{R}^d$, that is, $\sum_{i=1}^{n} S_i = \mathbb{R}^d$, the summation here denotes addition of sets ($S_1 + S_2 = \{x : x = x_1 + x_2, x_1 \in S_1, x_2 \in S_2\}$). We may now define the following norm

$$\|w\| := \min\left\{\sum_{i=1}^{n} \|v_i\|_{(i)} \;:\; v_i \in S_i,\ \forall i \in \mathbb{N}_n,\ \sum_{i=1}^{n} v_i = w\right\} \ \forall w \in \mathbb{R}^d. \tag{3.2.7}$$

This is indeed a norm, since the subspaces span $\mathbb{R}^d$, and that the above minimum is attained. The (k, s) support TV norms can be written in the form (3.2.7) by specifying all n norms to be the ℓ_2 norm and the linear subspaces to correspond to the constraints on the supports.

We note that this definition is equivalent to an infimal convolution of n norms. Let δ_S denote the indicator function of a subspace S and the infimal convolution $(f_1 \square \ldots \square f_n)$ of n functions as $\square_{i=1}^{n} f_i$. Using this notation, the norm $\|\cdot\|$ can be written equivalently as $\|\cdot\| = \square_{i=1}^{n} \left(\|\cdot\|_{(i)} + \delta_{S_i}\right)$. We may derive the general form of the dual norm $\|\cdot\|^*$ of $\|\cdot\|$ by a direct application of standard duality results from convex analysis.

Lemma 3.1. *Let $\|\cdot\|_{(1)}, \ldots, \|\cdot\|_{(n)}$ be norms on $\mathbb{R}^d$ with duals $\|\cdot\|_{(1)*}, \ldots, \|\cdot\|_{(n)*}$, respectively, and let $S_1, \ldots, S_n$, be linear subspaces of $\mathbb{R}^d$ such that $\sum_{i=1}^{n} S_i = \mathbb{R}^d$. Then the dual norm of*

$\|\cdot\|$ defined in (3.2.7) is given by

$$\|u\|^* = \max_{i=1}^{n} \min\left\{\|u-q\|_{(i)*} : q \in S_i^{\perp}\right\} = \max_{i=1}^{n} \left(\|\cdot\|_{(i)*} \,\square\, \delta_{S_i^{\perp}}\right)(u) \tag{3.2.8}$$

for all $u \in \mathbb{R}^d$. The unit ball of $\|\cdot\|^*$ equals $B_* = \bigcap_{i=1}^{n}\left(B_{i*} + S_i^{\perp}\right)$ where B_{i*} denotes the unit ball of $\|\cdot\|_{(i)*}$ for $i = 1, \ldots, n$.

Proof: Denote *convex conjugate* or *Fenchel conjugate* of a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ by f^* [Bauschke and Combettes, 2011]. It is known that the convex conjugate of a norm equals the indicator function of its dual unit ball. Thus it holds that

$$\delta_{B_*} = \left(\square_{i=1}^{n}\left(\|\cdot\|_{(i)} + \delta_{S_i}\right)\right)^* . \tag{3.2.9}$$

Moreover, the conjugate of an infimal convolution equals the sum of conjugates [Bauschke and Combettes, 2011, Prop. 13.21]. The converse duality also holds under Slater type conditions [Bauschke and Combettes, 2011, Thm. 15.3]. Applying these facts successively, we obtain that

$$\delta_{B_*} = \sum_{i=1}^{n}\left(\|\cdot\|_{(i)} + \delta_{S_i}\right)^* = \sum_{i=1}^{n}\left(\|\cdot\|_{(i)}^* \,\square\, \delta_{S_i}^*\right) = \sum_{i=1}^{n}\left(\delta_{B_{i*}} \,\square\, \delta_{S_i}^*\right) . \tag{3.2.10}$$

We now use the facts that, for any subspace S, $\delta_S^* = \delta_{S^\perp}$ and that, for any nonempty sets $C, D \subseteq \mathbb{R}^d$, $\delta_C \,\square\, \delta_D = \delta_{C+D}$, obtaining that

$$\delta_{B_*} = \sum_{i=1}^{n}\left(\delta_{B_{i*}} \,\square\, \delta_{S_i^{\perp}}\right) = \sum_{i=1}^{n}\left(\delta_{B_{i*}+S_i^{\perp}}\right) . \tag{3.2.11}$$

It follows that $B_* = \bigcap_{i=1}^{n}\left(B_{i*} + S_i^{\perp}\right)$. The intersection of norm balls corresponds to maximum of the corresponding norms which gives the formula for $\|\cdot\|^*$. □

Equation (3.2.8) for the dual norm is interpreted as the maximum of the distances of x (with respect to the corresponding dual norms) from the orthogonal complements. We now specialize this formula to the case of (k, s) support TV norm.

Notation We define G_k as all subsets of $\{1, ..., d\}$ of cardinality at most k and M_s as all subsets of $\{1, ..., m\}$ of cardinality at most s. For every $I \in G_k$, we denote $I^c = \{1, ..., d\}\backslash I$ and for every $J \in M_s$, $J^c = \{1, ..., m\}\backslash J$. We denote D_{J^c} as the submatrix of D with only the rows indexed by J^c and for every $u \in \mathbb{R}^d$, u_I is the subvector of u with only the elements indexed by I.

It is the case that r in Equation (3.2.6) can be assumed to be at most $|G_k||M_s|$ (by grouping components with the same (I, J) pattern and applying the triangle inequality). We can now

reduce the dual norm to

$$(\|x\|_{k,s}^{sptv})^* = \max_{(I,J)\in G_k\times M_s} \min\{\|x-q\|_2 : q \in S_{I,J}^{\perp}\} = \max_{(I,J)\in G_k\times M_s} E_{I,J}(x) \tag{3.2.12}$$

where $S_{I,J} = \{x \,|\, D_{J^c}x = 0 \text{ and } x_{I^c} = 0\}$, $S_{I,J}^{\perp} = \text{range}(D_{J^c}^{\top}) + \{x \,|\, x_I = 0\}$, and $E_{I,J}$ is an energy function we will derive (cf. Equation (3.2.14)). Before proceeding we use the described subspaces to note the conditions for which $\|x\|_{k,s}^{sptv}$ is a full fledged norm.

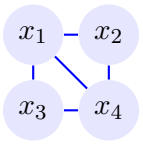
Proposition 1. *If*

$$\sum_{\substack{I\subseteq\{1,\ldots,d\},|I|=k\\ J\subseteq\{1,\ldots,m\},|J|=s}} S_{I,J} = \mathbb{R}^d \tag{3.2.13}$$

then $\text{span}\, C_{k,s}^2 = \mathbb{R}^d$.

This condition will depend on the choice of D, k and s. We choose D to be the transpose of the incidence matrix of a directed graph $G_d = (\mathcal{V}_d, \mathcal{E}_d)$, with the vertices corresponding to the elements of x. Furthermore $G = (\mathcal{V}, \mathcal{E})$ is an undirected graph with vertices $\mathcal{V} = \mathcal{V}_d$ and an unordered set of the same edges as $\mathcal{E}_d$. For a given J, we can consider the graph G_{J^c}, specified by the incidence matrix D_{J^c} as the original graph with $|J|$ edges removed. The notation presented is illustrated in Figure 3.1.

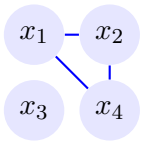
$$D = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \Rightarrow$$



(a)

$$J = \{3,2\}, J^c = \{1,4,5\}$$

$$D_{J^c} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \Rightarrow$$



(b)

Figure 3.1 – (a) Example of D matrix for a graph, and (b) an example D_{J^c} for a given instance of J. The graph in (b) has two subgraphs, one with nodes x_1, x_2, x_4 and the other the singleton, x_3.

We consider the linear constraints specified by $D_{J^c}x = 0$. Each row of the transpose incidence matrix, D_{J^c}, represents an edge $\mathcal{E}_{dij} = (i,j)$. Coupled with the constraint each of these rows corresponds to a constraint $x_i = x_j$. We note that this constraint is independent of the ordering on the graph. For any two vertices a, b of the undirected graph G, if there exists a path between a and b then $x_a = x_b$. More formally, if we divide G_{J^c} into all of its disjoint subgraphs denoted by $G_\gamma = (\mathcal{V}_\gamma, \mathcal{E}_\gamma)$,

$$G_{J^c} = \bigcup_{\gamma\in\Gamma} G_\gamma, \qquad (a,b) \in \mathcal{V}_\gamma \times \mathcal{V}_\gamma \Rightarrow x_a = x_b\ .$$

Thus for any disjoint subgraph of G_{J^c} we can take any tree containing the vertices of the subgraph and the associated incidence matrix will be a representation of the subspace associated with the components represented by those vertices.

Since each disjoint subgraph will have an independent set of constraints on its associated variables we can subdivide the linear constraints specifying $S^{\perp}_{I,J}$. Divide the graph corresponding to D_{J^c} into all disjoint subgraphs enumerated by $\Gamma = \{1, ...p\}$. Let $D_{J^c_\gamma}$ be the incidence matrix corresponding to each subgraph. Then $S_{I,J} = \left\{x \,|\, D_{J^c_\gamma} x = 0, \forall \gamma \in \Gamma, \text{ and } x_{I^c} = 0\right\}$ and $S^{\perp}_{I,J} = \sum_{\gamma \in \Gamma} \text{range}(D^{\top}_{J^c_\gamma}) + \{x \,|\, x_I = 0\}$. A direct computation yields the projection on each subgraph V_γ as

$$P_\gamma = D_{J^c_\gamma}(D_{J^c_\gamma})^{+} = \mathbf{I} - \tfrac{1}{n_\gamma}\mathbf{1}$$

if the subgraph has n_γ vertices. P_γ is exactly a centering matrix that projects orthogonal to the vector of all ones.

To compute the value of $E_{I,J}(x)$ we can split the parameters of x into independent groups, since the projection and thereby the residual of components corresponding to vertices in disjoint groups will have independent contribution. The components of $\text{Proj}_{S_{I,J}}(x)$ at I^c must be zero. Moreover, the members of any group that contains a vertex from I^c will be zero. We can therefore compute $E_{I,J}(x)$ independently for each disjoint group, and only for the groups that do not contain a vertex in I^c. For each disjoint group the contribution to $E^2_{I,J}(x)$ is

$$E^2_\gamma(x) = \|(\mathbf{I} - P_\gamma)x\|^2 = \frac{1}{n_\gamma}\left(\sum_{i \in V_\gamma} x_i\right)^2 . \tag{3.2.14}$$

A graph based version of the combinatorial optimization problem is as follows. Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ and $I \subset \mathcal{V}$, $J \subset \mathcal{E}$, remove edges J and all disjoint subgraphs containing a vertex in I^c to obtain a graph G_{IJ}. The energy over this graph, $E^2_{I,J}$ can be computed as the sum of E^2_γ over all disjoint subgraphs in G_{IJ}. The dual norm is then given by Equation (3.2.12).

We can additionally show that we can limit M_s to maximum cardinality sets (cardinality s) and G_k to maximum cardinality sets (cardinality k). Indeed, adding indexes in I or J cannot decrease $S_{I,J}$ and hence cannot decrease the norm of the projection in Equation (3.2.12). Thus we can narrow the problem to removing s edges and $d - k$ nodes (with their associated subgraphs).

We have now reduced the computation of the dual norm to a graph partitioning problem. Graph partition problems are often NP-hard, and we show this to be the case here as well:

Theorem 3.1. *Computation of the (k, s) support total variation dual norm is NP-hard*

Proof: We show that the (k, s)-support TV dual norm is an NP-hard problem, by reduction from minimum weight multiway cut problem [Vazirani, 2001]. Let the dual norm computation problem be denoted $P(z, D, s, k)$, where z is the input for the dual norm. We limit to the set of inputs where $k = d$, where d is the cardinality of z and D is an incidence matrix of a graph $G = (V, E)$ with vertex weights $z_i = w(v_i)$, $v_i \in V$. Additionally for simplifcation in later steps let $e = d - s$. This problem is refered to simply as $P1(G, e)$ and can be stated

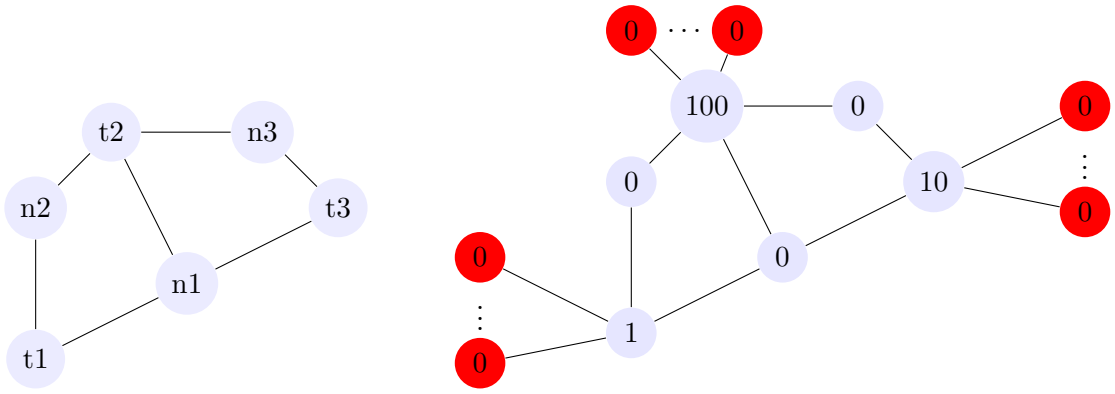


Figure 3.2 – Original unweighted graph of the 3-way mincut problem and the augmented weighted graph we construct as the input graph for problem P1.

as follows: given a graph $G = (V, E)$ partition G, $G_p = (V, E_p)$ obtained by removing any e edges from E which maximizes

$$\max \sum_{V_i \in \{G_1, G_2, \ldots, G_k\}} \frac{1}{|G_i|} \left(\sum_{v_j \in V_i} w(v_j) \right)^2,$$

where $G_p = G_1 \cup G_2 \cup \cdots \cup G_k$ and elements of G are disjoint. V_i is the vertex set for G_i and $w(v_j)$ is the weight for vertex v_j.

The minimum 3-way cut problem (which we denote P_{M3}) is NP-hard. The problem can be stated as follows: given a graph $G = (V, E)$ and terminals $t_1, t_2, t_3 \in V$, find a minimum set of edges $E' \subseteq E$ such that the removal of E' from E disconnects each terminal t_i from the others. Furthermore the decision problem,denoted P_{M3D}, is to find out if its possible to disconnect t_1, t_2, t_3 by removing no more than e edges, where e is a part of the input. This problem is NP-complete. We show that any instances of P_{M3D} can be reduced in polynomial time to an instance of $P1$. Given an instance of P_{M3D} we have a graph G, terminals t_1, t_2, t_3 and integer e. We construct a new graph G_{aug} as follows

- We add weights to the vertices of the graph G, weighting non-terminal nodes 0 and the terminal nodes $1, 10, 100$ in any order.
- For each terminal we add N (the choice of N is described later on) more vertices to be its neighbor and weight these vertices 0. These augmented vertices are connected to the original graph only at the terminal vertices.

Figure 3.2 shows an example of an instance of P_{M3D} and the constructed augmented graph. We can now compute $P1(G_{aug}, e)$.

If $N > |V|$ (the number of vertices in G) then none of the edges connected to the new vertices of G_{aug} will be removed by $P1$ since disconnecting two terminals from each other will always improve the result more than disconnecting one of the 0 nodes in the augmented vertices. Denoting the number of nodes in G (original graph) n, if $P1$ disconnects terminals $t1$, $t2$, and

$t3$ the solution takes the form: $\frac{1}{N+n_1} + \frac{100}{N+n_2} + \frac{10000}{N+n_3}$ where $n_1 + n_2 + n_3 = n$. We can lower bound this value as $\frac{1}{N+n} + \frac{100}{N+n} + \frac{10000}{N+n} = \frac{10101}{N+n}$.

If $P1$ does not disconnect the terminals the solution can take on one of 4 forms, each of which can be upper bounded. For example,

$$\frac{11^2}{2N+n_1} + \frac{10000}{N+n_2} < \frac{11^2}{2N} + \frac{10000}{N} = \frac{10060.5}{N}. \tag{3.2.15}$$

Similarly for the other 3 cases, $\frac{1}{N+n_1} + \frac{101^2}{2N+n_2} < \frac{5101.5}{N}$, $\frac{1^2}{N+n_1} + \frac{110^2}{2N+n_2} < \frac{6051}{N}$, $\frac{111^2}{3N+n} < \frac{4107}{N}$, where $n_1 + n_2 = n$.
Examining the above inequalities we state that if terminals $t1, t2$, and $t3$ are not connected the solution will be at most $\frac{10060.5}{N}$. For $N > \frac{10060.5}{40.5}n$, the inequality $\frac{10101}{N+n} > \frac{10060.5}{N}$ always holds. Thus if it is possible to disconnect the terminals with e edges $P1$ will produce a value greater than $\frac{10060.5}{N}$ answering P_{M3D}. Since solutions of P_{M3D} are obtained in polynomial calls to $P1$, $P1$ is NP-hard. □

Corollary 3.1. *Computation of the (k, s) support total variation norm is NP-hard.*

In light of this Theorem, we are unable to incorporate the (k, s) support total variation norm in a regularized risk setting. Instead in the sequel we examine a tractable approximation with bounds that scale well for the family of graphs of interest.

3.2.3 Approximating the Norm

Although special cases where s equals m or 1 are tractable, the general case for arbitrary values of s leads to an NP-hard graph partitioning problem for the dual norm, implying the norm itself is intractable. We thus relax the problem by taking instead the intersection of the k-support norm ball and the convex relaxation of total variation. This leads to the following penalty

$$\Omega_{sptv}(w) = \max\{\|w\|_k^{sp}, \tfrac{1}{\sqrt{s}\|D\|}\|Dw\|_1\}, \tag{3.2.16}$$

where $\|\cdot\|$ denotes the spectral norm. We can bound the error of this approximation as follows:

Proposition 2. *For every $w \in \mathbb{R}^d$, it holds that*

$$\Omega_{sptv}(w) \leq \|w\|_{k,s}^{sptv} . \tag{3.2.17}$$

Moreover, suppose that $\mathrm{range}(D^\top) = \mathbb{R}^d$ *and that for every $I \in G_k$ the submatrix D_{*I} has at least $m - s$ zero rows. Then it holds that*

$$\|w\|_{k,s}^{sptv} \leq \sqrt{1 + \frac{s\|D\|^2\|(D^\top)^+\|_\infty^2}{k}}\, \Omega_{sptv}(w) \tag{3.2.18}$$

where $\|\cdot\|_\infty$ *is the norm on* $\mathbb{R}^{m\times d}$ *induced by* ℓ_∞*, that is,* $\|A\|_\infty = \max_{i=1}^{m} \sum_{j=1}^{d} |A_{ij}|$.

Proof: First, note that $\|\cdot\|_k^{sp} \leq \|\cdot\|_{k,s}^{sptv}$. This follows directly from the definition of $\|\cdot\|_{k,s}^{sptv}$, since

$$\|w\|_k^{sp} = \left\|\sum_{i=1}^{r} v_i\right\|_k^{sp} \leq \sum_{i=1}^{r} \|v_i\|_k^{sp} = \sum_{i=1}^{r} \|v_i\|_2 \tag{3.2.19}$$

for every $v_i \in \mathbb{R}^d$ such that $\|v_i\|_0 \leq k$, $i = 1, \ldots, r$, and $w = \sum_{i=1}^{r} v_i$. Now let $v_i \in \mathbb{R}^d$ such that $\|Dv_i\|_0 \leq s$, $i = 1, \ldots, r$, and $w = \sum_{i=1}^{r} v_i$. Then

$$\|Dw\|_1 = \left\|\sum_{i=1}^{r} Dv_i\right\|_1 \leq \sum_{i=1}^{r} \|Dv_i\|_1 \leq \sum_{i=1}^{r} \sqrt{s}\|Dv_i\|_2 \leq \sqrt{s}\|D\| \sum_{i=1}^{r} \|v_i\|_2 \,. \tag{3.2.20}$$

The above two inequalities imply Equation (3.2.17).

For Equation (3.2.18), it suffices to show the dual inequality. Recall from Argyriou et al. [2012b] that the norm defined by $\|u\|_{(k)}^{(2)} := \left(\sum_{i=1}^{k} (|u|_i^{\downarrow})^2\right)^{\frac{1}{2}}$ is the dual of $\|\cdot\|_k^{sp}$. This is the ℓ_2 norm of the largest k entries in $|u|$, and is known as the 2-k *symmetric gauge norm* [Bhatia, 1997]. Thus, for every $a, w \in \mathbb{R}^d$, it holds that

$$\langle x - D^\top a, w\rangle \leq \|x - D^\top a\|_{(k)}^{(2)} \|w\|_k^{sp} \leq \|x - D^\top a\|_{(k)}^{(2)} \Omega_{sptv}(w) \tag{3.2.21}$$

$$\langle D^\top a, w\rangle = \langle a, Dw\rangle \leq \|a\|_\infty \|Dw\|_1 \leq \sqrt{s}\|D\|\|a\|_\infty \Omega_{sptv}(w) \tag{3.2.22}$$

Adding up and taking the infima with respect to a, we obtain

$$\langle x, w\rangle \leq \inf_{a\in\mathbb{R}^d} \left\{\|x - D^\top a\|_{(k)}^{(2)} + \sqrt{s}\|D\|\|a\|_\infty\right\} \Omega_{sptv}(w). \tag{3.2.23}$$

and hence

$$\Omega_{sptv}^*(x) \leq \inf_{a\in\mathbb{R}^d} \left\{\|x - D^\top a\|_{(k)}^{(2)} + \sqrt{s}\|D\|\|a\|_\infty\right\} \,. \tag{3.2.24}$$

Next we pick I to be the set of indexes corresponding to the largest k elements of $|x|$. We also pick

$$a = (D^\top)^+ c, \qquad c_i = \begin{cases} \operatorname{sgn}(x_i)\|x_{I^c}\|_\infty & \text{if } i \in I \\ x_i & \text{if } i \in I^c \,. \end{cases} \tag{3.2.25}$$

Since range$(D^\top) = \mathbb{R}^d$, it holds that $D^\top a = c$ and hence we obtain

$$\|x - D^\top a\|_{(k)}^{(2)} + \sqrt{s}\|D\|\|a\|_\infty = \sqrt{\sum_{i\in I}(|x_i| - \|x_{I^c}\|_\infty)^2} + \sqrt{s}\|D\|\|(D^\top)^+ c\|_\infty \tag{3.2.26}$$

$$\leq \sqrt{\sum_{i\in I}(x_i^2 - \|x_{I^c}\|_\infty^2)} + \sqrt{s}\|D\|\|(D^\top)^+\|_\infty\|x_{I^c}\|_\infty \tag{3.2.27}$$

$$= \sqrt{\sum_{i\in I} x_i^2 - k\|x_{I^c}\|_\infty^2} + \sqrt{s}\|D\|\|(D^\top)^+\|_\infty\|x_{I^c}\|_\infty \leq \sqrt{1 + \frac{s\|D\|^2\|(D^\top)^+\|_\infty^2}{k}}\,\|x_I\|_2. \tag{3.2.28}$$

By the hypothesis, we may choose $J \in M_s$ such that $D_{J^c I} = 0$. Then

$$\|x_I\|_2 = \max_{K\in M_s} \|\operatorname{Proj}_{\text{null}(D_{K^c I})}(x_I)\|_2 \leq (\|x\|_{k,s}^{sptv})^* \tag{3.2.29}$$

□

We note that we can fulfill the technical condition on the range of D^T by augmenting the incidence matrix in a manner that does not change the result of the regularized risk minimization. The condition that the submatrix D_{*I} has at least $m - s$ zero rows has an intuitive interpretation when D is the transpose of an incidence matrix of a graph. It means that any group of k vertices in the graph involves at most s edges. This is true in many cases of interest, such as grid structured graphs if s is proportional to k. The term involving $\|(D^\top)^+\|_\infty^2$ is at most linear in the number of vertices. $\|D\|^2$ corresponding to the maximum eigenvalue of the graph Laplacian is bounded above by a constant for a given structure (e.g. 2-D with neighborhood of 4).

We have proposed a tractable approximation to the (k, s) support total variation norm, which was shown to be NP-hard. We now discuss some optimization strategies for this approximate penalty and demonstrate several experiments showing its utility.

3.2.4 Optimization

Denoting $\hat{f}(w)$ as a loss function, $\Omega_{sptv}(w)$ as given by Equation (3.2.16), and $\lambda > 0$. It can be shown that, given appropriate parameter selection, the solution to a regularized risk minimization of $\hat{f}(w)$ constrained by $\Omega_{sptv}(w) \leq \lambda$ will be equivalent to optimizing any of the

following objectives for some regularization parameters $\lambda_1, \lambda_2 > 0$. [2]

$$\min_w \hat{f}(w) + \lambda_1 (\|w\|_k^{sp})^2 + \lambda_2 TV(w) \tag{3.2.30}$$

$$\min_w \hat{f}(w) + \lambda_1 \|w\|_k^{sp} + \lambda_2 TV(w) \tag{3.2.31}$$

$$\min_w \hat{f}(w) + \lambda_2 TV(w) \quad s.t. \ \|w\|_k \leq \lambda_1 \tag{3.2.32}$$

We analyze several optimization strategies for optimizing the prescribed objectives: Iterated FISTA with a smoothed $TV(w)$, FISTA with an approximate computation of the $\|w\|_k^{sp} + TV(w)$, and the Excessive Gap Method. A common concern in TV related optimization is the convergence. The former two methods have previously shown good empirical and theoretical convergence [Dohmatob et al., 2014, Dubois et al., 2014] and we describe specifics of their implementation with our objective below. However,these approaches do not provide optimality guarantees on the solution. For solving Equation (3.2.32) we may apply the Excessive Gap Method, which has convergence guarantees on the duality gap. We describe the non-trivial analysis required for applying the excessive gap method on our objective, which also requires the newly derived k-support ball projection operator in Section 3.2.4.1. We note that this section constitutes a preliminary proposal demonstrating our objectives can be optimized with state-of-the-art convex optimization methods. A detailed analysis of the optimization is beyond the scope of this work, and we utilize a combination of the methods described throughout our experiments.

In Iterated FISTA, we may utilize the proximal operator for k-support along with Nesterov smoothing on the $TV(w)$ term to make it differentiable [Dohmatob et al., 2014, Nesterov, 2004]. We can follow a strategy of repeatedly solving a FISTA problem with progressively decreasing smoothing parameter on the $TV(w)$ term as per [Dubois et al., 2014], who provide analysis of such an approach, which they call CONESTA. This technique can be used to solve any of Equations (3.2.31), (3.2.30), (3.2.32) given the relevant proximal mapping discussed in Section 3.2.4.1

We can estimate the proximal operator of $\lambda_1 \|w\|_k^{sp} + \lambda_2 TV(w)$ using an accelerated proximal gradient method in the dual, as described in Beck and Teboulle [2009], and the projection operator onto the $\|w\|_k^{sp}$ dual ball given in Chatterjee et al. [2014]. This allows us another approach of directly applying FISTA, but with the inexact proximal operator in order to solve Equation (3.2.31).

To apply the Excessive Gap Method to k-support TV regularizations we note the primal and the dual of Equation (3.2.32) can be written as $\min_{\|w\|_{ksp} \leq \lambda_1} f(w) = \max_{\|u\|_\infty < 1} \phi(u)$ where the primal is given $f(w) = \hat{f}(w) + \max_{\|u\|_\infty < 1} \{\langle Dw, u \rangle\}$, and the dual is given by $\phi(u) = -\hat{\phi}(u) + \langle Dw_u^*, u \rangle +$

[2] The proof of this statement follows from the fact that optimization subject to the intersection of two constraints has a Lagrangian that is exactly a regularized risk minimization with the two corresponding penalties each with their own Lagrange multiplier.

$\hat{f}(w_u^*)$ with $w_u^* = \underset{\|w\|_k^{sp} \leq \lambda_1}{\arg\min} \langle Dw, u \rangle + \hat{f}(x)$. We can now smooth the primal function

$$f_\mu(w) = \hat{f}(w) + \max_{\|u\|_\infty < 1} \{\langle Dw, u \rangle - \mu \|u\|^2\} = \hat{f}(w) + \langle Dw, u_\mu(x) \rangle - \mu \|u_\mu(x)\|^2$$

The excessive gap method now allows us to take successive approximations of $f_\mu(x)$ with a decreasing sequence of μ while maintaining a bound on the duality gap proportional to μ. To apply the excessive gap method we need the smooth approximations $u_\mu(x)$ and the gradient mappings $T_\mu(x)$, defined in [Nesterov, 2005]. We can obtain these using the simple projection of a vector, z, onto the ℓ_∞ ball,which we denote $P_{\|\cdot\|_\infty \leq 1}(z)$, obtained by truncating all values above magnitude 1. The relevant operations are then given by

$$u_\mu(w) = \underset{\|u\|_\infty \leq 1}{\arg\min} \{\langle Dw, u \rangle - \mu \|u\|^2\} = P_{\|\cdot\|_\infty \leq 1}\left(\frac{Dw}{2\mu}\right)$$

$$T_\mu(u) = \underset{\|u\|_\infty \leq 1}{\arg\max} \left\{\langle \nabla \phi, y - u \rangle - \frac{L_\phi}{2} \|y - u\|^2\right\} = P_{\|\cdot\|\infty \leq 1}\left(u + \frac{Dx(u)}{L_\phi}\right)$$

The sub-problem of finding $x(u)$ can be solved using an accelerated projected gradient method and the projection onto the k-support ball derived in Section 3.2.4.1.

3.2.4.1 Proximal Operators Associated With The k-support Norm

The proximal operator for $(\|w\|_k^{sp})^2$, associated with Equation (3.2.30) is given by McDonald et al. [2014]. The proximal operator for $\|w\|_k^{sp}$, associated with Equation (3.2.31), is given by Chatterjee et al. [2014]. In turn we can obtain the projection on the dual ball using Moreau decomposition [Parikh et al., 2014]. The projection onto the $\|w\|_k^{sp}$ ball (proximal of the indicator function) is not yet addressed in the literature to the best of our knowledge and we show below how to obtain this projection. We define δ_{C_λ} as the indicator function on the k-support ball of size λ, C_λ. We note that k-support norm is given by

$$\|w\|_k^{sp} = \left(\sum_{i=1}^{k-r-1} (|w|_i^\downarrow)^2 + \frac{1}{r+1}\left(\sum_{i=k-r}^{d} |w|_i^\downarrow\right)^2\right)^{\frac{1}{2}} \tag{3.2.33}$$

where $|w|_i^\downarrow$ is the ith largest element of w. The projection onto $\|w\|_k^{sp}$ is given by:

Theorem 3.2. *Given $\lambda > 0$ and $x \in R^p$, if $\|x\|_k^{sp} < \lambda$, then the projection,$w^* = prox_{\delta_{C_\lambda}}(x)$,is simply x. If $\|x\|_k^{sp} > \lambda$, define $D_r = \sum_{i=1}^{k-r-1} (|x|^\downarrow)^2, T_{r,l} = \sum_{i=k-r}^{l} |x|^\downarrow$, and $n = l - k + r + 1$, and construct the equation for $\beta_{r,l}$:*

$$\beta^2 D_r + \left(\frac{(\beta+1)\beta(r+1)T_{r,l}}{n + \beta(r+1)}\right)^2 - \lambda^2(\beta+1)^2 = 0. \tag{3.2.34}$$

The projection onto the k-support ball is given by finding r, l which satisfy the conditions:

$$|x|^{\downarrow}_{k-r-1} > \frac{(\beta+1)T_{r,l}}{n+\beta(r+1)} \geq |x|^{\downarrow}_{k-r} \ , \ |x|^{\downarrow}_{l} > \frac{T_{r,l}}{n+\beta(r+1)} \geq |x|^{\downarrow}_{l+1}, \tag{3.2.35}$$

where β is a non-negative solution to Equation (3.2.34). Furthermore the binary search specified in Chatterjee et al. [2014, Algorithm 2] with Equation (3.2.34) can be used to find the appropriate r and l in O(log(k) log(d-k)).

Proof Sketch: Argyriou et al. [2012b, Algorithm 1] specifies conditions on the proximal map of $\frac{1}{2\beta}(\|w\|_k^{sp})^2$. For a given β there must be a corresponding λ such that $\|w\|_k^{sp} = \lambda$, and therefore $\|prox_{\frac{1}{2\beta}(\|w\|_k^{sp})^2}(x)\|_k^{sp} = \lambda$. Substituting Equation (3.2.33) and explicit form and constraints for $prox_{\frac{1}{2\beta}(\|w\|_k^{sp})^2}(x)$ in Argyriou et al. [2012b, Algorithm 1] we obtain Equation (3.2.34) when the constraints are satisfied. Chatterjee et al. [2014, Theorem 3], holds since the constraints are the same. □

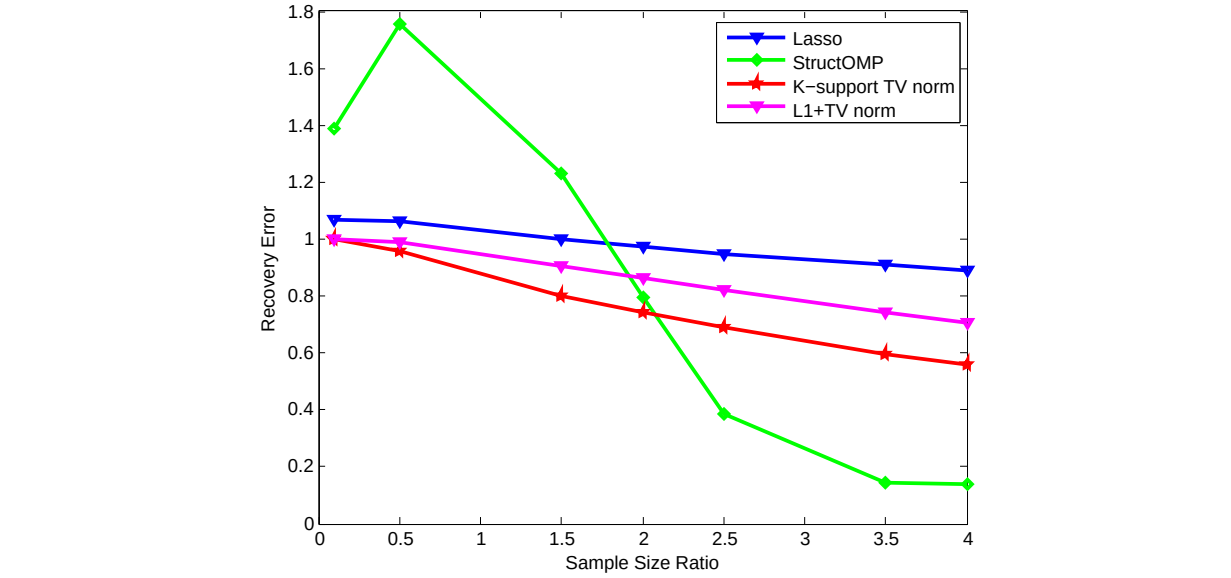
3.3 Experimental Results

We evaluate the effectiveness of the introduced penalty on signal recovery and classification problems. We consider a sparse image recovery problem from compressed sensing, a small training sample classification task using MNIST, an M/EEG prediction task, and classification and recovery task for fMRI and synthetic data. We compare our regularizer against several common regularizers (ℓ_1 and ℓ_2) and popular structured regularizers for problems with similar structure. In recent work TV+ℓ_1, which adds the TV and ℓ_1 constraints, has been heavily utilized for data with similar spatial assumptions [Dohmatob et al., 2014, Gramfort et al., 2013] and is thus one of our main benchmarks. Source code for learning with the k-support/TV regularizer is available at `https://github.com/eugenium/StructuredSparsityRegularization`.

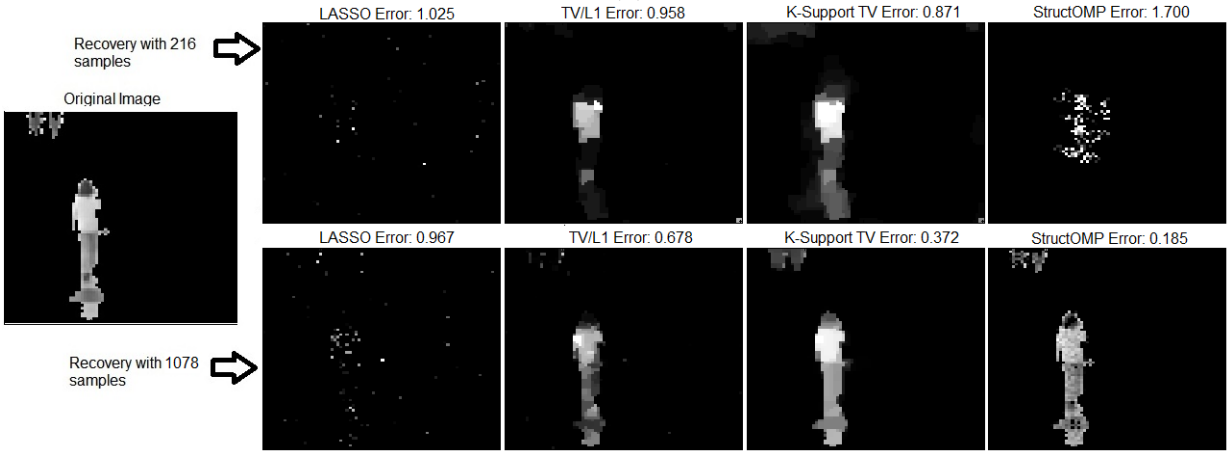
3.3.1 Background Subtracted Image Recovery

We apply k-support total variation regularization to a background subtracted image reconstruction problem frequently used in the structured sparsity literature [Baldassarre et al., 2012a, Huang et al., 2009]. We use a similar setup to Baldassarre et al. [2012a]. Here we apply m random projections to a background-subtracted image along with Gaussian noise, and reconstruct the image using the projections and projection matrices. Our evaluation metric for the recovery is the mean squared pixel error. For this experiment we utilize the a squared loss function and the iterative FISTA with smoothed TV described in Section 3.2.4.

We selected 50 images from the background segmented dataset and converted them to grayscale. We use squared loss and k-support total variation to reconstruct the original images. We com-



(a)



(b)

Figure 3.3 – (a) Average model error for background subtracted image reconstruction for various sample sizes. (b) Image example for different methods and sample sizes. k-support/TV regularization gives the best recovery error for 216 samples, and gives smoother recovery results than the other methods for both sample sizes.

pute normalized recovery error for different number of samples m and compare our regularizer to LASSO, TV+ℓ_1, and StructOMP. The latter is a structured regularizer which performs best on this problem in Huang et al. [2009]. The average normalized recovery error is shown for different sample sizes in Figure 3.3.1(a). We used a separate set of images to set the parameters for each method.

In terms of recovery error we note that k-support total variation substantially outperforms LASSO and TV+ℓ_1, and outperforms StructOMP for low sample sizes. Further examination of the images reveals other advantages of the k-support total variation regularizer. An example for one image recovery scenario is shown at 2 different sample sizes in Figure 3.3.1(b). Here we can see that at low sample sizes StructOMP and LASSO can completely fail in terms of creating a visually coherent reconstruction of the image. TV+ℓ_1 recovery at the low sample size improves upon the latter methods, producing smooth regions, but still not resembling the human shape pictured in the original image. k-support total variation has better visual quality at this low sample complexity, due to its ability to retain multiple groups of correlated variables in addition to the smoothness prior. For the case of a larger number of samples, illustrated by the bottom row of Figure 3.3.1(b), we note that although the recovery performance of StructOMP is better (lower error), the visual quality of the k-support total variation regualrizer produces smoother and more coherent image segments.

3.3.2 Low Sample Complexity MNIST Classification

We consider a simple classification problem using the MNIST data set [LeCun and Cortes, 2010]. We select a very small subset of data to train with in order to demonstrate the effectiveness of our regularizer. We train a one versus all classifier for each digit. In the case of each digit we take 9 negative training samples, one from each other digit, and 9 positive training samples of the digit. We use a validation set consisting of 8000 examples to perform parameter selection. We use a regularized risk function consisting of the form (3.2.30) and logistic loss. Optimization for a single parameter setting took on the order of one second for a MatLab implementation on a 2.8 GHz core. We choose the best model parameters from $k \in \{1, 2^3, 2^5, 2^7, 2^9, d\}$, $\lambda_1 \in \{\frac{10^5}{N}, .., \frac{10^2}{N}\}$, and $\lambda_2 \in \{0, \frac{10^3}{N}, .., \frac{10^{-1}}{N}\}$, where N is the training set size. Here d corresponds to the image size (28×28) and the cases $k = 1$ and $k = d$ correspond the ℓ_1 and ℓ_2 norm, respectively, when $\lambda_2 = 0$. We test on the entire MNIST test set of 10000 images. We optimize a logistic loss function combined with our k-support total variation norm and compare to results from ℓ_1, ℓ_2, k-support norm, and TV/ℓ_1 penalties combined with logistic loss. We perform optimization using FISTA on the k-support norm [Argyriou et al., 2012b, Nesterov, 2004] and a smoothing applied to the total variation. For the graph structure, specified by D, we use a grid graph with each pixel having a neighborhood consisting of the 4 adjacent pixels. We obtain surprisingly high classification accuracy using just 18 training examples. The results in Table 3.1 show classification accuracy for each one versus all classifier and the average of the classifiers. In all but two cases the k-support TV norm outperforms the other regularizers. We note that for the digit 9 classification the difference between the best classifier and k-support/TV is not statistically significant

Class.	ℓ_1	ℓ_2	**KS**	ℓ_1**+TV**	**KS+TV**
D0	$93.62 \pm .01$	$93.49 \pm .01$	$93.68 \pm .02$	$96.22 \pm .01$	$\mathbf{96.27 \pm .01}$
D1	$90.1 \pm .02$	$89.56 \pm .02$	$90.08 \pm .02$	$90.57 \pm .02$	$\mathbf{92.18 \pm .02}$
D2	$78.28 \pm .03$	$77.28 \pm .03$	$78.25 \pm .03$	$\mathbf{81.47 \pm .02}$	$81.39 \pm .03$
D3	$68.58 \pm .02$	$68.05 \pm .02$	$68.60 \pm .02$	$71.63 \pm .02$	$\mathbf{73.25 \pm .02}$
D4	$83.81 \pm .01$	$82.55 \pm .01$	$83.76 \pm .01$	$84.69 \pm .01$	$\mathbf{84.79 \pm .01}$
D5	$73.7 \pm .03$	$73.2 \pm .02$	$73.69 \pm .03$	$74.52 \pm .02$	$\mathbf{74.95 \pm .02}$
D6	$93.48 \pm .01$	$93.37 \pm .01$	$93.51 \pm .01$	$93.71 \pm .01$	$\mathbf{94.08 \pm .01}$
D7	$88.88 \pm .02$	$87.21 \pm .02$	$88.85 \pm .02$	$91.67 \pm .01$	$\mathbf{92.59 \pm .01}$
D8	$70.79 \pm .02$	$72.07 \pm .03$	$72.75 \pm .02$	$73.23 \pm .02$	$\mathbf{73.10 \pm .02}$
D9	$85.48 \pm .02$	$85.61 \pm .02$	$85.49 \pm .02$	$85.5 \pm .03$	$85.60 \pm .03$

Table 3.1 – Accuracy for One versus All classifiers on MNIST using only 18 training examples and standard error computed on the test set. In all but two cases, k-support/TV regularization gives the best performance with significance. For digit '9' k-support/TV regularization is statistically tied for best performance.

3.3.3 M/EEG Prediction

We apply k-support total variation regularization to an M/EEG prediction problem from Backus et al. [2011], Zaremba et al. [2013], using the preprocessing from Zaremba et al. [2013]. This results in data samples with 60 channels, each consisting of a time-series presumed to be independent across channels. Following Zaremba et al. [2013] we report results for subject 8 from this dataset. For the total variation graph structure, we impose constraints for adjacent samples within each channel, while values from different channels are not connected within the graph. In the original work a latent variable SVM with delay parameter h is used to improve alignment of the samples. We consider only the case for $h = 0$, which reduces to the standard SVM. To directly compare our results we utilize hinge loss with a constant C of 2×10^4, the same regularization value used in Zaremba et al. [2013]. Thus we optimize the following objective

$$R(w) = \frac{C}{N} \sum_{i=1}^{N} \max\{0, 1 - y_i \langle w, x_i \rangle\} + (1 - \lambda)(\|w\|_k^{sp})^2 + \lambda \|Dw\|_1 \tag{3.3.1}$$

Where λ allows us to easily trade off between k-support and total variation norms, while maintaining a fixed weight for our regularizer comparable to Zaremba et al. [2013]. We use $k = 2500$ (approximately 80% of the dimensions) and $\lambda = 0.1$. Table 3.2 shows the mean and standard deviation for the classification accuracy. We use the same partitioning of the data as described in [Zaremba et al., 2013], and on average obtain an improvement over the original results. We note that TV+ℓ_1 regularization has relatively poor performance. We hypothesize this is because the data used is very noisy and not very sparse.

3.3.4 Prediction and Identification in fMRI analysis

In this section we demonstrate the advantages of our sparse regularization method in the analysis of fMRI neuro-imaging data. Brain activation in response to stimuli is normally assumed to be sparse and locally contiguous, thus our proposed regularizer is ideal for de-

Classifier	Mean Acc.	Acc std.
SVM [Zaremba et al., 2013]	65.44%	2.29%
ksp-TV SVM	66.84%	3.42%
TV-ℓ_1 SVM	60.70%	4.66%

Table 3.2 – M/EEG accuracy for SVM, k-support total variation regularized SVM, and TV+ℓ_1 regularized SVM computed over 5 folds. k-support/TV regularization yields the best results on average.

scribing our prior assumptions on this signal. An important aspect of analysing fMRI data is the ability to demonstrate how the predictive variables identified as important by an estimator correspond to relevant brain regions. Regularized risk minimization is one of few approaches which can handle the multivariate nature of this problem. However, in the presence of many highly correlated variables, such as those in brain regions with many adjacent voxels being activated by a stimulus, using sparse regularization alone there may be many possible solutions with near equivalent predictive performance for small training sample size. Furthermore, from a practical standpoint, overly sparse solutions can be difficult to interpret when attempting to determine an implicated brain region. Thus regularization here allows us to not only converge to a good solution with lower sample complexity, but obtain more interpretable models from amongst the space of solutions with good prediction. Related to interpretability is solution stability, solutions which are more stable under different samples of training data, with regards to implicated voxels/regions allow the practitioner to make a more trustworthy interpretations of the model [Misyrlis et al., 2014a, Yan et al., 2014]. We evaluate our approach taking all these factors into account.

We first analyze our method using a synthetic simulation of a signal similar to brain activation patterns. This gives us the opportunity to assess the true support recovery performance, which we cannot obtain with real data. We then analyze a popular block-design fMRI dataset from a study on face and object representation in the human ventral temporal cortex [Dohmatob et al., 2014] and perform experiments on predicting and in turn utilizing the predictive models for identifying the relevant regions of interest. We attempt to classify scans taken when a user is shown a pair of scissor vs. when they observe scrambled pixels. We demonstrate that we can obtain improved accuracy, solution interpretability, and stability characteristics compared to previously applied sparse regularization methods incorporating spatial priors. For these experiments we use logistic loss and the $TV_I(w)$ penalty, which has been shown to work better in fMRI analysis. Optimization is done using FISTA and estimated proximal operator. As our baseline we focus on TV+ℓ_1 which has been recently popularized for fMRI applications as well as TV+ℓ_1+ℓ_2, which has been considered in structural MRI [Dubois et al., 2014].

We consider the estimation of an ideal weight vector with both spatial correlation and sparsity similar to brain activation patterns with spatial correlations between neurons which are active and not-active and the activated neurons often occuring in adjacent regions of the brain. We construct a 25x25 image with 84% of coefficients set to zero. The non-sparse portion of the image corresponds to Gaussian blobs. This image will serve as a set of parameters w we wish to recover. Figure 3.4 shows this ideal parameter vector. We construct data samples

$X = Yw + \varepsilon$. Where Y is a sample from $\{-1, 1\}$ and ε is Gaussian noise. We take 150 training samples, 100 validation samples, and 1000 test samples. We consider a binary classification setting using only ℓ_1, ℓ_2, or k-support regularizers, Smooth-Lasso [Hebiri et al., 2011], TV+ℓ_1 regularizer, TV+ℓ_1+ℓ_2, and our k-support TV regularizer. For each of these scenarios we perform model selection using grid search and select the model with the highest accuracy on the validation set. We repeat this experiment with a new set of training, validation, and test samples 15 times so that we may obtain statistical significance results. The test set accuracy results for each method are shown in Table 3.4. For each competing method we perform a Wilcoxon signed-rank test against the k-support total variation results. In all listed cases the test rejects the null hypothesis (at a significance level of $p < 0.05$) that the samples come from the same distribution. We assess the support recovery of competing method by measuring the area under the precision-recall curve for different support thresholds. Finally we measure stability using Pearson correlation between weight vectors from different trials.

Description	**Test Acc. (p-value)**	**Supp. Recovery**	**Stability**
ℓ_2	67.8%(7E-4)	0.388	0.173
ℓ_1	68.4%(7E-4)	0.377	0.220
k-support	68.1%(7E-4)	0.398	0.217
Smooth-LASSO	77.0%(7E-4)	0.407	0.464
TV+ℓ_1	80.2%(9E-3)	0.739	0.620
TV+ℓ_1+ℓ_2	81.5%(2E-2)	0.796	0.688
k-support/TV	**82.2%**	**0.816**	**0.719**

Table 3.3 – Average test accuracy, support recovery, and test accuracy results for 15 trials of synthetic data along with p-value for a Wilcoxon signed-rank test performed for each method against the k-support/TV result, below 0.05 for all cases. k-support/TV has both the highest accuracy, highest support recovery as well as the highest stability. Here stability is measured by average pairwise Pearson correlation between folds.

In Figure 3.4 we visualize the weight vector and precision-recall curve produced by the various regularization methods for one trial. We can see that in Figure 3.4 the k-support norm alone does a poor job at reconstructing a model with any of these local correlations in place. The Smooth-Lasso, TV+ℓ_1 and TV+ℓ_1+ℓ_2 regularizers do a substantially better job at indicating the areas of interest for this task but the k-support/TV regularizer produces more precise regions with fewer spurious patterns and substantially better classification accuracy and support recovery. We can see an additional advantage of the k-support/TV regularizer over the other methods in terms of stability of the results across trials. Figure 3.4(c) also shows the effectiveness of the k-support/TV regularizer for varying target weight vectors.

In the analysis of fMRI data we are often concerned with using the estimator to identify the predictive regions. Specifically the linear model is often mapped back to a brain volume and used for analysis. In this context regularization can not only improve predictive performance, but it can provide more interpretable brain maps. We prefer solutions which clearly indicate the areas of interest. Well converged $TV + \ell_1$ solutions can overemphasize the sparsity. With

the k variable we can encourage a less sparse solution, that may be more interpretable and include more highly correlated variables. Figure 3.5 shows this effect for maps of varying k values (note that $k = 1$ corresponds to $TV + \ell_1$).

We note that unlike the elastic-net penalty the k in k-support has an interpretable parameter setting for mixing sparsity and ℓ_2. We can interpret the k in our regularizer as an estimate of the number of voxel locations active in the brain. Thus we can set k based on prior knowledge. We fix the value of k to 500 representing approximately 2% sparsity, this allows us to directly compare to the state of the art method for sparse regularization in fMRI, $TV + \ell_1$, with an equal sized search space in model-selection. Below we show the accuracy and stability results for $TV + \ell_1$, $TV + \ell_1 + \ell_2$, and our TV+k-support.

Description	**Test Acc. (p-value)**	**Stability**
TV+ℓ_1	84.72 (8E-4)	0.132
TV+ℓ_1+ℓ_2	86.06 (0.15)	0.186
k-support/TV	87.91	0.415

Table 3.4 – Average test accuracy results for 20 trials along with p-value for a Wilcoxon signed-rank test performed for each method against the k-support/TV result. Solution stability is measured by averaging pairwise Spearman correlations between solutions from different folds of training data. We note that our accuracy is statistically significantly better than TV+ℓ_1 and we do much better in terms of solution stability.

Since the size of the data is small we often have equivalent average accuracies in model selection, we break ties based on intra-fold stability as measured by average pairwise Spearman correlations of the resulting weight vectors. Our result beats $TV + \ell_1$ in terms of accuracy . Compared to $TV + \ell_1 + \ell_2$ we have better classification accuracy, but not with a high statistical significance, however we obtain much more stable solutions and have more interpretable parameter settings. We describe another advantage of our approach compared to the competing methods below.

An additional issue in interpreting brain maps is where to threshold. Many sparse regularizers, even those such as ℓ_1 only have asymptotic guarantees for sparse solutions; in practice we threshold values at a specific value. This is particularly problematic when we add TV into the objective. Here we suggest a heuristic motivated by the properties of the k-support norm. As we can see in Equation (3.2.33) the k-support norm can be shown to a combination of ℓ_2 penalties on the highest magnitude $k - r - 1$ terms and ℓ_1 penalty applied to the rest. Here r is the unique integer in $\{0, \ldots, k-1\}$ satisfying

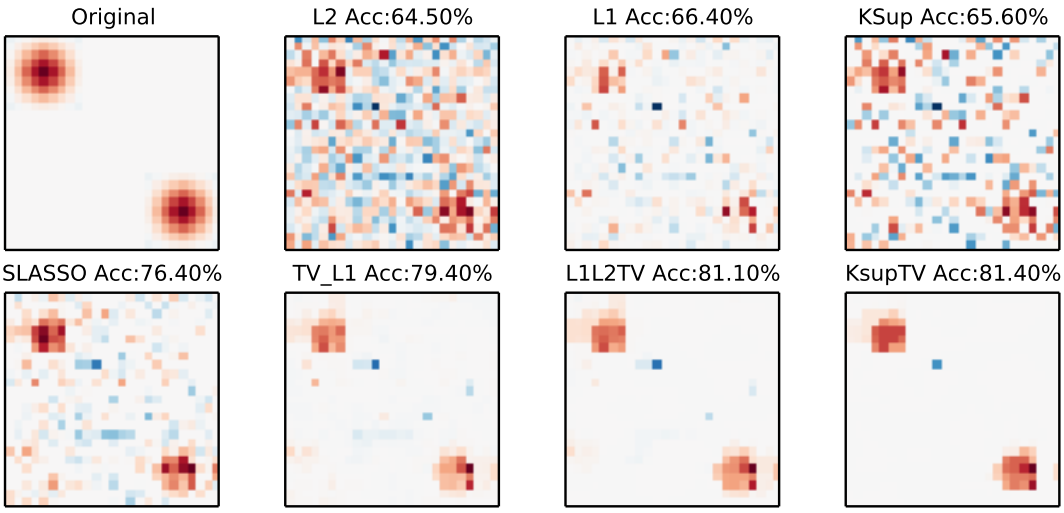
$$|w|^{\downarrow}_{k-r-1} > \frac{1}{r+1} \sum_{i=k-r}^{d} |w|^{\downarrow}_i \geq |w|^{\downarrow}_{k-r}. \tag{3.3.2}$$

Empirically we can show that the value of $k - r - 1$ for the solution grows from 0 as the optimization progress as seen in Figure 3.5. This can be loosely interpreted as the algorithm

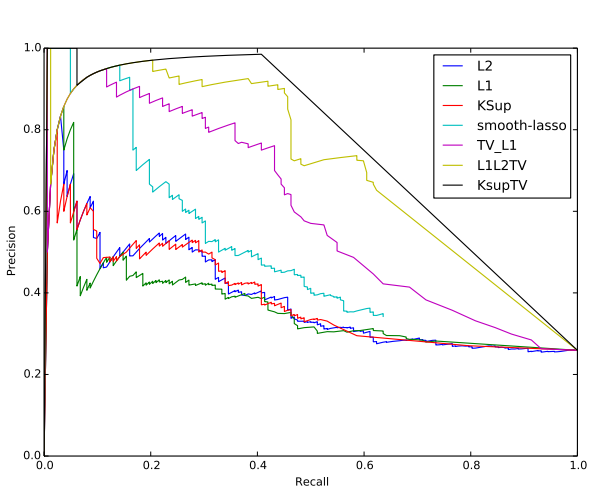
starting with ℓ_1 optimization,which attempts to push variables to zero, but as we progress we have flexibility to move onto parts of the k-support ball where specific key variables fall into the ℓ_2 term, while we still attempt to squash the remaining terms with ℓ_1. This property of the optimization of our penalty implies a visualization heuristic for the final solution of taking the top $k-r-1$ variables. Another view on this heuristic comes from the implicit delineation implied by Equation (3.3.4). For k much smaller than d and $k-r-1$ greater than 0 the definition of r implies the $k-r-1^{th}$ largest magnitude parameter will be a large factor $(\frac{d-k+r}{1+r})$ bigger than the mean of the rest of the parameters below it. Figure 3.3.4 illustrates thresholding based on a fixed threshold value and our heuristic of thresholding based on the final $k-r-1$ value in k-support TV optimization.

3.4 Conclusions

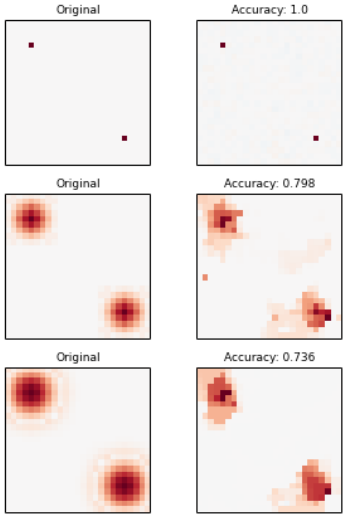
We have introduced a novel norm that incorporates spatial smoothness and correlated sparsity. This norm, called the (k,s) *support total variation norm*, extends both the total variation penalty which is a standard in image processing and the recently proposed k-support norm from machine learning. The (k,s) support TV norm is the *tightest convex penalty* that combines *sparsity*, ℓ_2 and *total variation* constraints jointly. We have derived a variational form for this norm for arbitrary graph structures. We have also expressed the dual norm as a combinatorial optimization problem on the graph. This graph problem is shown to be NP-hard motivating the use of a relaxation, which is shown to be equivalent to the weighted combination of a k-support norm and a total variation penalty. We have shown that this norm approximates the (k,s) support TV norm within a factor that depends on properties of the graph as well as on the parameters k and s, and that this bound scales well for grid structured graphs. Moreover, we have demonstrated that joint k support and TV regularization can be applied on a diverse variety of learning problems, such as classification with small samples, neural imaging and image recovery. These experiments have illustrated the utility of penalties combining k-support and total variation structure on problems where spatial structure, feature selection and correlations among features are all relevant. We have shown that this penalty has several unique properties that make it an excellent tool analysis of fMRI data. Some of our additional contributions include a generalized formulation of the dual norm of a norm which is the infimal convolution of norms, the first algorithm for projecting onto the k-support norm ball, and first analysis that notes interesting practical properties of the r variable of the k-support norm.



(a)



(b)



(c)

Figure 3.4 – (a) (*left top to bottom right*) ideal weight vector, weight vector obtained with ℓ_1, ℓ_2, k-support norm, TV+ℓ_1, and k-support/TV regularizer, and weight vector with combined total variation and k-support norm regularizer. The k-support/TV regularization gives the highest accuracy, support recovery, stability, and most closely approximates the target pattern. (b) Illustrates the improved precision-recall for k-support/TV versus the other methods on the support recovery for different thresholds. (c) Recovered support for varying ideal weight vector. This demonstrates that the k-support/TV regularization works well for a wide range of sparsity, correlation, and smoothness.

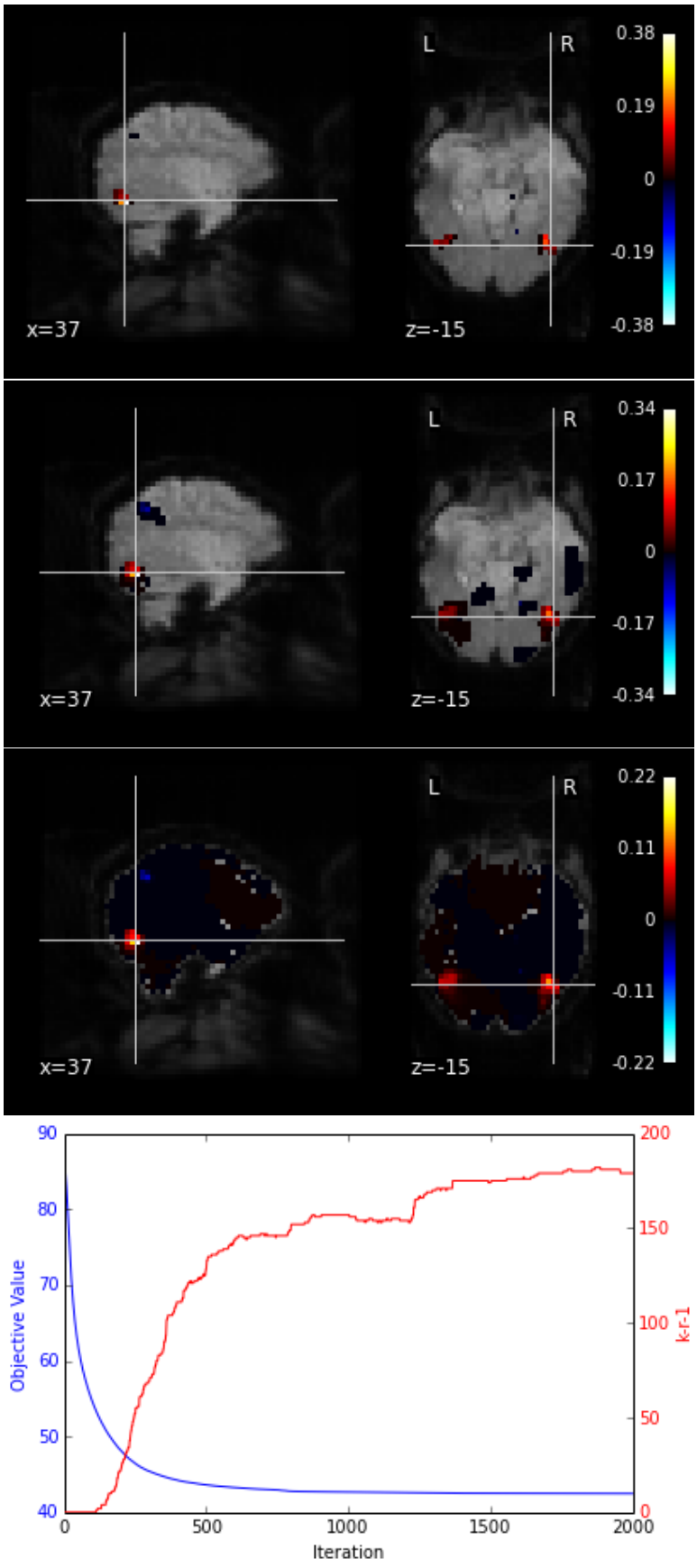


Figure 3.5 – Output map for k=1 (TV-ℓ_1), k=50, and k=500, in each case the Lateral Occipital Cortex is indicated, Objective value of TV+k-support (k=500) and $k - r - 1$ over iterations.

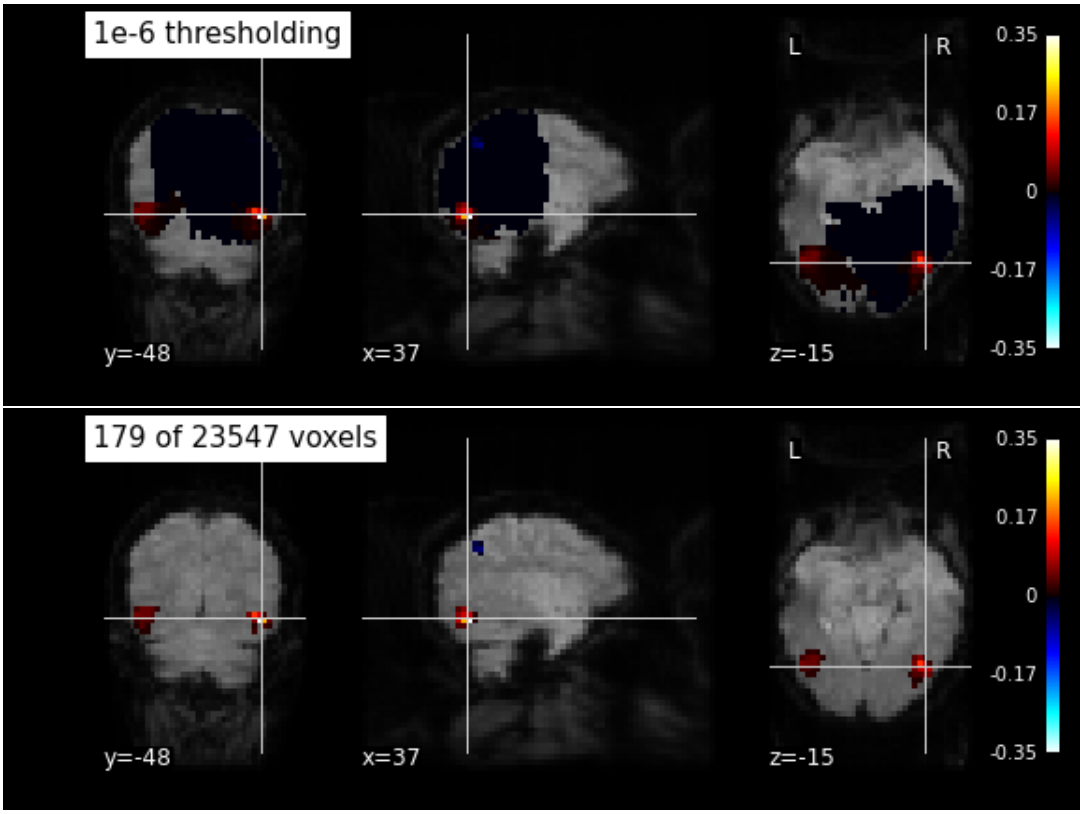


Figure 3.6 – Output map for fixed thresholding and thresholding based on converged $k-r-1$ value

Chapter 4

Testing for Differences in Gaussian Graphical Models: Applications to Brain Connectivity

In this Chapter we introduce the use of Gaussian Graphical Models for determining functional connectivity. We identify the difference connectivity problem and the role of determining significance results on edges. We propose a hypothesis test which can be used in this setting.

Contents

4.1 Introduction

Gaussian graphical models describe well interactions in many real-world systems. For instance, correlations in brain activity reveal brain interactions between distant regions, a process know as *functional connectivity*. Functional connectivity is an interesting probe on brain mechanisms as it persists in the absence of tasks (the so-called "resting-state") and is thus applicable to study populations of impaired subjects, as in neurologic or psychiatric diseases [Castellanos et al, 2013]. From a formal standpoint, Gaussian graphical models are well suited to estimate brain connections from functional Magnetic Resonance Imaging (fMRI) signals [Smith et al., 2011, Varoquaux et al., 2010]. A set of brain regions and related functional

connections is then called a functional *connectome* [Castellanos et al, 2013, Varoquaux and Craddock, 2013]. Its variation across subjects can capture cognition [Richiardi et al., 2011, Shirer et al., 2012] or pathology [Castellanos et al, 2013, Kelly et al., 2012]. However, the effects of pathologies are often very small, as resting-state fMRI is a weakly-constrained and noisy imaging modality, and the number of subjects in a study is often small given the cost of imaging. Statistical power is then a major concern [Button et al., 2013]. The statistical challenge is to increase the power to detect differences between Gaussian graphical models in the small-sample regime.

In these settings, estimation and comparison of Gaussian graphical models fall in the range of high-dimensional statistics: the number of degrees of freedom in the data is small compared to the dimensionality of the model. In this regime, sparsity-promoting ℓ_1-based penalties can make estimation well-posed and recover good estimation performance despite the scarcity of data [Bühlmann and van de Geer, 2011, Danaher et al., 2014, Friedman et al., 2008, Meinshausen and Bühlmann, 2006, Tibshirani, 1996b]. These encompass sparse regression methods such as the lasso or recovery methods such as basis pursuit, and can be applied to estimation of Gaussian graphical models with approaches such as the graphical lasso [Friedman et al., 2008]. There is now a wide body of literature which demonstrates the statistical properties of these methods [Bühlmann and van de Geer, 2011]. Crucial to applications in medicine or neuroscience, recent work characterizes the uncertainty, with confidence intervals and p-values, of the parameters selected by these methods [G'Sell et al., 2013, Janková and van de Geer, 2015, Javanmard and Montanari, 2014, Lockhart et al., 2014]. These works focus primarily on the lasso and graphical lasso.

Approaches to estimate statistical significance on sparse models fall into several general categories: (a) non-parameteric sampling based methods which are inherently expensive and have difficult limiting distributions [Bühlmann and van de Geer, 2011, Da Mota et al., 2014, Narayan and Allen, 2015], (b) characterizations of the distribution of new parameters that enter a model along a regularization path [G'Sell et al., 2013, Lockhart et al., 2014], or (c) for a particular regularization parameter, debiasing the solution to obtain a new consistent estimator with known distribution [Janková and van de Geer, 2015, Javanmard and Montanari, 2014, Van de Geer et al., 2014]. While some of the latter work has been used to characterize confidence intervals on network edge selection, there is no result, to our knowledge, on the important problem of identifying differences in networks. Here the confidence on the result is even more critical, as the differences are the direct outcome used for neuroscience research or medical practice, and it is important to provide the practitioner a measure of the uncertainty.

Here, we consider the setting of two datasets known to have very similar underlying signals, but which individually may not be very sparse. A motivating example is determining the difference in brain networks of subjects from different groups: population analysis of connectomes [Kelly et al., 2012, Varoquaux and Craddock, 2013]. Recent literature in neuroscience [Markov et al., 2013] has suggested functional networks are not sparse. On the other hand, differences in connections across subjects should be sparse. Indeed the link between functional and anatomical brain networks [Honey et al., 2009] suggests they should not differ drastically from one subject to another. From a neuroscientific standpoint we are interested in deter-

mining which edges between two populations (e.g. autistic and non-autistic) are different. Furthermore we want to provide confidence-intervals on our results. We particularly focus on the setting where one dataset is larger than the other. In many applications it is more difficult to collect one group (e.g. individuals with specific pathologies) than another.

We introduce an estimator tailored to this goal: the debiased multi-task fused lasso. We show that, when the underlying parameter differences are indeed sparse, we can obtain a tractable Gaussian distribution for the parameter difference. This closed-form distribution underpins accurate hypothesis testing and confidence intervals. We then use the relationship between nodewise regression and the inverse covariance matrix to apply our estimator to learning differences of Gaussian graphical models.

This Chapter is organized as follows. In Section 4.2 we review previous work on learning of GGMs and the debiased lasso. Section 4.3 discusses a joint debiasing procedure that specifically debiases the difference estimator. In Section 4.3.1 we introduce the debiased multi-task fused lasso and show how it can be used to learn parameter differences in linear models. In Section 4.3.2, we show how these results can be used for GGMs. In Section 4.4 we validate our approach on synthetic and fMRI data.

4.2 Debiased Lasso and Structure Learning

Debiased Lasso A central starting point for our work is the debiased lasso [Javanmard and Montanari, 2014, Van de Geer et al., 2014]. Here one considers the linear regression model, $Y = \boldsymbol{X}\beta + \epsilon$, with data matrix $\boldsymbol{X}$ and output Y, corrupted by $\epsilon \sim N(0, \sigma_\epsilon^2 I)$ noise. The lasso estimator is formulated as follows:

$$\hat{\beta}^\lambda = \arg\min_\beta \frac{1}{n}\|Y - \boldsymbol{X}\beta\|^2 + \lambda\|\beta\|_1. \tag{4.2.1}$$

The KKT conditions give $\hat{k}^\lambda = \frac{1}{n}\boldsymbol{X}^T(Y - \boldsymbol{X}\beta)$, where $\hat{k}$ is the subgradient of $\lambda\|\beta\|_1$. The debiased lasso estimator [Javanmard and Montanari, 2014, Van de Geer et al., 2014] is then formulated as $\hat{\beta}_u^\lambda = \hat{\beta}^\lambda + \boldsymbol{M}\hat{k}^\lambda$ for some $\boldsymbol{M}$ that is constructed to give guarantees on the asymptotic distribution of $\hat{\beta}_u^\lambda$. Note that this estimator is not strictly unbiased in the finite sample case, but has a bias that rapidly approaches zero (w.r.t. n) if $\boldsymbol{M}$ is chosen appropriately, the true regressor β is indeed sparse, and the design matrix satistifes a certain restricted eigenvalue property [Javanmard and Montanari, 2014, Van de Geer et al., 2014]. We decompose the difference of this debiased estimator and the truth as follows:

$$\hat{\beta}_u^\lambda - \beta = \frac{1}{n}\boldsymbol{M}\boldsymbol{X}^T\epsilon - (\boldsymbol{M}\hat{\Sigma} - I)(\hat{\beta} - \beta). \tag{4.2.2}$$

The first term is Gaussian and the second term is responsible for the bias. Using Holder's inequality the second term can be bounded by $\|\boldsymbol{M}\hat{\Sigma} - I\|_\infty\|\hat{\beta} - \beta\|_1$. The first part of which we can bound using an appropriate selection of $\boldsymbol{M}$ while the second part is bounded by our implicit sparsity assumptions coming from lasso theory [Bühlmann and van de Geer, 2011].

Two approaches from the recent literature discuss how one can select $\boldsymbol{M}$ to appropriately debias this estimate. In Van de Geer et al. [2014] it suffices to use nodewise regression to learn an inverse covariance matrix which guarantees constraints on $\|\boldsymbol{M}\hat{\Sigma} - I\|_\infty$. A second approach by Javanmard and Montanari [2014] proposes to solve a quadratic program to directly minimize the variance of the debiased estimator while constraining $\|\boldsymbol{M}\hat{\Sigma} - I\|_\infty$ to induce sufficiently small bias.

Intuitively the construction of $\hat{\beta}_u^\lambda$ allows us to trade variance and bias via the $\boldsymbol{M}$ matrix. This allows us to overcome a naive bias-variance tradeoff by leveraging the sparsity assumptions that bound $\|\hat{\beta} - \beta\|_1$. In the sequel we expand this idea to the case of debiased parameter difference estimates and sparsity assumptions on the parameter differences.

In the context of GGMs, the debiased lasso can gives us an estimator that asymptotically converges to the partial correlations. As highlighted by Waldorp [2014] we can thus use the debiased lasso to obtain difference estimators with known distributions. This allows us to obtain confidence intervals on edge differences between Gaussian graphical models. We discuss this further in the sequel.

Gaussian Graphical Model Structure Learning A standard approach to estimating Gaussian graphical models in high dimensions is to assume sparsity of the precision matrix and have a constraint which limits the number of non-zero entries of the precision matrix. This constraint can be achieved with a ℓ_1-norm regularizer as in the popular graphical lasso [Friedman et al., 2008]. Many variants of this approach that incorporate further structural assumptions have been proposed [Danaher et al., 2014, Honorio and Samaras, 2010, Mohan et al., 2012].

An alternative solution to inducing sparsity on the precision matrix indirectly is neighborhood ℓ_1 regression from Meinshausen and Bühlmann [2006]. Here the authors make use of a long known property that connects the entries of the precision matrix to the problem of regression of one variable on all the others [Marsaglia, 1964]. This property is critical to our proposed estimation as it allows relating regression models to finding edges connected to specific nodes in the GGM.

GGMs have been found to be good at recovering the main brain networks from fMRI data [Smith et al., 2011, Varoquaux et al., 2010]. Yet, recent work in neuroscience has showed that the structural wiring of the brain did not correspond to a very sparse network [Markov et al., 2013], thus questioning the underlying assumption of sparsity often used to estimate brain network connectivity. On the other hand, for the problem of finding differences between networks in two populations, sparsity may be a valid assumption. It is well known that anatomical brain connections tend to closely follow functional ones [Honey et al., 2009]. Since anatomical networks do not differ drastically we can surmise that two brain networks should not differ much even in the presence of pathologies. The statistical method we present here leverages sparsity in the difference of two networks, to yield well-behaved estimation and hypothesis testing in the low-sample regime. Most closely related to our work, Zhao et al. [2014] recently considers a different approach to estimating difference networks, but does not

consider assigning significance to the detection of edges.

4.3 Debiased Difference Estimation

In many applications one may be interested in learning multiple linear models from data that share many parameters. Situations such as this arise often in neuroimaging and bioinformatics applications. We can often improve the learning procedure of such models by incorporating fused penalties that penalize the $\|\cdot\|_1$ norm of the parameter differences or $\|\cdot\|_{1,2}$ which encourages groups of parameters to shrink together. These methods have been shown to substantially improve the learning of the joint models. However, the differences between model parameters, which can have a high sample complexity when there are few of them, are often pointed out only in passing [Chen et al., 2011, Danaher et al., 2014, Honorio and Samaras, 2010]. On the other hand, in many situations we might be interested in actually understanding and identifying the differences between elements of the support. For example when considering brain networks of patients suffering from a pathology and healthy control subjects, the difference in brain connectivity may be of great interest. Here we focus specifically on accurately identifying differences with significance.

We consider the case of two tasks (e.g. two groups of subjects), but the analysis can be easily extended to general multi-task settings. Consider the problem setting of data matrices $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, which are $n_1 \times p$ and $n_2 \times p$, respectively. We model them as producing outputs Y_1 and Y_2, corrupted by diagonal gaussian noise ϵ_1 and ϵ_2 as follows

$$Y_1 = \boldsymbol{X_1}\beta_1 + \epsilon_1, \quad Y_2 = \boldsymbol{X_2}\beta_2 + \epsilon_2 \tag{4.3.1}$$

Let S_1 and S_2 index the elements of the support of β_1 and β_2, respectively. Furthermore the support of $\beta_1 - \beta_2$ is indexed by S_d and finally the union of S_1 and S_2 is denoted S_a. Using a squared loss estimator producing independent estimates $\hat{\beta}_1, \hat{\beta}_2$ we can obtain a difference estimate $\hat{\beta}_d = \hat{\beta}_1 - \hat{\beta}_2$. In general if S_d is very small relative to S_a then we will have a difficult time to identify the support S_d. This can be seen if we consider each of the individual components of the prediction errors. The larger the true support S_a the more it will drown out the subset which corresponds to the difference support. This can be true even if one uses ℓ_1 regularizers over the parameter vectors. Consequently, one cannot rely on the straightforward strategy of learning two independent estimates and taking their difference. The problem is particularly pronounced in the common setting where one group has fewer samples than the other. Thus here we consider the setting where $n_1 > n_2$ and possibly $n_1 \gg n_2$.

Let $\hat{\beta}_1$ and $\hat{\beta}_2$ be regularized least squares estimates. In our problem setting we wish to obtain confidence intervals on debiased versions of the difference $\hat{\beta}_d = \hat{\beta}_1 - \hat{\beta}_2$ in a high-dimensional setting (in the sense that $n_2 < p$), we aim to leverage assumptions about the form of the true β_d, primarily that it is sparse, while the independent $\hat{\beta}_1$ and $\hat{\beta}_2$ are weakly sparse or not

sparse. We consider a general case of a joint regularized least squares estimation of $\hat{\beta}_1$ and $\hat{\beta}_2$

$$\min_{\beta_1,\beta_2} \frac{1}{n_1}\|Y_1 - \boldsymbol{X_1}\beta_1\|^2 + \frac{1}{n_2}\|Y_2 - \boldsymbol{X_2}\beta_2\|^2 + R(\beta_1, \beta_2) \tag{4.3.2}$$

We note that the differentiating and using the KKT conditions gives

$$\hat{k}^\lambda = \hat{k}_1 + \hat{k}_2 = \frac{1}{n_1}\boldsymbol{X_1}^T(Y - \boldsymbol{X_1}\beta_1) + \frac{1}{n_2}\boldsymbol{X_2}^T(Y - \boldsymbol{X_2}\beta_2) \tag{4.3.3}$$

where $\hat{k}^\lambda$ is the (sub)gradient of $R(\beta_1, \beta_2)$. Substituting Equation (4.3.1) we can now write

$$\hat{\Sigma}_1(\hat{\beta}_1 - \beta_1) + \hat{k}_1 = \frac{1}{n_1}\boldsymbol{X_1}^T\epsilon_1 \quad \text{and} \quad \hat{\Sigma}_2(\hat{\beta}_2 - \beta_2) + \hat{k}_2 = \frac{1}{n_2}\boldsymbol{X_2}^T\epsilon_2 \tag{4.3.4}$$

We would like to solve for the difference $\hat{\beta}_1 - \hat{\beta}_2$ but the covariance matrices may not be invertible. We introduce matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, which will allow us to isolate the relevant term. We will see that in addition these matrices will allow us to decouple the bias and variance of the estimators.

$$\boldsymbol{M}_1\hat{\Sigma}_1(\hat{\beta}_1 - \beta_1) + \boldsymbol{M}_1\hat{k}_1 = \frac{1}{n_1}\boldsymbol{M}_1\boldsymbol{X_1}^T\epsilon_1 \quad \text{and} \quad \boldsymbol{M}_2\hat{\Sigma}_2(\hat{\beta}_2 - \beta_2) + \boldsymbol{M}_2\hat{k}_2 = \frac{1}{n_2}\boldsymbol{M}_2\boldsymbol{X_2}^T\epsilon_2 \tag{4.3.5}$$

subtracting these and rearranging we can now isolate the difference estimator plus a term we add back controlled by $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$

$$(\hat{\beta}_1 - \hat{\beta}_2) - (\beta_1 - \beta_2) + \boldsymbol{M}_1\hat{k}_1 - \boldsymbol{M}_2\hat{k}_2 = \frac{1}{n_1}\boldsymbol{M}_1\boldsymbol{X_1}^T\epsilon_1 - \frac{1}{n_2}\boldsymbol{M}_2\boldsymbol{X_2}^T\epsilon_2 - \Delta \tag{4.3.6}$$

$$\Delta = (\boldsymbol{M}_1\hat{\Sigma}_1 - I)(\hat{\beta}_1 - \beta_1) - (\boldsymbol{M}_2\hat{\Sigma}_2 - I)(\hat{\beta}_2 - \beta_2) \tag{4.3.7}$$

Denoting $\beta_d := \beta_1 - \beta_2$ and $\beta_a := \beta_1 + \beta_2$, we can reformulate Δ:

$$\Delta = \frac{(\boldsymbol{M}_1\hat{\Sigma}_1 - I + \boldsymbol{M}_2\hat{\Sigma}_2 - I)}{2}(\hat{\beta}_d - \beta_d) + \frac{(\boldsymbol{M}_1\hat{\Sigma}_1 - \boldsymbol{M}_2\hat{\Sigma}_2)}{2}(\hat{\beta}_a - \beta_a) \tag{4.3.8}$$

Here, Δ will control the bias of our estimator. Additionally, we want to minimize its variance,

$$\frac{1}{n_1}\boldsymbol{M}_1\hat{\Sigma}_1\boldsymbol{M}_1\hat{\sigma}_1^2 + \frac{1}{n_2}\boldsymbol{M}_2\hat{\Sigma}_2\boldsymbol{M}_2\hat{\sigma}_2^2. \tag{4.3.9}$$

We can now overcome the limitations of simple bias variance trade-off by using an appropriate regularizer coupled with an assumption on the underlying signal β_1 and β_2. This will in turn make Δ asymptotically vanish while maximizing the variance.

Since we are interested in pointwise estimates, we can focus on bounding the infinity norm of Δ.

$$\|\Delta\|_\infty \leq \frac{1}{2}\underbrace{\|\boldsymbol{M}_1\hat{\Sigma}_1 + \boldsymbol{M}_2\hat{\Sigma}_2 - 2I\|_\infty}_{\mu_1}\underbrace{\|\hat{\beta}_d - \beta_d\|_1}_{l_d} + \frac{1}{2}\underbrace{\|\boldsymbol{M}_1\hat{\Sigma}_1 - \boldsymbol{M}_2\hat{\Sigma}_2\|_\infty}_{\mu_2}\underbrace{\|\hat{\beta}_a - \beta_a\|_1}_{l_a} \tag{4.3.10}$$

We can control the maximum bias by selecting $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ appropriately. If we use an appropriate regularizer coupled with sparsity assumptions we can bound the terms l_a and l_d and use this knowledge to appropriately select $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ such that the bias becomes neglibile. If we had only the independent parameter sparsity assumption we can apply the results of the debiased lasso and estimate $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ independently as in Javanmard and Montanari [2014]. In the case of interest where β_1 and β_2 share many weights we can do better by taking this as an assumption and applying a sparsity regularization on the difference by adding the term $\lambda_2\|\beta_1-\beta_2\|_1$. Comparing the decoupled penalty to the fused penalty proposed we see that l_d would decrease at a given sample size. We now show how to jointly estimate $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ so that $\|\Delta\|_\infty$ becomes negligible for a given n, p and sparsity assumption.

4.3.1 Debiasing the Multi-Task Fused Lasso

Motivated by the inductive hypothesis from neuroscience described above we introduce a consistent low-variance estimator, the debiased multi-task fused lasso. We propose to use the following regularizer $R(\beta_1,\beta_2) = \lambda_1\|\beta_1\|_1 + \lambda_1\|\beta_2\|_1 + \lambda_2\|\beta_1-\beta_2\|_1$. This penalty has been referred to in some literature as the multi-task fused lasso [Chen et al., 2011]. We propose to then debias this estimate as shown in (4.3.6). We estimate the $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ matrices by solving the following QP for each row m_1 and m_2 of the matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$.

$$\begin{aligned} &\min_{m_1,m_2} \frac{1}{n_1} m_1^T\hat{\Sigma}_1 m_1 + \frac{1}{n_2} m_2^T\hat{\Sigma}_2 m_2 \\ s.t. \quad &\|\boldsymbol{M}_1\hat{\Sigma}_1 + \boldsymbol{M}_2\hat{\Sigma}_2 - 2I\|_\infty \le \mu_1, \quad \|\boldsymbol{M}_1\hat{\Sigma}_1 - \boldsymbol{M}_2\hat{\Sigma}_2\|_\infty \le \mu_2 \end{aligned} \tag{4.3.11}$$

This directly minimizes the variance, while bounding the bias in the constraint. We now show how to set the bounds. The following analysis is used to show the conditions under which the debias multi-task fused lasso achieves a negligible bias.

Let $\beta = [\beta_1;\beta_2]$,$\beta_d = \beta_1 - \beta_2$,$\beta_a = \beta_1 + \beta_2$. Let $S_{1,2}$ bet the support of β. Define $\boldsymbol{X}_N = [\boldsymbol{X}_1/\sqrt{n_1}, 0; 0, \boldsymbol{X}_2/\sqrt{n_2}]$

Lemma 4.1. *(Basic Inequality)* $\|\boldsymbol{X}_N(\hat{\beta}-\beta)\|_2^2+\lambda_1\|\hat{\beta}\|_1+\lambda_2\|\hat{\beta}_d\| \le 2\epsilon^T\boldsymbol{X}_N(\hat{\beta}-\beta)+\lambda_1\|\beta\|_1+\lambda_2\|\beta_d\|_1$

This follows from the fact that $\hat{\beta}$ is the minimizer of the fused lasso objective.

The term, $\epsilon^T\boldsymbol{X}_N(\hat{\beta}-\beta)$, commonly known as the empirical process term [Bühlmann and van de Geer, 2011] can be bound as follows:

$$\begin{aligned} &2|\epsilon^T\boldsymbol{X}_N(\hat{\beta}-\beta)| = 2|\epsilon_1^T\boldsymbol{X}_1(\hat{\beta}_1-\beta_1)/n_1 + \epsilon_2^T\boldsymbol{X}_2(\hat{\beta}_2-\beta_2)/n_2| \le \\ &2\|\hat{\beta}_1-\beta_1\|_1 \max_{1\le j\le p}|\epsilon_1^T\boldsymbol{X}_1^{(j)}/n_1| + 2\|\hat{\beta}_2-\beta_2\|_1 \max_{1\le j\le p}|\epsilon_2^T\boldsymbol{X}_2^{(j)}/n_2| \end{aligned}$$

Where we utilize holder's inequality in the last line. We define the random event $\mathcal{F}$ for which the following holds: $\max_{1\le j\le p}|\epsilon_1^T\boldsymbol{X}_1^{(j)}/n_1| \le \lambda_0$ and $\max_{1\le j\le p}|\epsilon_1^T\boldsymbol{X}_1^{(j)}/n_1| \le \lambda_0$. furthermore we can select $2\lambda_0 \le \lambda_1$

Lemma 4.2. *Suppose* $\hat{\Sigma}_{j,j} = 1$ *for both* $\boldsymbol{X}_1$ *and* $\boldsymbol{X}_2$ *then we have for all* $t > 0$ *and* $n_1 > n_2$

$$\lambda_0 = 2\sigma_2\sqrt{\frac{t^2 + \log p}{n_2}} \tag{4.3.12}$$

$$P(\mathcal{F}) = 1 - 2\exp(-t^2/2) \tag{4.3.13}$$

Proof: This follows directly from the [Bühlmann and van de Geer, 2011, Lemma 6.2] and taking $n_1 > n_2$. □

This allows us to get rid of the empirical process term on $\mathcal{F}$, with an appropriate choice of λ_1.

Given a set, S, denote β_S the vector of equal size to β but all elements not in S set to zero. We can now show the following

Lemma 4.3. *We have on* $\mathcal{F}$ *with* $\lambda_1 \geq 2\lambda_0$

$$\begin{aligned} &2\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 + \lambda_1\|\hat{\beta}_{S_{1,2}^c}\|_1 + 2\lambda_2\|\hat{\beta}_{d,S_d^c}\|_1 \\ &\leq 3\lambda_1\|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1 + 2\lambda_2\|\hat{\beta}_{d,S_d} - \beta_{d,S_d}\|_1 \end{aligned} \tag{4.3.14}$$

Proof: Following [Bühlmann and van de Geer, 2011, Lemma 6.3] we start with the basic inequality on $\mathcal{F}$. Which gives

$$\begin{aligned} &2\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 + 2\lambda_1\|\hat{\beta}\|_1 + 2\lambda_2\|\hat{\beta}_d\| \\ &\leq \lambda_1\|\hat{\beta} - \beta\|_1 + 2\lambda_1\|\beta\|_1 + 2\lambda_2\|\beta_d\|_1 \end{aligned} \tag{4.3.15}$$

Since we assume the truth is in fact sparse,

$$\|\hat{\beta}_d - \beta_d\|_1 = \|\hat{\beta}_{d,S_d} - \beta_{d,S_d}\|_1 + \|\hat{\beta}_{d,S_d^C}\|_1 \tag{4.3.16}$$

$$\|\hat{\beta} - \beta\|_1 = \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1 + \|\hat{\beta}_{S_{1,2}^C}\|_1 \tag{4.3.17}$$

Furthermore,

$$\|\hat{\beta}\|_1 \geq \|\beta_{S_{1,2}}\|_1 - \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1 + \|\hat{\beta}_{S_{1,2}^C}\|_1 \tag{4.3.18}$$

$$\|\hat{\beta}_d\|_1 \geq \|\beta_{d,S_d}\|_1 - \|\hat{\beta}_{d,S_d} - \beta_{d,S_d}\|_1 + \|\hat{\beta}_{d,S_d^C}\|_1 \tag{4.3.19}$$

Substituting (4.3.18), (4.3.19), and (4.3.17) into (4.3.15) and rearranging completes the proof. □

From the lemma above we can now formulate the following the bounds in (4.3.20)

Proposition 3. *Take* $\lambda_1 > 2\sqrt{\frac{\log p}{n_2}}$ *and* $\lambda_2 = O(\lambda_1)$. *Denote* s_d *the difference sparsity,* $s_{1,2}$ *the parameter sparsity* $|S_1| + |S_2|$, $c > 1$,$a > 1$, *and* $0 < m \ll 1$. *When the compatibility condition [Bühlmann and van de Geer, 2011, Ganguly and Polonik, 2014] holds the following bounds gives* $l_a u_2 = o(1)$ *and* $l_d u_1 = o(1)$ *and thus* $\|\Delta\|_\infty = o(1)$ *with high probability.*

$$\mu_1 \leq \frac{1}{c\lambda_2 s_d n_2^m} \quad \textit{and} \quad \mu_2 \leq \frac{1}{a(\lambda_1 s_{1,2} + \lambda_2 s_d) n_2^m} \tag{4.3.20}$$

Proof: We first consider the bound associated with l_a

$$\begin{aligned} \lambda_1 \|\hat{\beta}_a - \beta_a\|_1 \leq \lambda_1 \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1 + \lambda_1 \|\hat{\beta}_{S_{1,2}^c}\|_1 \leq \\ 4\lambda_1 \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1 + 2\lambda_2 \|\hat{\beta}_{S_d} - \beta_{S_d}\|_1 - 2\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 \qquad (4.3.21) \\ \leq 4\lambda_1 \sqrt{s_{1,2}} \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_2 + 2\lambda_2 \sqrt{s_d} \|\hat{\beta}_{S_d} - \beta_{S_d}\|_2 \\ -2\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 \qquad (4.3.22) \end{aligned}$$

Invoking the compatibility assumption [Bühlmann and van de Geer, 2011, Ganguly and Polonik, 2014, Javanmard and Montanari, 2014] with compatibility constant $\phi_{\min}$

$$\begin{aligned} \leq \frac{4\lambda_1 \sqrt{s_{1,2}}}{\phi_{min}} \|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2 + \frac{2\lambda_2 \sqrt{s_d}}{\phi_{min}} \|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2 \\ -2\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 \qquad (4.3.23) \\ \leq \frac{4\lambda_1^2 s_{1,2}}{\phi_{min}^2} + \frac{2\lambda_2^2 s_d}{\phi_{min}^2} \qquad (4.3.24) \end{aligned}$$

The bound u_2 now follows by inverting the expression shown and adding a factor of n_2^m where $m \ll 1$.

Now we consider the bound for l_d.

$$\begin{aligned} \lambda_2 \|\hat{\beta}_d - \beta_d\|_1 = \lambda_2 \|\hat{\beta}_{d,S} - \beta_{d,S}\|_1 + \lambda_2 \|\hat{\beta}_{d,S^c}\|_1 \qquad (4.3.25) \\ \leq 2\lambda_2 \|\hat{\beta}_{d,S} - \beta_{d,S}\|_1 + 3\lambda_1 \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1/2 \qquad (4.3.26) \\ -\|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 - \lambda_1 \|\hat{\beta}_{S_{1,2}^c}\|_1/2 \qquad (4.3.27) \end{aligned}$$

In the domain of interest $n_1 \gg n_2$ if we select $\lambda_2 = O(\lambda_1)$ we can see the relevant terms related to the parameter support become small with respect to terms with $S_{1,2}$. Thus the error on the difference should dominate. In this region we can have $3\lambda_1 \|\hat{\beta}_{S_{1,2}} - \beta_{S_{1,2}}\|_1/2 - \lambda_1 \|\hat{\beta}_{S_{1,2}^c}\|_1/2 \leq c\lambda_2 \|\hat{\beta}_{d,S} - \beta_{d,S}\|_1$ where $c > 0$.

$$\begin{aligned} \lambda_2 \|\hat{\beta}_d - \beta_d\|_1 \leq 2\lambda_2 \|\hat{\beta}_{d,S} - \beta_{d,S}\|_1 - \|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 \qquad (4.3.28) \\ \leq 2c\lambda_2 \sqrt{s_d} \|\hat{\beta}_{d,S} - \beta_{d,S}\|_2 - \|\boldsymbol{X}_N(\hat{\beta} - \beta)\|_2^2 \qquad (4.3.29) \end{aligned}$$

Invoking the compatibility assumption [Bühlmann and van de Geer, 2011]

$$\leq 2c\lambda_2\sqrt{s_d}\|\boldsymbol{X}_N(\hat{\beta}-\beta)\|_2/\phi_{min} - \|\boldsymbol{X}_N(\hat{\beta}-\beta)\|_2^2 \tag{4.3.30}$$

$$\leq \frac{c^2\lambda_2^2 s_d}{\phi_{min}^2} + \|\boldsymbol{X}_N(\hat{\beta}-\beta)\|_2^2 - \|\boldsymbol{X}_N(\hat{\beta}-\beta)\|_2^2 \tag{4.3.31}$$

Thus $\|\hat{\beta}_d - \beta_d\|_1 \leq \frac{c^2\lambda_2 s_d}{\phi_{min}^2}$ and use of the bound prescribed gives $l_d u_1 = o(1)$.

□

Using the prescribed $\boldsymbol{M}$s obtained with (4.3.11) and 4.3.20 we obtain an unbiased estimator given by (4.3.6) with variance (4.3.9)

4.3.2 GGM Difference Structure Discovery with Significance

The debiased lasso and the debiased multi-task fused lasso, proposed in the previous section, can be used to learn the structure of a difference of Gaussian graphical models and to provide significance results on the presence of edges within the difference graph. We refer to these two procedures as Difference of Neighborhoods Debiased Lasso Selection and Difference of Neighborhoods Debiased Fused Lasso Selection.

We recall that the conditional independence properties of a GGM are given by the zeros of the precision matrix and these zeros correspond to the zeros of regression parameters when regressing one variable on all the other. By obtaining a debiased lasso estimate for each node in the graph, Waldorp [2014] notes this leads to a sparse unbiased precision matrix estimate with a known asymptotic distribution. Subtracting these estimates for two different datasets gives us a difference estimate whose zeros correspond to no difference of graph edges in two GGMs. We can similarly use the debiased multi-task fused lasso described above and the joint debiasing procedure to obtain a test statistic for the difference of networks. We now formalize this procedure.

Notation Given GGMs $j = 1, 2$. Let $\boldsymbol{X}_j$ denote the random variable in $\mathbb{R}^p$ associated with GGM j. We denote $X_{j,v}$ the random variable associated with a node, v of the GGM and $\boldsymbol{X}_{j,v^c}$ all other nodes in the graph. We denote $\hat{\beta}_{j,v}$ the lasso or multi-task fused lasso estimate of $\boldsymbol{X}_{j,v^c}$ onto $X_{j,v}$, then $\hat{\beta}_{j,dL,v}$ is the debiased version of $\hat{\beta}_{j,v}$. Finally let $\beta_{j,v}$ denote the unknown regression, $X_{j,v} = \boldsymbol{X}_{j,v^c}\beta_{j,v} + \epsilon_j$ where $\epsilon_j \sim \mathbf{N}(0, \sigma_j\mathbf{I})$. Define $\beta^i_{D,v} = \hat{\beta}^i_{1,dL,v} - \hat{\beta}^i_{2,dL,v}$ the test statistic associated with the edge v, i in the difference of GGMs $j = 1, 2$.

Proposition 4. *Given the $\hat{\beta}^i_{D,v}$, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ computed as in Javanmard and Montanari [2014] for the debiased lasso or as in Section 4.3.1 for the debiased multi-task fused lasso. When the respective assumptions of these estimators are satisfied the following holds w.h.p.*

$$\hat{\beta}^i_{D,v} - \beta^i_{D,v} = W + o(1) \quad \textit{where} \quad W \sim \mathbf{N}(0, [\sigma_1^2\boldsymbol{M}_1\hat{\Sigma}_1\boldsymbol{M}_1^T + \sigma_2^2\boldsymbol{M}_2\hat{\Sigma}_2\boldsymbol{M}_2^T]_{i,i}) \tag{4.3.32}$$

Algorithm 1 Difference Network Selection with Neighborhood Debiased Lasso

$V = \{1, ..., P\}$
NxP Data Matrices, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$
Px(P-1) Output Matrix $\boldsymbol{B}$ of test statistics
for $v \in V$ **do**
  Estimate unbiased $\hat{\sigma}_1, \hat{\sigma}_2$ from $X_{1,v}, X_{2,v}$
  for $j \in \{1, 2\}$ **do**
    $\beta_j \leftarrow SolveLasso(\boldsymbol{X}_{j,v^c}, X_{j,v})$
    $\boldsymbol{M}_j \leftarrow MEstimator(\boldsymbol{X}_{j,v^c})$
    $\beta_{j,U} \leftarrow \beta_j + \boldsymbol{M}_j \boldsymbol{X}_{j,v^c}^T (X_{j,v} - \boldsymbol{X}_{j,v^c}\beta_j)$
  end for
  $\sigma_d^2 \leftarrow diag(\frac{\hat{\sigma_1}^2}{n_1}\boldsymbol{M}_1^T\hat{\Sigma}_1\boldsymbol{M}_1 + \frac{\hat{\sigma_2}^2}{n_2}\boldsymbol{M}_2^T\hat{\Sigma}_2\boldsymbol{M}_2)$
  for $j \in v^c$ **do**
    $\boldsymbol{B}_{v,j} = (\beta_{1,U,j} - \beta_{2,U,j})/\sqrt{\sigma_{d,j}^2}$
  end for
end for

Algorithm 2 Difference Network Selection with Neighborhood Debiased Fused Lasso

$V = \{1, ..., P\}$
NxP Data Matrices, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$
Px(P-1) Output Matrix $\boldsymbol{B}$ of test statistics
for $v \in V$ **do**
  Estimate unbiased $\hat{\sigma}_1, \hat{\sigma}_2$ from $X_{1,v}, X_{2,v}$
  $\beta_1, \beta_2 \leftarrow FusedLasso(\boldsymbol{X}_{1,v^c}, X_{1,v}, \boldsymbol{X}_{2,v^c}, X_{2,v})$
  $\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow MEstimator(\boldsymbol{X}_{1,v^c}, \boldsymbol{X}_{2,v^c})$
  for $j \in \{1, 2\}$ **do**
    $\beta_{j,U} \leftarrow \beta_j + \boldsymbol{M}_j \boldsymbol{X}_{j,v^c}^T (X_{j,v} - \boldsymbol{X}_{j,v^c}\beta_j)$
  end for
  $\sigma_d^2 \leftarrow diag(\frac{\hat{\sigma_1}^2}{n_1}\boldsymbol{M}_1^T\hat{\Sigma}_1\boldsymbol{M}_1 + \frac{\hat{\sigma_2}^2}{n_2}\boldsymbol{M}_2^T\hat{\Sigma}_2\boldsymbol{M}_2)$
  for $j \in v^c$ **do**
    $\boldsymbol{B}_{v,j} = (\beta_{1,U,j} - \beta_{2,U,j})/\sqrt{\sigma_{d,j}^2}$
  end for
end for

This follows directly from the asymptotic consistency of each individual $\hat{\beta}^i_{j,dL,v}$ for the debiased lasso and multi-task fused lasso.

We can now define the the null hypothesis of interest as $H_0 : \boldsymbol{\Theta}_{1,(i,j)} = \boldsymbol{\Theta}_{2,(i,j)}$. Obtaining a test statistic for each element $\beta^i_{D,v}$ allows us to perform hypothesis testing on individual edges, all the edges, or groups of edges (controlling for the FWER). We summarize the Neighbourhood Debiased Lasso Selection process in Algorithm 1 and the Neighbourhood Debiased Multi-Task Fused Lasso Selection in Algorithm 2 which can be used to obtain a matrix of all the relevant test statistics.

4.4 Experiments

4.4.1 Simulations

We generate synthetic data based on two Gaussian graphical models with 75 vertices. Each of the individual graphs have a sparsity of 19% and their difference sparsity is 3%. We construct the models by taking two identical precision matrices and randomly removing some edges from both. We generate synthetic data using both precision matrices. We use $n_1 = 800$ samples for the first dataset and vary the second dataset $n_2 = 20, 30, ...150$.

We perform a regression using the debiased lasso and the debiased multi-task fused lasso on each node of the graphs. As an extra baseline we consider the projected ridge method from the R package "hdi" [Dezeure et al., 2015]. We use the debiased lasso of Javanmard and Montanari [2014], where we set $\lambda = k\hat{\sigma}\sqrt{\log p/n}$. We select c by 3-fold cross validation $k = \{0.1, ..100\}$ and $\boldsymbol{M}$ as prescribed in Javanmard and Montanari [2014] which we obtain by solving a quadratic program. $\hat{\sigma}$ is an unbiased estimator of the noise variance. For the debiased lasso we let both $\lambda_1 = k_1\hat{\sigma_2}\sqrt{\log p/n_2}$ and $\lambda_2 = k_2\hat{\sigma_2}\sqrt{\log p/n_2}$, and select based on 3-fold cross-validation from the same range as k. $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are obtained as in Equation (4.3.11)

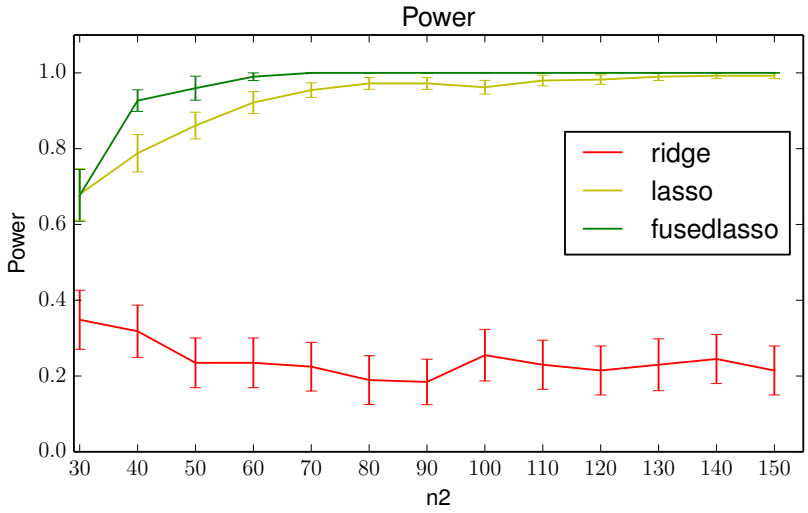


Figure 4.1 – Power of the test for different number of samples in the second simulation, with $n_1 = 800$. The debiased fused lasso has highest statistical power.

Method	FP	TP(Power)	Cov S	Cov S_d^c	len S	len S_d^c
Deb. Lasso	3.7%	80.6%	96.2%	92%	2.199	2.195
Deb. Fused Lasso	0.0%	93.3%	100%	98.6%	2.191	2.041
Ridge Projection	0.0%	18.6%	100%	100%	5.544	5.544

Table 4.1 – Comparison of Debiased Lasso, Debiased Fused Lasso, and Projected Ridge Regression for edge selection in difference of GGM. The significance level is 5%, $n_1 = 800$ and $n_2 = 60$. All methods have false positive below the significance level and the debiased fused lasso dominates in terms of power. The coverage of the difference support and non-difference support is also best for the debiased fused lasso, which simultaneously has smaller confidence intervals on average.

with the bounds (4.3.20) being set with $c = a = 2$, $s_d = 2, s_{1,2} = 15, m = 0.01$, and the cross validated λ_1 and λ_2. In both debiased lasso and fused multi-task lasso cases we utilize the Mosek QP solver package to obtain $\boldsymbol{M}$. For the projected ridge method we use the hdi package to obtain two estimates of β_1 and β_2 along with their upper bounded biases which are then used to obtain p-values for the difference.

We report the false positive rate, the power, the coverage and interval length as per Van de Geer et al. [2014] for the difference of graphs. In these experiments we aggregate statistics to demonstrate power of the test statistic, as such we consider each edge as a separate test and do not perform corrections. Table 4.1 gives the numerical results for $n_2 = 60$: the power and coverage is substantially better for the debiased fused multi-task lasso, while at the same time the confidence interval smaller.

Figure 4.1 shows the power of the test for different values of n_2. The fused lasso outperforms the other methods substantially. Projected ridge regression is particularly weak, in this scenario, as it uses a worst case p-value obtained using an estimate of an upper bound on the bias [Dezeure et al., 2015].

4.4.2 Autism Dataset

Correlations in brain activity measured via fMRI reveal functional interactions between remote brain regions [Lindquist et al., 2008]. In population analysis, they are used to measure how connectivity varies between different groups. Such analysis of brain function is particularly important in psychiatric diseases, that have no known anatomical support: the brain functions in a pathological aspect, but nothing abnormal is clearly visible in the brain tissues. Autism spectrum disorder is a typical example of such ill-understood psychiatric disease. Resting-state fMRI is accumulated in an effort to shed light on this diseases mechanisms: comparing the connectivity of autism patients versus control subjects. The ABIDE (Autism Brain Imaging Data Exchange) dataset [Di Martino et al., 2014] gathers rest-fMRI from 1,112 subjects, with 539 individuals suffering from autism spectrum disorder and 573 typical controls. We use the preprocessed and curated data.[1]

In a connectome analysis [Richiardi et al., 2011, Varoquaux and Craddock, 2013], each subject is described by a GGM measuring functional connectivity between a set of regions. We build a connectome from brain regions of interest based on a multi-subject atlas[2] of functional regions derived from resting-state fMRI [Varoquaux et al., 2011] (see Fig. 4.5).

We are interested in determining edge differences between the autism group and the control group. We use this data to show how our parametric hypothesis test can be used to determine differences in brain networks. Since no ground truth exists for this problem, we use permutation testing to evaluate the statistical procedures [Da Mota et al., 2014, Nichols and Holmes, 2002]. Here we permute the two conditions (e.g. autism and control group) to compute a p-value and compare it to our test statistics. This provides us with a finite sample strict control on the error rate: a non-parametric validation of our parametric test.

For our experiments we take 2000 randomly chosen volumes from the control group subjects and 100 volumes from the autism group subjects. We perform permutation testing using the de-biased lasso, de-biased multi-task fused lasso, and projected ridge regression. Parameters for the de-biased fused lasso are chosen as in the previous section. For the de-biased lasso we use the exact settings for λ and constraints on M provided in the experimental section of Javanmard and Montanari [2014]. Projected ridge regression is evaluated as in the previous section.

Figure 4.2 shows a comparison of three parametric approaches versus their analogue obtained with a permutation test. The chart plots the permutation p-values of each entry in the 38×39 $\boldsymbol{B}$ matrix against the expected parametric p-value. For all the methods the points are above the line indicating the tests are not breaching the expected false positive rates. However the de-biased lasso and ridge projecting are very conservative and lead to few detections. The de-biased multi-task fused lasso yields far more detections on the same dataset, within the expected false positive rate or near it.

[1] http://preprocessed-connectomes-project.github.io/abide/

[2] https://team.inria.fr/parietal/research/spatial_patterns/spatial-patterns-in-resting-state/

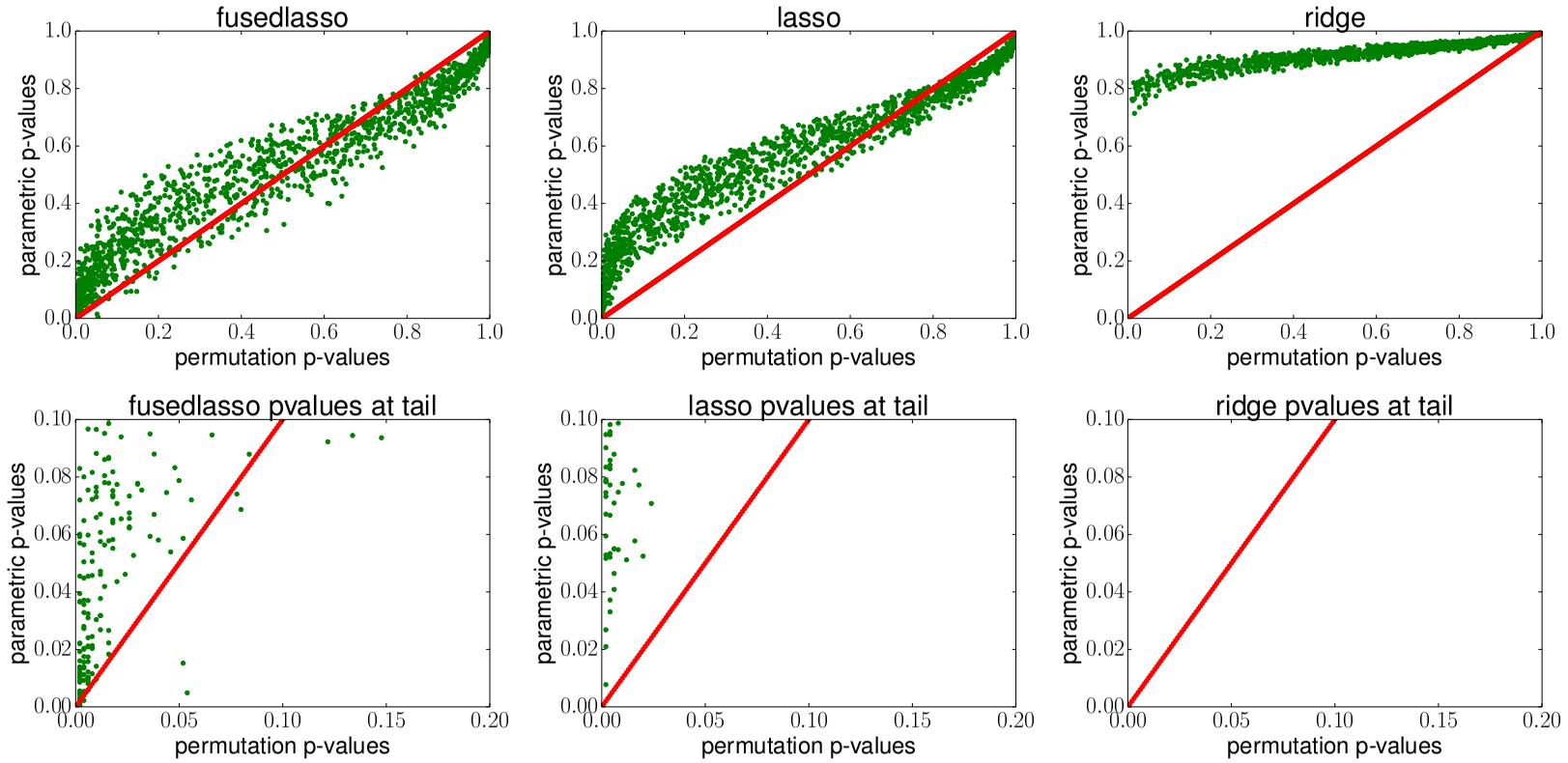


Figure 4.2 – Permutation testing comparing debiased fused lasso, debiased lasso, and projected ridge regression on the ABIDE dataset. The chart plots the permutation p-values of each method on each possible edge against the expected parametric p-value. The debiased lasso and ridge projection are very conservative and lead to few detections. The fused lasso yields far more detections on the same dataset, almost all within the expected false positive rate.

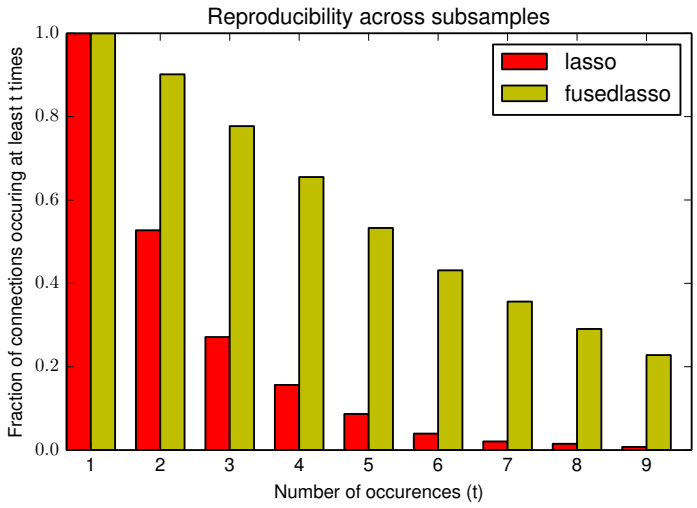


Figure 4.3 – Reproducibility of results from sub-sampling using uncorrected error rate. The fused lasso is much more likely to detect edges and produce stable results. Using corrected p-values no detections are made by lasso (figure in supplementary material).

We now analyse the reproducibility of the results by repeatedly sampling 100 subsets of the data (with the same proportions $n_1 = 2000$ and $n_2 = 100$), obtaining the matrix of test statistics, selecting edges that fall below the 5% significance level. Figure 4.3 shows how often edges are selected multiple times across subsamples. We report results with a threshold on uncorrected p-values as the lasso procedure selects no edges with multiple comparison correction (supplementary materials give FDR-corrected results for the de-biased fused multi-task lasso selection). Figure 4.5 shows a connectome of the edges frequently selected by the de-biased fused multi-task lasso (with FDR correction).

We show the corrected reproducibility results in Figure 4.6. For multiple testing correction in our experiments We use the Benjamin-Hochberg FDR procedure.

4.5 Conclusions

We have shown how to characterize the distribution of differences of sparse estimators and how to use this distribution for confidence intervals and p-values on GGM network differences. For this purpose, we have introduced the de-biased multi-task fused lasso. We have demonstrated on synthetic and real data that this approach can provide accurate p-values and a sizable increase of statistical power compared to standard procedures. The settings match those of population analysis for functional brain connectivity, and the gain in statistical power is direly needed to tackle the low sample sizes [Button et al., 2013].

Future work calls for expanding the analysis to cases with more than two groups as well as considering a $\ell_{1,2}$ penalty sometimes used at the group level [Varoquaux et al., 2010]. Additionally the squared loss objective optimizes excessively the prediction and could be modified to lower further the sample complexity in terms of parameter estimation.

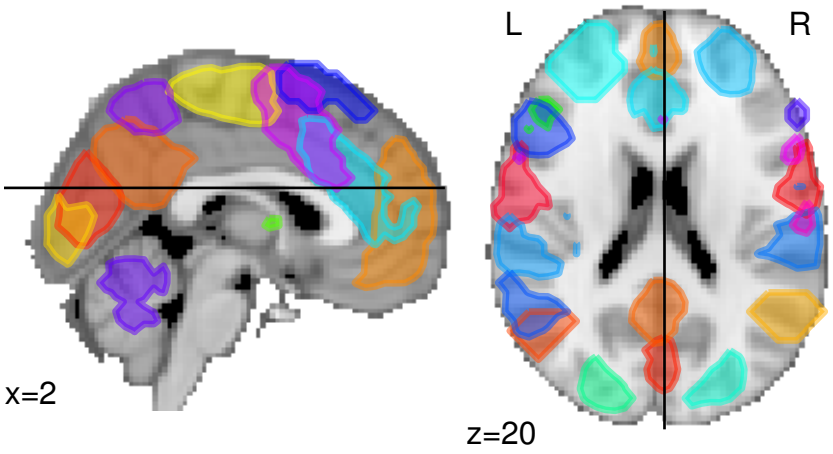


Figure 4.4 – Outlines of the regions of the MSDL atlas.

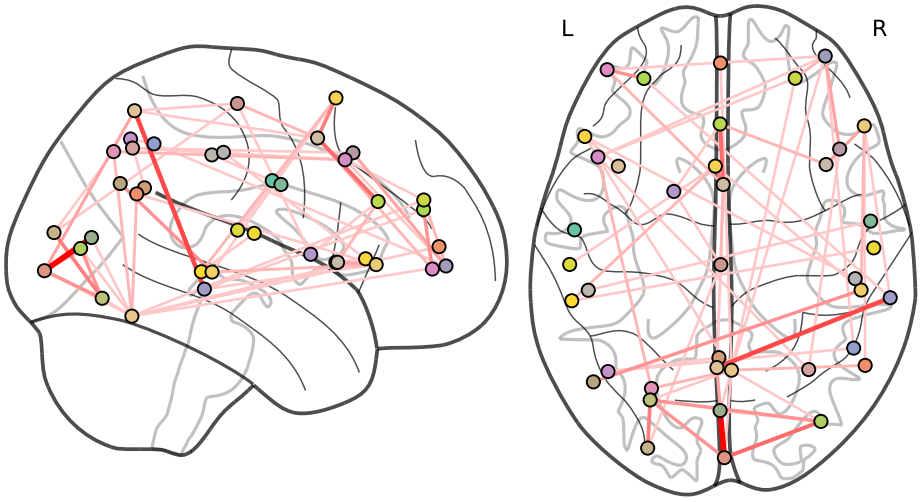


Figure 4.5 – Connectome of repeatedly picked up edges in 100 trials. We only show edges selected more than once. Darker red indicates more frequent selection.

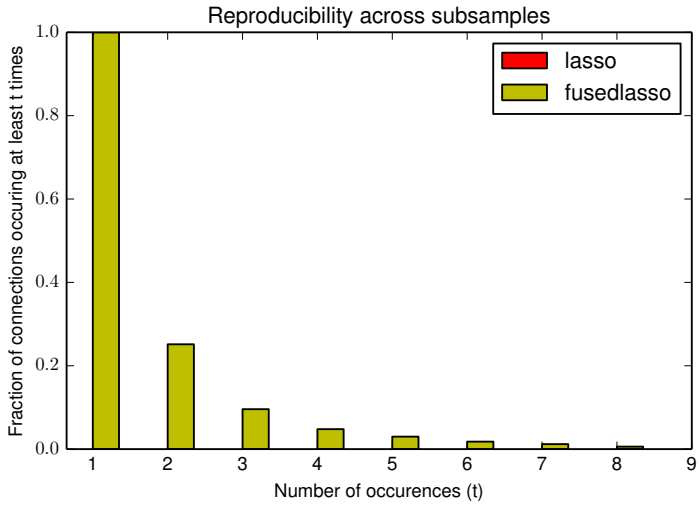


Figure 4.6 – Reproducibility of results from sub-sampling using FDR of 5% Reproducibility of results from subsampling, debiased lasso does not produce any significant edge differences that correspond to a 5% error rate

Chapter 5

Learning to Discover Sparse Graphical Model Structures

In the previous section we have seen that construction of novel structured sparsity penalties for GGMs can be a challenging process. We remark that in the case of Sparse Gaussian Graphical Models a generative model of the underlying data assumptions may often be relatively straightforward. In this section we ask whether one can leverage this fact and use learning approaches to construct efficient estimators of undirected graphical model structures and how well such estimators can generalize.

Contents

5.1 Introduction

Probabilistic graphical models provide a powerful framework to describe the dependencies between a set of variables. Many applications infer the structure of a probabilistic graphical model from data to elucidate the relationships between variables. These relationships are

often represented by an undirected graphical model also known as a Markov Random Field (MRF). We focus on a common MRF model, Gaussian graphical models (GGMs). GGMs are used in structure-discovery settings for rich data such as neuroimaging, genetics, or finance [Belilovsky et al., 2016b, Friedman et al., 2008, Mohan et al., 2012, Ryali et al., 2012]. Although multivariate Gaussian distributions are well-behaved, determining likely structures from few examples is a difficult task when the data is high dimensional. It requires strong priors, typically a sparsity assumption, or other restrictions on the structure of the graph, which now make the distribution difficult to express analytically and use.

A standard approach to estimating structure with GGMs in high dimensions is based on the classic result that the zeros of a precision matrix correspond to zero partial correlation, a necessary and sufficient condition for conditional independence [Lauritzen, 1996]. Assuming only a few conditional dependencies corresponds to a sparsity constraint on the entries of the precision matrix, leading to a combinatorial problem. Many popular approaches to learning GGMs can be seen as leveraging the ℓ_1-norm to create convex surrogates to this problem. Meinshausen and Bühlmann [2006] use nodewise ℓ_1 penalized regressions, while other estimators penalize the precision matrix directly [Cai et al., 2011, Friedman et al., 2008, Ravikumar et al., 2011], the most popular being the graphical lasso

$$f_{gl}(\hat{\mathbf{\Sigma}}) = \arg\min_{\mathbf{\Theta} \succ 0} - \log |\mathbf{\Theta}| + \mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\mathbf{\Theta}\right) + \lambda \|\mathbf{\Theta}\|_1 \tag{5.1.1}$$

which can be seen as a penalized maximum-likelihood estimator. Here $\mathbf{\Theta}$ and $\hat{\mathbf{\Sigma}}$ are the precision and sample covariance matrices, respectively. A large variety of alternative penalties extend the priors of the graphical lasso [Danaher et al., 2014, Ryali et al., 2012, Varoquaux et al., 2010]. However, this strategy faces several challenges. Constructing novel surrogates for structured-sparsity assumptions on MRF structures is difficult, as priors need to be formulated and incorporated into a penalized maximum likelihood objective which then calls for the development of an efficient optimization algorithm, often within a separate research effort. Furthermore, model selection in a penalized maximum likelihood setting is difficult as regularization parameters are often unintuitive.

We propose to learn the estimator. Rather than manually designing a specific graph-estimation procedure, we frame this estimator-engineering problem as a learning problem, selecting a function from a large flexible function class by risk minimization. This allows us to construct a loss function that explicitly aims to recover the edge structure. Indeed, sampling from a distribution of graphs and empirical covariances with desired properties is often possible, even when this distribution is not analytically tractable. As such we can perform empirical risk minimization to select an appropriate function for edge estimation. Such a framework gives more control on the assumed level of sparsity (as opposed to graph lasso) and can impose structure on the sampling to shape the expected distribution, while optimizing a desired performance metric.

For particular cases we show that the problem of interest can be solved with a polynomial function, which is learnable with a neural network [Andoni et al., 2014]. Motivated by this fact, as well as theoretical and empricial results on learning smooth functions approximating

solutions to combinatorial problems [Cohen et al., 2016, Vinyals et al., 2015], we propose to use a particular convolutional neural network as the function class. We train it by sampling small datasets, generated from graphs with the prescribed properties, with a primary focus on sparse graphical models. We estimate from this data small-sample covariance matrices ($n < p$), where n is the number of samples and p is the dimensionality of the data. Then we use them as training data for the neural network (Figure 5.2) where target labels are indicators of present and absent edges in the underlying GGM. The learned network can then be employed in various real-world structure discovery problems.

In Section 5.1.1 we review the related work. In Section 5.2 we formulate the risk minimization view of graph-structure inference and describe how it applies to sparse GGMs. Section 5.2.3 describes and motivates the deep-learning architecture we chose to use for the sparse GGM problem in this work. In Section 5.3 we describe the details of how we train an edge estimator for sparse GGMs. We then evaluate its properties extensively on simulation data. Finally, we show that this edge estimator trained only on synthetic data can obtain state of the art performance at inference time on real neuroimaging and genetics problems, while being much faster to execute than other methods.

5.1.1 Related Work

Lopez-Paz et al. [2015] analyze learning functions to identify the structure of directed graphical models in causal inference using estimates of kernel-mean embeddings. As in our work, they demonstrate the use of simulations for training while testing on real data. Unlike our work, they primarily focus on finding the causal direction in two node graphs with many observations.

Our learning architecture is motivated by the recent literature on deep networks. Vinyals et al. [2015] have shown that neural networks can learn approximate solutions to NP-hard combinatorial problems, and the problem of optimal edge recovery in MRFs can be seen as a combinatorial optimization problem. Several recent works have proposed neural architectures for graph input data [Duvenaud et al, 2015, Henaff et al., 2015, Li et al., 2016]. These are based on multi-layer convolutional networks, as in our work, or multi-step recurrent neural networks. The input in our approach can be viewed as a complete graph, while the output is a sparse graph, thus none of these are directly applicable. Related to our work, Balan et al. [2015] use deep networks to approximate a posterior distribution. Finally, Gregor and LeCun [2010], Xin et al. [2016] use deep networks to approximate steps of a known sparse recovery algorithm.

Bayesian approaches to structure learning rely on priors on the graph combined with sampling techniques to estimate the posterior of the graph structure. Some approaches make assumptions on the decomposability of the graph [Moghaddam et al., 2009]. The G-Wishart distribution is a popular distribution which forms part of a framework for structure inference, and advances have been recently made in efficient sampling [Mohammadi and Wit, 2015]. These methods can still be rather slow compared to competing methods, and in the setting

of $p > n$ we find they are less powerful.

5.2 Methods

5.2.1 Learning an Approximate Edge Estimation Procedure

We consider MRF edge estimation as a learnable function. Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a matrix whose n rows are i.i.d. samples $x \sim P(x)$ of dimension p. Let $G = (V, E)$ be an undirected and unweighted graph associated with the set of variables in x. Let $\mathcal{L} = \{0, 1\}$ and $N_e = \frac{p(p-1)}{2}$ the maximum possible edges in E. Let $Y \in \mathcal{L}^{N_e}$ indicate the presence or absence of edges in the edge set E of G, namely

$$Y^{ij} = \begin{cases} 0 & x_i \perp x_j | x_{V \setminus i,j} \\ 1 & x_i \not\perp x_j | x_{V \setminus i,j}. \end{cases} \tag{5.2.1}$$

We define an approximate structure discovery method $g_w(\boldsymbol{X})$, which predicts the edge structure, $\hat{Y} = g_w(\boldsymbol{X})$, given a sample of data $\boldsymbol{X}$. We focus on $\boldsymbol{X}$ drawn from a Gaussian distribution. In this case, the empirical covariance matrix, $\hat{\boldsymbol{\Sigma}}$, is a sufficient statistic of the population covariance and therefore of the conditional dependency structure. We thus express our structure-recovery problem as a function of $\hat{\boldsymbol{\Sigma}}$: $g_w(\boldsymbol{X}) := f_w(\hat{\boldsymbol{\Sigma}})$. f_w is parametrized by w and belongs to the function class $\mathcal{F}$. Note that the graphical lasso in Equation (5.1.1) is an f_w for a specific choice of $\mathcal{F}$.

This view on the edge estimator now allows us to bring the selection of f_w from the domain of human design to the domain of empirical risk minimization over $\mathcal{F}$. Defining a distribution $\mathbb{P}$ on $\mathbb{R}^{p \times p} \times \mathcal{L}^{N_e}$ such that $(\hat{\boldsymbol{\Sigma}}, Y) \sim \mathbb{P}$, we would like our estimator, f_w, to minimize the expected risk

$$R(f) = \mathbb{E}_{(\hat{\boldsymbol{\Sigma}}, Y) \sim \mathbb{P}}[l(f(\hat{\boldsymbol{\Sigma}}), Y)]. \tag{5.2.2}$$

Here $l : \mathcal{L}^{N_e} \times \mathcal{L}^{N_e} \to \mathbb{R}^+$ is the loss function. For graphical model selection the 0/1 loss function is the natural error metric to consider [Wang et al., 2010]. The estimator with minimum risk is generally not possible to compute as a closed form expression for most interesting choices of $\mathbb{P}$, such as those arising from sparse graphs. In this setting, Eq. (5.1.1) achieves the information theoretic optimal recovery rate up to a constant for certain $\mathbb{P}$ corresponding to uniformly sparse graphs with a maximum degree, but only when the optimal λ is used and the non-zero precision matrix values are bounded away from zero [Ravikumar et al., 2011, Wang et al., 2010].

The design of the estimator in Equation (5.1.1) is not explicitly minimizing this risk functional. Thus modifying the estimator to fit a different class of graphs (e.g. small-world networks) while minimizing $R(f)$ is not obvious. Furthermore, in practical settings the optimal λ is unknown and precision matrix entries can be very small. We would prefer to directly minimize the risk functional. Desired structural assumptions on samples from $\mathbb{P}$ on the underlying graph, such as sparsity, may imply that the distribution is not tractable for analytic solutions. Meanwhile, we can often devise a sampling procedure for $\mathbb{P}$ allowing us to select an appropriate function

via empirical risk minimization. Thus it is sufficient to define a rich enough $\mathcal{F}$ over which we can minimize the empirical risk over the samples generated, giving us a learning objective over N samples $\{Y_k, \mathbf{\Sigma}_k\}_{k=1}^N$ drawn from $\mathbb{P}$: $\min_w \frac{1}{N}\sum_{k=1}^N l(f_w(\hat{\mathbf{\Sigma}}_k), Y_k)$. To maintain tractability, we use the standard cross-entropy loss as a convex surrogate, $\hat{l} : \mathbb{R}^{N_e} \times \mathcal{L}^{N_e}$, given by

$$\sum_{i \neq j} (Y^{ij} \log(f_w^{ij}(\hat{\mathbf{\Sigma}})) + (1 - Y^{ij}) \log(1 - f_w^{ij}(\hat{\mathbf{\Sigma}}))).$$

We now need to select a sufficiently rich function class for f_w and a method to produce appropriate $(Y, \hat{\mathbf{\Sigma}})$ which model our desired data priors. This will allow us to learn a f_w that explicitly attempts to minimize errors in edge discovery.

5.2.2 Discovering Sparse GGMs and Beyond

We discuss how the described approach can be applied to recover sparse Gaussian graphical models. A typical assumption in many modalities is that the number of edges is sparse. A convenient property of these GGMs is that the precision matrix has a zero value in the (i, j)th entry precisely when variables i and j are independent conditioned on all others. Additionally, the precision matrix and partial correlation matrix have the same sparsity pattern, while the partial correlation matrix has normalized entries.

We propose to simulate our *a priori* assumptions of sparsity and Gaussianity to learn $f_w(\hat{\mathbf{\Sigma}})$, which can then produce predictions of edges from the input data. We model $P(x|G)$ as arising from a sparse prior on the graph G and correspondingly the entries of the precision matrix $\mathbf{\Theta}$. To obtain a single sample of $\boldsymbol{X}$ corresponds to n i.i.d. samples from $\mathcal{N}(0, \mathbf{\Theta}^{-1})$. We can now train $f_w(\hat{\mathbf{\Sigma}})$ by generating sample pairs $(\hat{\mathbf{\Sigma}}, Y)$. At execution time we standardize the input data and compute the covariance matrix before evaluating $f_w(\hat{\mathbf{\Sigma}})$. The process of learning f_w for the sparse GGM is given in Algorithm 3.

Algorithm 3 Training a GGM edge estimator

for i $\in \{1, .., N\}$ **do**
  Sample $G_i \sim \mathbb{P}(G)$
  Sample $\mathbf{\Theta}_i \sim \mathbb{P}(\mathbf{\Theta}|G = G_i)$
  $\boldsymbol{X}_i \leftarrow \{x_j \sim N(0, \mathbf{\Theta}_i^{-1})\}_{j=1}^n$
  Construct $(Y_i, \hat{\mathbf{\Sigma}}_i)$ pair from $(G_i, \boldsymbol{X_i})$
end for
Select Function Class $\mathcal{F}$ (e.g. CNN)
Optimize: $\min_{f \in \mathcal{F}} \frac{1}{N}\sum_{k=1}^N \hat{l}(f(\hat{\mathbf{\Sigma}}_k), Y_k))$

A weakly-informative sparsity prior is one where each edge is equally likely with small probability, versus structured sparsity where edges have specific configurations. For obtaining the training samples $(\hat{\mathbf{\Sigma}}, Y)$ in this case we would like to create a sparse precision matrix, $\mathbf{\Theta}$, with the desired number of zero entries distributed uniformly. One strategy to do this and assure the precision matrices lie in the positive definite cone is to first construct an upper triangular

sparse matrix and then multiply it by its transpose. This process is described in detail in the experimental section. Alternatively, an MCMC based G-Wishart distribution sampler can be employed if specific structures of the graph are desired [Lenkoski, 2013].

The sparsity patterns in real data are often not uniformly distributed. Many real world networks have a small-world structure: graphs that are sparse and yet have a comparatively short average distance between nodes. These transport properties often hinge on a small number of high-degree nodes called hubs. Normally, such structural patterns require sophisticated adaptation when applying estimators like Eq. (5.1.1). Indeed, high-degree nodes break the small-sample, sparse-recovery properties of ℓ_1-penalized estimators [Ravikumar et al., 2011]. In our framework such structural assumptions appear as a prior that can be learned offline during training of the prediction function. Similarly priors on other distributions such as general exponential families can be more easily integrated. As the structure discovery model can be trained offline, even a slow sampling procedure may suffice.

5.2.3 Neural Network Graph Estimator

In this work we propose to use a neural network as our function f_w. To motivate this let us consider the extreme case when $n \gg p$. In this case $\hat{\mathbf{\Sigma}} \approx \mathbf{\Sigma}$ and thus entries of $\hat{\mathbf{\Sigma}}^{-1}$ or the partial correlation that are almost equal to zero can give the edge structure. We can show that a neural network is consistent with this limiting case.

Definition 5.1 ($\mathbb{P}$-consistency)**.** *A function class $\mathcal{F}$ is $\mathbb{P}$-consistent if $\exists f \in \mathcal{F}$ such that $\mathbb{E}_{(\hat{\mathbf{\Sigma}},Y)\sim\mathbb{P}}[l(f(\hat{\mathbf{\Sigma}}), Y)] \to 0$ as $n \to \infty$ with high probability.*

Proposition 5 (Existence of $\mathbb{P}$-consistent neural network graph estimator)**.** *There exists a feed forward neural network function class $\mathcal{F}$ that is $\mathbb{P}$-consistent.*

Proof: If the data is standardized, each entry of $\mathbf{\Sigma}$ corresponds to the correlation $\rho_{i,j}$. The partial correlation of edge (i, j) conditioned on nodes $\boldsymbol{Z}$, is given recursively as

$$\rho_{i,j|\boldsymbol{Z}} = (\rho_{i,j|\boldsymbol{Z}\setminus z_o} - \rho_{i,z_o|\boldsymbol{Z}\setminus z_o}\rho_{j,z_o|\boldsymbol{Z}\setminus z_o})\frac{1}{D}. \tag{5.2.3}$$

We may ignore the denominator, D, as we are interested in $\mathbb{I}(\rho_{i,j|\boldsymbol{Z}} = 0)$. Thus we are left with a recursive formula that yields a high degree polynomial. From Andoni et al. [2014, Theorem 3.1] using gradient descent, a neural network with only two layers can learn a polynomial function of degree d to arbitrary precision given sufficient hidden units. □

Remark 5.1. *Naïvely the polynomial from the recursive definition of partial correlation is of degree bounded by 2^{p-2}. In the worst case, this would seem to imply that we would need an exponentially growing number of hidden nodes to approximate it. However, this problem has a great deal of structure that can allow efficient approximation. Firstly, higher order monomials will go to zero quickly with a uniform prior on $\rho_{i,j}$, which takes values between 0 and 1, suggesting that in many cases a concentration bound exists that guarantees non-exponential*

growth. Furthermore, the existence result is shown already for a shallow network, and we expect a logarithmic decrease in the number of parameters to peform function estimation with a deep network [Cohen et al., 2016].

Moreover, there are a great deal of redundant computations in Eq. (5.2.3) and an efficient dynamic programming implementation can yield polynomial computation time and require only low order polynomial computations with appropriate storage of previous computation. Similarly we would like to design a network that would have capacity to re-use computations across edges and approximate low order polynomials. We also observe that the conditional independence of nodes i, j given $\boldsymbol{Z}$ can be computed equivalently in many ways by considering many paths through the nodes $\boldsymbol{Z}$. Thus we can choose any valid ordering for traversing the nodes starting from a given edge.

We now describe an efficient architecture for this problem which uses a series of shared operations at each edge. We consider a feedforward network where each edge i, j is associated with a vector, $o^k_{i,j}$, at each layer $k > 0$. For each edge, i, j, we start with a neighborhood of the 6 adjacent nodes, i, j, i-1, i+1, j-1, j+1 for which we take all corresponding edge values from the covariance matrix and construct $o^1_{i,j}$. We proceed at each layer to increase the nodes considered for each $o^k_{i,j}$, the output at each layer progressively increasing the receptive field making sure all values associated with the considered nodes are present. The entries used at each layer are illustrated in Figure 5.1. The receptive field here refers to the original covariance entries which are accessible by a given, $o^k_{i,j}$ [Luo et al., 2010]. The equations defining the process are shown in Figure 5.1. Here a neural network f_{w^k} is applied at each edge at each layer and a dilation sequence d_k is used. We call a network of this topology a D-Net of depth l. We use dilation here to allow the receptive field to grow fast, so the network does not need a great deal of layers. We make the following observations:

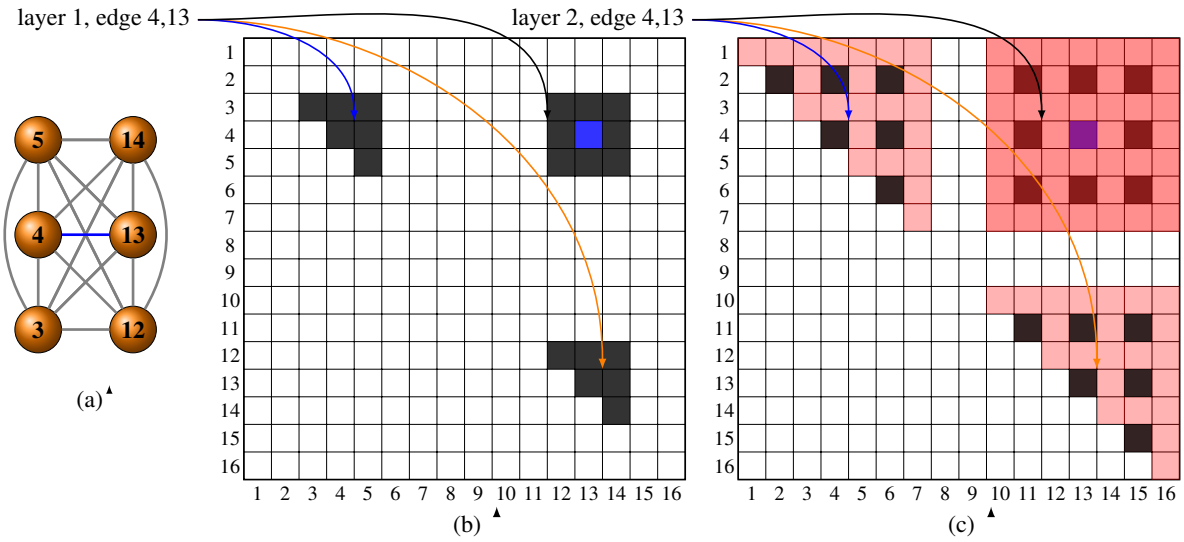
Proposition 6. *For general $\mathbb{P}$ it is a necessary condition for $\mathbb{P}$-consistency that the receptive field of D-Net covers all entries of the covariance, $\hat{\boldsymbol{\Sigma}}$, at any edge it is applied.*

Proof: Consider nodes i and j and a chain graph such that i and j are adjacent to each other in the matrix but are at the terminal nodes of the chain graph. One would need to consider all other variables to be able to explain away the correlation. Alternatively we can see this directly from expanding Eq. (5.2.3). □

Proposition 7. *A $p \times p$ matrix $\hat{\boldsymbol{\Sigma}}$ will be covered by the receptive field for a D-Net of depth $\log_2(p)$ and $d_k = 2^{k-1}$*

Proof: The receptive field of a D-Net with dilation sequence $d_k = 2^{k-1}$ of depth l is $O(2^l)$. We can see this as $o^k_{i,j}$ will receive input from $o^{k-1}_{a,b}$ at the edge of it's receptive field, effectively doubling it. It now follows that we need at least $\log_2(p)$ layers to cover the receptive field. □

Intuitively adjacent edges have a high overlap in their receptive fields and can easily share information about the non-overlapping components. This is analogous to a parametrized



$$
\begin{aligned}
o^0_{i,j} &= p_{i,j} \\
o^1_{i,j} &= f_{w^1}(o^0_{i,j}, o^0_{i\text{-}1,j}, o^0_{i,j\text{-}1}, o^0_{i+1,j\text{-}1}, ..) \\
o^2_{i,j} &= f_{w^2}(o^1_{i,j}, o^1_{i\text{-}d_2,j}, o^1_{i,j\text{-}d_2}, o^1_{i+d_2,j\text{-}d_2}, ..) \\
o^l_{i,j} &= f_{w^l}(o^{l\text{-}1}_{i,j}, o^{l\text{-}1}_{i\text{-}d_l,j}, o^{l\text{-}1}_{i,j\text{-}d_l}, o^{l\text{-}1}_{i+d_l,j\text{-}d_l}..) \\
\hat{y}_{i,j} &= \sigma(w^{l+1} o^l_{i,j})
\end{aligned}
$$

Figure 5.1 – (a) Illustration of nodes and edges "seen" at edge 4,13 in layer 1 and (b) Receptive field at layer 1. All entries in grey show the $o^0_{i,j}$ in covariance matrix used to compute $o^1_{4,13}$. (c) shows the dilation process and receptive field (red) at higher layers. Finally the equations for each layer output are given, initialized by the covariance entries $p_{i,j}$

message passing. For example if edge (i, j) is explained by node k, as k enters the receptive field of edge $(i, j-1)$, the path through (i, j) can already be discounted. In terms of Eq. (5.2.3) this can correspond to storing computations that can be used by neighbor edges from lower levels in the recursion.

As f_{w^k} is identical for all nodes, we can simultaneously implement all edge predictions efficiently as a convolutional network. We make sure that to have considered all edges relevant to the current set of nodes in the receptive field which requires us to add values from filters applied at the diagonal to all edges. In Figure 5.1 we illustrate the nodes and receptive field considered with respect to the covariance matrix. This also motivates a straightforward implementation using 2D convolutions (adding separate convolutions at i, i and j, j to each i, j at each layer to achieve the specific input pattern described) shown in Figure 5.2.

Ultimately our choice of architecture that has shared computations and multiple layers is highly scalable as compared with a naive fully connected approach and allows leveraging existing optimized 2-D convolutions. In preliminary work we have also considered fully connected layers but this proved to be much less efficient in terms of storage and scalibility than using deep convolutional networks.

Considering the general $n \gg p$ case is illustrative. However, the main benefit of making the computations differentiable and learned from data is that we can take advantage of the sparsity and structure assumptions to obtain more efficient results than naive computation of partial correlation or matrix inversion. As n decreases our estimate of $\hat{\rho}_{i,j}$ becomes inexact; a data-driven model that takes better advantage of the assumptions on the underlying distribution and can more accurately recover the graph structure.

The convolution structure is dependent on the order of the variables used to build the covariance matrix, which is arbitrary. Permuting the input data we can obtain another estimate of the output. In the experiments, we leverage these various estimate in an ensembling approach, averaging the results of several permutations of input. We observe that this generally yields a modest increase in accuracy, but that even a single node ordering can show substantially improved performance over competing methods in the literature.

5.3 Experiments

Our experimental evaluations focus on the challenging high dimensional settings in which $p > n$ and consider both synthetic data and real data from genetics and neuroimaging. In our experiments we explore how well networks trained on parametric samples generalize, both to unseen synthetic data and to several real world problems. In order to highlight the generality of the learned networks, we apply the same network to multiple domains. We train networks taking in 39, 50, and 500 node graphs. The former sizes are chosen based on the real data we consider in subsequent sections. We refer to these networks as DeepGraph-39, 50, and 500. In all cases we have 50 feature maps of 3×3 kernels. The 39 and 50 node network with 6 convolutional layers and $d_k = k + 1$. For the 500 node network with 8 convolutional layers

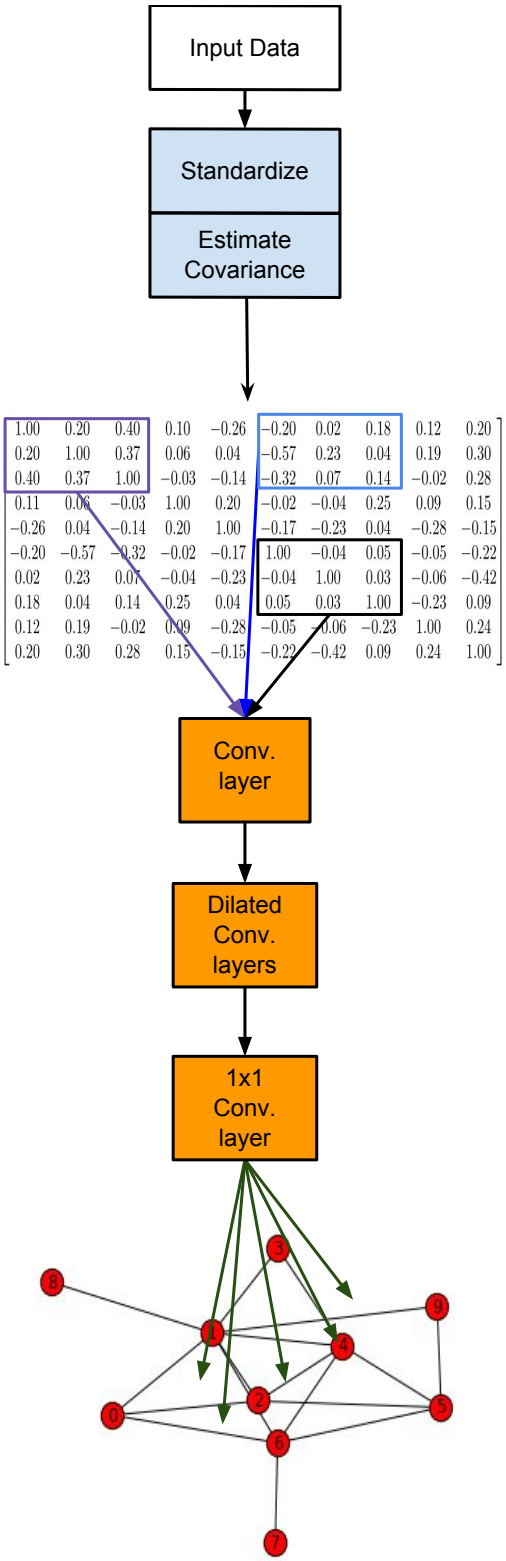


Figure 5.2 – Diagram of the DeepGraph structure discovery architecture used in this work. The input is first standardized and then the sample covariance matrix is estimated. A neural network consisting of multiple dilated convolutions Yu and Koltun [2016] and a final 1×1 convolution layer is used to predict edges corresponding to non-zero entries in the precision matrix.

and $d_k = 2^{k+1}$. We use ReLU activations. The last layer has 1×1 convolution and a sigmoid outputing a value of 0 to 1 for each edge.

We sample $P(X|G)$ with a sparse prior on $P(G)$ as follows. We first construct a lower diagonal matrix, L, where each entry has α probability of being zero. Non-zero entries are set uniformly between $-c$ and c. Multiplying LL^T gives a sparse positive definite precision matrix, $\mathbf{\Theta}$. This gives us our $P(\mathbf{\Theta}|G)$ with a sparse prior on $P(G)$. We sample from the Gaussian $\mathcal{N}(0, \mathbf{\Theta}^{-1})$ to obtain samples of $\boldsymbol{X}$. Here α corresponds to a specific sparsity level in the final precision matrix, which we set to produce matrices $92 - 96\%$ sparse and c chosen so that partial correlations range 0 to 1.

Each network is trained continously with new samples generated until the validation error saturates. For a given precision matrix we generate 5 possible $\boldsymbol{X}$ samples to be used as training data, with a total of approximately $100K$ training samples used for each network. The networks are optimized using ADAM [Kingma and Ba, 2015b] coupled with cross-entropy loss as the objective function (cf. Sec. 5.2.1). We use batch normalization at each layer. Additionally, we found that using the absolute value of the true partial correlations as labels, instead of hard binary labels, improves results.

Synthetic Data Evaluation To understand the properties of our learned networks, we evaluated them on different synthetic data than the ones they were trained on. More specifically, we used a completely different third party sampler so as to avoid any contamination. We use DeepGraph-39, which takes 4 hours to train, on a variety of settings. The same trained network is utilized in the subsequent neuroimaging evaluations as well. DeepGraph-500 is also used to evaluate larger graphs.

We used the `BDGraph` R-package to produce sparse precision matrices based on the G-Wishart distribution [Mohammadi and Wit, 2015] as well as the R-package `rags2ridges` [Peeters et al., 2015] to generate data from small-world networks corresponding to the Watts–Strogatz model [Watts and Strogatz, 1998]. We compared our learned estimator against the `scikit-learn` [Pedregosa et al, 2011] implementation of Graphical Lasso with regularizer chosen by cross-validation as well as the Birth-Death Rate MCMC (BDMCMC) method from Mohammadi and Wit [2015].

For each scenario we repeat the experiment for 100 different graphs and small sample observations showing the average area under the ROC curve (AUC), precision@k corresponding to 5% of possible edges, and calibration error (CE) [Mohammadi and Wit, 2015].

For graphical lasso we use the partial correlations to indicate confidence in edges; `BDGraph` automatically returns posterior probabilities as does our method. Finally to understand the effect of the regularization parameter we additionally report the result of graphical lasso under optimal regularizer setting on the testing data.

Our method dominates all other approaches in all cases with $p > n$ (which also corresponds to the training regime). For the case of random Gaussian graphs with n=35 (as in our training

data), and graph sparsity of 95%, we have superior performance and can further improve on this by averaging permutations. Next we apply the method to less straightforward synthetic data, such as that arising from small-world graphs which is typical of many applications. We found that, compared to baseline methods, our network performs particularly well with high-degree nodes and when the distribution becomes non-normal. In particular our method performs well on the relevant metrics with small-world networks, a very common family of graphs in real-world data, obtaining superior precision at the primary levels of interest. Figure 5.3 shows examples of random and Watts-Strogatz small-world graphs used in these experiments.

Training a new network for each number of samples can pose difficulties with our proposed method. Thus we evaluted how robust the network DeepGraph-39 is to input covariances obtained from fewer or more samples. We find that overall the performance is quite good even when lowering the number of samples to $n = 15$, we obtain superior performance to the other approaches (Table 5.1). We also applied DeepGraph-39 on data from a multivariate generalization of the Laplace distribution [Gómez, 1998]. As in other experiments precision matrices were sampled from the G-Wishart at a sparsity of 95%. Gómez [1998, Proposition 3.1] was applied to produce samples. We find that DeepGraph-39 performs competitively, despite the discrepancy between train and test distributions. Experiments with variable sparsity are considered in the supplementary material, which find that for very sparse graphs, the networks remain robust in performance, while for increased density performance degrades but remains competitive.

Using the small-world network data generator [Peeters et al., 2015], we demonstrate that we can update the generic sparse prior to a structured one. We re-train DeepGraph-39 using only 1000 examples of small-world graphs mixed with 1000 examples from the original uniform sparsity model. We perform just one epoch of training and observe markedly improved performance on this test case as seen in the last row of Table 5.1.

For our final scenario we consider the very challenging setting with 500 nodes and only $n = 50$ samples. We note that the MCMC based method fails to converge at this scale, while graphical lasso is very slow as seen in the timing performance and barely performs better than chance. Our method convincingly outperforms graphical lasso in this scenario as shown in Tabel 5.2. Here we additionally report precision at just the first 0.05% of edges since competitors perform nearly at chance at the 5% level.

We compute the average execution time of our method compared to Graph Lasso and BD-Graph on a CPU in Table 5.3. We note that we use a production quality version of graph lasso [Pedregosa et al, 2011], whereas we have not optimized the network execution, for which known strategies may be applied [Denton et al., 2014].

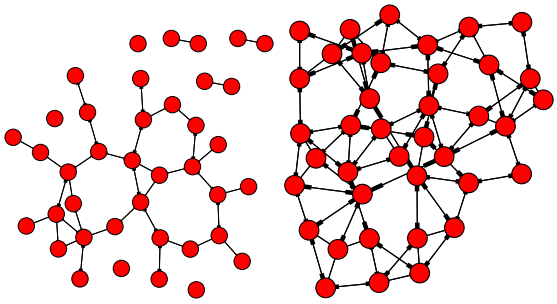


Figure 5.3 – Example of (a) random and (b) small world used in experiments

Method	Prec@0.05%	Prec@5%	AUC	CE
random	0.052 ± 0.002	0.053 ± 0.000	0.500 ± 0.000	0.05
Glasso	0.156 ± 0.010	0.055 ± 0.001	0.501 ± 0.000	0.05
Glasso (optimal)	0.162 ± 0.010	0.055 ± 0.001	0.501 ± 0.000	0.05
DeepGraph-500	0.449 ± 0.018	0.109 ± 0.002	0.543 ± 0.002	0.06
DeepGraph-500+Perm	**0.583 ± 0.018**	**0.116 ± 0.002**	**0.547 ± 0.002**	**0.06**

Table 5.2 – Experiment on 500 node graphs with only 50 samples repeated 100 times. This corresponds to the experimental setup of Gaussian Random Graphs (n=50,p=500). Improved performance in all metrics.

5.3.1 Synthetic Results on Sparsity

We investigate the affect of sparsity on DeepGraph-39 which has been trained with input that has sparsity $96\% - 92\%$ sparse. We find that DeepGraph performs well at the 2% sparsity level despite not seeing this at training time. At the same time performance begins to degrade for 15% but is still competitive in several categories. The results are shown in Table 5.7.

Cancer Genome Data We perform experiments on a gene expression dataset described in Honorio et al. [2012a]. The data come from a cancer genome atlas from 2360 subjects for various types of cancer. We used the first 50 genes from Honorio et al. [2012a, Appendix C.2] of commonly regulated genes in cancer. We evaluated on two groups of subjects, one with breast invasive carcinoma (BRCA) consisting of 590 subjects and the other colon adenocarcinoma (COAD) consisting of 174 subjects.

Evaluating edge selection in real-world data is challenging. We use the following methodology: for each method we select the top-k ranked edges, recomputing the maximum likelihood precision matrix with support given by the corresponding edge selection method. We then evaluate the likelihood on held-out data. We repeat this procedure for a range of k. We rely on Algorithm 0 in Hara and Takemura [2010] to compute the maximum likelihood precision given a support. The experiment is repeated for each of CODA and BRCA subject groups 150 times. Results are shown in Figure 5.5. In all cases we use 40 samples for edge selection and precision estimation. We compare with graphical lasso as well as the Ledoit-Wolf shrinkage

estimator [Ledoit and Wolf, 2004]. We additionally consider the MCMC based approach described in previous section. For graphical lasso and Ledoit-Wolf, edge selection is based on thresholding partial correlation [Balmand and Dalalyan, 2016].

Additionally, we evaluate the stability of the solutions provided by the various methods. In several applications a low variance on the estimate of the edge set is important. On Table 5.5, we report Spearman correlations between pairs of solutions, as it is a measure of a monotone link between two variables. DeepGraph has far better stability in the genome experiments and is competitive in the fMRI data.

Resting State Functional Connectivity We evaluate our graph discovery method to study brain functional connectivity in resting-state fMRI data. Correlations in brain activity measured via fMRI reveal functional interactions between remote brain regions. These are an important measure to study psychiatric diseases that have no known anatomical support. Typical connectome analysis describes each subject or group by a GGM measuring functional connectivity between a set of regions [Varoquaux and Craddock, 2013]. We use the ABIDE dataset [Di Martino et al., 2014], a large scale resting state fMRI dataset. It gathers brain scans from 539 individuals suffering from autism spectrum disorder and 573 controls over 16 sites.[1] For our experiments we use an atlas with 39 regions of interest described in Varoquaux et al. [2011].

We use the network DeepGraph-39, the same network and parameters from synthetic experiments, using the same evaluation protocol as used in the genomic data. For both control and autism patients we use time series from 35 random subjects to estimate edges and corresponding precision matrices. We find that for both the Autism and Control group we can obtain edge selection comparable to graph lasso for very few selected edges. When the number of selected edges is in the range above 25 we begin to perform significantly better in edge selection as seen in Fig. 5.5. We evaluated stability of the results as shown in Tab. 5.5. DeepGraph outperformed the other methods across the board.

ABIDE has high variability across sites and subjects. As a result, to resolve differences between approaches, we needed to perform 1000 folds to obtain well-separated error bars. We found that the birth-death MCMC method took very long to converge on this data, moreover the need for many folds to obtain significant results amongst the methods made this approach prohibitively slow to evaluate.

We show the edges returned by Graph Lasso and DeepGraph for a sample from 35 subjects (Fig. 5.6) in the control group. We also show the result of a large-sample result based on 368 subjects from graphical lasso. In visual evaluation of the edges returned by DeepGraph we find that they closely align with results from a large-sample estimation procedure. Furthermore we can see several edges in the subsample which were particularly strongly activated in both methods.

[1] http://preprocessed-connectomes-project.github.io/abide/

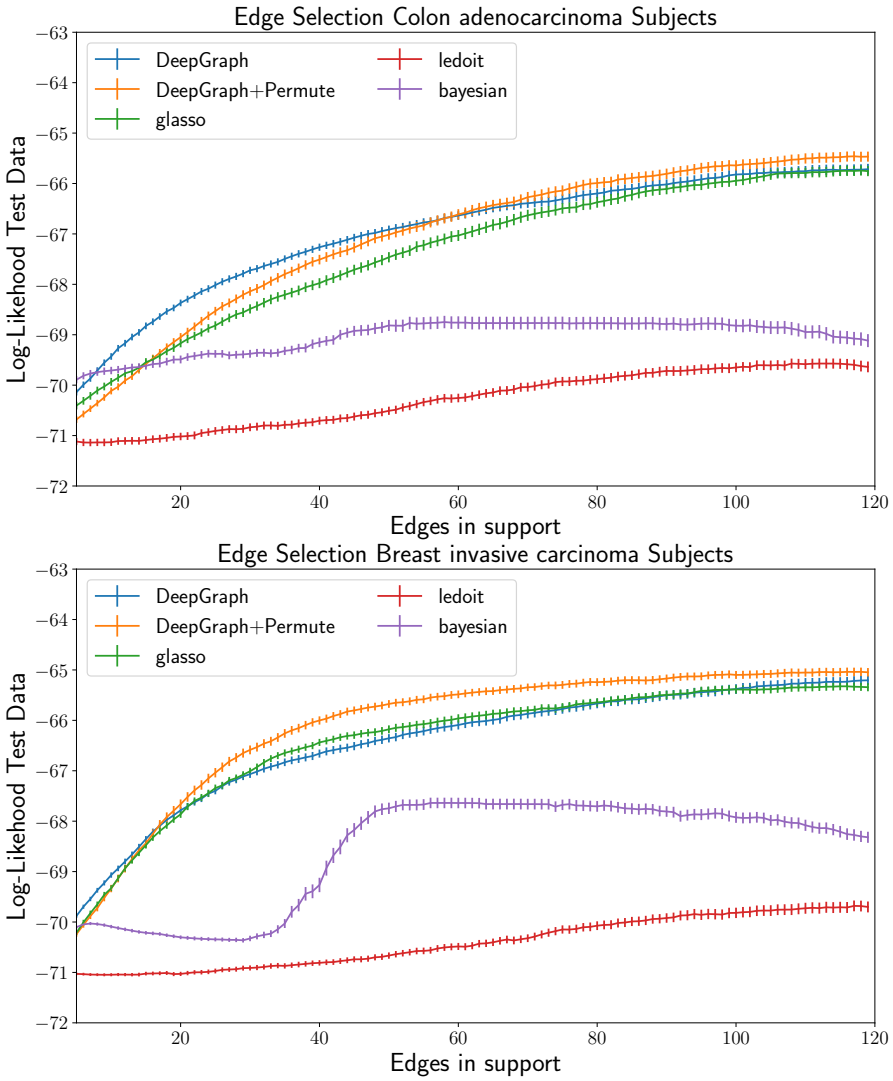


Figure 5.4 – Average test likelihood for COAD and BRCA subject groups in gene data using different number of selected edges. Each experiment is repeated 50 times. DeepGraph with averaged permutation dominates in all cases.

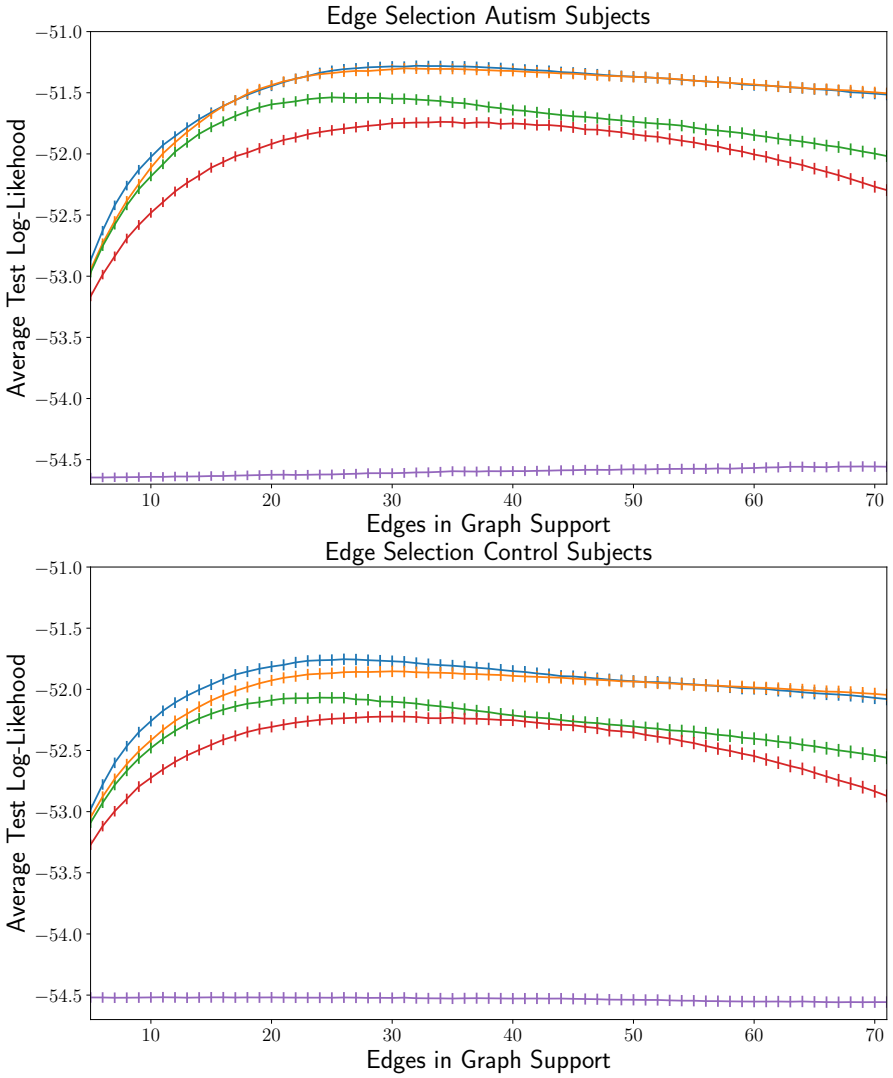


Figure 5.5 – Average test likelihood for neuroimaging data using different number of selected edges. It is repeated approximately 1500 times to obtain significant results due high variance in the data. DeepGraph+Permutation is superior or equal to competing methods.

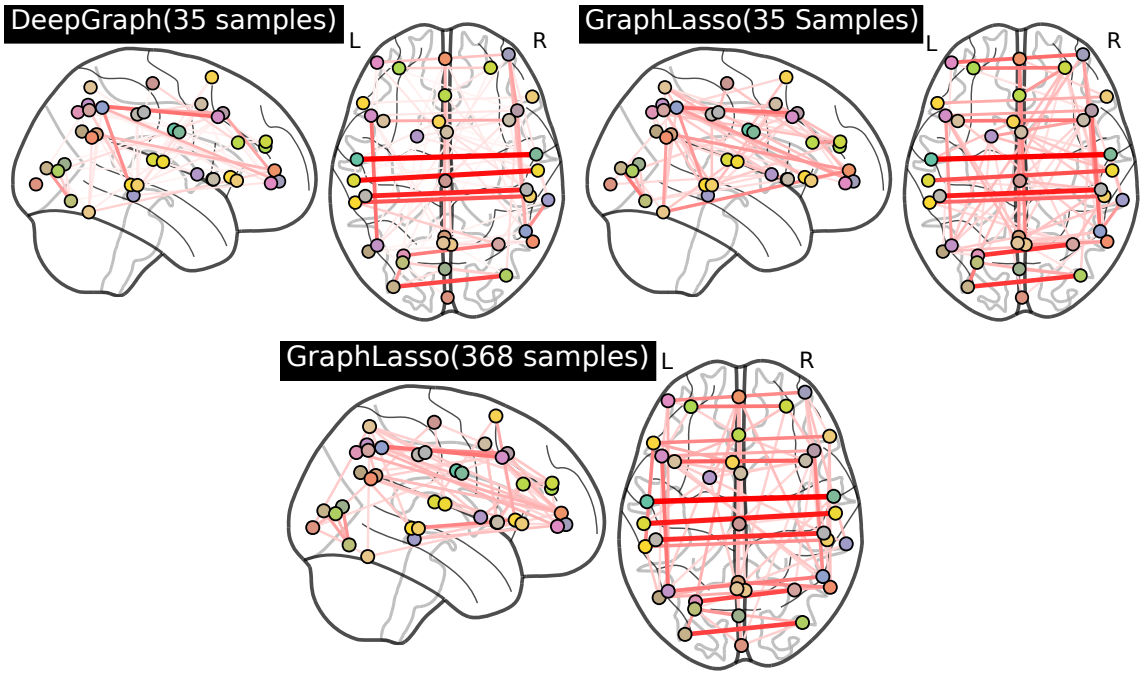


Figure 5.6 – Example solution from DeepGraph and Graph Lasso in the small sample regime on the same 35 samples, along with a larger sample solution of Graph Lasso for reference. DeepGraph is able to extract similar key edges as graphical lasso

5.3.2 Analyzing the Network Jacobian

In Graves [2012] a method for loosely obtaining the attention of a trained neural network is used. The jacobian matrix of partial derivatives can be used for a given input to get an idea of which entries in the input matrix most affect the given outputs. We use this approach to begin to analyze the behavior of our DeepGraph-39 network. We gave one empirical covariance matrix to the network which was generated from the underlying graph specified in 5.7. We first note the jacobian is concentrated in just a few values, the top activated edges for the input closely correspond to the higher value partials in the overall jacobian matrix.

For the most activated output edges we visualize the entries or edges in the input (which can also be thought of as entries in a weighted adjacency matrix) that are most activated. We do this by taking a softmax of all partials for a given output edge and only keeping outliers (2 standard deviations). In 5.7 we show the top partials for 4 of the top output edges. The relevant covariance entry corresponding to the edge at the output is almost always amongst the top partial values. We note that there is no explicit association of input relations outputs specified before or during training of the network so it has learned to associate the input and outputs. When there are other covariance entries pointed to they are usually connected to the relevant nodes either directly or indirectly. This may indicate the network is learning an algorithm to more succintly represent the connections of the two nodes.

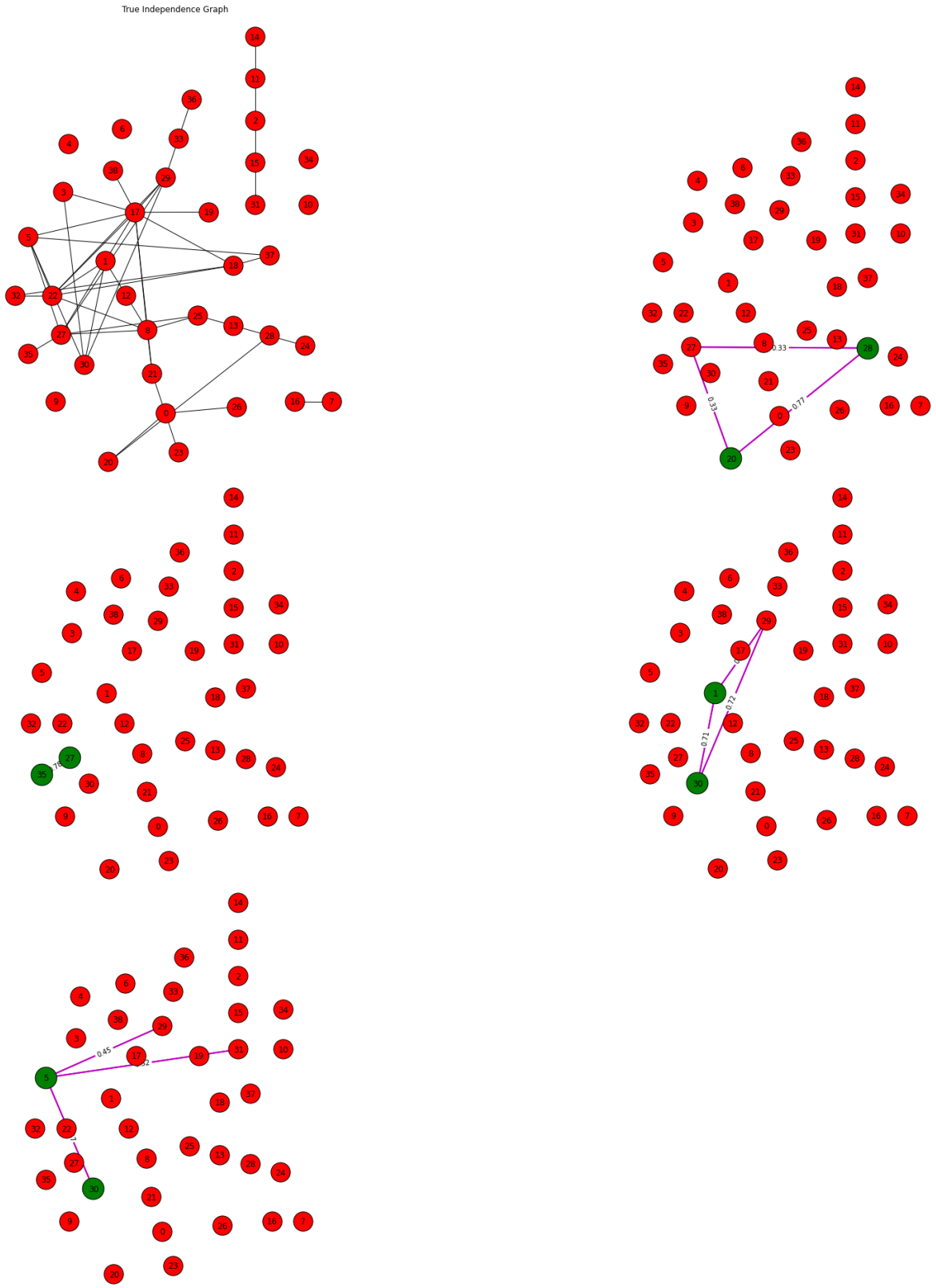


Figure 5.7 – True graph structure and top activated partials corresponding to the covariance input for top activated outputs of the network. The two green nodes indicate the connection being evaluated, the magenta edges show the top partials corresponding to the input. We see the network is learning to associate outputs and inputs (not specified in any way) and potentially explore correlated nodes amongst the considered ones

5.3.3 Predicting Covariance Matrices

Using our framework it is possible to attempt to directly predict an accurate covariance matrix given a noisy one constructed from few observations. This is a more challenging task than predicting the edges. In this section we show preliminay experiments which given an empirical covariance matrix from few observations attempts to predict a more accurate covariance matrix that takes into account underlying sparse data dependency structure.

One challenge is that outputs of our covariance predictor must be on the positive semidefinite cone, thus we choose to instead predict on the cholesky decompositions, which allows us to always produce positive definite covariances. We train a similar structure to DeepGraph-39 structure modifying the last layer to be fully connected linear layer that predicts on the cholesky decomposition of the true covariance matrices generated by our model with a squared loss.

We evaluate this network using the ABIDE dataset described in Section 5.3.1. The ABIDE data has a large number of samples allowing us to obtain a large sample estimate of the covariance and compare it to our estimator as well as graphical lasso and empirical covariance estimators. Using the large sample ABIDE empirical covariance matrix. We find that we can obtain competitive ℓ_2 and ℓ_∞ norm using few samples. We use 403 subjects from the ABIDE Control group each with a recording of $150 - 200$ samples to construct covariance matrix, totaling 77 330 samples (some correlated). This acts as our very approximate estimate of the population Σ. We then evaluate covariance estimation on 35 samples using the empirical covariance estimator, graphical lasso, and DeepGraph trained to output covariance matrices. We repeat the experiment for 50 different subsamples of the data. We see in 5.6 that the prediction approach can obtain competitive results. In terms of ℓ_2 graphical lasso performs better, however our estimate is better than empirical covariance estimation and much faster then graphical lasso. In some applications such as robust estimation a fast estimate of the covariance matrix (automatically embedding sparsity assumptions) can be of great use. For ℓ_∞ error we see the empirical covariance estimation outperforms graphical lasso and DeepGraph for this dataset, while DeepGraph performs better in terms of this metric.

We note these results are preliminary, as the covariance predicting networks were not heavily optimized, moreover the ABIDE dataset is very noisy even when pre-processed and thus even the large sample covariance estimate may not be accurate. We believe this is an interesting alternate application of our paper.

5.3.4 Synthetic Results on Sparsity

We investigate the affect of sparsity on DeepGraph-39 which has been trained with input that has sparsity $96\% - 92\%$ sparse. We find that DeepGraph performs well at the 2% sparsity level despite not seeing this at training time. At the same time performance begins to degrade for 15% but is still competitive in several categories. The results are shown in Table 5.7. Future investigation can consider how alternate variation of sparsity at training time will affect these

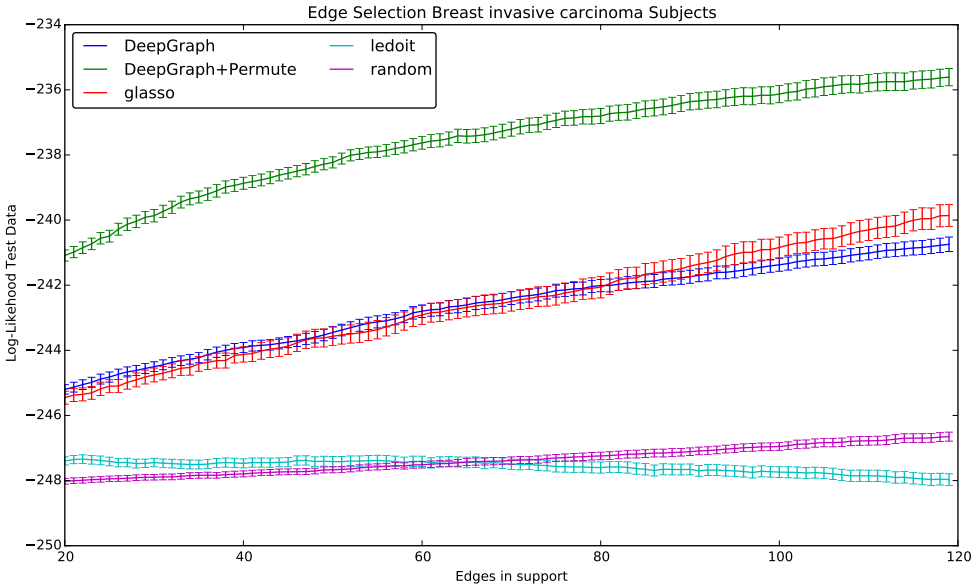


Figure 5.8 – Average test likelihood over 50 trials of applying a network trained for 500 nodes, used on a 175 node problem

results.

5.3.5 Application of Larger Network on Smaller Input

We perform preliminary investigation of application of a network trained for a larger number of nodes to a smaller set of nodes. Specifically, we consider the breast invasive carcinoma groups gene data. We now take all 175 valid genes from Appendix C.2 of [Honorio et al., 2012a]. We take the network trained on 500 nodes in the synthetic experiments section. We use the same experimental setup as in the gene experiments. The 175×175 covariance matrix from 40 samples and padded to the appropriate size. We observe that DeepGraph has similar performance to graph lasso while permuting the input and ensembling the result gives substantial improvement.

5.3.6 Permutation as Ensemble Method

The proposed method is not permutation invariant, meaning different permutations of the input may give different results. The algorithm will be biased by the training procedure to learn permutation-invariance, but ultimately the predictions for different input permutations may differ in their prediction. One can consider how to structure the input such that more relevant data points are closer together. On the other hand we argue that not being permutation invariant is a potential advantage of this method. Indeed as discussed in Section 5.2.3, permuting the input and averaging several permutations can produce an improved result empirically. We interpret this as a typical ensembling method. This can be an advantage of the proposed architecture as we are able to easily use standard ensemble techniques. We perform

an experiment to further verify that indeed the permutation of the input (and subsequent inverse permutation) allows us to produce separate classifiers that have uncorrelated errors.

We use the setup from the synthetic experiments with DeepGraph-39 in Section 5.3 with $n = 35$ and $p = 39$. We construct 20 permutation matrices as in the experimental section. Treating each as a separate classifier we compute the correlation coefficient of the errors on 50 synthetic input examples. We find that the average correlation coefficient of the errors of two classifiers is 0.028 ± 0.002, suggesting they are uncorrelated. Finally we note the individual errors are relatively small, as can already be inferred from our extensive experimental results in Section 5.3. We however compute the average absolute error of all the outputs across each permutation for this set of inputs as 0.03, notably the range of outputs is 0 to 1. Thus since prediction error differ at each permutation but are accurate we can average and yield a lower total prediction error.

Finally we note that our method is extremely efficient computationally thus averaging the results of several permutations is practical even as the graph becomes large.

5.4 Discussion and Conclusions

Learned graph estimation outperformed strong baselines in both accuracy and speed. Even in cases that deviate from standard GGM sparsity assumptions (e.g. Laplace, small-world) it performed substantially better. When fine-tuning on the target distribution performance further improves. Most importantly the learned estimator generalizes well to real data finding relevant stable edges. We also observed that the learned estimators generalize to variations not seen at training time (e.g. different n or sparsity), which points to this potentialy learning generic computations. This also shows potential to more easily scale the method to different graph sizes. One could consider transfer learning, where a network for one size of data is used as a starting point to learn a network working on larger dimension data.

Penalized maximum likelihood can provide performance guarantees under restrictive assumptions on the form of the distribution and not considering the regularization path. In the proposed method one could obtain empirical bounds under the prescribed data distribution. Additionally, at execution time the speed of the approach can allow for re-sampling based uncertainty estimates and efficient model selection (e.g. cross-validation) amongst several trained estimators.

We have introduced the concept of learning an estimator for determining the structure of an undirected graphical model. A network architecture and sampling procedure for learning such an estimator for the case of sparse GGMs was proposed. We obtained competitive results on synthetic data with various underlying distributions, as well as on challenging real-world data. Empirical results show that our method works particularly well compared to other approaches for small-world networks, an important class of graphs common in real-world domains. We have shown that neural networks can obtain improved results over various statistical methods on real datasets, despite being trained with samples from parametric distributions. Our

approach enables straightforward specifications of new priors and opens new directions in efficient graphical structure discovery from few examples.

Experimental Setup	Method	Prec@5%	AUC	CE
Gaussian Random Graphs ($n = 35, p = 39$)	Glasso	0.361 ± 0.011	0.624 ± 0.006	0.07
	Glasso (optimal)	0.384 ± 0.011	0.639 ± 0.007	0.07
	BDGraph	0.441 ± 0.011	0.715 ± 0.007	0.28
	DeepGraph-39	0.463 ± 0.009	0.738 ± 0.006	0.07
	DeepGraph-39+Perm	**0.487 ± 0.010**	**0.740 ± 0.007**	0.07
Gaussian Random Graphs ($n = 100, p = 39$)	Glasso	0.539 ± 0.014	0.696 ± 0.006	0.07
	Glasso (optimal)	0.571 ± 0.011	0.704 ± 0.006	0.07
	BDGraph	**0.648 ± 0.012**	**0.776 ± 0.007**	0.16
	DeepGraph-39	0.567 ± 0.009	0.759 ± 0.006	0.07
	DeepGraph-39+Perm	0.581 ± 0.008	0.771 ± 0.006	0.07
Gaussian Random Graphs ($n = 15, p = 39$)	Glasso	0.233 ± 0.010	0.566 ± 0.004	0.07
	Glasso (optimal)	0.263 ± 0.010	0.578 ± 0.004	0.07
	BDGraph	0.261 ± 0.009	0.630 ± 0.007	0.41
	DeepGraph-39	0.326 ± 0.009	0.664 ± 0.008	0.08
	DeepGraph-39+Perm	**0.360 ± 0.010**	**0.672 ± 0.008**	0.08
Laplace Random Graphs ($n = 35, p = 39$)	Glasso	0.312 ± 0.012	0.605 ± 0.006	0.07
	Glasso (optimal)	0.337 ± 0.011	0.622 ± 0.006	0.07
	BDGraph	0.298 ± 0.009	0.687 ± 0.007	0.36
	DeepGraph-39	0.415 ± 0.010	0.711 ± 0.007	0.07
	DeepGraph-39+Perm	**0.445 ± 0.011**	**0.717 ± 0.007**	0.07
Gaussian Small-World Graphs (n=35,p=39)	Glasso	0.387 ± 0.012	0.588 ± 0.004	0.11
	Glasso (optimal)	0.453 ± 0.008	0.640 ± 0.004	0.11
	BDGraph	0.428 ± 0.007	0.691 ± 0.003	0.17
	DeepGraph-39	**0.479 ± 0.007**	0.709 ± 0.003	0.11
	DeepGraph-39+Perm	0.453 ± 0.007	**0.712 ± 0.003**	0.11
	DeepGraph-39+update	**0.560 ± 0.008**	**0.821 ± 0.002**	0.11
	DeepGraph-39+update+Perm	0.555 ± 0.007	0.805 ± 0.003	0.11

Table 5.1 – For each case we generate 100 sparse graphs with 39 nodes and data matrices sampled (with n samples) from distributions with those underlying graphs. DeepGraph outperforms other methods in terms of AP, AUC, and precision at 5% (the approximate true sparsity). In terms of precision and AUC DeepGraph has better performance in all cases except $n > p$.

	50 nodes (s)	500 nodes (s)
`sklearn` GraphLassoCV	4.81	554.7
`BDgraph`	42.13	N/A
DeepGraph	***0.27***	***5.6***

Table 5.3 – Avg. execution time over 10 trials for 50 and 500 node problem on a CPU for Graph Lasso, BDMCMC, and DeepGraph

Experimental Setup	Method	Prec@5%	AUC	CE
Gaussian Random Graphs (n=35,p=39,sparsity=2%)	Glasso	0.464 ± 0.038	0.726 ± 0.021	0.02
	Glasso (optimal)	0.519 ± 0.035	0.754 ± 0.019	0.02
	BDGraph	0.587 ± 0.033	0.811 ± 0.017	0.15
	DeepGraph-39	0.590 ± 0.026	0.810 ± 0.019	0.03
	DeepGraph-39+Perm	0.598 ± 0.026	0.831 ± 0.017	0.03
Gaussian Random Graphs (n=35,p=39,sparsity=15%)	Glasso	0.732 ± 0.046	0.562 ± 0.013	0.32
	Glasso (optimal)	0.847 ± 0.029	0.595 ± 0.011	0.33
	BDGraph	0.861 ± 0.015	0.654 ± 0.013	0.33
	DeepGraph-39	0.678 ± 0.032	0.643 ± 0.012	0.33
	DeepGraph-39+Perm	0.792 ± 0.023	0.660 ± 0.011	0.33

Table 5.4 – For each scenario we generate 100 graphs with 39 nodes, and corresponding data matrix sampled from distributions with those underlying graphs. The number of samples is indicated by n.

	Gene BRCA	Gene COAD	ABIDE Control	ABIDE Autistic
Graph Lasso	$0.25 \pm .003$	0.34 ± 0.004	$0.21 \pm .003$	$\mathbf{0.21 \pm .003}$
Ledoit-Wolfe	0.12 ± 0.002	0.15 ± 0.003	$0.13 \pm .003$	$0.13 \pm .003$
Bdgraph	0.07 ± 0.002	0.08 ± 0.002	N/A	N/A
DeepGraph	$\mathbf{0.48 \pm 0.004}$	$\mathbf{0.57 \pm 0.005}$	$\mathbf{0.23 \pm .004}$	$0.17 \pm .003$
DeepGraph +Permute	0.42 ± 0.003	0.52 ± 0.006	$0.19 \pm .004$	$0.14 \pm .004$

Table 5.5 – Average Spearman correlation results for real data showing stability of solution amongst 50 trials

	mean $\|\hat{\Sigma} - \Sigma\|_2^2$	mean $\|\hat{\Sigma} - \Sigma\|_\infty$
Empirical	0.0267	0.543
Graph Lasso	0.0223	0.680
DeepGraph	0.0232	0.673

Table 5.6 – Covariance prediction of ABIDE data. Averaged over 50 trials of 35 samples from the ABIDE Control data

Experimental Setup	Method	Prec@5%	AUC	CE
Gaussian Random Graphs (n=35,p=39,sparsity=2%)	Glasso	0.464 ± 0.038	0.726 ± 0.021	0.02
	Glasso (optimal)	0.519 ± 0.035	0.754 ± 0.019	0.02
	BDGraph	0.587 ± 0.033	0.811 ± 0.017	0.15
	DeepGraph-39	0.590 ± 0.026	0.810 ± 0.019	0.03
	DeepGraph-39+Perm	0.598 ± 0.026	0.831 ± 0.017	0.03
Gaussian Random Graphs (n=35,p=39,sparsity=15%)	Glasso	0.732 ± 0.046	0.562 ± 0.013	0.32
	Glasso (optimal)	0.847 ± 0.029	0.595 ± 0.011	0.33
	BDGraph	0.861 ± 0.015	0.654 ± 0.013	0.33
	DeepGraph-39	0.678 ± 0.032	0.643 ± 0.012	0.33
	DeepGraph-39+Perm	0.792 ± 0.023	0.660 ± 0.011	0.33

Table 5.7 – For each scenario we generate 100 graphs with 39 nodes, and corresponding data matrix sampled from distributions with those underlying graphs. The number of samples is indicated by n.

Chapter 6

Conclusion

High dimensional inference with few samples is a thriving field that is developing rapidly. In this thesis we have advanced the use of structured sparsity priors for predictive modeling, variable selection, and graph structure discovery. We demonstrated how our techniques can be particularly applied and expanded in the domain of fMRI data analysis.

This chapter summarizes the contributions made in this thesis followed by some directions for future research.

6.1 Summary of contributions

In Chapter 3, we propose a novel penalty for estimating an interpretable and stable predictive linear model under graph constraints. Our primary contributions in this Chapter are

- A novel penalty for estimating an interpretable and stable predictive linear model under graph constraints.
- A proof that tight convex relaxation of correlated sparsity and total variation is intractable.
- A proposal for a looser convex relaxation that corresponds to k-support norm and total variation.
- An empirical validation of k-support norm and total variation penalty on neuroimaging and other data.
- Insights on how the k parameter in the k-support norm can lead to better control of the support size of the solution.

In Chapter 5 and Chapter 4 we make several contributions for the problem of inferring graphical model structure from few observations under sparse constraints. In Chapter 4 we propose a novel statistical test which determines whether two related sets of data have significant differences in their underlying conditional independence structures. Our contributions here include

- To highlight the problem of identifying difference, with significance, in brain connectivity patterns using sparse estimators. This problem has only been directly addressed in several concurrent works.
- To highlight recent literature on debiasing sparse estimators as a path to obtaining difference estimates and their confidence intervals
- To show how to debias a fused lasso estimate leading to a tractable characterization of the difference estimate
- To show how both the joint estimator and independent debiased lasso estimators can be used to infer graph structure differences.

Subsequently in 5 we propose a novel approach for graph structure discovery, we make the following contributions

- We formulate graph structure discovery as a supervised learning problem and show that this is natural.
- We propose a simple sampling approach to sample graphs and corresponding covariance matrices.
- We show empirically that we can learn efficient estimators both in speed and accuracy
- We propose an efficient architecture for this problem.
- We lay foundations for analysis of the proposed architecture used for inference.
- We identify several directions for further scaling the estimators.

We now highlight the future research directions for this work.

6.2 Directions for Future Research

In the last section of this thesis, we draw some directions for future research based on the contributions and limitations of our work. We propose several research paths that can be followed.

Convex Relaxations of Total Variation and Correlated Sparsity

Extension For Non-Grid Graphs and Those Discovered from Data The applications shown in Chapter 3 in this thesis have been largely for grid structured data. However, as noted the model permits a total variation constraint on a generic graph. For example in genomics or social networking data. Additional extensions can consider using the structure discovery methods proposed to first find graph structures, and subsequently make predicitons using them.

Comparisons to Hard-Thresholding Another approach to obtaining solutions under discrete constraint sets is the application of so called hard-thresholding methods [Kyrillidis et al., 2012]. These attempt to directly solve the problem in an approximate way using combinatorial methods instead of obtaining a relaxation and solving it exactly. It would be of interest to compare the hard-thresholding solution to those obtained by our proposed convex relaxation.

Hypothesis Testing for Differences of Gaussian Graphical Models

Scalability of the algorithm The current algorithm and its implementation can be improved by efficiently reusing the computations between the fused lasso solvers and debiasing matrix estimators for each node. Particularly a bottleneck is the computation of the Quadratic Program at each node. This does not take advantage of the structure of the problem and is a generic solution. A naive approach to improve on this can warm-start the debiasing matrix estimator using the previous nodes solution.

Extension to likelihood based methods In our work on the debiasing we consider the pseudo-likelihood approach of solving independent ℓ_1 penalized problems. We can consider recent extensions of the debiasing approach to the graphical lasso and how this can be applied in the joint fused lasso estimator.

Applications to Genetics Datasets Genetics data is another area where we commonly need to identify networks. Applications of our structure discovery methods on genetics datasets can be of interest, especially since some networks in these datasets are better understood than those studied in neuroscience.

Learning to Discover Sparse Graphical Models

Learnability of graph structure discovery In our work we have laid the foundation for an analysis of the learnability of the graph structure discovery problem at least in the setting of Gaussian Graphical Models. However, our analysis only shows that there exists a class of deep models that can represent the solutions. We believe that this work can be extended to show that such a class of models exists which can learn an approximation with ϵ precision and a finite number of parameters.

Additionally scaling the current approach to larger problems as as addressed in Hsieh et al. [2013]. This technique could potentially replace the graph lasso solvers used for the subproblems in Hsieh et al. [2013], which relies on clustering the variables and then solving subproblems with graphical lasso.

Application to Covariance and Inverse Covariance Estimation As shown in the preliminary results in Chapter 5 our method shows promise to be able to learn to predict directly positive definite covariance estimates from noisy ones under a sparse prior. There are many potential applications for this, for example in Gaussian Process Regression the key bottleneck is the inversion of a covariance matrix. If a structure such as sparsity on this inverse can be assumed then. Furthermore even in neuroscience communities there are some who correlation instead of partial correlation for the analysis of neuroimaging data, but denoising the covariance estimate can still be useful. Similarly in certain financial data the correlation from noisy data is the ultimate target. Finally another example of the utility of a fast alternative to graph lasso is robust covariance estimation. This is a procedure that often requires a covariance estimator as a sub-problem, graph lasso can be used in the case of sparsity assumptions.

Dynamic Graphs Extending our model to graphs which follow dynamic behavior, edges and nodes progressively changing over time, is a promising direction. Assuming a generative model can be constructed of the underlying dynamics additional temporal aspects of the inference architecture can be handled by embedding the current network in a recurrent network.

Applications to Action Recognition Action recognition from videos can be seen as a problem that requires relational reasoning between entities as well as how this changes over time. Integrating a structure discovery in the inference pipeline can be a promising direction for improving performance on this task.

Appendix A

The k-support norm in fMRI: A primer on sparse regularization in fMRI

Experiments in Chapter 3 apply sparse regularization to build predictive models of fMRI data. The dataset considered in Chapter 3 has well known tasks and associated regions which explain them in the neuroscience literature [Haxby et al., 2001]. In this Chapter, as a primer on sparse regularization in fMRI, we describe and demonstrate the application of traditional sparse penalties as well as the k-support norm, which forms the foundation of Chapter 3, to a real world fMRI dataset with consequences on medical understanding of addiction. We consider sparse regularization in both the regression and classification settings performing experiments on fMRI scans from cocaine-addicted as well as healthy control subjects. We show that in many cases, use of the k-support norm leads to better predictive performance, solution stability, and interpretability as compared to other standard approaches. We additionally analyze the advantages of using the absolute loss function versus the standard squared loss which leads to significantly better predictive performance for the regularization methods tested in almost all cases. Our results support the use of the k-support norm for fMRI analysis and on the clinical side, the generalizability of the I-RISA model of cocaine addiction. The materials in this appendix are based on work published in [Belilovsky et al., 2015b, Gkirtzou et al., 2013a, Misyrlis et al., 2014b] .

A.1 Overview

The main challenges in statistical fMRI data analysis Bartels and Zeki [2005], Hardoon et al. [2007], Honorio et al. [2012b], Song et al. [2011] are (i) the curse of dimensionality (ii) a small number of samples, due to the high cost of fMRI acquisition, and (iii) high levels of noise, such as system noise and random neural activity.

A general approach for analyzing functional magnetic resonance imaging (fMRI) data is based on pattern recognition and statistical learning. By predicting some cognitive variables related to brain activation maps, this approach aims at decoding brain activity. This approach takes into account the multivariate information between voxels and is a way to assess how precisely some cognitive information is encoded by the activity of neural populations within the whole

brain. However, this approach relies on a prediction function that is plagued by the curse of dimensionality, since there are generally far more features (voxels) than samples. To address this problem, different methods have been proposed, such as, among others, univariate feature selection and regularization techniques [Jenatton et al., 2012].

Sparsity regularizers are key statistical methods for improving predictive performance in the event that the number of observations is substantially smaller than the dimensionality of the data while the underlying signal is known to be sparse. This is the case in fMRI analysis where brain activity is known to occur in only a subset of regions for a given task. In this paper we compare the most frequently applied sparsity regularizer developed in the statistics literature, LASSO [Tibshirani, 1996b] and it's extension the elastic net [Zou and Hastie, 2005], with the k-support norm [Argyriou et al., 2012a], a recently introduced method which tends to retain correlated variables while simultaneously enforcing sparsity.

The k-support norm has an intrinsic parameter, $k \in \{1, \ldots, d\}$, where d is the dimensionality of the data, that controls the degree of sparsity. When used with squared loss, k-support regularization specializes to the LASSO when $k = 1$ and ridge regression when $k = d$. The k-support norm has previously been used in [Argyriou et al., 2012a] for classification. We first evaluate the k-support norm in an fMRI volume classification setting in which we predict a binary task, based on an fMRI volume. We then extend this analysis to a regression problem, predicting a task-variable based on the fMRI volume.

We focus on comparing LASSO and elastic net with the k-support norm in order to establish the latter regularizer's superiority in analyzing fMRI data in the context of a classification task. We then consider a regression setting and use two loss functions, namely the squared error and the absolute error functions. The advantage of the absolute error loss is that it is more robust, in that it penalizes outliers less than the squared loss, while still retaining convexity, which guarantees finding the global optimum. In this setting we compare ℓ_1 regularization with the k-support norm and demonstrate marked improvement. We compare the methods not only in their predictive accuracy but also in the interpretability and stability of their results which is critical in fMRI data analysis.

Although we consider a specific neuroscience application of validating a model of human drug addiction, this approach is more generally applicable and can be used in many other neuroscience studies involving interpretation of fMRI data.

The primary neuroscientific motivation for most of our experiments in this article is the exploration of human drug addiction. Basic studies have led to a theoretical model of human drug addiction, characterized by Impaired Response Inhibition (RI) and Salience Attribution (SA) (hence, I-RISA) [Goldstein and Volkow, 2002]. According to the model, the skew in SA is predictive of impaired RI, together contributing to excessive drug use and relapse, core clinical symptoms of cocaine addiction. We use the fMRI data from a SA task (drug Stroop) in order to predict behavioral data in a RI task (color-word Stroop) collected at a different time, hence providing further evidence to support the I-RISA model.

A.2 Methods

We summarize the regularizers considered in this Chapter in Table A.1 below

Table A.1 – A summary of the regularizers considered in this work.

Regularizer	$J(w)$
LASSO [Tibshirani, 1996b]	$\lambda_1\|w\|_1$
Elastic net [Zou and Hastie, 2005]	$\lambda_1\|w\|_1 + \lambda_2\|w\|_2^2$
k-support [Argyriou et al., 2012a]	$\lambda\|w\|_k^{sp}$ (see Equation (3.2.33))

The k-support norm is closely related to the elastic net, in that it can be bounded to within a constant factor of the elastic net, but it leads to different sparsity patterns. One can see from Equation (3.2.33) that the norm trades off a squared ℓ_2 penalty for the largest components with an ℓ_1 penalty for the smallest components.

A difficulty in using sparse regularizers is that they tend to lead to non-smooth functions which can cause difficulties when using gradient based convex optimization procedures. For this class of functions proximal methods are a very popular way to quickly find optimal solutions with the bottleneck generally being the computation of the proximal mapping. Among many advantages of the k-support norm, it has an easy to compute proximal operator given in Argyriou et al. [2012a].

While initial experiments have shown promising results with the k-support norm for a range of machine learning problems [Argyriou et al., 2012a], to the best of our knowledge the studies discussed here are the first applications to fMRI.

For classification we consider squared loss: $f(w, X, y) = \|y - Xw\|_2^2$. Here we set the labels for the discriminative task to $y \in \{-1, 1\}$ and predict new examples, x_n, as $y_n = sign(x_n w)$. In the regression setting we consider two loss functions: the squared error and the absolute error $f(w, X, y) = \|y - Xw\|_1$. Here y corresponds to the output task-variable. In practice, we approximate the absolute error with a Huber type smoothing around zero to ensure differentiability as described in [Blaschko, 2013]. The advantage of the absolute error loss in regression is that it is more robust, in that it penalizes outliers less than squared loss, while still retaining convexity which guarantees finding the global optimum.

Optimization of objectives containing sparse regularizers are not trivial since they generally contain non-smooth terms which are not compatible with classic optimization techniques such as stochastic gradient descent. Optimization of the LASSO and elastic net has been extensively studied in the literature [Efron et al., 2004, Friedman et al., 2010]. The k-support norm is a relatively new approach and does not have extensive analysis with regards to optimization. However a proximal operator is provided in [Argyriou et al., 2012a]. This is a fundamental building block of many non-smooth optimization techniques a popular one being Fast Iterative Threshold-Shrinkage Algorithm (FISTA) [Argyriou et al., 2012a, Beck and Teboulle, 2009, Huang et al., 2011a,b]. The method is designed for optimizing the sum of a smooth and non-smooth convex function. It requires only the gradient of the smooth function, a proximal operator for the non-smooth function, and an upper bound on the

Lipschitz constant of the gradient of the smooth function. For each of the loss functions considered here, these quantities are known.

A.3 Experimental Results

Results are presented on three fMRI datasets. The first consists of fMRI scans of a subject viewing a movie. The second and third dataset each consist of fMRI scans from control and cocaine-addicted subjects [Goldstein et al., 2009, Honorio et al., 2012b].

Free-Viewing Dataset This dataset consists of a set of fMRI scans from a healthy subject in a free-viewing setting. Data collection was previously described in [Bartels and Zeki, 2004, Bartels et al., 2008], while the pre-processing followed [Blaschko et al., 2009]. The discriminative task in the first data set is the prediction of a "Temporal Contrast" variable computed from the content of a movie presented to the subject [Blaschko et al., 2011]. This dataset was employed for preliminary quantitative evaluation due to its larger sample size.

Cocaine Classification Dataset The overall neuropsychological experiment, referred to as the fMRI drug-Stroop task Goldstein et al. [2007], follows a block design with each subject (either control or cocaine-addicted) performing the same task repeatedly, during a total of six sessions where there are two varying conditions: (i) the monetary reward, as well as (ii) the word that cues the task (which can be a drug word or a neutral word). The sessions consist of an initial screen displaying a monetary reward and then presenting a sequence of forty words in four different colors (yellow, blue, red or green). The subject was instructed to press one of four buttons matching the color of the word they had just read. The subjects were rewarded for correct performance depending on the monetary condition. In our experiments we use sessions with the same monetary reward (50) and the only varying condition is the type of cue words shown (drug words or neutral words) leading to a total of 2 sessions per subject. The discriminative task is to determine whether a subject is cocaine-addicted or a healthy control subject [Goldstein et al., 2009, Honorio et al., 2012b].

Cocaine Regression Dataset The overall neuropsychological experiment follows a block design with each subject (either control or cocaine-addicted) performing the same task repeatedly, during a total of eight session where there are two varying conditions: monetary reward and cue word (drug word or neutral word). Individual sessions follow the same protocol as described in the Cocaine Classification Dataset. In this experiment the monetary reward varies (50, 25, 1 and 0) as well as the type of cue words shown (drug words or neutral words) resulting in a total of 8 sessions per subject.

We use the behavioral responses of the same subjects in a color-word task [Moeller et al., 2012], a classic task of inhibitory control. In this task the subjects indicated the ink-color of color-words printed in either their congruent or incongruent colors [Moeller et al., 2012, Figure 1(a)].

Four colors and words (red, blue, yellow and green) were used in all possible combinations. Both congruent and incongruent stimuli were presented randomly. The subjects performed four consecutive runs of this task. As there were 12 incongruent events in each run of 200 events, each subject's data contained up to 48 incongruent events. For 38 control subjects and 74 cocaine abusers, we use the fMRI data from the drug-word task, to predict color-word behavioral variables such as the difference in subject performance accuracy between congruent and incongruent events.

A.3.1 Classification

In our first experiment we use the free-viewing dataset in a classification task [Blaschko et al., 2011]. The performance of the different sparse regularization techniques, shown in Figure A.1, is evaluated as the mean correlation over 100 trials of random permutation of the data described in [Blaschko et al., 2009]. In each trial, 80% of the data are used to train the method, while the remaining 20% are used to evaluate the performance. More specifically, The top of Figure A.1 shows the mean correlation between LASSO and elastic net against the number of non-zero variables (i.e voxels), while the bottom of Figure A.2 shows the mean correlation for the k-support norm against different k values–which are correlated with the number of non-zero coefficients. LASSO achieves a maximum mean correlation of 0.1198 for 44 non-zero variables, elastic net a maximum mean correlation of 0.1189 for 866 non-zero variables, while k-support norm a maximum of 0.129 for $k = 800$. This is substantially higher than was previously reported in [Blaschko et al., 2011].

Next we evaluate interpretability in the classification setting for the cocaine classification dataset. We use 16 cocaine addicted individuals and 17 control subjects. These were the subjects that complied to the following requirements: motion $<$ 2mm translation, $<$ 2° rotation and at least 50% performance of the subject in an unrelated task [Goldstein et al., 2009]. We visualize the brain regions predicted when applying the LASSO and the k-support norm to this data. For each, we have selected slices through the brain that maximize the sum of the absolute values of the weights predicted by the respective methods. These results are presented in Figure A.2 and discussed in the next section.

The main area of activity shown in Figure A.2 is the rostral anterior cingulate cortex (rostral ACC). It has been shown to be deactivated during the drug Stroop as compared to baseline in cocaine users vs. controls. This is even when performance, task interest, and engagement are matched between the groups [Goldstein et al., 2009] and its activity is normalized by oral methylphenidate [Goldstein et al., 2010]–which similarly to cocaine blocks the dopamine transporters increasing extracellular dopamine–an increase that was associated with lower task-related impulsivity (errors of commission). This region was responsive (showed reduction in drug cue reactivity) to pharmacotherapeutic interventions in cigarette smokers [Culbertson et al., 2011, Franklin et al., 2011], and may be a marker of treatment response in other psychopathology (e.g., depression). The LASSO does not show a meaningful sparsity pattern (Figure A.2).

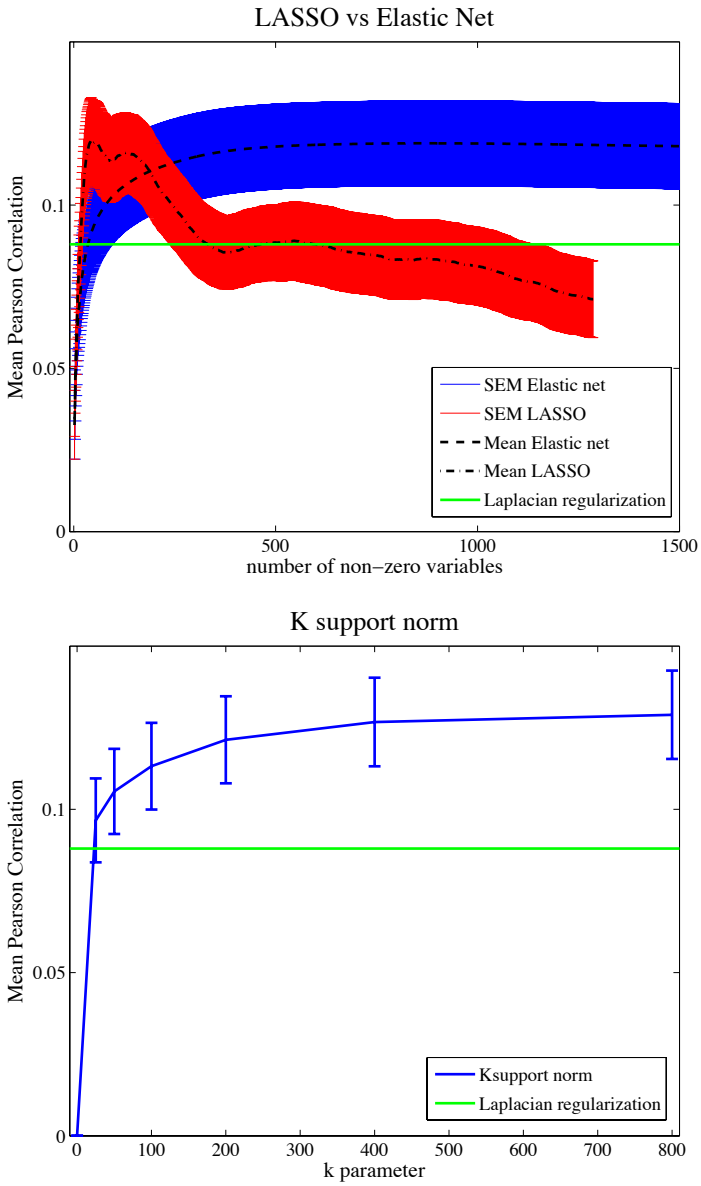


Figure A.1 – Mean Pearson correlations between the label and prediction on the hold-out data over 100 trials for the dataset described in Blaschko et al. [2009] (higher values indicate better performance). Error bars show the standard deviation. The LASSO achieves its best performance with a sparsity level substantially lower than the elastic net, as it suppresses correlated voxels (top Figure). The k-support norm performs better than the LASSO, elastic net, or Laplacian regularization reported in Blaschko et al. [2011], and is a promising candidate for sparsity in fMRI analysis (bottom Figure). (Figure best viewed in color.)

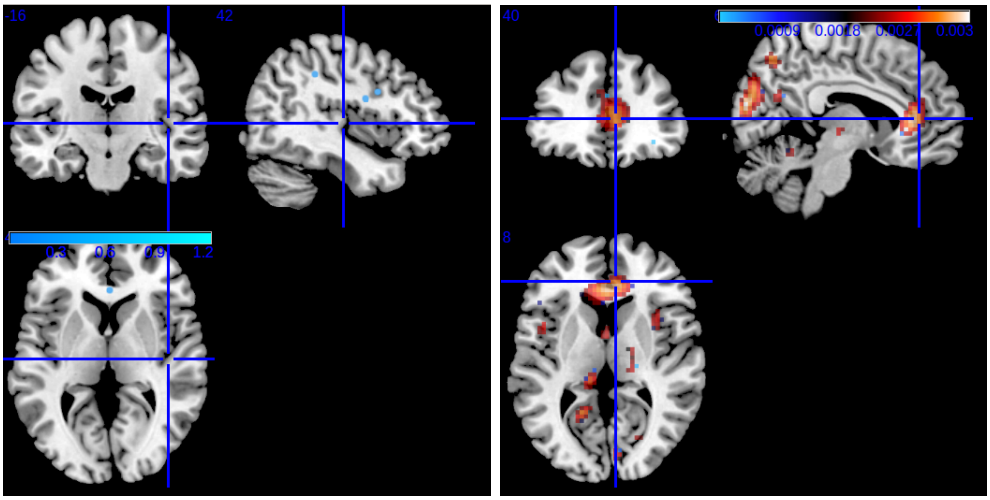


Figure A.2 – A visualization of the areas of the brain selected by the LASSO and by the k-support norm applied to the data described in [Goldstein et al., 2009]. The LASSO leads to overly sparse solutions that do not lend themselves to easy interpretation (Left), while the k-support norm does not suppress correlated voxels, leading to interpretable and robust solutions (Right). A medical interpretation of the result presented in the left figure is given in Section A.3.1. (Figure best viewed in color.)

To further understand the differences in brain activity of addicted and not addicted patients we next extend our analysis to the cocaine regression dataset.

A.3.2 Regression

In this section we present our regression experiments on the cocaine dataset. Our experiments aim at providing empirical evidence for the support of the I-RISA model.

We use the Cocaine Regression Dataset described in Sec A.3 in two experiments both predicting color-word behavioral variables.

In **experiment 1** we use the fMRI contrast drug > neutral words, averaged over monetary reward condition, to predict the conflict effect in the subjects' reaction time on the color-word task, defined as the difference in time between correctly performing the task for congruent and incongruent events. We use the Insula, Hippocampus Complex, Amygdala and ACC, part of the brain's limbic (emotion) circuit, as regions of interest (ROIs) for this experiment. These regions are chosen on the basis of previous studies on independent datasets that showed limbic system modulation by drug-related cues, e.g. drug words [Chase et al., 2011].

In **experiment 2** we use the fMRI contrast 50 > 0, averaged over word type condition (drug or neutral), in order to predict the subjects' responses on the color-word task, defined as the difference in percent accuracy between performing the task for congruent and incongruent events. We use the Basal Ganglia and Thalamus, part of the brain's reward circuit, as ROIs for this experiment. We chose these ROIs on the basis of previous studies on independent datasets that showed reward system modulation by primary and secondary reinforcers,

Mean Correlation, D>N, Conflict effect on Reaction Time

Control Subjects

Norm / Loss	Squared	Absolute	p
LASSO	0.16 (0.02)	0.27 (0.02)	<0.01
k-support	0.22 (0.02)	0.24 (0.02)	<0.05
p	<0.001	0.21	

Cocaine-Addicted Subjects

Norm / Loss	Squared	Absolute	p
LASSO	0.27 (0.01)	0.37 (0.01)	<0.001
k-support	0.33 (0.01)	0.36 (0.02)	<0.001
p	<0.001	0.96	

Table A.2 – Mean (SE) correlation over 500 random permutations of the samples between the predicted and the actual conflict effect on the reaction times for drug > neutral using the limbic ROI, for all combinations of regularizers and loss functions. The p-values were computed with a Wilcoxon signed rank test between the 500 correlation values for the two combinations of regularizer and loss function in the preceding rows or columns. Based on the p-values, there is a statistically significant difference between absolute loss predictions and squared loss predictions and between LASSO and k-support norm with the squared loss function in both cocaine and control subjects.

including money [Liu et al., 2011].

For each experiment we perform 500 trial with an 85% / 15% random split between training and test sets. For each trial we perform model selection on the training set. That is, for each combination of parameters ($\lambda \in \{10^i : i = -2, \ldots, 8\}$ for LASSO, $\lambda \in \{10^i : i = -2, \ldots, 8\}$, $k \in \{1, 2, 3, 6, 12, 100, 200, 300, 600\}$ for k-support norm), we do a leave-one-subject-out cross validation on the samples that constitute the training set. We measure the correlation between the predicted and the true response variables on the training set. The parameter setting that leads to the highest correlation is used on the whole training set in order to learn a set of weights for each method, which are then applied on the test set. Finally, we measure the correlation between the predicted and the true response variables on the test set. We report the mean correlation on the holdout test samples and its standard error across the 500 random permutations. We note that the same sample randomization is used for both LASSO and k-support norm.

We compare the performance of the two methods in Table A.2 for the first experiment and Table A.3 for the second experiment.

With the squared loss function, the k-support norm outperforms LASSO for almost all cases, while when combined with the absolute loss function, the regularizers do not significantly differ in their predictive performance. The absolute loss function, for both regularizers, leads to correlations that are significantly higher than those with the squared loss function in almost all cases.

We report the fraction of non-zero weights that were selected by each method for over 50% of the 500 trials in Tables A.4 and A.5 for the first and the second experiment respectively.

Mean Correlation, 50>0, Conflict effect on Accuracy

Control Subjects

Norm / Loss	Squared	Absolute	p
LASSO	0.25 (0.02)	0.09 (0.02)	<0.001
k-support	0.26 (0.02)	0.09 (0.02)	<0.001
p	0.42	0.78	

Cocaine-Addicted Subjects

Norm / Loss	Squared	Absolute	p
LASSO	0.22 (0.02)	0.42 (0.02)	<0.001
k-support	0.27 (0.01)	0.41 (0.02)	<0.001
p	<0.001	0.78	

Table A.3 – Mean (SE) correlation over 500 random permutations of the samples between the predicted and the actual response variables for 50> 0 using the Basal Ganglia, Thalamus ROI, for all combinations of regularizers and loss functions. The p-values were computed with a Wilcoxon signed rank test between the 500 correlation values for the two combinations of regularizer and loss function in the preceding rows or columns. Based on the p-values there is a statistically significant difference between absolute loss predictions and squared loss predictions and between k-support and LASSO with the squared loss in cocaine-addicted subjects only.

Voxel Selection Stability, D>N, Conflict effect on Reaction Time

	Control		Cocaine-Addicted	
Norm / Loss	Squared	Absolute	Squared	Absolute
LASSO	0.0004	0.0007	0	0.0023
k-support	0.0029	0.0018	0.0058	0.0734

Table A.4 – Voxel Selection stability over 500 random permutations of the samples for drug > neutral using the limbic ROI, for all combinations of regularizers and loss functions. The fraction of voxels which are selected for more than 50% of the 500 trials are presented. The higher values reported for k-support norm indicate that it makes more stable voxel selection than LASSO over different training sets.

We average the weights assigned to the voxels over the 500 permutations and then compute the cumulative distribution function (CDF) for those weights. We threshold the CDF at 0.9 and visualize the weights of the voxels up to that threshold in Fig. A.3. The overly sparse solutions of the LASSO lead to models that cannot be interpreted as easily as the solutions of the k-support norm method.

In the presence of correlated features, the degree of sparsity of the solution can be tuned with the k-support norm in order to include several highly correlated features. In contrast, LASSO tends to pick one representative feature with no guarantee of consistency in feature selection across different splits of the data samples into training and test sets. In all cases the fraction of non-zero weights selected by the k-support norm is higher than that of LASSO, indicating that the k-support norm method leads to more stable solutions as compared to those obtained with LASSO.

Voxel Selection Stability, 50>0, Conflict effect on Accuracy

	Control		Cocaine-Addicted	
Norm / Loss	Squared	Absolute	Squared	Absolute
LASSO	0.0004	0.0050	0.0008	0.0013
k-support	0.0037	0.0083	0.0223	0.0122

Table A.5 – Voxel Selection stability over 500 random permutations of the samples for 50> 0 using the Basal Ganglia, Thalamus ROI, for all combinations of regularizers and loss functions. The fraction of voxels which are selected for more than 50% of the 500 trials are presented. The higher values reported for k-support norm indicate that it makes more stable voxel selection than LASSO over different training sets.

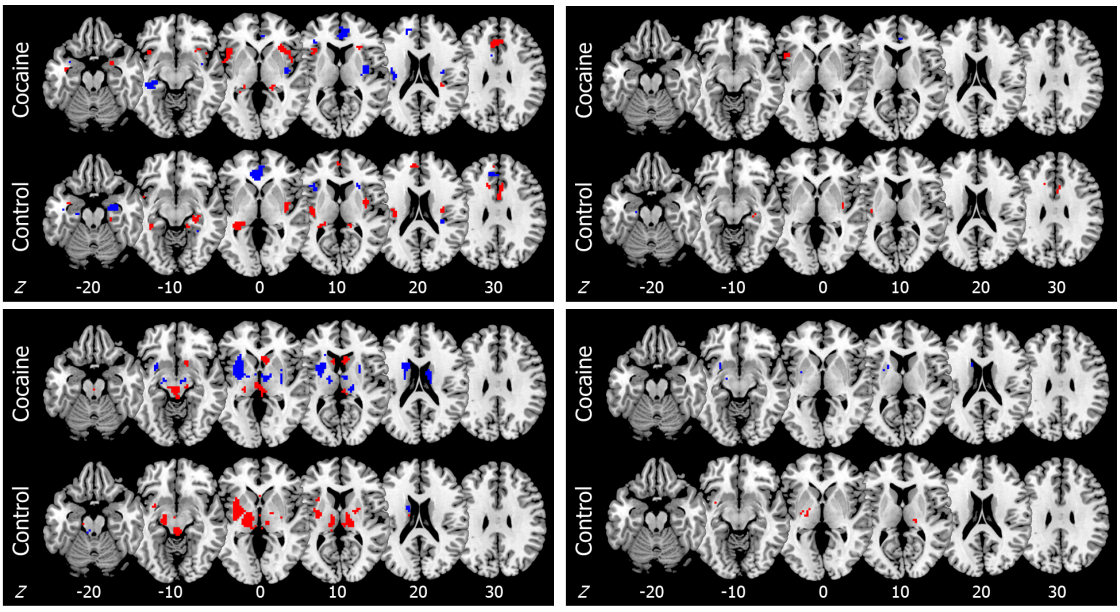


Figure A.3 – Visualization of the most predictive voxels in Exp. 1 (upper left & upper right) and Exp. 2 (bottom left & bottom right) over the 500 permutations. Red areas indicate regions of substantially increased activity and blue regions of subtantially decreased activity. The degree of sparsity of the solution can be tuned with the k-support norm, thus leading to models (upper right, bottom right) that are easier to interpret than those of LASSO (upper left, bottom left). (Best viewed in color)

A.4 Discussion

In our classification experiments we have shown that the k-support norm can give better predictive performance than the LASSO and elastic net, while having favorable mathematical and computational properties. Furthermore, the brain regions implicated in addiction by the k-support norm coincide with previous results on addiction indicating that the k-support norm is additionally useful for generating sparse, but correlated, regions suitable for interpretation in a medical-research setting

In our regression experiments, in almost all cases, the k-support norm outperforms LASSO in predicting the behavioral measures given fMRI data when combined with squared loss, while when combined with the absolute loss, the predictive accuracy of the two regularizers does not differ significantly. The absolute loss led to higher predictions than squared loss for both regularizers for almost all cases. The LASSO leads to sparse solutions, since it tends to pick one feature per group of correlated features. On the other hand, the k-support norm allows calibrating the cardinality of the solutions and thus can select more interpretable groupings of correlated features and also leads to more stable results across different training sets. Thus, our results support the further exploration of the k-support norm for fMRI analysis. Furthermore, we demonstrate that we can predict real valued behavioral variables measured in an inhibitory control task given fMRI data from a *different* task, designed to capture emotionally-salient reward.

On the medical side, we also provide further evidence to support the I-RISA model of drug addiction, whereby the skew in SA in cocaine abusers, as indexed by fMRI response to drug words and monetary rewards, two motivationally salient stimuli, is predictive of RI, as indexed by response slowing and accuracy on a task requiring inhibitory control (the color-word Stroop). Specifically, we show that in cocaine users, response to drug words in voxels located in limbic brain regions, such as the anterior insula and ACC implicated in emotion processing and emotion regulation, was predictive of slower responses on the RI task (Exp. 1), while response to money in voxels located in reward-related brain regions, such as the putamen implicated in habits, was predictive of lower accuracy on the RI task (Exp. 2).

A.4.1 Conclusions

In this Chapter, we have investigated the applicability of sparsity regularizers in fMRI analyses. We have shown that the k-support norm can give better predictive performance than the LASSO and elastic net, while having favorable mathematical and computational properties. Furthermore, the brain regions implicated in addiction by the k-support norm coincide with previous results on addiction, indicating that the k-support norm is additionally useful for generating sparse, but correlated, regions suitable for interpretation in a medical-research setting.

Appendix B

Joint Embeddings of Scene Graphs and Images

Much of the thesis has focused on leveraging structured sparsity assumptions to determine graphical model structures and on using known graph structured assumptions to improve prediction and inference. Here we consider a related problem where there is already a known sparse graphical data and an associated data from a different modality. The goal is to learn representations that associate well the graphs to data in the related modality. We consider a particularly challenging task where the label space for nodes and edges is very large with respect to the number of samples. Our problem is motivated from the recent developments in associating textual descriptions with natural scenes. Multimodal representations of text and images have become popular in recent years. Text however has inherent ambiguities when describing visual scenes, leading to the recent development of datasets with detailed graphical descriptions in the form of scene graphs. We consider the task of joint representation of semantically precise scene graphs and images. We propose models for representing scene graphs and aligning them with images. We investigate methods based on bag-of-words, subpath representations, as well as neural networks. Our investigation proposes and contrasts several models which can address this task and highlights some unique challenges in both designing models and evaluation. This section corresponds to work that has been presented in Belilovsky et al. [2017a].

B.1 Introduction

With recent advances in perceptual tasks, attention in computer vision has been brought to problems requiring greater levels of semantic interpretation of images. Joint modeling of text and vision has led to great improvements in performance on caption generation, visual question answering, and retrieval. Text, however, often has many inherent ambiguities and, for some tasks, connecting a more precise description of image content to visual representation can be of great interest.

Compact representation of semantically precise descriptions of visual information are of great interest and can be potentially used for a variety of downstream tasks from image retrieval, generation, and visual question answering. Until recently study of multimodal embeddings of images has focused on connecting sentence level descriptions and images. One recently

popular method of describing the content of images is based on the scene graph, a detailed description of the underlying image content. Recently, datasets with detailed scene graph annotations have become available [Antol et al., 2015, Krishna et al., 2017]. In this work we make a first step to analyse the joint embedding of images and scene graph into a shared latent space.

We investigate several strategies for performing the embedding scene graphs. We propose as a baseline a bag of words embedding that only considers the scene objects and then consider a subpath representation as well as a graph neural network which can take advantage of the structural information within the scene, we find that, for the data we consider, subpath representations provide the best results on a retrieval task.

Related to our work Fisher et al. [2011] proposes efficient kernels for retrieving images based on a scene graph. Johnson et al. [2015] propose to use scene graphs for the task of image retrieval. Their model uses a probabilistic inference framework in comparing graphs. Lu et al. [2016] consider the closely related problem of visual relationship detection. Most recently, Teney et al. [2016] proposed to graph structured models for visual question answering. Their work considers a model similar to the Graph Neural networks [Li et al., 2016] which we also consider. The problem addressed is different as it involves graph matching instead of embedding a whole graph, furthermore visual features are used as annotations, while we consider categorical annotations.

B.2 Joint Representations of Scene Graphs and Images

A scene graph [Johnson et al., 2015, Krishna et al., 2017] is defined by its objects, their attributes, and relationships. Consider a set of object classes, $\mathcal{C}$, attributes $\mathcal{A}$, and relationships $\mathcal{R}$. Let a scene graph $G = (O, E)$ be a directed graph. For $o \in O$, an object in an image I, $o = (c, A)$, where $c \in \mathcal{C}$ is the class of the object and $A \subseteq \mathcal{A}$. For $t \in E$, a labeled directed edge $t = (o, s, r)$ where $r \in \mathcal{R}$ and $o, s \in O$. Define $Nbr(o)$ as the set of all (s, r) such that $(o, s, r) \in E$.

A joint representation of a scene graph, $\boldsymbol{g}$, and image,$\boldsymbol{x}$, should provide embedding functions, $\boldsymbol{f}_i(\boldsymbol{x})$ and $\boldsymbol{f}_g(\boldsymbol{g})$ which produce continuous vector representations in $\mathbb{R}^D$ for input images and scene graphs, along with a similarity metric, s (commonly the inner product in the embedding space). These vector representations should respect semantic similarities, such that for images $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ it will be the case that $s(\boldsymbol{x}_i, \boldsymbol{g}) > s(\boldsymbol{x}_j, \boldsymbol{g})$ if the graph is semantically closer to the image $\boldsymbol{x}_i$ than $\boldsymbol{x}_j$. Here we consider several possible choices for embedding the graph.

Bag of Words A bag of words model takes a frequency count of nodes in the scene graphs, and does not consider the relationship information or the association of attributes. This is a natural baseline and the analogue in the text domain has shown strong performance in many joint vision and language tasks [Frome et al., 2013]. Here we take the vocabulary V to be of size $|\mathcal{C}|$. Let e_o represent the one hot encoding for object class o. The embedding is defined

by the matrix $\boldsymbol{W_g} \in \mathbb{R}^{D \times V}$ and given simply as $\boldsymbol{f}_g(\boldsymbol{g}_r) = \boldsymbol{W_g} \sum_{o \in O} e_o$. This is then rescaled to unit norm.

SubPath Representations We consider the use of a graph path representation [Swamidass et al., 2005]. Here we augment the count of node frequency by additionally considering subpaths up to length l. This allows structural information to be used in the final embedding. The final embedding is constructed as in the case of the bag of words $\boldsymbol{W_g}$ with V now the size of all unique subpaths in the dataset. Similar to the literature on text representations paths can be seen as an analog of n-grams which can still provide strong baselines in text classification [Joulin et al., 2016]. In the base case of order 1 paths that only consider the nodes it reduces to the bag of words model.

Graph Neural Network Another strategy is similar in spirit to recent work proposed in Li et al. [2016], Teney et al. [2016] which maintains a state vector for each node and uses a recurrent procedure that updates each node state based on its neighbor, progressively propagating information. Below the update sequence is defined per object.

$$h^0_{g,o} = \boldsymbol{W}_g e_o \qquad\qquad h^{i+1}_{g,o} = \tanh\left(\boldsymbol{W}_o h^i_{g,o} + \boldsymbol{W}_p \left(\sum_{(s,r) \in Nbr(o)} \boldsymbol{V}_r h^i_{g,s}\right)\right)$$

For each node we obtain its representation by performing an embedding and then updating the representation by adding a term for the neighbors based on the maximum path between any 2 nodes. In practice we will take a maximum of $i = 3$ steps. $\boldsymbol{V}_r$ is a separate term associated with each relationship, we consider only the most common relationships and fold others into one category. The final graph representation can be obtained by summing the node states and normalizing. In this work we focus on the object and their interactions but attributes can additionally be incorporated as edges in the graph.

Image embedding and loss function The image embedding we utilize are the VGG-19 fc7 10-crop features,x, For the image embedding we use VGG-19 fc7 10-crop features, as in Kiros et al. [2014], denoted $\boldsymbol{x}$, projected as $\boldsymbol{f}_{W_m}(\boldsymbol{x}) = W_m \boldsymbol{x}$ and normalized. If we let W_G describe all parameters of the encoding model given a set of images,$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ and corresponding scene graphs $\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_N$, and the similarity measure $s(\boldsymbol{x}, \boldsymbol{g}) = \boldsymbol{f}_{W_m}(\boldsymbol{x}) \cdot \boldsymbol{f}_{W_G}(\boldsymbol{g})$ the following contrastive loss function is used to align the image and scene graph

$$\min_{W_m, W_G} \sum_{\boldsymbol{x}_k, \boldsymbol{g}_k, \boldsymbol{g}_c} \max\{0, \alpha - s(\boldsymbol{x}_k, \boldsymbol{g}_k) + s(\boldsymbol{x}_k, \boldsymbol{g}_c)\} + \sum_{\boldsymbol{g}_k, \boldsymbol{x}_k, \boldsymbol{x}_c} \max\{0, \alpha - s(\boldsymbol{x}_k, \boldsymbol{g}_k) + s(\boldsymbol{x}_c, \boldsymbol{g}_k)\}$$

Where $\alpha > 0$ is a scalar defining the size of the margin of the two hinge losses.

B.3 Evaluating Joint Embeddings

Evaluating joint embeddings can be challenging. A common approach in caption/image embedding is the use of a retrieval task [Vendrov et al., 2016] that involves ranking a large dataset of images by relevance for a query. However, although posed as a retrieval task the score is often only known for only one ground truth image. This problem is exacerbated for the case of highly detailed descriptions such as a paragraphs or scene graphs. Given that scene graphs can become very large (some having over 50 labeled objects) it becomes increasingly easy to match images simply based on object counts, while at the same time there can be many images in the result set which are indeed very similar to the ground truth. We thus take a different approach to the evaluation of scene graph to image retrieval than shown in Johnson et al. [2015]. Since images in the test set have associated scene graphs our evaluation leverages existing graph similarity metrics to allow comparisons to all the images in the retrieval set. For the case of scene graphs we can construct a metric based on the path kernel [Borgwardt, 2007]. Using this similarity metric, we can compute the Normalized Discounted Cumulative Gain (NDCG) [Wang et al., 2013] to evaluate the retrieval performance considering the returned ranking of all images in the search space to their underlying graph similarity score.

We use a dataset of images and scene graphs from Johnson et al. [2015], Lu et al. [2016]. The data consists of 5000 images with carefully curated scene graphs. We first perform a first level analysis, described in the Appendix, to determine that our visual features can indeed discriminate structural information in the scene graph.

We first perform a basic evaluation of whether structural information related to the scene graph is extractable from the image features (VGG fc7) we have selected to use. We consider the top occurring edges and construct binary classification problems for each of the top edges attempting to predict it's presence or absence from the visual features. We use a random forest classifier and consider the AUC. We find that 3 out of 10 have chance performance with the remaining classifiers obtaining an average AUC of $55 \pm 0.5\%$. This first order analysis indicates that there is discernible structural information in the visual features used, although expectedly the rate is rather low. We note that a given image may have a large number of interactions which can together give noticeable improvement on tasks such as retrieval.

We now evaluate the proposed joint representation approaches on the retrieval task. We use the 4000 train and 1000 test images from the splits specified in Johnson et al. [2015]. We use the objective described previously with batches of size 500 and optimize using the ADAM optimizer [Kingma and Ba, 2015a], ℓ_2 regularization, and $\alpha = 0.4$.

For evaluation we compute the mean Normalized Discounted Cumulative Gain (NDCG) for each test image using a path graph kernel of order 3 as our relevance metric. For each test graph we embed the graph in the joint embedding space and look for the nearest matching image out of the 999 possible remaining.

We consider results using 500 and 100 dimensional latent spaces. We also consider path

Methods	NDCG 5	NDCG 10	NDCG 20	medRank Gr2im
PathRep 3(500 latent)	**0.320**	**0.354**	**0.396**	9
PathRep 2(500 latent)	0.300	0.338	0.381	9
BOW (500 latent)	0.281	0.317	0.362	9
PathRep 3 (100 latent)	0.305	0.338	0.378	10
PathRep 2 (100 latent)	0.290	0.327	0.372	10
BOW (100 latent)	0.276	0.310	0.355	10
Graph NN (100 latent)	0.249	0.280	0.321	15
SG obj Johnson et al. [2015]	-	-	-	28
SG obj-attr-rel Johnson et al. [2015]	-	-	-	14

Table B.1 – Results for graph to image retrieval. NDCG is computed at the top 5,10, and 20 images. Medium rank is computed for the ground truth image retrieval

representations of order 2 and 3. We observe that the use of the graph neural network underperforms the more simple linear embedding of the bag of words features. This is analogous to observations in several text based tasks [Joulin et al., 2016] and highlights the difficulty in extracting semantic information in this challenging scenario. However we can see that the path representations indeed improve substantially the performance in terms of NDCG, highlighting that graph structural information is indeed useful and can be leveraged in this task.

For reference, we also report results for the median rank of the ground truth image on the same task from Johnson et al. [2015], which uses a different model based on object detections. We note in Johnson et al. [2015] the label space is limited to the top occurring objects, which in our embedding framework is not necessary and we utilize the full set of objects provided in the dataset. Notably the results using bag of words and path representations can improve on those of Johnson et al. [2015]. Additionally, it is possible to do image to graph retrieval with our model.

Bibliography

A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang. Learning polynomials with neural networks. In *Proceedings of the International Conference on Machine Learning*, 2014.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1457–1465. 2012a.

A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In *Advances Neural Information Processing Systems*, pages 1466–1474, 2012b.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012a.

F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012b.

A. Backus, O. Jensen, E. Meeuwissen, M. van Gerven, and S. Dumoulin. Investigating the temporal dynamics of long term memory representation retrieval using multivariate pattern analyses on magnetoencephalography data. Technical report, 2011.

A. K. Balan, V. Rathod, K. Murphy, and M. Welling. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, 2015.

L. Baldassarre, J. Morales, A. Argyriou, and M. Pontil. A general framework for structured sparsity via proximal optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 82–90, 2012a.

L. Baldassarre, J. Mourao-Miranda, and M. Pontil. Structured sparsity models for brain decoding from fMRI data. In *Proceedings of The International Workshop on Pattern Recognition in Neuroimaging*, 2012b.

S. Balmand and A. S. Dalalyan. On estimation of the diagonal elements of a sparse precision matrix. *Electronic Journal of Statistics*, 10(1):1551–1579, 2016.

A. Bartels and S. Zeki. The chronoarchitecture of the human brain–natural viewing conditions reveal a time-based anatomy of the brain. *NeuroImage*, 22(1):419–433, 2004. ISSN 1053-8119.

A. Bartels and S. Zeki. Brain dynamics during natural viewing conditions–a new guide for mapping connectivity in vivo. *NeuroImage*, 24(2):339–349, 2005.

A. Bartels, S. Zeki, and N. Logothetis. Natural vision reveals regional specialization to local motion and to contrast-invariant, global flow in the human brain. *Cerebral Cortex*, 18(3): 705–717, 2008.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in mathematics. Springer, 2011.

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18 (11):2419–2434, 2009.

E. Belilovsky, A. Argyriou, G. Varoquaux, and M. Blaschko. Convex relaxations of penalties for sparse correlated variables with bounded total variation. *Machine Learning*, 100(2-3): 533–553, 2015a.

E. Belilovsky, K. Gkirtzou, M. Misyrlis, A. B. Konova, J. Honorio, N. Alia-Klein, R. Z. Goldstein, D. Samaras, and M. B. Blaschko. Predictive sparse modeling of fmri data for improved classification, regression, and visualization using the k-support norm. *Computerized Medical Imaging and Graphics*, 46:40–46, 2015b.

E. Belilovsky, W. Bounliphone, M. B. Blaschko, I. Antonoglou, and A. Gretton. A test of relative similarity for model selection in generative models. *International Conference on Representation Learning*, 2016a.

E. Belilovsky, G. Varoquaux, and M. B. Blaschko. Hypothesis testing for differences in Gaussian graphical models: Applications to brain connectivity. In *Advances Neural Information Processing Systems*, 2016b.

E. Belilovsky, G. Varoquaux, and M. B. Blaschko. Testing for differences in gaussian graphical models: Applications to brain connectivity. *Advances in Neural Information Processing Systems*, 2016c.

E. Belilovsky, M. Blaschko, J. R. Kiros, R. Urtasun, and R. Zemel. Joint embeddings of scene graphs and images. *International Conference on Representation Learning (ICLR) Workshop Track*, 2017a.

E. Belilovsky, K. Kastner, G. Varoquaux, and M. Blaschko. Learning to discover graphical model structures. *International Conference on Machine Learning*, 2017b.

I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. Neural combinatorial optimization with reinforcement learning. *International Conference on Representation Learning Workshop Track*, 2017.

Y. Bengio, I. J. Goodfellow, and A. Courville. Deep learning. *Nature*, 521:436–444, 2015.

R. Bhatia. *Matrix Analysis*. Graduate Texts in Mathematics. Springer, 1997.

M. Blaschko, J. Shelton, A. Bartels, C. Lampert, and A. Gretton. Semi-supervised kernel canonical correlation analysis with application to human fMRI. *Pattern Recognition Letters*, 32(11):1572–1583, 2011. ISSN 0167-8655.

M. B. Blaschko. A note on k-support norm regularized risk minimization. Technical report, 2013. arXiv:1303.6390.

M. B. Blaschko, J. A. Shelton, and A. Bartels. Augmenting feature-driven fMRI analyses: Semi-supervised learning and resting state activity. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 126–134. 2009.

K. M. Borgwardt. *Graph kernels*. PhD thesis, Ludwig Maximilians University Munich, Germany, 2007.

D. Borsook, E. A. Moulton, K. F. Schmidt, and L. R. Becerra. Neuroimaging revolutionizes therapeutic approaches to chronic pain. *Molecular Pain*, 3(1):25, 2007.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

P. Bühlmann, M. Kalisch, and L. Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278, 2014.

K. Button et al. Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14:365, 2013.

T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

F. X. Castellanos et al. Clinical applications of the functional connectome. *Neuroimage*, 80: 527, 2013.

H. W. Chase, S. B. Eickhoff, A. R. Laird, and L. Hogarth. The neural basis of drug stimulus processing and craving: An activation likelihood estimation meta-analysis. *Biological Psychiatry*, 70(8):785–793, 2011.

S. Chatterjee, S. Chen, and A. Banerjee. Generalized Dantzig selector: Application to the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1934–1942, 2014.

S. Chen, Y. Xing, and J. Kang. Latent and abnormal functional connectivity circuits in autism spectrum disorder. *Frontiers in neuroscience*, 11, 2017.

X. Chen, Q. Lin, S. Kim, J. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse learning. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: a tensor analysis. In *Conference On Learning Theory*, 2016.

C. S. Culbertson, J. Bramen, M. S. Cohen, E. D. London, R. E. Olmstead, J. J. Gan, M. R. Costello, S. Shulenberger, M. A. Mandelkern, and A. L. Brody. Effect of bupropion treatment on brain activation induced by cigarette-related cues in smokers. *Archives of General Psychiatry*, 68(5):505–515, 2011.

B. Da Mota, V. Fritsch, G. Varoquaux, T. Banaschewski, G. J. Barker, A. L. Bokde, U. Bromberg, P. Conrod, et al. Randomized parcellation based inference. *NeuroImage*, 89:203–215, 2014.

P. Danaher, P. Wang, and D. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society (B)*, 76(2):373–397, 2014.

A. P. Dawid. Conditional independence in statistical theory. *Royal Statistical Society*, 1979.

A. P. Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972.

E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, 2014.

R. Dezeure, P. Bühlmann, L. Meier, and N. Meinshausen. High-dimensional inference: Confidence intervals, p-values and R-software hdi. *Statistical Science*, 30(4):533–558, 11 2015. doi: 10.1214/15-STS527.

A. Di Martino et al. The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19:659, 2014.

E. Dohmatob, A. Gramfort, B. Thirion, and G. Varoquaux. Benchmarking solvers for TV-l1 least-squares and logistic regression in brain imaging. In *Proceedings of The International Workshop on Pattern Recognition in Neuroimaging*, 2014.

M. Dubois, F. Hadj-Selem, T. Lofstedt, M. Perrot, C. Fischer, V. Frouin, and E. Duchesnay. Predictive support recovery with TV-elastic net penalty and logistic regression: An application to structural MRI. In *Proceedings of The International Workshop on Pattern Recognition in Neuroimaging*, 2014.

D. K. Duvenaud et al. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, 2015.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM Transactions on Graphics (TOG)*, volume 30, page 34. ACM, 2011.

R. A. Fisher. The distribution of the partial correlation coefficient. *Metron*, 3:329–332, 1924.

T. R. Franklin, Z. Wang, Y. Li, J. J. Suh, M. Goldman, F. W. Lohoff, J. Cruz, R. Hazan, W. Jens, J. A. Detre, W. Berrettini, C. P. O'Brien, and A. R. Childress. Dopamine transporter genotype modulation of neural responses to smoking cues: Confirmation in a new cohort. *Addiction Biology*, 16(2):308–322, 2011. ISSN 1369-1600.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

A. Ganguly and W. Polonik. Local neighborhood fusion in locally constant Gaussian graphical models. *arXiv:1410.8766*, 2014.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *International Conference of Machine Learning*, 2017.

K. Gkirtzou, J. Honorio, D. Samaras, R. Goldstein, and M. B. Blaschko. fMRI analysis of cocaine addiction using k-support sparsity. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*, pages 1078–1081, 2013a.

K. Gkirtzou, J. Honorio, D. Samaras, R. Z. Goldstein, and M. B. Blaschko. fMRI analysis of cocaine addiction using k-support sparsity. In *IEEE International Symposium on Biomedical Imaging*, pages 1078–1081, 2013b.

R. Goldstein and N. Volkow. Drug addiction and its underlying neurobiological basis: Neuroimaging evidence for the involvement of the frontal cortex. *The American Journal of Psychiatry*, 159(10):1642, 2002.

R. Goldstein, D. Tomasi, S. Rajaram, L. Cottone, L. Zhang, T. Maloney, F. Telang, N. Alia-Klein, and N. Volkow. Role of the anterior cingulate and medial orbitofrontal cortex in processing drug cues in cocaine addiction. *Neuroscience*, 144(4):1153–1159, 2007.

R. Goldstein, N. Alia-Klein, D. Tomasi, J. Carrillo, T. Maloney, P. Woicik, R. Wang, F. Telang, and N. Volkow. Anterior cingulate cortex hypoactivations to an emotionally salient task in cocaine addiction. *PNAS*, 106(23):9453, 2009.

R. Z. Goldstein, P. A. Woicik, T. Maloney, D. Tomasi, N. Alia-Klein, J. Shan, J. Honorio, D. Samaras, R. Wang, F. Telang, G.-J. Wang, and N. D. Volkow. Oral methylphenidate normalizes cingulate activity in cocaine addiction during a salient cognitive task. *Proceedings of the National Academy of Sciences*, 107(38):16667–16672, 2010.

E. Gómez. A multivariate generalization of the power exponential family of distributions. *Communications in Statistics-Theory, Methods*, 27(3), 1998.

A. R. Goncalves, P. Das, S. Chatterjee, V. Sivakumar, F. J. Von Zuben, and A. Banerjee. Multi-task sparse structure learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 451–460, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2662091.

A. Gramfort, B. Thirion, and G. Varoquaux. Identifying predictive regions from fMRI with TV-L1 prior. In *Proceedings of The International Workshop on Pattern Recognition in Neuroimaging*, pages 17–20, 2013.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the International Conference on Machine Learning*, 2010.

M. G. G'Sell, J. Taylor, and R. Tibshirani. Adaptive testing for the graphical lasso. *arXiv:1307.4765*, 2013.

H. Hara and A. Takemura. A localization approach to improve iterative proportional scaling in Gaussian graphical models. *Commun Stat Theory Methods*, 39(8-9):1643–1654, 2010.

D. R. Hardoon, J. Mourão Miranda, M. Brammer, and J. Shawe-Taylor. Unsupervised analysis of fMRI data using kernel canonical correlation. *NeuroImage*, 37(4):1250–1259, 2007.

J. Haxby, M. Gobbini, M. Furey, A. Ishai, J. Schouten, and P. Pietrini. Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex. *Science*, 293(5539):2425–2430, Sept. 2001. doi: 10.1126/science.1063736.

M. Hebiri, S. Van De Geer, et al. The smooth-lasso and other 1+ 2-penalized methods. *Electronic Journal of Statistics*, 5:1184–1226, 2011.

M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv:1506.05163*, 2015.

C. Honey, O. Sporns, L. Cammoun, X. Gigandet, et al. Predicting human resting-state functional connectivity from structural connectivity. *Proceedings of National Academy of Sciences*, 106:2035, 2009.

J. Honorio and D. Samaras. Multi-task learning of Gaussian graphical models. In *Proceedings of the International Conference on Machine Learning*, 2010.

J. Honorio, D. Samaras, N. Paragios, R. Goldstein, and L. E. Ortiz. Sparse and locally constant Gaussian graphical models. In *Advances in Neural Information Processing Systems*, pages 745–753, 2009.

J. Honorio, T. Jaakkola, and D. Samaras. On the statistical efficiency of $\ell_{1,p}$ multi-task learning of Gaussian graphical models. *arXiv:1207.4255*, 2012a.

J. Honorio, D. Tomasi, R. Z. Goldstein, H.-C. Leung, and D. Samaras. Can a single brain region predict a disorder? *IEEE Transactions on Medical Imaging*, 31(11):2062–2072, Nov 2012b. ISSN 0278-0062. doi: 10.1109/TMI.2012.2206047.

J. J. Hopfield and D. W. Tank. "neural" computation of decisions in optimization problems. *Biological cybernetics*, 52(3):141–152, 1985.

C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in neural information processing systems*, pages 3165–3173, 2013.

J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proceedings of the International Conference on Machine Learning*, pages 417–424, 2009.

J. Huang, S. Zhang, H. Li, and D. Metaxas. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding*, 115(12):1610–1622, 2011a.

J. Huang, S. Zhang, and D. Metaxas. Efficient mr image reconstruction for compressed mr imaging. *Medical Image Analysis*, 15(5):670–679, 2011b.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.

J. Janková and S. van de Geer. Confidence intervals for high-dimensional inverse covariance estimation. *Electron. J. Statist.*, 9(1):1205–1229, 2015. doi: 10.1214/15-EJS1031.

A. Javanmard and A. Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, E. Eger, F. Bach, and B. Thirion. Multiscale mining of fmri data with hierarchical structured sparsity. *SIAM Journal on Imaging Sciences*, 5(3):835–856, 2012.

J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678. IEEE, 2015.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2016.

C. Kelly, B. B. Biswal, R. C. Craddock, F. X. Castellanos, and M. P. Milham. Characterizing variation in the functional connectome: Promise and pitfalls. *Trends in Cognitive Science*, 16:181, 2012.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2015a.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015b.

R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Neural Information Processing Deep Learning Workshop*, 2014.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1): 32–73, 2017.

A. Kyrillidis, G. Puy, and V. Cevher. Hard thresholding with norm constraints. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3645–3648. Ieee, 2012.

S. L. Lauritzen. *Graphical models*. Oxford University Press, 1996.

Y. LeCun and C. Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

A. Lenkoski. A direct sampler for G-Wishart variates. *Statistics*, 2(1):119–128, 2013.

G. Letac, H. Massam, et al. Wishart distributions for decomposable graphs. *The Annals of Statistics*, 35(3):1278–1323, 2007.

Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. *International Conference on Learning Representations*, 2016.

M. A. Lindquist et al. The statistical analysis of fMRI data. *Stat. Sci.*, 23(4):439–464, 2008.

X. Liu, J. Hairston, M. Schrier, and J. Fan. Common and distinct networks underlying reward valence and processing stages: A meta-analysis of functional neuroimaging studies. *Neuroscience & Biobehavioral Reviews*, 35(5):1219–1236, 2011.

R. Lockhart, J. Taylor, R. J. Tibshirani, and R. Tibshirani. A significance test for the lasso. *Ann. Stat.*, 42:413, 2014.

P.-L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *Ann. Stat.*, 41(6):3022–3049, 2013.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of cause-effect inference. In *Proceedings of the International Conference on Machine Learning*, 2015.

C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016.

W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, 2010.

J. Mairal and B. Yu. Supervised feature selection in graphs with path coding penalties and network flows. *Journal of Machine Learning*, 14(1):2449–2485, 2013.

N. T. Markov, M. Ercsey-Ravasz, D. C. Van Essen, K. Knoblauch, Z. Toroczkai, and H. Kennedy. Cortical high-density counterstream architectures. *Science*, 342(6158): 1238406, 2013.

G. Marsaglia. Conditional means and covariances of normal variables with singular covariance matrix. *Journal of the American Statistical Association*, 59(308):1203–1204, 1964.

A. M. McDonald, M. Pontil, and D. Stamos. New perspectives on k-support and cluster norms. *arXiv:1403.1481*, 2014.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Ann. Stat.*, pages 1436–1462, 2006.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

V. Michel, A. Gramfort, G. Varoquaux, E. Eger, and B. Thirion. Total variation regularization for fMRI-based prediction of behavior. *IEEE Trans. Med. Imaging*, 30(7):1328–1340, 2011.

M. Misyrlis, A. Konova, M. Blaschko, J. Honorio, N. Alia-Klein, R. Goldstein, and D. Samaras. Predicting cross-task behavioral variables from fMRI data using the k-support norm. In *Sparsity Techniques in Medical Imaging*, 2014a.

M. Misyrlis, A. B. Konova, M. B. Blaschko, J. Honorio, N. Alia-Klein, R. Z. Goldstein, and D. Samaras. Predicting cross-task behavioral variables from fMRI data using the k-support norm. In *Sparsity Techniques in Medical Imaging*. 2014b.

S. J. Moeller, D. Tomasi, J. Honorio, N. D. Volkow, and R. Z. Goldstein. Dopaminergic involvement during mental fatigue in health and cocaine addiction. *Translational Psychiatry*, 2(10):e176, 2012.

B. Moghaddam, E. Khan, K. P. Murphy, and B. M. Marlin. Accelerating Bayesian structural inference for non-decomposable Gaussian graphical models. In *Neural Information Processing Systems*, 2009.

A. Mohammadi and E. C. Wit. Bayesian structure learning in sparse Gaussian graphical models. *Bayesian Analysis*, 10(1):109–138, 2015.

K. Mohan, M. Chung, S. Han, D. Witten, S.-I. Lee, and M. Fazel. Structured learning of Gaussian graphical models. In *Advances in Neural Information Processing Systems*, pages 620–628, 2012.

T. Moreau and J. Bruna. Understanding trainable sparse coding via matrix factorization. *International Conference on Learning Representations*, 2017.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

M. Narayan and G. I. Allen. Mixed effects models to find differences in multi-subject functional connectivity. *bioRxiv:027516*, 2015.

Y. Nesterov. *Introductory lectures on convex optimization*. Springer, 2004.

Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.

T. E. Nichols and A. P. Holmes. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002.

E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. *International Conference on Computer Vision*, 2017.

E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2208–2221, Sep. 2019.

N. Parikh, S. Boyd, et al. Foundations and trends in optimization. *Foundations and Trends in Theoretical Computer Science*, 8(1-2), 2014.

F. Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning*, 12:2825–2830, 2011.

C. Peeters, A. Bilgrau, and W. van Wieringen. rags2ridges: Ridge estimation of precision matrices from high-dimensional data. *R package*, 2015.

R. A. Poldrack, J. A. Mumford, and T. E. Nichols. *Handbook of functional MRI data analysis*. Cambridge University Press, 2011.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *EJS*, 5:935–980, 2011.

J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville. Decoding brain states from fMRI connectivity graphs. *NeuroImage*, 56:616–626, 2011.

P. Rigollet. 18. s997: High dimensional statistics. *Lecture Notes, Cambridge, MA, USA: MIT OpenCourseWare*, 2015.

A. Roverato. Hyper inverse wishart distribution for non-decomposable graphs and its application to bayesian inference for gaussian graphical models. *Scandinavian Journal of Statistics*, 29(3):391–411, 2002.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, Nov. 1992.

S. Ryali, T. Chen, K. Supekar, and V. Menon. Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage*, 59(4):3852–3861, 2012.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

W. R. Shirer, S. Ryali, E. Rykhlevskaia, V. Menon, and M. D. Greicius. Decoding subject-driven cognitive states with whole-brain connectivity patterns. *Cerebral Cortex*, 22(1): 158–165, 2012.

K. A. Smith. Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34, 1999.

S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, et al. Network modelling methods for fMRI. *NeuroImage*, 54:875, 2011.

S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao. Comparative study of SVM methods combined with voxel selection for object category classification on fMRI data. *PLoS One*, 6(2):e17191, 2011.

S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(suppl 1):i359–i368, 2005.

D. Teney, L. Liu, and A. v. d. Hengel. Graph-structured representations for visual question answering. *Computer Vision and Pattern Recognition*, 2016.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58:267–288, 1996a.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, pages 267–288, 1996b.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, 2005.

S. Van de Geer, P. Bühlmann, Y. Ritov, R. Dezeure, et al. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3): 1166–1202, 2014.

G. Varoquaux and R. C. Craddock. Learning and comparing functional connectomes across subjects. *NeuroImage*, 80:405–415, 2013.

G. Varoquaux, A. Gramfort, J.-B. Poline, and B. Thirion. Brain covariance selection: Better individual functional connectivity models using population prior. In *Advances in Neural Information Processing Systems*, 2010.

G. Varoquaux, A. Gramfort, F. Pedregosa, V. Michel, and B. Thirion. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Information Processing in Medical Imaging*, 2011.

V. Vazirani. *Approximation Algorithms*. Springer, 2001.

I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *International Conference on Representation Learning*, 2016.

O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In *Neural Information Processing Systems*, 2015.

O. Vinyals et al. Order matters: Sequence to sequence for sets. *International Conference on Learning Representations*, 2016.

L. Waldorp. Testing for graph differences using the desparsified lasso in high-dimensional data. *Statistics Survey*, 2014.

H. Wang, S. Z. Li, et al. Efficient gaussian graphical model determination under g-wishart prior distributions. *Electronic Journal of Statistics*, 6:168–198, 2012.

W. Wang, M. J. Wainwright, and K. Ramchandran. Information-theoretic bounds on model selection for gaussian markov random fields. In *IEEE International Symposium on Information Theory*, pages 1373–1377, 2010.

Y. Wang, L. Wang, Y. Li, D. He, T. Liu, and W. Chen. A theoretical analysis of normalized discounted cumulative gain (NDCG) ranking measures. In *Conference on Learning Theory*, 2013.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 (6684):440–442, 06 1998.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, 2009.

B. Xin, Y. Wang, W. Gao, and D. Wipf. Maximal sparsity with deep networks? *Advances in Neural Information Processing Systems*, 2016.

S. Yan, X. Yang, C. Wu, Z. Zheng, and Y. Guo. Balancing the stability and predictive performance for multivariate voxel selection in fMRI study. In *Brain Informatics and Health*, pages 90–99, 2014.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations*, 2016.

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.

W. Zaremba, M. P. Kumar, A. Gramfort, and M. B. Blaschko. Learning from M/EEG data with variable brain activation delays. In *Information Processing in Medical Imaging*, pages 414–425, 2013.

S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101 (2):253–268, 2014.

H. Zou and T. Hastie. Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Apprentissage de graphes structuré et parcimonieux dans des données de haute dimension avec applications à l'imagerie cérébrale

Mots clefs : apprentissage statistique, Gaussian graphical models, neuroimagerie, apprentissage profond

Résumé : Cette thèse présente de nouvelles méthodes d'apprentissage structuré et parcimonieux sur les graphes, ce qui permet de résoudre une large variété de problèmes d'imagerie cérébrale, ainsi que d'autres problèmes en haute dimension avec peu d'échantillon. La première partie de cette thèse propose des relaxation convexe de pénalité discrète et combinatoriale impliquant de la parcimonie et bounded total variation d'un graphe, ainsi que la bounded ℓ_2. Ceux-ci sont dévelopé dans le but d'apprendre un modèle linéaire interprétable et on démontre son efficacacité sur des données d'imageries cérébrales ainsi que sur les problèmes de reconstructions parcimonieux.

Les sections successives de cette thèse traite de la découvertre de structure sur des modèles graphiques "undirected" construit à partir de peu de données. En particulier, on se concentre sur des hypothèses de parcimonie et autres hypothèses de structures dans les modèles graphiques gaussiens. Deux contributions s'en dégagent.On construit une approche pour identifier les différentes entre des modèles graphiques gaussiens (GGMs) qui partagent la même structure sous-jacente. On dérive la distribution de différences de paramètres sous une pénalité jointe quand la différence des paramètres est parcimonieuse. On montre ensuite comment cette approche peut être utilisée pour obtenir des intervalles de confiances sur les différences prises par le GGM sur les arêtes. De là, on introduit un nouvel algorithme d'apprentissage lié au problème de découverte de structure sur les modèles graphiques non dirigées des échantillons observés. On démontre que les réseaux de neurones peuvent être utilisés pour apprendre des estimateurs efficacaces de ce problèmes. On montre empiriquement que ces méthodes sont une alternatives flexible et performantes par rapport aux techniques existantes.

Title : Structured Sparse Learning on Graphs in High-Dimensional Data with Applications to NeuroImaging

Keywords : machine learning, Gaussian graphical models, neuroimaging, deep learning

Abstract : This dissertation presents novel structured sparse learning methods on graphs that address commonly found problems in the analysis of neuroimaging data as well as other high dimensional data with few samples. The first part of the thesis proposes convex relaxations of discrete and combinatorial penalties involving sparsity and bounded total variation on a graph as well as bounded ℓ_2 norm. These are developed with the aim of learning an interpretable predictive linear model and we demonstrate their effectiveness on neuroimaging data as well as a sparse image recovery problem.

The subsequent parts of the thesis considers structure discovery of undirected graphical models from few observational data. In particular we focus on invoking sparsity and other structured assumptions in Gaussian Graphical Models (GGMs). To this end we make two contributions. We show an approach to identify differences in Gaussian Graphical Models (GGMs) known to have similar structure. We derive the distribution of parameter differences under a joint penalty when parameters are known to be sparse in the difference. We then show how this approach can be used to obtain confidence intervals on edge differences in GGMs. We then introduce a novel learning based approach to the problem structure discovery of undirected graphical models from observational data. We demonstrate how neural networks can be used to learn effective estimators for this problem. This is empirically shown to be flexible and efficient alternatives to existing techniques.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France