Leveraging Sparsity for Genetic and Wireless Cognitive Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Juan Andrés Bazerque

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Georgios B. Giannakis, Advisor

August 2013

Acknowledgments

First and foremost, my deepest gratitude goes to my advisor, Prof. Georgios B. Giannakis. I would like to thank him for giving me the opportunity to be part of his prestigious research group. His advice has been extraordinary by all means, dedicating extensive amounts of time to guide me through my research, improve my technical writing and oral presentations, and even read books together to introduce me to deeper understanding of the discipline. He explains what are intricate things to me in the most easy ways, and encourages me to go beyond that.

Due thanks go to Profs. Jarvis Haupt, Mos Kaveh, Chad Myers, and Nikos Sidiropoulos for agreeing to serve on my committee.

Through my graduate studies, I had the opportunity to collaborate with several individuals and I have benefited from their vision, idea, and insights. I would like to extend my gratitude to Dr. Danielle Angelosante, Dr. Emiliano Dall'Anesse, Dr. Seung-Jun Kim, Prof. Ketan Rajawat, and Prof. Hao Zhu. I am particularly appreciative of Prof. Xiaodong Cai whose tutelage during his sabbatical year at the University of Minnesota facilitated, if not made possible, my research on the area of biological sciences, and who deserves due credit for his collaboration on the contents of Chapters 2 and 3 of this thesis. Special thanks also go to Dr. Gonzalo Mateos for his enlightening and congenial collaboration that led to the contents of Chapters 4 and 5. I would also like to acknowledge the NSF and AFOSR grants that supported financially our research.

The material in this thesis has benefited from numerous additional inputs, discussions, and talks from current and former members of the Spincom group: Brian Baingana, Dr. Alfonso Cano, Yiannis Delis, Dr. Shahrokh Farahmand, Dr. Pedro Forero, Dr. Nikolaos Gatsis, Dr. Vassilis Kekatos, Dr. Seung-Jun Kim, Prof. Geert Leus, Dr. Guobing Li, Morteza Mardani, Prof. Antonio Marques, Dr. Eric Msechu, Prof. Yannis Schizas, Nasim Yahya Soltani, Dr. Yingqun Yu, Dr. Tairan Wang, Dr. Yuchen Wu, and Yu Zhang. I will not also forget Prof. A. Ribeiro and his family: I want to give them special thanks for among many other things their hospitality and help upon my arrival to Minneapolis. At this point, I would also like to thank Profs. Gregory Randall, and Juan Martony at the Instituto de Ingeniería Eléectrica, Universidad de la República, who encouraged me to pursue graduate studies abroad.

To my family and friends in Uruguay, especially to my father who has always led me by love and example, and to my friends in Minnesota whose company made my days more enjoyable. Finally, I lovingly dedicate this thesis to Ana and Juli, who have been a strong reason for me to try my best.

Juan Andrés Bazerque, Minneapolis, August 21, 2013.

Abstract

Sparse graphical models can capture uncertainty of interconnected systems while promoting parsimony and simplicity - two attributes that can be utilized to identify the topology and control processes defined on networks. This thesis advocates such models in the context of learning the structure of *gene-regulatory networks*, for which it is argued that single nucleotide polymorphisms can be seen as perturbation data that are critical to identify edge directionality. Applied to the immune-related gene network, these models facilitate the discovery of new regulation pathways.

Learning gene-regulating interactions is critical not only to understand how cells differentiate and behave, but also to decipher mechanisms triggering diseases with a genetic component. The impact here is on the development of a new generation of drugs designed to target specific genes. In particular, the genetic interactions of an uncharacterized chemical compound are identified by comparing its effect on the fitness of *Saccharomyces cerevisiae* (yeast) to that of double-deletion knockouts. As *drug targeting* is limited by expensive and time-involving laboratory tests, a judicious design of experiments is instrumental in order to reduce the required number of diagnostic mutant strains. During in-vitro experiments with 82 test-drugs, an orderly data reduction of 30% was shown possible without altering the identification of the primary chemical-genetic interactions.

Sparsity in *wireless cognitive networks* emerges due to the geographical distribution of sources, and also due to the scarcity of the radio frequency spectrum used for transmission. In this context, sparsity is leveraged for mapping the interference temperature across space while identifying unoccupied frequency bands. This is achieved by a novel so-terms nonparametric basis pursuit (NBP) method, which entails a basis expansion model with coefficients belonging to a function space. The spatial awareness markedly impacts spectral efficiency, especially when cognitive radios collaborate to reach consensus in a decentralized manner. Tested in a simulated communication setting, NBP captures successfully both shadowing as well as path-loss effects. In additional tests with real-field RF measurements, the spectrum maps reveal the frequency bands utilized for transmission and also reveal the position of the sources.

Finally, a *blind* NBP alternative is introduced to yield a Bayesian nuclear-norm regularization approach for matrix completion. In this context, it becomes possible to incorporate prior covariance information which enables smoothing and prediction. Blind NBP can be further applied to impute missing entries of third- or higher-order data arrays (tensors). These attracted features of blind NBP are illustrated for network flow prediction and imputation of missing entries in three-way ribonucleic-acid (RNA) sequencing arrays and magnetic-resonance-imaging (MRI) tensors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Networks abound in nature as biological systems at different levels, from genes regulating each other via their expressed proteins, to proteins binding one another to form signalling pathways, cells communicating via the release of hormones, or even humans interacting in their social environment. There are also multiple examples of engineered networks, as those of cognitive radios collaborating to increase spectral efficiency, aerial and terrestrial multi-robot systems coordinating a surveillance mission, clusters of microprocessors distributing the shared complexity of a computational task, or renewable sources delivering energy to the end consumers through the smart grid. The list goes on well beyond these examples since networked interactions are inherent to virtually any system with multiple components, and these are ubiquitous [103].

Sparsity is an attribute naturally present in most network applications since owing to proximity or shared functionality, each node interacts with a subset of neighbors only, which gives rise to sparse graph comprising a relatively small number of edges when compared with the maximal power graph formed by all possible node pairs. These premises set sparse networks as the common denominator and motivate the statistical models and learning techniques dealt with in this thesis, which is centered around gene-regulatory and wireless cognitive networks.

In the context of gene regulatory networks (GRN), a sparse structural equation model (SEM) is proposed with the goal of identifying the topology of signed, directed and cyclic networks. Such networks reveal activating or inhibitory genetic interactions and identify protein stabilizing loops. The main contribution in this aspect is three-fold: i) the incorporation of genotype data as perturbations that provably resolve direction ambiguities; ii) an algorithm to solve the sparsity-promoting regularized maximum likelihood SEM estimator, and iii) real data testing on a network of immune-related human genes, depicted in Fig. 1.1, to reveal previously uncharacterized regulation paths.

Also in the context of gene interactions, but now with a focus on drug discovery, a sparse model is proposed to capture chemical-genetic interactions of a new test drug. Accounting for these interactions accommodates multiple gene-targets, and contributes to a tailored design of experiments that markedly reduces the number of diagnostic mutant strains required for identifying the deletion mutants which are sensitive to the drug.

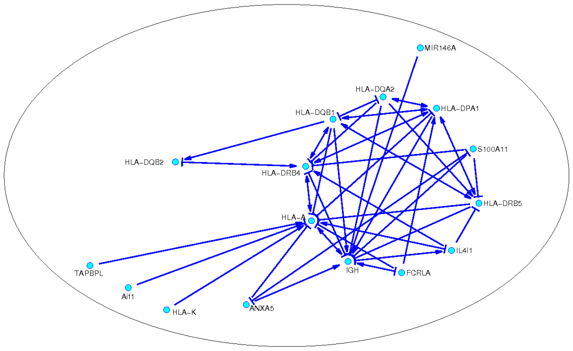


Figure 1.1: Gene regulatory network of immune-related human genes with cycles, edge directionality, and signed interactions.

Switching attention to wireless cognitive radio (CR) networks, an RF cartography approach is proposed for spectrum sensing with the goal of revealing unoccupied frequency bands, but lifting the prevalent and restrictive assumption that occupancy is space invariant. A novel nonparametric basis pursuit (NBP) method is proposed to leverage sparsity for revealing the unoccupied bands, and map the distribution of power across space, time, and frequency. A distributed NBP algorithm is devised to obtain the spectrum maps through the collaboration of CRs communicating with their neighbors in a decentralized fashion.

When a blind approach is considered to fit the NBP model, and the space of feasible functions is constrained to a linear span of Kronecker deltas, NBP reduces to low-rank matrix imputation via nuclear norm regularization. In a more general setup, the proposed blind NBP develops into a Bayesian inference method for matrix completion that incorporates covariance information and thus enables smoothing and prediction. In addition, blind NBP can be applied to impute missing entries of third- or higher-order data arrays (tensors). This property can be further appreciated by noticing that extending the methods for matrix nuclear-norm regularization to low rank tensors is not straightforward, because higher-order PARAFAC and Tucker decompositions, which define the rank and singular values of a tensor, are basically unrelated. These extra capabilities of blind NBP are illustrated with applications to network flow prediction and imputation of missing data corresponding to magnetic resonance imaging (MRI) and the three-way ribonucleic acid (RNA) sequencing.

1.1 Motivation and context

From a high-level viewpoint, the research themes of this thesis can be divided in four thrusts: [T1] inference of gene regulatory networks; [T2] Design of experiments for drug targeting; [T3] Spectrum cartography in cognitive radio networks; and, [T4] tensor rank regularization. Further motivation and context for each of these thrusts is given in this section, with their detailed description, tests, and specific results deferred to the corresponding chapters of this thesis that are delineated next.

1.1.1 Sparsity-aware inference of gene regulatory networks

Genes in living organisms do not function in isolation, but may interact with each other and act together forming intricate networks [111]. Deciphering the structure of gene regulatory networks is crucial for understanding gene functions and cellular dynamics, as well as for system-level modeling of individual genes and cellular functions. Although physical interactions among individual genes can be experimentally deduced (e.g., by identifying transcription factors and their regulatory target genes or discovering protein-protein interactions), such an experimental approach is time consuming and labor intensive. Given the explosive number of combinations of genes involved in any possible gene interaction, such an approach may not be practically feasible to reconstruct or “reverse engineer” gene networks. On the other hand, technological advances allow for high-throughput measurement of gene expression levels to be carried out efficiently and in a cost-effective manner. These genome-wide expression data reflect the state of the underlying network in a specific condition and provide valuable information that can be fruitfully exploited to infer the network structure.

Indeed, a number of computational methods have been developed to infer gene networks from gene expression data. One class leverages a similarity measure, such as the correlation or mutual information present in pairs of genes, to construct a so-termed co-expression or relevance network [16, 32]. Another approach relies on Gaussian graphical models with edges being present (absent) if the corresponding gene pairs are conditionally dependent (respectively independent), given expression levels of all other genes [62, 155]. While the approach based on Gaussian graphical models entails undirected graphs, directed acyclic graphs (DAGs) or Bayesian networks have also been employed to infer the dependency structure among genes [72, 159]. The fourth approach employs linear regression models and associated inference methods to find the dependency among genes and to infer gene networks [28, 60, 75, 156], [103]. Finally, while these approaches use gene expression data in the steady state, several methods exploiting time series expression data have also been reported; see e.g., [141, 166] and references therein.

Recently, gene expression data from gene-knockout experiments have been combined with time series comprising gene expression data with perturbations to considerably improve the accuracy of network inference [189]. When a gene is knocked out or silenced, expression levels of other genes are perturbed. Different from using gene expression levels of the original network alone, comparing gene expression levels in the perturbed network with those in the original network reveals extra information about the underlying network structure. Gene perturbations can be performed with other experimental approaches such as controlled gene over-expression and treatment of cells with certain chemical compounds [60, 75]. However, these gene perturbation experiments may not be feasible for all genes or organisms. To overcome this hurdle, one can exploit naturally occurring genetic variations that can be viewed as perturbations to gene networks [152]. More importantly, such genetic variations enable inference of the causal relationship between different genes or between genes and certain phenotypes.

Structural equation models combining phenotype and genetic perturbation data

Several approaches are available to capitalize on both genetic variations and gene expression data for inference of gene networks. The first approach models a gene network as a Bayesian network, and then infers the network by incorporating prior information about the network obtained from expression quantitative trait loci (eQTLs) [198–200]. In the second approach, a likelihood test is employed to search for a casual model that "best" explains the observed gene expression and eQTL data [13, 45, 108, 128, 134]. The third approach relies on the SEM to infer gene [115, 117, 126, 187] or phenotype networks [41, 58, 79, 96, 153, 183, 196]. While these approaches focus on inference of gene networks incorporating information from eQTL, another approach employs both phenotype and QTL genotype data to jointly decipher the phenotype network and identify eQTLs that are causal for each phenotype [135]. Logsdon and Mezey [117] proposed an adaptive Lasso (AL) [201] based algorithm to infer gene networks modeled with an SEM. They compared the performance of a number of methods using simulated directed acyclic or cyclic networks. Their simulations showed that the AL-based algorithm outperformed all other methods tested. Despite its superiority over other methods, the AL-based algorithm does not fully exploit the structure of the SEM. Therefore, it is expected that a more systematic inference algorithm may significantly improve performance of the SEM-based approach.

Motivated by the fact that gene networks or more general biochemical networks are sparse [75, 97, 175, 177], a sparse SEM is advocated in this chapter to infer gene networks from both gene expression and eQTL data. Incorporating network sparsity constraints, a sparsity-aware maximum likelihood (SML) algorithm is developed for network topology inference. The core technique used is to maximize the likelihood function regularized by the ℓ_1-norm of the parameter vector determining the network structure. The ℓ_1-norm controls complexity of the SEM, and thus yields a sparse network. The key innovative element of the SML algorithm is a block coordinate ascent method derived to maximize the ℓ_1-regularized likelihood function, which makes the SML algorithm computationally efficient. The simulations provided demonstrate that the novel SML algorithm offers significantly better performance than the two state-of-the-art algorithms: the AL [117], and the QDG algorithm [134]. The SML algorithm is further applied to infer a human network of 39 human genes related to the immune function.

1.1.2 Design of experiments for revealing chemical-genetic interactions

Recent advances on pharmacology include the development of a new generation of drugs for cancer therapy that target specific tumor cells, thus avoiding the extensive tissue damage induced by conventional chemotherapy [26]. Drug specificity is effected by acting over cells with cancer-related mutations only, while silencing genes that repress apoptosis to trigger the programmed cell death.

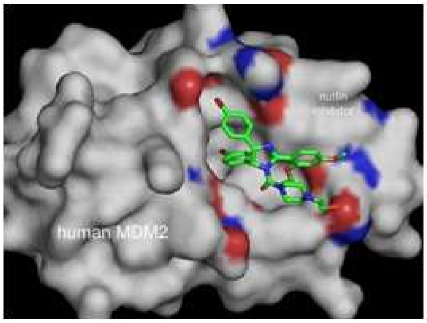


Figure 1.2: A binding chemical compound renders protein MDM2 nonfunctional [61].

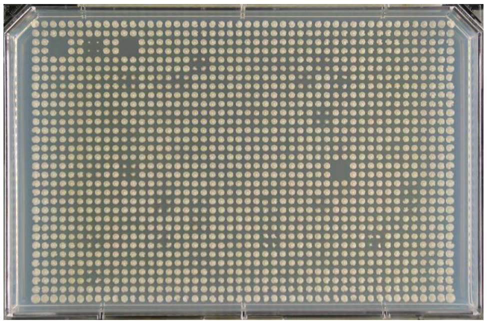


Figure 1.3: Fitness phenotype profiles of double-deletion mutants of Saccharomyces cerevisiae (yeast) [52].

A critical step in the development of such targeted therapy is to identify the target genes of a drug. To this end, a number of alternatives have been developed, including the association method [95], haploinsufficiency profiling [77], chemical genomics [137], and the gene network-based method [60]. The chemical genomics approach in particular, has been shown effective in identifying target genes for a number of chemical compounds [90, 137, 138]. In this latter approach, the chemical-genetic interaction profile of a chemical compound obtained by treating a set of single mutants with the compound is compared against a library of genetic interaction profiles of a set of double-deletion mutants (shown in Fig. 1.3). Specifically, perturbations introduced by a certain chemical compound emulate the effect of a gene knockout, by binding to the protein expressed from the gene and thus rendering it nonfunctional [15] (Fig. 1.2). Hence, the loss of functionality of a gene corresponding to the target of an inhibitory compound models the primary effect of the compound. The fitness of the double deletion mutants, obtained by crossing the mutated target gene with a set of viable mutants, yields a genetic interaction profile for the target gene that resembles the chemical-genetic interaction profile of its inhibitory compound. On this ground, correlating chemical-genetic interaction profile of a chemical compound with a set of genetic interaction profiles offers the potential to reveal the target genes of a drug.

The main challenge facing advances of such a drug-targeting approach resides in the lack of time [12] and cost efficiency [139]. Indeed, this approach is time-consuming and expensive since for each drug thousands of deletion mutant strains need to be screened for sensitivity. Hence, a high-impact objective is to reduce the number of experiments for each drug without compromising the accuracy of target prediction.

Toward this objective, a sparse linear model is put forth to describe chemical-genetic interactions, while at the same time accommodating multiple possible targets per chemical compound. As a result, the main result of this contribution consists in designing the most informative subset of experiments to carry out under a prescribed budget. Specifically, if one can only afford acquiring a limited number of entries per chemical-genetic profile of a new test-drug, the proposed design of experiments aims to unveil the optimal subset of entries to acquire.

1.1.3 Spectrum cartography in decentralized cognitive-radio networks

The cognitive radio paradigm endeavors to mitigate the scarcity of spectral resources for wireless communications through intelligent sensing and agile resource allocation techniques [130], [89]. The motivating reason is that although most of the available spectrum has been licensed to primary users (PUs) for exclusive usage, it is often significantly underutilized depending on the time and location that communication takes place [2]. CRs aim to learn the RF landscape, and identify the unused spectral resources (often called white space or spectrum holes) in the time, frequency, and space domains through spectrum sensing.

Sensing the ambient interference spectrum is of paramount importance to the operation of CR networks (Fig. 1.4), since it enables spatial frequency reuse and allows for dynamic spectrum allocation; see, e.g., [74], [129] and references therein. Collaboration among CRs can markedly improve sensing performance [145], and is key to revealing opportunities for spatial frequency reuse [136]. Pertinent existing approaches have mostly relied on detecting spectrum occupancy per radio, and do not account for spatial changes in the radio frequency (RF) ambiance,

Figure 1.4: Heterogeneous network of CRs collaborating on spectrum cartography.

especially at the intended receiver(s) which may reside several hops away from the sensed area.

The impact of this chapter's novel field estimator to CR networks is a collaborative sensing scheme whereby receiving CRs cooperate to estimate the distribution of power at spatial location $\mathbf{x}$ and frequency f, namely the power spectrum density (PSD) map $\Phi(\mathbf{x}, f)$, as depicted in Fig. 1.5, from local periodogram measurements.

Toward this goal, the following basis expansion model (BEM) is adopted for the target map

$$\Phi(\mathbf{x}, f) = \sum_{\nu=1}^{N_b} g_\nu(\mathbf{x}) b_\nu(f) \tag{1.1}$$

with $\mathbf{x} \in \mathbb{R}^2$, $f \in \mathbb{R}$, and the L_2-norms $\{||b_\nu(f)||_{L_2} = 1\}_{\nu=1}^{N_b}$ normalized to unity.

Bases $\{b_\nu(f)\}_{\nu=1}^{N_b}$ are preselected, while functions $g_\nu(\mathbf{x})$ are to be estimated based on noisy samples of Φ. This way, the model-versus-data balance is calibrated by introducing a priori knowledge on the dependence of the map Φ w.r.t. variable f, or more generally a group of variables, while trusting the data to dictate the functions $g_\nu(\mathbf{x})$ of the remaining variables $\mathbf{x}$.

Consider selecting N_b basis functions using the *basis pursuit* approach [44], which entails an extensive set of bases thus rendering N_b overly large and the model overcomplete. This motivates augmenting the variational LS problem with a suitable sparsity-promoting penalty, which endows the map estimator with ability to discard factors $g_\nu(\mathbf{x})b_\nu(f)$ in (1.1), only keeping a few bases that "better" explain the data. This attribute is inherited because the novel kernel-based method of this chapter induces group sparsity in the coefficients of the optimal finitely-parameterized g_ν.

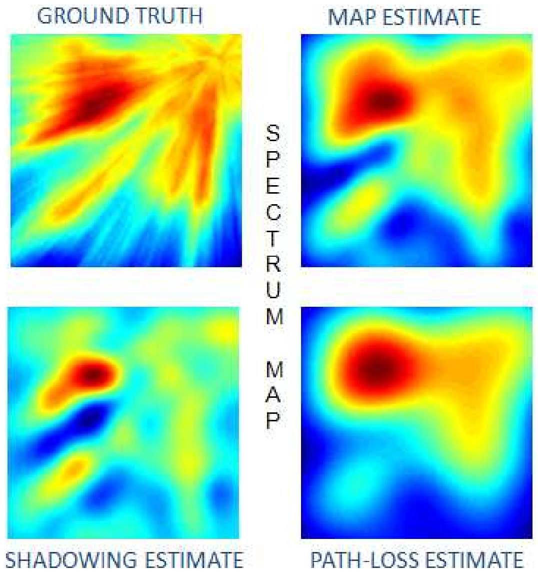


Figure 1.5: Spectrum maps obtained via nonparametric basis pursuit.

The spectrum cartography method should be precise enough to identify spectrum holes, which justifies adopting the known bases to capture the PSD frequency dependence in (1.1). As far as the spatial dependence is concerned, the model must account for path loss, fading, mobility, and shadowing effects, all of which vary with the propagation medium. For this reason, it is prudent to let the data dictate the spatial component of (1.1). Knowing the spectrum at any location allows remote CRs to reuse dynamically idle bands. It also enables CRs to adapt their transmit-power so as to minimally interfere with licensed transmitters. The kernel-based PSD maps here provide an alternative to [17], where known bases are used both in space and frequency. Different from [8] and [17], the field estimator here does not presume a spatial covariance model or pathloss channel model. Moreover, it captures general propagation characteristics including both shadowing and fading.

Nonparametric basis pursuit

The fitting criterion for (1.1) relies on reproducing kernel Hilbert spaces (RKHSs), which provide an orderly analytical framework for nonparametric regression, with the optimal kernel-based function estimate emerging as the solution of a regularized variational problem [191]. The pivotal role of RKHS is further appreciated through its connections to “workhorse” signal processing tasks, such as the Nyquist-Shannon sampling and reconstruction result that involves sinc kernels [132]. Alternatively, spline kernels replace sinc kernels, when smoothness rather than bandlimitedness is to be present in the underlying function space [182].

Kernel-based function estimation can be also seen from a Bayesian viewpoint. RKHS and linear minimum mean-square error (LMMSE) function estimators coincide when the pertinent covariance matrix equals the kernel Gram matrix. This equivalence has been leveraged in the context of field estimation, where spatial LMMSE estimation referred to as Kriging, is tantamount to two-dimensional RKHS interpolation [53]. Finally, RKHS based function estimators can be linked with Gaussian processes (GPs) obtained upon defining their covariances via kernels [146].

Recent advances in sparse signal recovery and regression motivate a sparse kernel-based learning (KBL) redux introduced in this thesis chapter. Building blocks of sparse signal processing include the (group) least-absolute shrinkage and selection operator (Lasso) and its weighted versions [88], compressive sampling [36], and nuclear norm regularization [68]. The common denominator behind these operators is the sparsity on a signal’s support that the ℓ_1-norm regularizer induces. Exploiting sparsity for KBL leads to several innovations regarding the selection of multiple kernels [105, 127], additive modeling [114, 147], collaborative filtering [6], matrix and tensor completion via dictionary learning [24], as well as nonparametric basis selection [21]. In this context, the spectrum cartography approach is immersed into NBP, which is understood in a wider sense as a framework unifying and advancing a number of *sparse* KBL approaches.

Yet another seemingly unrelated, but increasingly popular theme in contemporary statistical learning and signal processing, is that of matrix completion [68], where data organized in a matrix can have missing entries due to e.g., limitations in the acquisition process. This chapter builds on the assertion that imputing missing entries amounts to interpolation, as in classical sampling theory, but with the low-rank constraint replacing that of bandlimitedness. From this point of view, RKHS interpolation via a blind NBP emerges as the prudent framework for matrix completion that allows effective incorporation of a priori information via kernels [6], including sparsity attributes.

Sparse KBL and its various forms contribute to computer vision [162,185], cognitive radio sensing [21], management of user preferences [6], bioinformatics [167], econometrics [114, 147], and forecasting of electric prices, load, and renewables (e.g., wind speed) [102], to name a few.

Decentralized cognitive-radio networks

Spectrum cartography is further developed into a distributed algorithm that capitalizes on the equivalence of NBP and group-Lasso. Consider the classical problem of linear regression, where a vector $\mathbf{y} \in \mathbb{R}^n$ of observations is given along with a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of inputs. Suppose the p features are split into N_b disjoint *factors* (groups of features) such that the coefficient vector is $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_{N_b}']' \in \mathbb{R}^p$, where $'$ denotes transposition and $\boldsymbol{\zeta}_\nu$ corresponds to the coefficients of factor ν. The *group* least-absolute shrinkage and selection operator (Lasso) [190] is a model selection and estimation technique used to select relevant factors in linear regression, and yields

$$\hat{\boldsymbol{\zeta}}_{\text{glasso}} := \arg\min_{\boldsymbol{\zeta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\zeta}\|_2^2 + \mu \sum_{\nu=1}^{N_b} \|\boldsymbol{\zeta}_\nu\|_2 \tag{1.2}$$

where $\mu \geq 0$ is a tuning parameter typically chosen via model selection techniques such as cross-validation (CV); see e.g., [88, 190]. If $\mu = 0$, no sparsity is enforced since (1.2) reduces to LS. As μ increases, more sub-vector estimates $\boldsymbol{\zeta}_f$ become zero due to the effect of the group sparsity-encouraging penalty, and the corresponding factors drop out of the model. When $N_b = p$, (1.2) becomes the Lasso [178] that performs variable – rather than factor – selection.

Finding $\hat{\zeta}_{\text{glasso}}$ requires solving (iteratively) for any given value of μ a second-order cone program (SOCP). While standard SOCP solvers can be invoked to this end, an increasing amount of effort has been put recently into developing fast algorithms that capitalize on the unique properties of the group-Lasso; see e.g. [190], [144], [82], [70], [194].

Typically, the training set is assumed to be centrally available, so that it can be jointly processed to obtain $\hat{\zeta}_{\text{glasso}}$. However, collecting all data in a central location may be prohibitive in contemporary applications of interest. A decentralized alternative is desired, in which the CRs communicate with neighbors within their coverage area, eliminating the need of a dedicated infrastructure, and facilitating scalability specially for big data network applications [103].

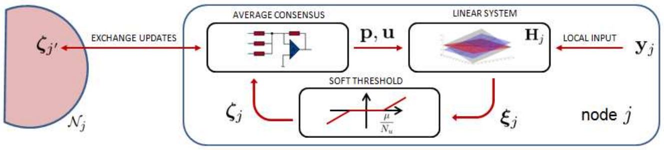


Figure 1.6: Schematic diagram for the distributed NBP algorithm running at a local CR.

Having this context in mind, a consensus-based distributed algorithm is developed for group-Lasso, which can be specialized for the Lasso as well. The resulting optimization is recast as a convex *constrained* minimization and it is iteratively solved using the alternating-direction method of multipliers (AD-MoM) [25, p. 253]. This way, provably convergent parallel recursions are derived to update each CR's local estimate, that entail simple vector soft-thresholding operations. This becomes possible by leveraging the closed-form solution that is available in the orthonormal case [194], [144], and evidences the factor-level sparsity encouraging property of group-Lasso. A schematic of the resulting distributed algorithm is shown in Fig. 1.6, where it is indicated that on a per iteration basis, CRs only exchange their current local estimate with their neighbors. By specializing to a dummy single node network, a novel centralized group-Lasso solver is obtained as a byproduct. Different from [190] and [144], the algorithm here can handle a nonorthonormal matrix $\mathbf{X}$, and does not require an inner Newton-Raphson recursion per iteration. By comparing the centralized algorithm with its distributed counterparts of it is shown that the latter effectively split the computational burden across nodes.

1.1.4 Bayesian rank regularization for tensor completion and extrapolation

Returning to the analysis of blind NBP, the fourth and last chapter of this thesis focuses on imputation of data arrays. Imputation of missing data is a basic task arising in various Big Data applications as diverse as medical imaging [73], bioinformatics [10], as well as social and computer networking [48, 119]. The key idea rendering recovery feasible is the "regularity" present among missing and available data. Low rank is an attribute capturing this regularity, and can be readily exploited when data are organized in a matrix. A natural approach to *low-rank matrix completion* problem is minimizing the rank of a target matrix, subject to a constraint on the error in fitting the observed entries [34]. Since rank minimization is generally NP-hard [184], the nuclear norm has been advocated recently as a convex surrogate to the rank [68]. Beyond tractability, nuclear-norm minimization enjoys good performance both in theory as well as in practice [34].

The goal of this chapter is to impute missing entries of tensors (also known as multi-way arrays), which are high-order generalizations of matrices frequently encountered in chemometrics, medical imaging, and networking [50, 104]. Leveraging the low-rank structure for tensor completion is challenging, since even computing the tensor rank is NP-hard [87]. Defining a nuclear norm surrogate is not obvious either, since singular values as defined by the Tucker decomposition are not generally related with the rank. Traditional approaches to finding low-dimensional representations of tensors include unfolding the multi-way data and applying matrix factorizations such as the singular-value decomposition (SVD) [10, 46, 180] or, employing the parallel factor (PARAFAC) decomposition [107, 176]. In the context of tensor completion, an approach falling under the first category can be found in [73], while imputation using PARAFAC was dealt with in [5].

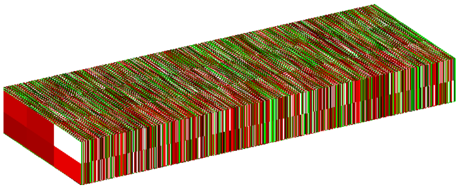


Figure 1.7: Three-way RNA sequencing data modeled as a Poisson counting process with missing entries.

The imputation approach presented in this chapter builds on a novel regularizer accounting for the tensor rank, that relies on redefining the matrix nuclear norm in terms of its low-rank factors. The contribution is two-fold. First, it is established that the low-rank inducing property of the regularizer carries over to tensors by promoting sparsity in the factors of the tensor's PARAFAC decomposition. In passing, this analysis allows for drawing a neat connection with the atomic-norm in [40]. The second contribution is the incorporation of prior information by using a Bayesian approach that endows tensor completion with extra smoothing and prediction capabilities. A parallel analysis in the context of RKHS further explains these acquired capabilities, provides an alternative means of obtaining the prior information, and establishes a useful connection with collaborative filtering approaches [6] when reduced to the matrix case. It is also in this context that low-rank tensor imputation meets NBP.

While LS is typically utilized as the fitting criterion for matrix and tensor completion, implicitly assuming Gaussian data, the adopted probabilistic framework supports the incorporation of alternative data models. Targeting count processes available in the form of network flows, social media interactions, or the genome sequencing tensor-data in Fig. 1.7, all of them modeled as Poisson distributed, the maximum a posteriori (MAP) estimator is expressed in terms of the Kullback-Leibler (K-L) divergence [48].

1.2 Thesis outline and contributions

The four thrusts described in the previous section are presented in Chapters 2-4. The sparse SEM for gene regulatory networks is introduced in Chapter 2, together with the main identifiability result, which justifies the use of eQTL perturbations, and with a detailed development of the ℓ-1 regularized algorithm for maximum likelihood estimation of SEMs. The methods in this chapter are compared to the state of the art through extensive simulations, and are used to infer a network of 39 immune-related genes from real data acquired from human blood cells. Edges in the so inferred network are compared with documented gene interactions in the literature, finding support for all discoveries but for three edges that, according to the theoretically imposed false-alarm constraints, are speculated to reveal newly discovered regulation paths.

Chapter 3 deals with drug discovery, based on the sparse model for chemical-genetic interactions, which also accommodates multiple drug targets. Then the correlation rule for detection is presented, the criterion for design of experiments is postulated, and a semi-definite program is formulated. The methods in this chapter are tested with real data to deduce that 30% reduction in the number of experiments is possible if a single mismatch is admitted over the set of 10% identified target-genes, thus corroborating the performance gains over a randomized design, which can only afford 4% data reduction at 10% mismatch. In addition, a comprehensive list of 10 target-genes for 82 test- drugs is included, further establishing that the primary target-gene identification is unaltered by the 30% data reduction induced by the proposed design.

Spectrum cartography is the subject of Chapter 4. It begins with a review of RKHS and sparse kernel-based learning, to introduce the overcomplete basis model and the variational NBP estimator. The collaborative algorithm for distributed sparse estimation is also developed here along with the blind NBP method for matrix completion. The methods in Chapter 4 are tested in a simulated communication setting, which reveals that the use of a double-kernel learning approach is successful in capturing both shadowing and path-loss effects. A test on real RF measurements is also included in this chapter, showing that the sparsity pattern obtained with NBP reveals the frequency bands utilized for transmission, and that the field estimates can also localize the PU sources. An application of blind NBP is further explored for network traffic flow prediction based on historical data obtained from the Internet 2 repository.

Chapter 5 is dedicated to low-rank tensor based inference. A novel regularizer on the factors of the PARAFAC decomposition is proposed, and shown equivalent to the matrix nuclear norm in the case of two-way arrays. The sparsifying effect of such a regularizer on the PARAFAC outer-products is demonstrated by showing equivalence with an $\ell_{2/3}$ minimization problem, and with the atomic norm. The resulting estimator is endowed with extra smoothing and extrapolation capabilities by reconstructing it in a Bayesian setup, which is proved equivalent to blind NBP in the matrix case. Two algorithms are developed to process Gaussian and Poisson data, respectively. These algorithms are tested with real data to impute missing entries of two third-order tensors containing MRI scans and RNA sequencing arrays.

1.3 Published results

The results on gene regulatory networks have been accepted for publication in PLOS Computational Biology [39]. Spectrum cartography and the distributed algorithms have been published in the IEEE Transactions on Signal Processing [17], [21], [122], in the IEEE Signal Processing Magazine [18], in the Elsevier Journal on Physical Communication [55], and in the EURASIP Journal on Signal Processing [56]. The work on low-rank tensor imputation has been submitted and is currently under review for the IEEE Transactions on Signal Processing [24]. All these results have also been disseminated at flagship international conferences, where eight articles have been accepted for presentation [20], [19] [23], [22], [38], [57], [123], and [124].

1.4 Notational conventions

The notation adopted throughout uses bold lowercase and capital letters for vectors $\mathbf{a}$ and matrices $\mathbf{A}$, respectively, with superscript T denoting transposition. Tensors are underlined as e.g., $\underline{\mathbf{X}}$, and their slices carry a subscript as in $\mathbf{X}_p$; see also Fig. 5.1. Both the matrix and tensor Frobenius norms are represented by $\|\cdot\|_F$. Symbols $\otimes$, $\odot$, $\circledast$, and $\circ$, denote the Kroneker, Kathri-Rao, Hadamard (entry-wise), and outer product, respectively. Vector diag$(\mathbf{M})$ collects the diagonal entries of $\mathbf{M}$, whereas the diagonal matrix diag$(\mathbf{v})$ holds the entries of $\mathbf{v}$ on its diagonal. The ℓ_q norm of vector $\mathbf{x} \in \mathbb{R}^p$ is $\|\mathbf{x}\|_q := (\sum_{i=1}^p |x_i|^q)^{1/q}$ for $q \geq 1$; and $\|\mathbf{M}\|_F := \sqrt{\text{tr}\,(\mathbf{M}\mathbf{M}')}$ is the matrix Frobenious norm. Positive-definite matrices will be denoted by $\mathbf{M} \succ \mathbf{0}$. The $p \times p$ identity matrix will be represented by $\mathbf{I}_p$, while $\mathbf{0}_p$ will denote the $p \times 1$ vector of all zeros, and $\mathbf{0}_{p\times q} := \mathbf{0}_p\mathbf{0}_q'$. Similar notation will be adopted for vectors (matrices) of all ones. The i-th vector of the canonical basis in $\mathbb{R}^n$ will be denoted by $\mathbf{e}_i$, $i = 1, \ldots, n$.

Chapter 2

Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations

Consider expression levels of N_g genes from N individuals measured using e.g., microarray or RNA-seq. Let $\mathbf{y}_i := [y_{i1}, \ldots, y_{iN_g}]^T$ denote the $N_g \times 1$ vector collecting the expression levels of these N_g genes of individual i. Suppose that a set of perturbations to these genes has been also observed. These perturbations can be due to naturally occurring genetic variations near or within the genes, gene copy number changes, gene knockdown by RNAi or controlled gene over-expression. In this chapter, focus is placed on genetic variations observed at eQTLs, although the network model and the inference method described in the next section are also applicable to cases where other perturbations are available. As in [117], it is assumed that each gene has at least one *cis*-eQTL so that the structure of the underlying gene network is uniquely identifiable. Let $\mathbf{x}_i := [x_{i1}, \ldots, x_{iN_q}]^T$ denote the genotype of $N_q \geq N_g$ eQTLs of individual i. The goal is to infer the network structure of the N_g genes from the available gene expression measurements $\mathbf{y}_i$, $i = 1, \ldots, N$, and eQTL observations $\mathbf{x}_i$, $i = 1, \ldots, N$.

As in [115, 117], the gene network is postulated to obey the SEM

$$\mathbf{y}_i = \mathbf{B}\mathbf{y}_i + \mathbf{F}\mathbf{x}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, N \tag{2.1}$$

where $N_g \times N_g$ matrix $\mathbf{B}$ contains unknown parameters defining the network structure; $N_g \times N_q$ matrix $\mathbf{F}$ captures the effect of each eQTL; $N_g \times 1$ vector $\boldsymbol{\mu}$ accounts for possible model bias; and $N_g \times 1$ vector $\boldsymbol{\epsilon}_i$ captures the residual error, which is modeled as a zero-mean Gaussian vector with covariance $\sigma^2\mathbf{I}$, where $\mathbf{I}$ denotes the $N_g \times N_g$ identity matrix. It is assumed that no self-loops are present per gene, which implies that the diagonal entries of $\mathbf{B}$ are zero. As mentioned in [117], lack of self-loops and a diagonal covariance matrix of $\boldsymbol{\epsilon}_i$ are commonly assumed in almost all graph-based network inference methods. It is further assumed that the loci of N_q eQTLs have been determined using an existing eQTL method, but the effective size of each eQTL is unknown. Therefore, $\mathbf{F}$ has N_q unknown entries whose locations are known and $N_gN_q - N_q$ remaining zero entries (for instance $\mathbf{F}$ is a diagonal matrix when $N_q = N_g$).

The network inference task is to estimate $N_g(N_g - 1)$ unknown entries of $\mathbf{B}$, and as a byproduct, the N_q unknown entries of $\mathbf{F}$. Without any knowledge about the network, no restriction is imposed on the structure specified by $\mathbf{B}$. Therefore, the network is considered as a general directed graph that can possibly be a directed cyclic graph (DCG) or a DAG. Network inference is challenging since the number of unknowns to be estimated is very large for a moderately large N_g. Note that under the assumption that each gene has at least one *cis*-eQTL, the “Recovery” Theorem in [117] guarantees that the network is identifiable for both DCGs and DAGs.

As discussed in [75, 97, 175, 177], gene regulatory networks or more general biochemical networks are sparse meaning that a gene directly regulates or is regulated by a small number of genes relative to the total number of genes in the network. Taking into account sparsity, only a relatively small number of the entries of $\mathbf{B}$ are nonzero. These nonzero entries determine the network structure and the regulatory effect of one gene on other genes. The SEM in (2.1) under the aforementioned sparsity assumption will be henceforth referred to as the sparse SEM. Exploiting the sparsity inherent to the network, an efficient and powerful algorithm for network inference will be developed in the ensuing section.

Sparsity-aware inference method

Upon defining $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, and $\mathbf{E} := [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N]$, the SEM in (2.1) can be compactly written as $\mathbf{Y} = \mathbf{BY} + \mathbf{FX} + \boldsymbol{\mu}\mathbf{1}^T + \mathbf{E}$, where **1** is the $N \times 1$ vector of all-ones.

Given $\mathbf{X}$ and $\mathbf{Y}$, the log-likelihood function can be written as

$$\begin{aligned}\log p(\mathbf{Y}|\mathbf{X};\mathbf{B},\mathbf{F},\boldsymbol{\mu}) =& \frac{N}{2}\log|\det(\mathbf{I}-\mathbf{B})|^2 - \frac{NN_g}{2}\log(2\pi\sigma^2)\\ &- \frac{1}{2\sigma^2}\|\mathbf{Y}-\mathbf{B}\mathbf{Y}-\mathbf{F}\mathbf{X}-\boldsymbol{\mu}_\epsilon\mathbf{1}^T\|_F^2 \end{aligned} \tag{2.2}$$

where $\det(\cdot)$ denotes matrix determinant, and $\|\cdot\|_F$ denotes the Frobenius norm.

As mentioned earlier, $\mathbf{B}$ is a sparse matrix having most entries equal to zero. In order to obtain a sparse estimate of $\mathbf{B}$, the natural approach is to maximize the log likelihood regularized by the weighed ℓ_1-norm term $\|\mathbf{B}\|_{1,W} := \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} w_{ij}|B_{ij}|$, where B_{ij} denotes the (i,j)th entry of $\mathbf{B}$. In a linear regression model, it is well known that the ℓ_1-regularized least-squares estimation also known as Lasso [178] can yield a sparse estimate of the regression coefficient vector. Similarly, the ℓ_1-regularized maximum likelihood (ML) approach used here is expected to shrink most of the entries of $\mathbf{B}$ toward zero, thereby yielding a sparse matrix. It is easy to show that maximizing $\log p(\mathbf{Y}|\mathbf{X};\mathbf{B},\mathbf{F},\boldsymbol{\mu})$ with respect to (w.r.t.) $\boldsymbol{\mu}$ yields $\hat{\boldsymbol{\mu}} = (\mathbf{I}-\mathbf{B})\bar{\mathbf{y}} - \mathbf{F}\bar{\mathbf{x}}$, where $\bar{\mathbf{y}} = \sum_{n=1}^{N}\mathbf{y}_n/N$ and $\bar{\mathbf{x}} = \sum_{n=1}^{N}\mathbf{x}_n/N$. Upon defining $\tilde{\mathbf{y}}_n := \mathbf{y}_n - \bar{\mathbf{y}}$, $\tilde{\mathbf{x}}_n := \mathbf{y}_n - \bar{\mathbf{x}}$, $\tilde{\mathbf{Y}} := [\tilde{\mathbf{y}}_1,\ldots,\tilde{\mathbf{y}}_N]$, $\tilde{\mathbf{X}} := [\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_N]$, and substituting $\hat{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$ in (2.2), the proposed ℓ_1-penalized ML estimation approach yields

$$\begin{gathered}(\hat{\mathbf{B}},\hat{\mathbf{F}}) = \arg\max_{\mathbf{B},\mathbf{F}} N\sigma^2\log|\det(\mathbf{I}-\mathbf{B})| - \frac{1}{2}\|\tilde{\mathbf{Y}}-\mathbf{B}\tilde{\mathbf{Y}}-\mathbf{F}\tilde{\mathbf{X}}\|_F^2 - \lambda\|\mathbf{B}\|_{1,W} \\ \text{subject to } \ B_{ii}=0, \forall i=1,\ldots,N_g,\ F_{jk}=0,\ \forall(j,k)\in\mathcal{S}_q\end{gathered} \tag{2.3}$$

where $\mathcal{S}_q$ denotes the set of row and column indices of the entries of $\mathbf{F}$ known to be zero. As assumed earlier, each phenotype has at least one *cis*-eQTL that has been identified, which implies that the locations of nonzero entries of $\mathbf{F}$ or equivalently the set $\mathcal{S}_q$ is known. However, our sparse SEM and inference method are also applicable to more general cases where some or all phenotypes have *cis*-eQTLs that have not been identified. In these cases, the locations of nonzero entries of $\mathbf{F}$ corresponding to the unidentified *cis*-eQTLs are unknown. We can form a weighted ℓ_1-norm of the entries of $\mathbf{F}$ excluding those corresponding to the identified *cis*-eQTL and then add a penalty term involving this ℓ_1-norm to the objective function in (2.3). This new optimization problem can be solved efficiently using a method modified from the one solving (2.3), as described in Appendix A.

Weights w_{ij} in the penalty term are introduced to improve estimation accuracy in line with the AL [201]. They are selected as $1/\tilde{B}_{ij}$, where $\tilde{B}_{ij}$ is found using a preliminary estimate of $\mathbf{B}$

obtained via ridge regression as

$$(\tilde{\mathbf{B}}, \tilde{\mathbf{F}}) = \arg\min_{\mathbf{B},\mathbf{F}} \frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2 + \rho\|\mathbf{B}\|_F^2$$

$$\text{subject to } \; B_{ii} = 0, \forall i = 1, \ldots, N_g, \; F_{jk} = 0, \; \forall (j,k) \in \mathcal{S}_q. \tag{2.4}$$

The sparsity-controlling parameters λ in (2.3) and ρ in (2.4) are selected via cross validation (CV), while σ^2 is estimated as the sample variance of the error using $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$. In adaptive Lasso based linear regression [201], Zou suggested using the ordinary least squares (OLS) estimate to determine the weights; if the OLS estimate does not exist due to, e.g., collinearity, Zou suggested the estimate obtained from ridge regression, although it remains to show if the ridge regression estimate is consistent in this case and if the resulting adaptive Lasso yields the desired oracle properties. If OLS is used for estimating $\mathbf{B}$ and $\mathbf{F}$ in the SEM, the solution usually does not exist since the number of unknowns is typically larger than the number of samples. However, even in this case the solution can always be obtained from ridge regression as in (2.4). Moreover, every entry of the solution is typically nonzero, which yields a finite weight for every variable, and thus every variable will be included in the following ℓ_1-penalized ML procedure. An alternative approach is to replace the weighed ℓ_1-norm in (2.3) with an unweighted ℓ_1-norm to obtain a preliminary estimate of $\mathbf{B}$ and then calculate the weights from this preliminary estimate, as in [117]. However, the unweighted ℓ_1-penalized ML procedure may shrink many variables to zero and exclude them from the weighted ℓ_1-penalized ML estimator, possibly yielding a biased estimate. For this reason, the inference method in this chapter uses ridge regression to determine $\{w_{ij}\}$, with the additional advantage of (2.4) admitting a closed-form solution.

A block diagram of the novel inference algorithm, abbreviated as the sparsity-aware maximum likelihood (SML) algorithm, is depicted in Figure 2.2. The first and third blocks in Figure 2.2 perform cross-validation to select optimal parameters ρ and λ to be used in (2.3) and (2.4), respectively (see the description of the cross-validation procedure in Appendix A.) The third block produces weights $\{w_{ij}\}$ and error-variance estimate $\hat{\sigma}_e^2$ after solving (2.4). Finally, the fourth block takes data $\mathbf{X}$ and $\mathbf{Y}$ together with λ, $\{w_{ij}\}$ and $\hat{\sigma}_e^2$ and solves (2.3) to yield $\hat{\mathbf{B}}$, representing the SML estimator for $\mathbf{B}$ in (2.1) and revealing the genetic-interaction network. As it will be described in the Methods section, (2.4) is separable across rows of $\mathbf{B}$ and $\mathbf{F}$, and each row of $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$ becomes available in closed form [cf. (2.11)-(2.12)]. The ℓ_1-regularized ML problem (2.3) is solved efficiently using a novel block coordinate ascent iterative scheme given by (2.14)-(2.19) in the Methods section. Precise description of the overall SML algorithm is also presented in the Methods section as Algorithm 1, which was used to yield an executable computer program.

2.1 Identifiability of SEMs with full rank perturbation data

The following proposition assesses identifiability of the network topology, and highlights the key role of the eQTL perturbation data $\mathbf{X}$ as the enabler to resolve direction ambiguities.

Proposition 2.1 *Assume that data $\mathbf{X}$ and $\mathbf{Y}$ abide to an SEM model $\mathbf{Y} = \mathbf{BY} + \mathbf{FX}$, for some matrix $\mathbf{B}$ with diagonal entries $b_{ii} = 0$ and diagonal matrix $\mathbf{F}$ with diagonal entries $f_{ii} \neq 0$. Assume also that $\mathbf{X}$ has full column rank.*

Then, $\mathbf{B}$ and $\mathbf{F}$ are unique. Moreover, they are expressible in terms of $\mathbf{X}$ and $\mathbf{Y}$ as $\mathbf{F} = \left(Diag\left[\left(\mathbf{YX}^{\dagger}\right)^{-1}\right]\right)^{-1}$ and $\mathbf{B} = \mathbf{I} - \mathbf{F}(\mathbf{YX}^{\dagger})^{-1}$.

Proposition 2.1 establishes that if the perturbation data is rich enough; i. e., presenting sufficient variation across samples, then the matrix $\mathbf{B}$ and thus the topology and edge direction can be uniquely identified. The importance of the perturbation data is emphasized by the counterexample in Fig. 2.1.

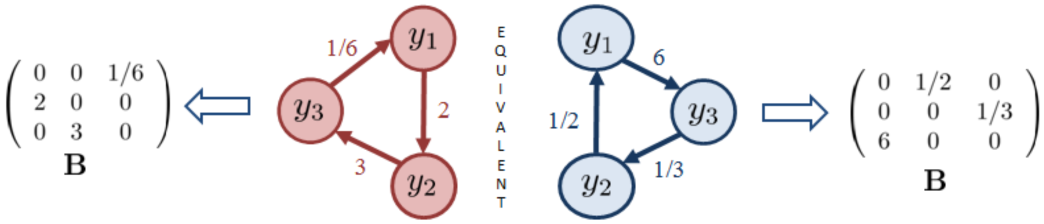

$\begin{pmatrix} 0 & 0 & 1/6 \\ 2 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}$
B
1/6
y_1
y_3
2
3
y_2
EQUIVALENT
y_1
6
1/2
y_3
y_2
1/3
$\begin{pmatrix} 0 & 1/2 & 0 \\ 0 & 0 & 1/3 \\ 6 & 0 & 0 \end{pmatrix}$
B

Figure 2.1: Example of two equivalent networks representing the same data $\mathbf{Y}$ in the absence of perturbations.

The two graphs in figure 2.1 are undistinguishable from expression data only. Indeed, for any matrix $\mathbf{Y}$ such that $\mathbf{Y} = \mathbf{BY}$ is verified for matrix $\mathbf{B}$ at the left in figure 2.1, $\mathbf{Y} = \mathbf{BY}$ is also satisfied for the matrix $\mathbf{B}$ at the right. This corresponds to setting $\mathbf{X}$ to zero, in which case the hypothesis of Proposition 2.1 does not hold.

Proof: Rewriting the SEM as $(\mathbf{I} - \mathbf{B})\mathbf{Y} = \mathbf{F}\mathbf{X}$, and with $\mathbf{X}$ and $\mathbf{F}$ having full row rank, it follows that $(\mathbf{I} - \mathbf{B})\mathbf{Y}$ has full row rank. Hence $(\mathbf{I} - \mathbf{B})$ is invertible, and thus the system of equations $\mathbf{Y} = \mathbf{A}\mathbf{X}$ is solved by $\mathbf{A}^{\star} := (\mathbf{I} - \mathbf{B})^{-1}\mathbf{F}$.

On the other hand, the system $\mathbf{Y} = \mathbf{A}\mathbf{X}$ admits the solution $\mathbf{A} = \mathbf{Y}\mathbf{X}^{\dagger}$, where $\mathbf{X}^{\dagger}$ stands for the Moore-Penrose pseudo-inverse of $\mathbf{X}$. Since $\mathbf{X}$ is full row rank, the solution is unique, hence

$$\mathbf{A}^{\star} = \mathbf{Y}\mathbf{X}^{\dagger} \tag{2.5}$$

The diagonal of elements of $\mathbf{B}$ are null and $\mathbf{F}^{-1}$ is a diagonal matrix, therefore the diagonal entries of $(\mathbf{A}^{\star})^{-1} = \mathbf{F}^{-1}(\mathbf{I} - \mathbf{B})$ coincide with those of $\mathbf{F}^{-1}$, implying

$$\mathbf{F} = \left(\mathrm{Diag}\left[(\mathbf{A}^{\star})^{-1}\right]\right)^{-1} \tag{2.6}$$

In addition $\mathbf{F}^{-1}(\mathbf{A}^{\star})^{-1} = (\mathbf{I} - \mathbf{B})$, so that

$$\mathbf{B} = \mathbf{I} - \mathbf{F}^{-1}(\mathbf{A}^{\star})^{-1}. \tag{2.7}$$

Substituting (2.5) into (2.6) and (2.7), it follows that $\mathbf{B}$ and $\mathbf{F}$ are functions of $\mathbf{Y}\mathbf{X}^{\dagger}$, which proves uniqueness and completes the proof.

2.2 Simulation studies and performance of inference algorithms

In their simulation studies, Logsdon and Mezey [117] compared the performance of their AL-based algorithm with that of several other algorithms including the PC-algorithm [98, 169], the QDG algorithm [134], the QTLnet algorithm [135], and the NEO algorithm [13]. In two out of four simulation setups, the AL outperformed all other algorithms; and in the other two simulation setups, the AL and QDG algorithms exhibited comparable performance, but consistently outperformed the other two algorithms. Logsdon and Mezey [117] also considered other existing algorithms [113, 115], but these were deemed either computationally too demanding [113] or prohibitively complex [115]. For these reasons, the AL and QDG algorithms are regarded as state-of-the-art in the field. Their performance was compared against this chapter's SML algorithm.

Following the setup of Logsdon and Mezey [117], two types of acyclic gene networks were simulated first: one with 10 genes and another with 30 genes. Specifically, a random DAG of 10 or 30 nodes with an expected $N_e = 3$ edges per node was generated by creating directed edges between two randomly picked nodes. Care was taken to avoid any cycle in the simulated graph. If an edge from node j to node i was emerging, B_{ij} was generated from a random variable uniformly distributed over the interval $(0.5, 1)$ or $(-1, -0.5)$; otherwise, $B_{ij} = 0$. The genotype per eQTL was simulated from an F2 cross. Values 1 and 3 were assigned to two homozygous genotypes, respectively, and 2 to the heterozygous genotype. Hence, X_{ij} was generated as a ternary random variable taking values $\{1, 3, 2\}$ with corresponding probabilities $\{0.25, 0.25, 0.5\}$. Matrix $\mathbf{F}$ was the $N_g \times N_g$ identity matrix, E_{ij} was sampled from a Gaussian distribution with zero mean and variance 10^{-2}, and $\boldsymbol{\mu}$ was set to zero. Finally, $\mathbf{Y}$ was calculated from $\mathbf{Y} = (\mathbf{I} - \mathbf{B})^{-1}(\mathbf{FX} + \mathbf{E})$.

For each type of gene network, 100 realizations or replicates of the network were generated, and then the SML, the AL and the QDG algorithms were run to infer the network topology. When running the SML algorithm, 10-fold CV was employed to determine the optimal values of parameters λ and ρ and then use these values to infer the network. An edge from gene j to i was deemed present if $\hat{B}_{ij} \neq 0$. The AL algorithm also automatically ran using CV to determine the values of its parameters. For 100 replicates of the network, N_t counted the total number of edges, $\hat{N}_t$ denoted the total number of edges detected by the inference algorithm. Among $\hat{N}_t$ detected edges, N_{true} stands for the number of true edges presented in the simulated networks, and N_{false} for the number of false edges. The power of detection (PD) was then found as N_{true}/N_t, and the false discovery rate (FDR) as $N_{false}/\hat{N}_t$. The PD and the FDR of the SML, AL, and QDG algorithms for different sample sizes are depicted in Figure 2.3. It is seen from Figures 2.3(a) and (c) that the PD of the SML algorithm exceeds 0.9 for both networks across all sample sizes, whereas the PD of the AL algorithm is about 0.65 for $N_g = 10$ and 0.35 for $N_g = 30$. The PD of the QDG algorithm is even lower ranging from 0.22 to 0.33. As shown in Figures 2.3(b) and (d) , the FDR of the SML algorithm is on the order of 10^{-3} for most sample sizes, and is much lower than that of the AL and QDG algorithms, which is about 0.3 for $N_g = 10$ and over the range from 0.31 to 0.6 for $N_g = 30$.

Two types of *cyclic* networks were subsequently simulated: one with 10 genes and the other with 30 genes. The average number of edges per gene is again equal to 3. The same procedures used in simulating acyclic networks described earlier were employed, except that DCGs instead of DAGs were simulated. Again, 100 replicates for each type of the networks were randomly generated. The

PD and the FDR of three algorithms are depicted in Figure 2.4. As shown in Figure 2.4(a) and (c) , the PD of the SML algorithm is between 0.83 and 0.9, whereas the PD of the AL algorithm is about 0.52 for $N_g = 10$ and 0.29 for $N_g = 30$, and the PD of the QDG algorithm is between 0.16 and 0.28. As shown in Figures 2.4(b) and (d) , the FDR of the SML algorithm is < 0.01, which is much smaller than that of the AL and QDG algorithms over the range from 0.33 to 0.68. For the convenience of comparison, the results in Figures 2.3 and 2.4 at sample size 500 are summarized in Table 2.1.

As confirmed by Figures 2.3 and 2.4, the SML algorithm offers much better performance in terms of PD and FDR than the AL and QDG algorithms. However, these results were obtained for gene networks of small size. To test performance of the SML algorithm for networks of relatively large size, an acyclic network of 300 genes was simulated with an expected $N_e = 1$ edge per node, and randomly generated 10 replicates of the network. PD and FDR of the SML and AL algorithms obtained from these replicates are depicted in Figure 2.5. The PD of SML exceeds 0.99 across all sample sizes from 100 to 1,000, whereas that of the AL algorithm is about 0.04 for sample sizes from 100 to 500, and gradually increases to 0.42 at the sample size of 1,000. The FDR of SML stays below 10^{-4} for sample sizes from 400 to 1,000, whereas the FDR of the AL algorithm is on the order of 10^{-2} for the same sample size. When the sample size is relatively small (in the range from 100 to 300), the FDR of SML is higher than that of the AL algorithm, but it is still relatively small (< 0.2). Note that the AL algorithm essentially does not work for sample sizes $N \leq 500$, since its power is too small. All simulation results show that the novel SML algorithm significantly outperforms the AL and QDG algorithms in terms of PD and FDR.

An extra set of simulations assessing the stability of SML is described in the section of “Stability of model selection under CV perturbations” in Appendix A. As an alternative to CV, stability selection (STS) [125] provides a means of selecting an appropriate sparsity level to guarantee that the FDR is less than a theoretical upper bound. The STS procedure was applied to the SML algorithm as described in Appendix A, and was used with the selection probability cutoff $\delta = 0.8$ and an upper bound or target FDR=0.1 in simulations for the networks in Figures 2.3[(c) and (d)] and 2.4 [(c) and (d)]. As shown in Figure A.3, the FDR of the STS is indeed much smaller than the target FDR and almost uniform across different sample sizes, but the PD of the STS is smaller than that of CV. In fact, the FDR of the STS is on the same order as that of the CV except at the sample size of 100 for the DAG. As seen from these simulation results, although the STS guarantees a FDR upper bound, this upper bound is loose for the simulation setups tested, which may sacrifice detection power. Nevertheless, the STS procedure can select a set of stable variables as described in [125] and verified by our simulations.

So far, all the simulated data were generated with noise variance $\sigma^2 = 0.01$. Next, the performance of SML was analyzed for simulated networks of 30 genes, when σ^2 was increased to 0.05 and N_e was changed from 3 to 1 or 5. Reducing N_e from 3 to 1 improved the performance of SML for most of the sample sizes, as it is depicted in Figure 2.6, withstanding the increase in the noise variance. Increasing N_e at constant σ^2, or increasing σ^2 at constant N_e degraded the performance, most notably in the later case. Comparing Figure 2.6 with Figures 2.3 and 2.4 [(c) and (d)] demonstrates that in both cases the SML estimates still achieve higher detection power and lower FDR than those estimates obtained with the AL algorithm for $N_e = 3$ and $\sigma^2 = 0.01$.

2.3 Inference of a network of immune-related human genes

Pickrell *et al.* [143] used RNA-Seq technology to sequence RNA from 69 lymphoblastoid cell lines derived from unrelated Nigerian individuals extensively genotyped by the International HapMap Project [69]. For each gene, they evaluated possible associations between its gene expression level calculated from RNA-Seq reads and all 3.8 million single nucleotide polymorphisms (SNPs) using the genotypes from phases II and III of the HapMap Project. At FDR=0.1, they identified 929 genes or putative new exons that have eQTLs within 200kb of the gene or the exon. From these 929 genes, 39 genes that are related to immune functions were selected manually by an expert as mentioned in the Acknowledgements section; expression levels and the genotypes of the eQTLs of these 39 genes in 69 individuals were used to infer the underlying regulatory network.

Pickrell *et al.* normalized expression values using quantile normalization before performing eQTL mapping. They also provided a data set that contains the number of reads mapped to each of 929 genes. This data set was obtained and the number of reads for each of 39 genes was normalized with the length of the gene to yield expression value. Such kind of values may better reflect the real expression values than the values normalized with quantile normalization, and thus they were used to infer the network. To ensure the quality of the data, the SAS ROBUSTREG procedure was applied to 69 expression values of each of 39 genes to detect outliers. The default M estimation method of the ROBUSTREG procedure was employed and the outliers were detected at a significance level of 0.05. Several outliers with values much larger than the remaining values were identified and were replaced with the largest non-outlier since it is closest to the outliers. More sophisticated means of revealing and imputing outliers are possible using robust statistical schemes; see e.g., [78]. The genotypes of the eQTLs of the 39 genes were downloaded from HapMap database using the SNP IDs for the eQTL provided by Pickrell *et al.*. About 12% genotypes are missing. These missing genotypes were imputed using the program IMPUTE2 [93]. The name and a brief description of each gene were obtained from DAVID [94] using the Ensembl gene IDs provided by Pickrell *et al.* Information of these 39 genes including their Ensembl gene IDs and names, a brief description of each gene, and HapMap SNP IDs of the associated eQTLs can be found in Table S1 in Appendix A.

The SML algorithm was run with the expression levels and genotypes of eQTLs of these 39 genes. An edge from gene j to i was detected if $\hat{B}_{ij} \neq 0$. To improve the reliability of the detected edges, the SML algorithm was run with stability selection at an FDR ≤ 0.1 using 100 random subsamples, yielding 13 directional edges as shown in Figure 2.7. The frequency of each edge detected in 100 runs is given in Table A.2. It is interesting to see from Figure 2.7 that only 9 genes are involved in the network, and the remaining 30 genes are not connected with any other genes and thus not shown in the figure. AL and QDG algorithms were also run with stability selection at an FDR ≤ 0.1 using 100 random subsamples. The edges detected by AL and QDG algorithms and their frequencies are included in Table A.2. The AL algorithm detected only one edge that was not detected by the SML algorithm. The QDG yielded 3 edges, one of which was also detected by the SML algorithm. Comparing the results of three algorithms shows that our SML algorithm detected more edges than the other two algorithms at the same FDR due to its higher detection power as confirmed also by the simulations. When the FDR was increased to ≤ 0.3, the SML algorithm with stability selection yielded a network of 16 genes that have 42 edges as shown in Figure A.4 in Appendix A. Since only 39 genes were used to construct the network, an edge between two genes may not necessarily imply a direct regulatory effect, but may reflect the fact that two genes are either directly linked or very close to each other in the real network that consists of all genes. Particularly, if two genes are co-regulated by another gene which is not included in the 39 genes, these two genes may have a unidirectional or bidirectional edge.

Most edges in Figure 2.7 are between major histocompatibility complex (MHC) genes (HLA-A, HLA-DPA1, HLA-DQA2, HLA-DQB1, HLA-DRB4 and HLA-DRB5), which is expected since these genes may interact with each other and/or be co-regulated. FCRLA is a member of Fc receptor-like family of genes. It is expressed in B cells and interacts with IgG and IgM [154, 193]. IGH, encoding the heavy chain of immunoglobulin, characterizes the B-cell origin of the samples. Hence, it is not surprising to see an edge between FCRLA and IGH. Interleukin-4-induced gene 1 (IL4I1) was first described in the mouse [49] and subsequently characterized in human B cells [43]. Human IL4I1 is expressed by antigen-presenting cells [29], which may allude to the edge between HLA-A and IL4I1, but this may be speculative since there is no edges between IL4I1 and MHC class II genes in the network. The edges between IGH and HLA-A and between IGH and HLA-DRB4 may reflect the coordinated effect of antibody and MHC as a response to antigens. In fact, IGH is connected to most of MCH genes in Figure A.4, which may imply the wide coordination between the two classes of molecules.

2.4 Methods

2.4.1 Ridge regression

Closed-form solution: Problem (2.4) can be solved row by row independently in closed form. Let $\mathbf{b}_i^T$, $\tilde{\mathbf{b}}_i^T$, $\mathbf{f}_i^T$, $\tilde{\mathbf{f}}_i^T$ and $\check{\mathbf{y}}_i^T$ denote the ith row of $\mathbf{B}$, $\tilde{\mathbf{B}}$, $\mathbf{F}$, $\tilde{\mathbf{F}}$, and $\tilde{\mathbf{Y}}$, respectively. Then, problem (2.4) is equivalent to the following problem

$$(\tilde{\mathbf{b}}_i, \tilde{\mathbf{f}}_i) = \arg\min_{\mathbf{b}_i,\mathbf{f}_i} \frac{1}{2}\|\check{\mathbf{y}}_i^T - \mathbf{b}_i^T\tilde{\mathbf{Y}} - \mathbf{f}_i^T\tilde{\mathbf{X}}\|_2^2 + \rho\|\mathbf{b}_i\|_2^2$$
$$\text{subject to } \; b_i(i) = 0, \; f_i(k) = 0, \; \forall k \text{ s.t. } (i,k) \in \mathcal{S}_q \tag{2.8}$$

where $b_i(j)$ stands for the jth element of $\mathbf{b}_i$ and $f_i(k)$ denotes the kth element of $\mathbf{f}_i$.

The constraints in (2.8) can be imposed directly by discarding elements of $\mathbf{b}_i$ and $\mathbf{f}_i$ known to be zero. To this end, define an $(N_g - 1) \times 1$ vector $\check{\mathbf{b}}_i := [b_i(1), \ldots, b_i(i-1), b_i(i+1) \ldots, b_i(N_g)]^T$ and a vector $\check{\mathbf{f}}_i$ collecting the entries of $\mathbf{f}_i$ whose indexes are not in $\mathcal{S}_q(i) := \{k \in \mathbb{N} : \; (i,k) \in \mathcal{S}_q\}$.

Let $\bar{\mathbf{b}}_i$ and $\bar{\mathbf{f}}_i$ denote the solution for $\check{\mathbf{b}}_i$ and $\check{\mathbf{f}}_i$, respectively. Similarly, let $\check{\mathbf{Y}}_i$ be a sub-matrix of $\tilde{\mathbf{Y}}$ formed by removing the ith row of $\tilde{\mathbf{Y}}$, and $\check{\mathbf{X}}_i$ collecting those rows of $\tilde{\mathbf{X}}$ whose indexes are not in $\mathcal{S}_q(i)$. Under these definitions, (2.8) is equivalent to

$$(\bar{\mathbf{b}}_i, \bar{\mathbf{f}}_i) = \arg\min_{\check{\mathbf{b}}_i, \check{\mathbf{f}}_i} \frac{1}{2}\|\check{\mathbf{y}}_i - \check{\mathbf{Y}}_i^T \check{\mathbf{b}}_i - \check{\mathbf{X}}_i^T \check{\mathbf{f}}_i\|_2^2 + \rho\|\check{\mathbf{b}}_i\|_2^2. \tag{2.9}$$

Minimizing for $\check{\mathbf{f}}_i$ first, one arrives at

$$\check{\mathbf{f}}_i = \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i \left(\check{\mathbf{y}}_i - \check{\mathbf{Y}}_i \check{\mathbf{b}}_i\right). \tag{2.10}$$

Substituting (2.10) into (2.9) after defining $\mathbf{P}_i := \mathbf{I} - \check{\mathbf{X}}_i^T \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i$, yields

$$\bar{\mathbf{b}}_i = \arg\min_{\check{\mathbf{b}}_i} \frac{1}{2}\|\mathbf{P}_i \check{\mathbf{y}}_i - \mathbf{P}_i \check{\mathbf{Y}}_i^T \check{\mathbf{b}}_i\|_2^2 + \rho\|\check{\mathbf{b}}_i\|_2^2,$$

which is a standard ridge regression problem with solution given by

$$\bar{\mathbf{b}}_i = \left(\check{\mathbf{Y}}_i \mathbf{P}_i \check{\mathbf{Y}}_i^T + \rho\mathbf{I}\right)^{-1} \check{\mathbf{Y}}_i^T \mathbf{P}_i \check{\mathbf{y}}_i. \tag{2.11}$$

Finally, substituting (2.11) into (2.10) yields

$$\bar{\mathbf{f}}_i = \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i \left(\mathbf{I} - \check{\mathbf{Y}}_i \left(\check{\mathbf{Y}}_i \mathbf{P}_i \check{\mathbf{Y}}_i^T + \rho\mathbf{I}\right)^{-1} \check{\mathbf{Y}}_i^T \mathbf{P}_i\right) \check{\mathbf{y}}_i. \tag{2.12}$$

Vectors $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{f}}_i$ are obtained by inserting zeros into $\bar{\mathbf{b}}_i$ and $\bar{\mathbf{f}}_i$ at appropriate positions specified by the constraints in (2.8). Collecting $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{f}}_i$, $i = 1, \ldots, N_g$, yields the solution of (2.4), namely $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$.

Parameter ρ is required to solve (2.4). A K-fold CV scheme is adopted for this purpose with typical choices of $K = 5$ or 10, as suggested in [88]. A detailed description of the CV procedure [88] is given in Appendix A.

2.4.2 ℓ_1-regularized ML method

Coordinate-ascent algorithm: Solving (2.3) is performed by a cyclic block-coordinate ascent iteration. Consider a specific cycle where estimates of $\mathbf{B}$ and $\mathbf{F}$ obtained in the previous cycle are denoted by $\hat{\mathbf{B}}$ and $\hat{\mathbf{F}}$, respectively. The first step of the cycle entails maximizing the objective function in (2.3) w.r.t. $\mathbf{F}$ with $\mathbf{B}$ fixed to $\hat{\mathbf{B}}$, which yields a new estimate of $\mathbf{F}$ denoted as $\hat{\mathbf{F}}^{\text{new}}$. This step coincides with the minimization of the objective function in (2.4) w.r.t. $\mathbf{F}$, which admits a closed-form solution per row given by (2.10). In each of the next $N(N-1)$ steps of the cycle, the objective function in (2.3) is maximized w.r.t. a single entry of $\mathbf{B}$, namely $B_{ij},\ i \neq j$, with the remaining entries of $\mathbf{B}$ equal to the corresponding entries of $\hat{\mathbf{B}}$ and $\mathbf{F} = \hat{\mathbf{F}}^{\text{new}}$. An expression for the new estimate of B_{ij}, $\hat{B}_{ij}^{\text{new}}$ is derived next.

Define matrix $\hat{\mathbf{B}}(B_{ij}) := \hat{\mathbf{B}} + \mathbf{e}_i\mathbf{e}_j^T(B_{ij} - \hat{B}_{ij})$ having all entries equal to those of $\hat{\mathbf{B}}$ except for its (i,j)th entry, which is replaced by the variable B_{ij}, where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the ith and jth canonical vectors in $\mathbb{R}^{Ng}$, respectively. Then, the objective in (2.3) can be written as

$$f_{ij}(B_{ij}) = N\hat{\sigma}^2 \log|\det(\mathbf{I} - \hat{\mathbf{B}}(B_{ij}))| - \frac{1}{2}\|\tilde{\mathbf{Y}} - \hat{\mathbf{B}}(B_{ij})\tilde{\mathbf{Y}} - \hat{\mathbf{F}}^{\text{new}}\tilde{\mathbf{X}}\|_F^2 - \lambda w_{ij}|B_{ij}|. \quad (2.13)$$

Upon re-arranging and discarding constant terms, (2.13) simplifies to

$$g_{ij}(B_{ij}) := N\hat{\sigma}^2 \log|\alpha_0 - c_{ij}B_{ij}| + \alpha_1 B_{ij} - \frac{1}{2}\alpha_2 B_{ij}^2 - \lambda w_{ij}|B_{ij}| \quad (2.14)$$

where c_{ij} denotes the (i,j)th co-factor of matrix $\mathbf{I} - \hat{\mathbf{B}}$, and $\{\alpha_l\}_{l=0}^2$ are defined as

$$\begin{aligned} \alpha_0 &:= \det(\mathbf{I} - \hat{\mathbf{B}}) + c_{ij}\hat{B}_{ij}, \\ \alpha_1 &:= \left[\left(\mathbf{I} - \hat{\mathbf{B}} + \mathbf{e}_i\mathbf{e}_j^T\hat{B}_{ij}\right)\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T - \hat{\mathbf{F}}^{\text{new}}\tilde{\mathbf{X}}\tilde{\mathbf{Y}}^T\right]_{ij} \\ \alpha_2 &:= \|\tilde{\mathbf{Y}}^T\mathbf{e}_j\|_2^2 \end{aligned}$$

with $[\cdot]_{ij}$ representing the $(i,j)^{\text{th}}$ entry of the matrix between brackets. For numerical stability and computational savings, all co-factors $c_{ij},\ j = 1,\dots N_g$, per row can be computed simultaneously by solving $(\mathbf{I} - \hat{\mathbf{B}})\mathbf{c}_i = \mathbf{e}_i$, with $\mathbf{c}_i := [c_{i1},\dots,c_{iN_g}]^T$. After an iteration step is completed and $\hat{B}_{ij}^{\text{new}}$ is computed, $\mathbf{c}_i$ can be updated using the matrix inversion lemma as $\mathbf{c}_i = \mathbf{c}_i/(1+\hat{B}_{ij}^{\text{new}} - \hat{B}_{ij})$ before updating $\hat{B}_{ij} = \hat{B}_{ij}^{\text{new}}$.

A new estimate of B_{ij} is formed by maximizing $g_{ij}(B_{ij})$ in (2.14). To this end, consider two cases with $c_{ij} = 0$ and $c_{ij} \neq 0$. If $c_{ij} = 0$, the logarithmic term can be dropped from (2.14) yielding a standard Lasso problem with solution

$$\hat{B}_{ij}^{\text{new}} = \frac{\text{sign}(\alpha_1)}{\alpha_2}\max\{|\alpha_1| - \lambda w_{ij}, 0\}. \quad (2.15)$$

When $c_{ij} \neq 0$, three hypotheses are tested, namely: i) $B_{ij} > 0$; ii) $B_{ij} = 0$; and, iii) $B_{ij} < 0$. For hypotheses i) and iii), the solution can be found in closed form after equating to zero the derivative of (2.14) w.r.t. B_{ij}. The roots found in both cases have to be tested against the corresponding hypothesis. Then, the surviving roots are grouped with $B_{ij} = 0$ as candidate solutions, and the candidate yielding the maximum $g_{ij}(B_{ij})$ is the new estimate $\hat{B}_{ij}$.

Specifically, under hypothesis i) where $B_{ij} > 0$, the derivative of $g_{ij}(B_{ij})$ in (2.14) takes the form $-N\sigma^2 c_{ij}/(\alpha_0 - c_{ij}B_{ij}) + (\alpha_1 - \lambda w_{ij}) - \alpha_2 B_{ij}$, which upon multiplication with $(\alpha_0 - c_{ij}B_{ij})/c_{ij}$ turns into

$$\begin{aligned} & -N\sigma^2 + \alpha_1\frac{\alpha_0}{c_{ij}} - \lambda w_{ij}\frac{\alpha_0}{c_{ij}} - \left(\alpha_2\frac{\alpha_0}{c_{ij}} + \alpha_1 - \lambda w_{ij}\right)B_{ij} + \alpha_2 B_{ij}^2 \\ & = p_0 - \lambda w_{ij}\frac{\alpha_0}{c_{ij}} - (p_1 - \lambda w_{ij})\,B_{ij} + \alpha_2 B_{ij}^2 \end{aligned} \tag{2.16}$$

under the definitions

$$\begin{aligned} p_0 &:= -N\sigma^2 + \alpha_1\frac{\alpha_0}{c_{ij}} \\ p_1 &:= -\alpha_1 + \alpha_2\frac{\alpha_0}{c_{ij}}. \end{aligned}$$

Consider the equation obtained by setting (2.16) equal to zero. If it has root(s), then they are given by

$$r_{ij}^{+} = \frac{1}{2\alpha_2}\left[p_1 - \lambda w_{ij} \pm \sqrt{(p_1 - \lambda w_{ij})^2 - 4\alpha_2\left(p_0 - \lambda w_{ij}\frac{\alpha_0}{c_{ij}}\right)}\right]. \tag{2.17}$$

Let B_{ij}^{+} stand for the set containing the positive root(s) in (2.17). If the equation does not have a solution, B_{ij}^{+} equals the empty set.

Similarly for hypothesis iii) where $B_{ij} < 0$, setting the derivative of (2.14) equal to zero, one obtains an equation. If this equation has root(s), they are given by

$$r_{ij}^{-} = \frac{1}{2\alpha_2}\left[p_1 + \lambda w_{ij} \pm \sqrt{(p_1 + \lambda w_{ij})^2 - 4\alpha_2\left(p_0 + \lambda w_{ij}\frac{\alpha_0}{c_{ij}}\right)}\right]. \tag{2.18}$$

Let B_{ij}^{-} denote the set containing the negative root(s) in (2.18). If the equation does not have a solution, B_{ij}^{-} becomes the empty set. Considering all three hypotheses, one arrives at

$$\hat{B}_{ij}^{\text{new}} = \underset{B_{ij}\in B_{ij}^{+}\cup B_{ij}^{-}\cup\{0\}}{\arg\max}\; g_{ij}(B_{ij}). \tag{2.19}$$

After a cycle is completed, the algorithm is checked for convergence by verifying whether the inequality $\|\hat{\mathbf{B}} - \hat{\mathbf{B}}^{\text{new}}\|_F^2/\|\mathbf{B}\|_F^2 + \|\hat{\mathbf{F}} - \hat{\mathbf{F}}^{\text{new}}\|_F^2/\|\mathbf{F}\|_F^2 < \varepsilon$ is satisfied, where ε is a prespecified small constant. If yes, the algorithm is stopped and $\hat{\mathbf{B}} = \hat{\mathbf{B}}^{\text{new}}$ and $\hat{\mathbf{F}} = \hat{\mathbf{F}}^{\text{new}}$ are output as the final estimates of $\mathbf{B}$ and $\mathbf{F}$; otherwise, $\hat{\mathbf{B}} = \hat{\mathbf{B}}^{\text{new}}$ and $\hat{\mathbf{F}} = \hat{\mathbf{F}}^{\text{new}}$ and one proceeds to execute the next cycle.

In order to increase the speed of the SML algorithm, the discarding rules proposed for sparse linear regression [65, 179] were adapted to the sparse SEM setup. Given λ, the discarding rules provide a means of computing a matrix $Q(\lambda)$, whose entries determining entries of $\mathbf{B}$ that can be set to zero *a priori* without be updated during the coordinate-ascent iterations. A detailed description of the discarding rules, together with the CV procedure to select the optimal λ, and the expression for the required $\lambda_{\max}$, that is, the minimum value of λ for which the solution to (2.3) is null, are provided in Appendix A.

2.4.3 SML algorithm

The overall SML approach described in the Methods section, including the ridge regression weights, the discarding rules, and the coordinate descent cycle is depicted step-by-step in Algorithm 1. The for-loop starting from line 8 and ending at the last line is the ℓ_1-regularized ML method for computing $\hat{\mathbf{B}}$ and $\hat{\mathbf{F}}$ in (2.3), which comprises the block coordinate ascent algorithm and discarding rules. In our computer program, these lines were written as a subroutine. Since the CV on line 7 needs to solve (2.3), the subroutine is also called on line 3 with λ varying from $\lambda_{\max}$ to $\lambda_{\min} = 10^{-4}\lambda_{\max}$. An additional subroutine implementing ridge regression was written to solve (2.4), and subsequently called on lines 1 and 2.

In Appendix A, three relevant extensions to the SML algorithm are described. First, stability selection [125] is applied to the SML, as an alternative to CV, to select the sparsity level so that the FDR is controlled. Second, the SML is extended to handle heteroscedasticity in the SEM error. Third, the SML is modified to enable inference of unknown eQTLs. In addition, Appendix A gives a description of the state-of-the-art AL-based and QDG algorithms that were considered for comparison with SML.

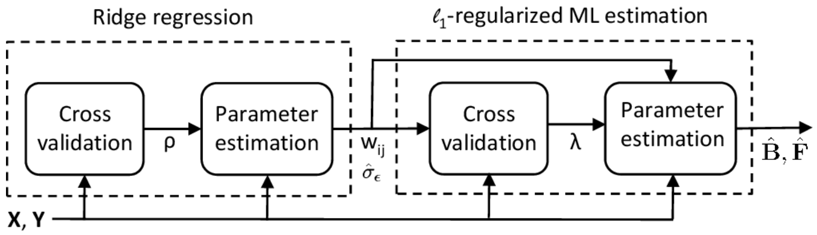


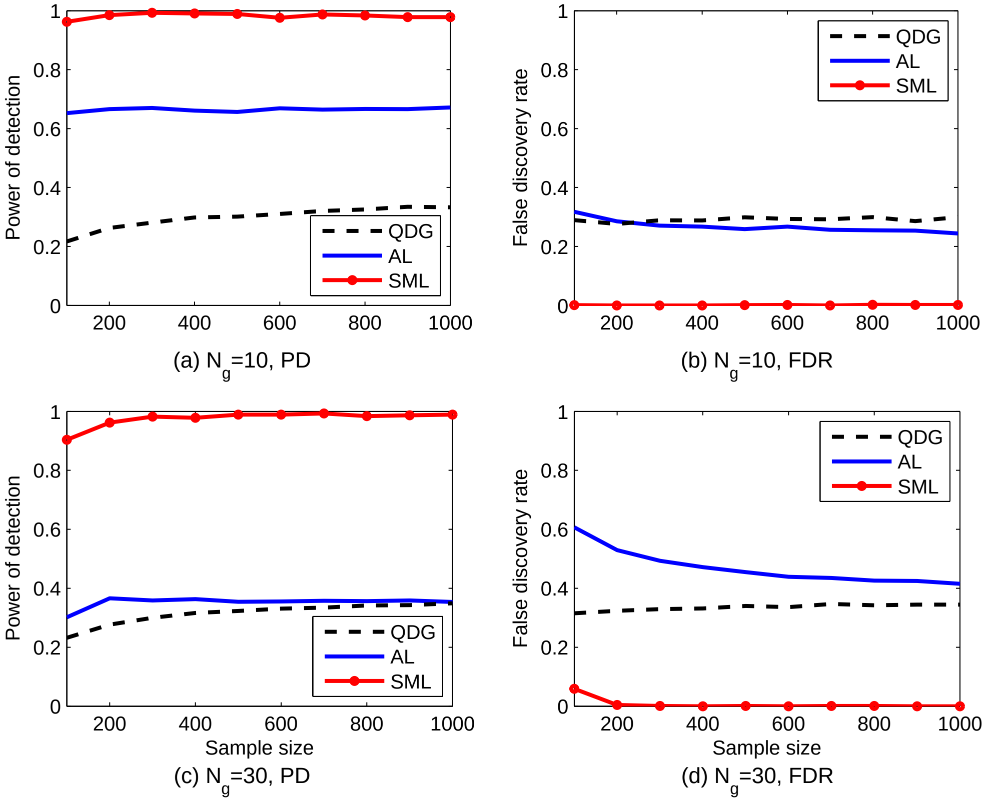
Figure 2.2: Block diagram of the sparsity-aware maximum likelihood (SML) algorithm. The first and third blocks perform cross-validation to select optimal parameters ρ and λ to be used in (2.3) and (2.4), respectively. The third block produces weights $\{w_{ij}\}$ and error-variance estimate $\hat{\sigma}_e^2$ after solving (2.4). Finally, the fourth block takes data $\mathbf{X}$ and $\mathbf{Y}$ together with λ, $\{w_{ij}\}$ and $\hat{\sigma}_e^2$ and solves (2.3) to yield $\hat{\mathbf{B}}$, which represents the SML estimator for $\mathbf{B}$ in (2.1) revealing the genetic-interaction network. A more detailed description of the SML algorithm is given in Algorithm 1 in the Methods section.

Table 2.1: Performance of SML, AL and QDG algorithms. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with a sample size of 500.

Network	N_g	PD			FDR		
		SML	AL	QDG	SML	AL	QDG
DAG	10	0.9887	0.6564	0.3014	0.0007	0.2586	0.2991
	30	0.9891	0.3544	0.3232	0.0010	0.4548	0.3403
DCG	10	0.8872	0.5330	0.2677	0.0067	0.3268	0.3783
	30	0.8931	0.2941	0.2254	0.0020	0.6086	0.5047

2.5 Summary

Integrating genetic perturbations with gene expression data for inference of gene networks not only improves inference accuracy, but also enables learning of causal regulatory relations among genes. Although much progress has been made recently on the development of inference methods that integrate both types of data, a truly efficient algorithm is missing. The SEM provides a systematic framework to integrate both types of data, and offers flexibility to model both directed cyclic as well as acyclic graphs. However, there is no systematically designed inference method for SEMs of



(a) N$_g$=10, PD

(b) N$_g$=10, FDR

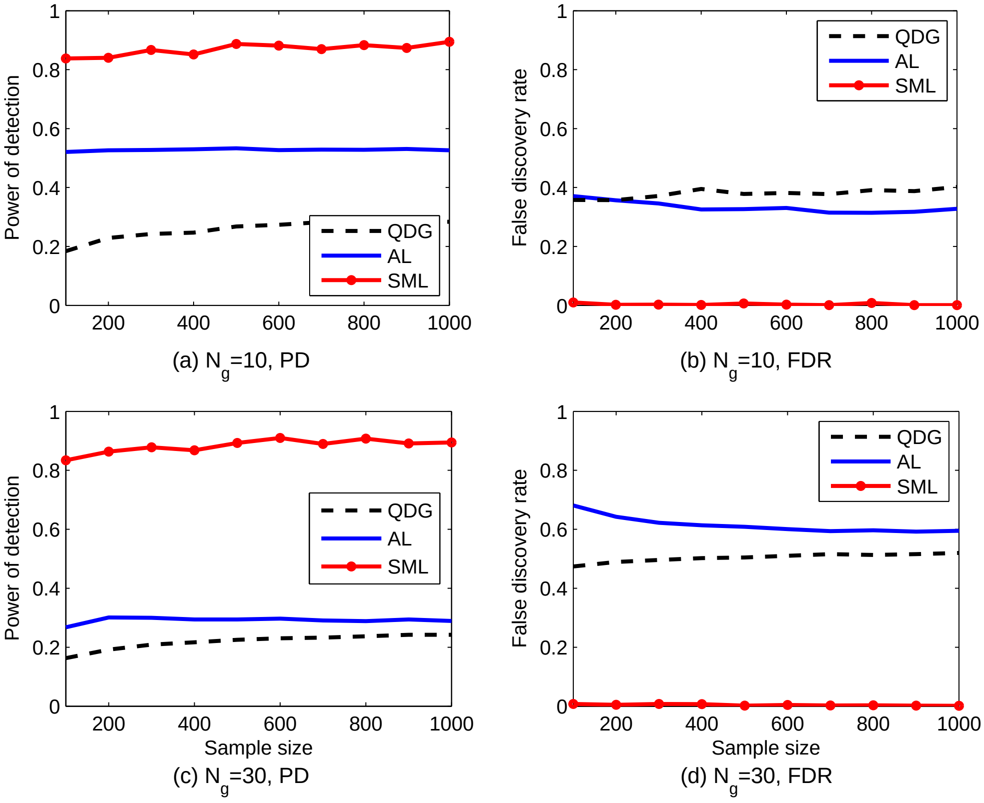
(c) N$_g$=30, PD

(d) N$_g$=30, FDR

Figure 2.3: Performance of SML, AL and QDG algorithms for directed *acyclic* networks of $N_g = 10$ [(a) and (b)] or 30 [(c) and (d)] genes. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).

relatively high dimension, which is particularly true for gene networks typically including hundreds or thousands of genes. Traditionally, inference for SEMs has relied on the ML or generalized least-squares methods implemented with a numerical optimization algorithm [27, 99]; but recently, Bayesian alternatives [109] have emerged too, based on Markov chain Monte Carlo simulations [37, 151]. These methods not only are computationally intensive, but also may be inaccurate for sparse SEMs of relatively high dimension, since they do not account for sparsity present in the model.

In the context of QTL mapping, Newton's method is employed in [126] to implement the ML method, while the genetic algorithm [81, 91] is used in [115, 187] to maximize the likelihood function, and in conjunction with a model selection method using a χ^2 test or Occam's window to search



(a) N_g=10, PD

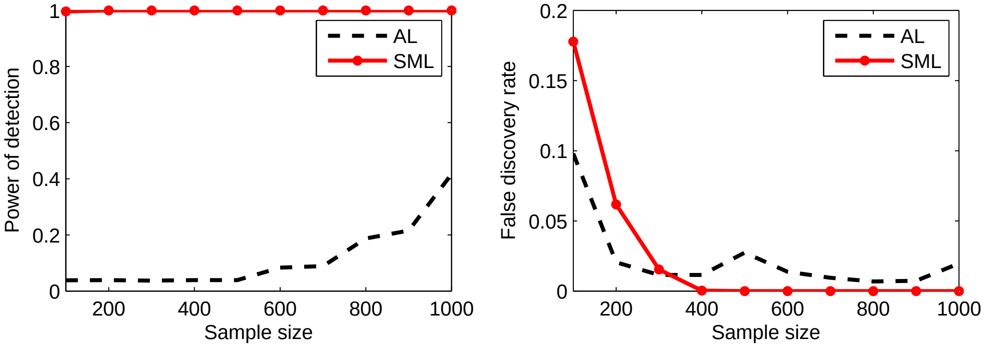
(b) N_g=10, FDR

(c) N_g=30, PD

(d) N_g=30, FDR

Figure 2.4: Performance of SML, AL and QDG algorithms for directed *cyclic* networks of $N_g = 10$ [(a) and (b)] or 30 [(c) and (d)] genes. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).

for the best network topology. These methods are not scalable to SEMs of relatively high dimension. The AL-based algorithm proposed in [117] is more efficient because it automatically incorporates model selection into the inference process, and also takes into account the sparsity present in gene networks. However, the AL-based scheme borrows the adaptive Lasso [201] optimally designed for the linear regression model instead of the SEM. In contrast, the SML algorithm proposed in this chapter directly maximizes the ℓ_1-regularized likelihood function of the SEM, which fully exploits the information present in the data and therefore improves inference accuracy. Moreover, the novel block coordinate ascent method combined with discarding rules can efficiently maximize the ℓ_1-regularized likelihood function, rendering the SML algorithm applicable to SEMs of high dimension. However, unlike the AL-based algorithm, the SML algorithm maximizes a non-convex



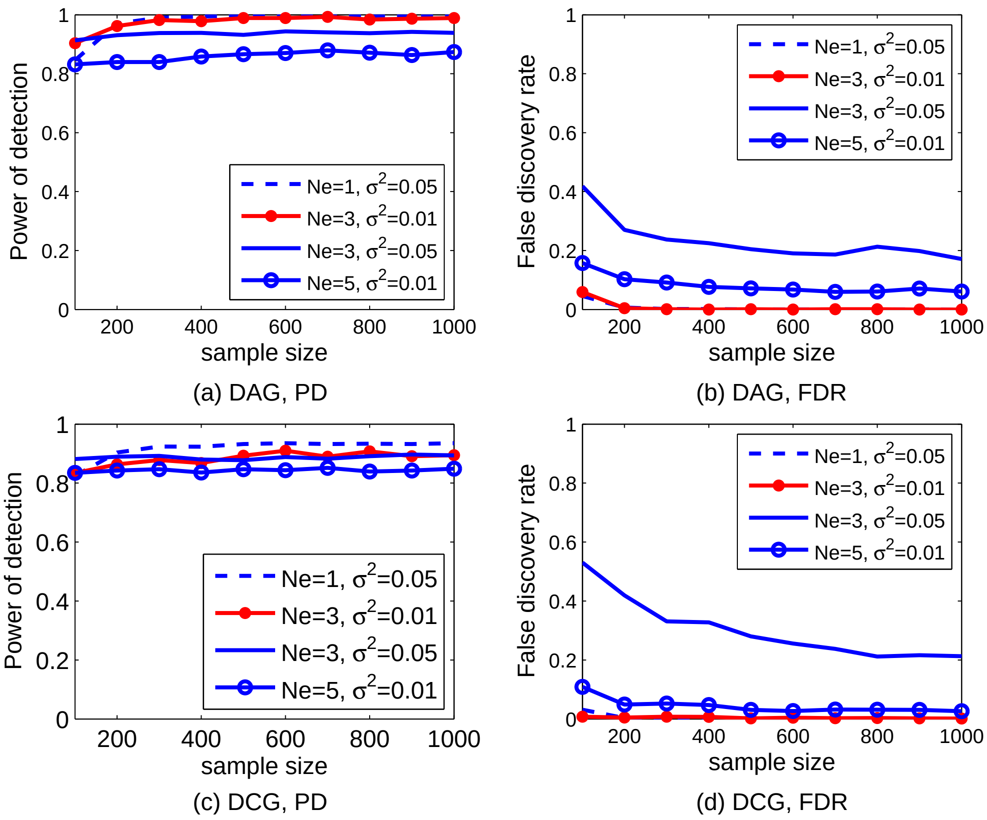
(a) N_g=300, PD

(b) N_g=300, FDR

Figure 2.5: Performance of the SML and AL algorithms for directed *acyclic* networks of $N_g = 300$ genes. Expected number of nodes per node is $N_e = 1$. PD and FDR were obtained from 10 replicates of the network with different sample sizes (N= 100 to 1,000).

objective function as given in (2.3). Although the "Recovery" Theorem in [117] guarantees the identifiability of the network, the algorithm can converge to a local maximum that may not necessarily be coincident with the global maximum corresponding to the optimal network. A common technique for alleviating this problem is to use multiple random initial values. We tested multiple initial values in our simulations and observed that the algorithm converged to the same solution. In Algorithm 1, we used the pathwise coordinate optimization strategy as used in [71], where the solution of (2.3) obtained with λ_i was used as the initial point for the run with $\lambda_{i+1} < \lambda_i$. The pertinence of this strategy is corroborated by simulated numerical tests, showing significant performance gains of the SML algorithm in terms of detection power and FDR when compared to the AL-based algorithm.

Comparisons in the Simulation Studies section, as summarized in Figures 2.3-2.6, demonstrated that the SML algorithm markedly outperforms two state-of-the-art algorithms: the AL [117] and QDG [134] algorithms. For three directed acyclic networks with number of genes $N_g = 10, 30$ and 300, respectively, the PD of the SML algorithm exceeds 0.9 for all sample sizes from 100 to 1,000, and is greater than 0.99 for most sample sizes. This is much greater than the PD of the AL and QDG algorithm that ranges from 0.004 to 0.67. In fact, The QDG algorithm was too time-consuming to obtain results for $N_g = 300$. The FDR of SML is on the order of 10^{-3} for most sample sizes, which is much smaller than those of the AL and QDG algorithms, that are between 0.25 and 0.6



(a) DAG, PD

(b) DAG, FDR

(c) DCG, PD

(d) DCG, FDR

Figure 2.6: Performance of the SML algorithms for DAGs [(a) and (b)] or DCGs [(c) and (d)] of N_g=30 genes with an expected number of nodes per node $N_e \in \{1{,}3{,}5\}$ and error variance $\sigma^2 \in \{0.01{,}0.05\}$. PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).

for $N_g = 10$ and 30. The FDR of the AL algorithm for $N_g = 300$ is between 0.02 and 0.1. The only case where the FDR of SML exceeds that of the AL algorithm is when $N_g = 300$, and the sample size $N < 400$. However, the AL algorithm essentially does not work in this case, since its PD is about 0.04. In the case of directed cyclic networks, all algorithms offer slightly degraded performance when compared to that of directed acyclic networks. However, the SML algorithm still considerably outperforms the AL and QDG algorithms.

Using a limited amount of available data [143], 39 genes related to the immune system and having one eQTL per gene were selected to infer a possible network among these genes. At an FDR ≤10% for the detected edges, a network of 9 out of 39 genes containing 13 edges were obtained. An edge between two genes in the inferred network may be an indication of the direct regulator

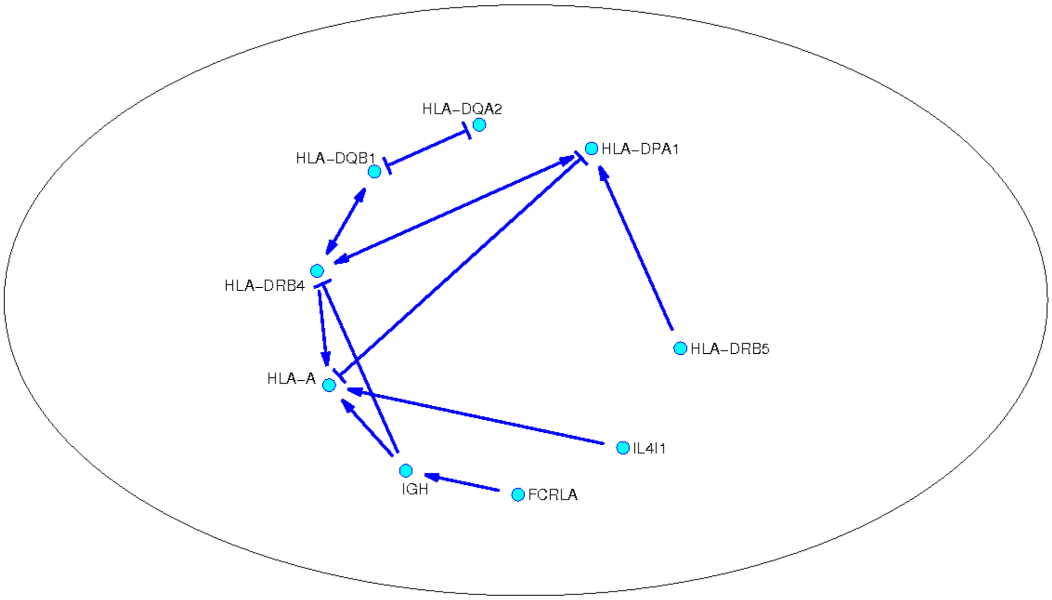


Figure 2.7: The network of 39 human genes inferred from gene expression and eQTL data with the SML algorithm. The 39 genes related to the immune function were chosen from [143] to have a reliable eQTL per gene. The SML algorithm was run with stability selection and edges were detected at an FDR < 0.1. See Table A.1 for the IDs and description of 39 genes. IGH in this figure corresponds to gene ID ENSG00000211897. Pointer $\dashv$ at the edge end stands for inhibitory effect and a $\rightarrow$ edge stands for activating effect.

effect, or indirect interaction or co-regulation mediated by some other genes that are not among the 39 genes. The majority of the edges were reasonably expected from the experimental results in the literature, while the remaining edges may represent new interactions to be elucidated.

Structural equation modeling has a long history of about a century, with well-documented contributions to various fields including biology, psychology, econometrics and other social sciences [27, 99, 140, 161]. The model considered in this chapter belongs to a class of SEMs with observed variables [27]. The SML algorithm is the first one that is systematically developed for inferring sparse SEMs with observed variables. It is expected to accelerate the application of high-dimensional SEMs not only in biology, but also in other fields.

Algorithm 1 : SML

```
Select the optimal value of $\rho$ in (2.4), $\rho_{\text{opt}}$, via cross validation
Solve (2.4) with $\rho_{\text{opt}}$ for $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{B}}$
Estimate $\hat{\sigma}^2$ as the sample variance of E=$\tilde{\mathbf{Y}} - \tilde{\mathbf{B}}\tilde{\mathbf{Y}} - \tilde{\mathbf{F}}\tilde{\mathbf{X}}$
Compute weights $w_{ij} = 1/[\tilde{\mathbf{B}}]_{ij}, i, j = 1, \dots, N_g$
Compute $Q(\lambda_{\max})$ via (A.2) $\forall i, j = 1, \dots, N_g$
Compute $\lambda_{\max}$ via (A.9)
Select the optimal value of $\lambda$, $\lambda_{\text{opt}}$, via cross validation
for $\lambda_l = \lambda_{\max}, \dots, \lambda_{\text{opt}}$ do
    Compute $\mathcal{S}_B(\lambda_l)$ via (A.4)
    Initialize $\hat{\mathbf{B}} = \tilde{\mathbf{B}}$, $\hat{\mathbf{F}} = \tilde{\mathbf{F}}$, $\varepsilon = 10^{-4}$ and $\text{err} = 10$
    while err$>\varepsilon$ do
        for $i = 1,\dots, N_g$ do
            Obtain $\hat{\mathbf{F}}^{\text{new}}$ by computing its row via (2.10) with $\mathbf{b}_i = \hat{\mathbf{b}}_i$
        end for
        for $i = 1,\dots, N_g$ do
            for $j = 1,\dots, N_g$ do
                if $\hat{B}_{ij} \notin \mathcal{S}_B(\lambda_l)$ then
                    Compute cofactor of $\mathbf{I} - \hat{\mathbf{B}}$, $c_{ij}$
                    if $c_{ij} = 0$ then
                        Compute $\hat{B}_{ij}^{\text{new}}$ via (2.15)
                    else
                        Compute $\hat{B}_{ij}^{\text{new}}$ via (2.19)
                    end if
                end if
            end for
        end for
        Compute $\text{err} = \|\hat{\mathbf{B}} - \hat{\mathbf{B}}^{\text{new}}\|_F^2/\|\mathbf{B}\|_F^2 + \|\hat{\mathbf{F}} - \hat{\mathbf{F}}^{\text{new}}\|_F^2/\|\mathbf{F}\|_F^2$
        Set $\hat{\mathbf{B}} = \hat{\mathbf{B}}^{\text{new}}$ and $\hat{\mathbf{F}} = \hat{\mathbf{F}}^{\text{new}}$
    end while
    Compute $Q_{ij}(\lambda_l)$ via (A.1) $\forall i, j = 1, \dots, N_g$
end for
Output $\hat{\mathbf{B}}$ and $\hat{\mathbf{F}}$.
```

Chapter 3

Design of Experiments for Sparse Chemical-Genetic Interactions

Let vector $\mathbf{y} \in \mathbb{R}^{N_c}$ denote the chemical-genetic interaction profile, whose entries represent the fitness phenotypes of N_c single-deletion yeast strains treated with drug d. As described in Fig. 1 (left), $\mathbf{y}$ is compared to the phenotype vector $\mathbf{x}_g \in \mathbb{R}^{N_c}$ of double-deletion mutants, which comes from knocking out gene g together with genes j, where $j = 1, \ldots, N_c$; that is, the gth column of the fitness matrix $\mathbf{X} \in \mathbb{R}^{N_c \times N_g}$ measured over the whole collection of double deletion mutants [52]. If only one gene $g^\star$ is assumed to be the target of d, then $g^\star$ is identified as the gene with profile $\mathbf{x}_{g^\star}$ exhibiting maximum correlation with $\mathbf{y}$ among all candidates $g \in \{1, \ldots, N_g\}$.

A more general setup is desired however, in which drug d can have multiple target genes. Such a model enables the detection of secondary targets, which in turn improves drug-design [54] and facilitates the study of its side effects [31]. Under this general setup, the primary goal is to learn the subset $\mathcal{G} \subset \{1, \ldots, N_g\}$ of gene targets for drug d. This subset selection task is undertaken by assuming that the chemical-genetic profile $\mathbf{y}$ abides to the sparse linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{3.1}$$

where the sparse vector $\boldsymbol{\beta}$ captures the information about target genes, with its entries being nonzero only for those indexes contained in $\mathcal{G}$; and with vector $\mathbf{e}$ accounting for model errors. The subset selection task is thus reduced to fitting (3.1) for a sparse $\hat{\boldsymbol{\beta}}$ that will yield a set $\hat{\mathcal{G}}$ of nonzero entries, revealing the gene targets for drug d.

In the single-target setup, only one entry of $\boldsymbol{\beta}$ is nonzero, and the optimal method to identify such an entry $\beta_{g^\star},\ g^\star \in \{1, \ldots, N_g\}$, in the sense of minimizing the least-squares cost $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$, is to pick the column $\mathbf{x}_{g^\star}$ of matrix $\mathbf{X}$ having the maximum correlation coefficient with profile $\mathbf{y}$. In the multi-target setup, the correlation rule is generalized to pick the columns of $\mathbf{X}$ for which the correlation with $\mathbf{y}$ exceeds a prescribed threshold τ. Thus, letting $\mathbf{r} := \mathbf{X}^T\mathbf{y}$ denote the correlation vector between $\mathbf{X}$ and $\mathbf{y}$, the nonzero entries of $\hat{\boldsymbol{\beta}}$ are obtained as

$$\hat{\beta}_g = \begin{cases} 1, & \text{if } r_g \geq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Remark 1. When $\boldsymbol{\beta}$ has multiple nonzero entries, the correlation rule (3.2) is suboptimal in identifying the nonzero entries of $\boldsymbol{\beta}$. This reason motivates well advanced subset selection methods using e.g., the Least-absolute shrinkage and selection operator (Lasso), which offers finite-sample as well as asymptotic performance guarantees [35]. However, the so-termed restricted isometry assumptions required for the aforementioned guarantees are not satisfied by the matrix $\mathbf{X}$ of double mutant profiles. This happens because genes present a certain degree of redundancy in their functionality to protect the cell [110], which causes the columns of $\mathbf{X}$ to have correlation coefficients as large as 0.8 [52]. The Elastic-Net augments the Lasso cost with a quadratic regularizer, which is particularly useful for sparse estimation when the regression matrix entails highly correlated columns [202]. In particular, it has been proved that Elastic-Net assigns equal values to those entries of $\hat{\boldsymbol{\beta}}$ corresponding to identical columns of $\mathbf{X}$, thus implicitly performing subset selection jointly with clustering. This attribute promotes the Elastic-net as a viable alternative to (3.2) for discovering target genes. However, such an option will not be pursued because the ultimate goal here is to advocate an optimal design of experiments based on a finite-sample estimation performance metric.[1] Instead, the simple closed-form expression of (3.2) will be utilized, since it allows for identifying the a-priori most informative experiments to perform under a constrained budget, as it is described in the ensuing section.

[1] For the Elastic-Net this is an open issue that goes beyond the scope of this work.

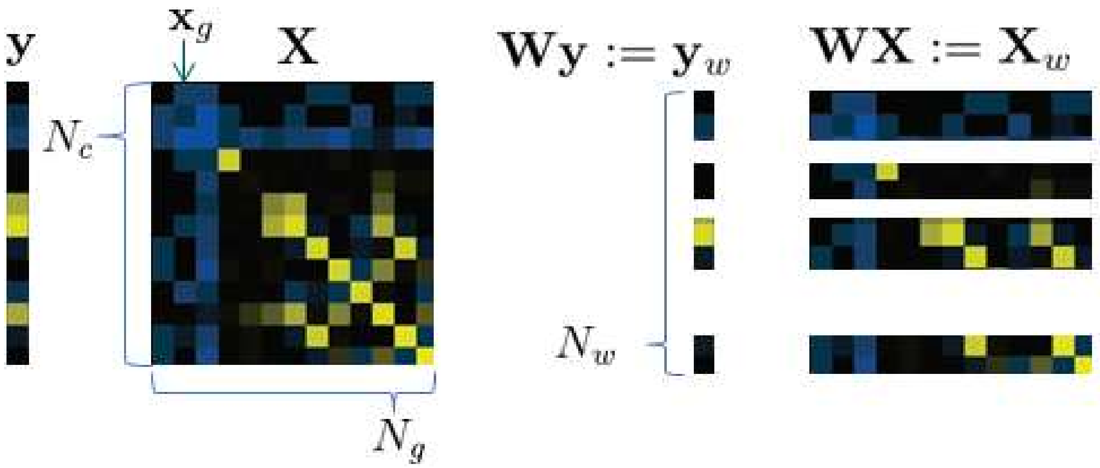

$\mathbf{y}$
$\mathbf{x}_g$
$\mathbf{X}$
N_c
N_g
$\mathbf{W}\mathbf{y} := \mathbf{y}_w$
$\mathbf{W}\mathbf{X} := \mathbf{X}_w$
N_w

Figure 3.1: Design of experiments with the goal of retaining the N_w most informative equations; (left) full system; (right) reduced system obtained after pre-multiplying by matrix $\mathbf{W}$.

3.1 Chemical-genetic design

The goal of this section is to systematically reduce the number of laboratory tests required for estimating $\boldsymbol{\beta}$. As depicted in Fig. 3.1, the specific objective is to select a subset of equations

$$\mathbf{y}_w = \mathbf{X}_w\boldsymbol{\beta} + \mathbf{e}_w \tag{3.3}$$

where $\mathbf{y}_w$ denotes a sub-vector formed with $N_w < N_c$ entries of $\mathbf{y}$, and $\mathbf{X}_w$ the matrix constructed using the corresponding rows of $\mathbf{X}$.

A key observation is that discarding equations in (3.1) can be effected by pre-multiplying both $\mathbf{y}$ and $\mathbf{X}$ with a diagonal matrix $\mathbf{W}$ having binary entries. Indeed, for an index $j \in \{1, \ldots, N_c\}$ with $w_{jj} = 0$, entry y_j and row $\mathbf{x}_j^T$ become irrelevant and are tacitly discarded, as the corresponding equation $w_{jj}y_j = w_{jj}\mathbf{x}_j^T\boldsymbol{\beta} + w_{jj}e_j$ becomes uninformative. Otherwise, those indexes with $w_{jj} = 1$ specify which equations in (3.1) are retained.

Consequently, the design of experiments (3.3) is reformulated as the problem of finding a diagonal binary matrix $\mathbf{W}$ under the criterion introduced next. For a particular $\mathbf{W}$, let $\hat{\boldsymbol{\beta}}_w \in \mathbb{R}^{N_g}$ denote the vector that results after replacing $\mathbf{r}_w := \mathbf{X}_w^T\mathbf{y}_w = \mathbf{X}^T\mathbf{W}\mathbf{y}$ by $\mathbf{r}$ in (3.2). As before, $\mathbf{r} = \mathbf{X}^T\mathbf{y}$ and corresponding $\hat{\boldsymbol{\beta}}$ stand for the vectors that would be obtained if the full data $\mathbf{y}$ and $\mathbf{X}$ were available. With the objective of rendering the support of $\hat{\boldsymbol{\beta}}_w$ as close as possible to that of $\hat{\boldsymbol{\beta}}$, the adopted criterion for selecting $\mathbf{W}$ is to minimize the expected distance between $\mathbf{r}_w$ and $\mathbf{r}$, that is

$$\begin{aligned} \min_{\mathbf{W}\in\{0,1\}^{N_c\times N_c}} & \ E\|\mathbf{r}-\mathbf{r}_w\|_2^2 \\ \text{s. to } & \ \mathbf{r}_w = \mathbf{X}^T\mathbf{W}\mathbf{y} \\ & \ \text{Tr}(\mathbf{W}) = N_w \\ & \ w_{jj'} = 0,\ \forall\, j, j',\ j \neq j' \end{aligned} \tag{3.4}$$

where N_w is the prescribed number of equations to be retained, and the expectation is taken with respect to (w.r.t.) the unknown vectors $\boldsymbol{\beta}$ and $\mathbf{e}$ in the linear model (3.1) for $\mathbf{y}$. Specifically, substituting (3.1) in (3.4) yields

$$\mathbf{r}-\mathbf{r}_w = \mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{X}\boldsymbol{\beta} + \mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{e},$$

and hence

$$\begin{aligned} E_{\beta,e}\|\mathbf{r}-\mathbf{r}_w\|^2 &= E_\beta\|\mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{X}\boldsymbol{\beta}\|^2 \\ &\quad + E_e\|\mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{e}\|^2 \\ &= \sigma_\beta^2\|\mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{X}\|_F^2 \\ &\quad + \sigma_e^2 \text{Tr}\left(\mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{X}\right) \end{aligned} \tag{3.5}$$

where $\|\cdot\|_F$ represents the Frobenius norm of a matrix.

Supposing a high-SNR regime where $\sigma_\beta \gg \sigma_e$, the last term in (3.5) containing the trace can be discarded, and (3.4) becomes

$$\begin{aligned} \min_{\mathbf{W}\in\{0,1\}^{N_c\times N_c}} & \ \|\mathbf{X}^T(\mathbf{I}-\mathbf{W})\mathbf{X}\|_F^2 \\ \text{s. to } & \ \text{Tr}(\mathbf{W}) = N_w \\ & \ w_{jj'} = 0,\ \forall\, j, j',\ j \neq j' \end{aligned} \tag{3.6}$$

Remark 2. The high-SNR regime was adopted to avoid parameter tuning. If instead the ratio $\sigma_e^2/\sigma_\beta^2$ is not negligible but can be estimated, the last term in (3.5) can be reincorporated into (3.6) without major modifications henceforth.

Remark 3. After averaging over the a-priori unavailable profile $\mathbf{y}$, implicitly present in (3.4), then (3.6) becomes dependent only on the data $\mathbf{X}$ at hand, which is necessary for the design of experiments to be of practical interest.

Remark 4. If the Elastic-Net is considered instead of (3.2), then (3.6) remains relevant, since it can be shown that the corresponding Karush-Kuhn-Tucker conditions yield equations expressible in terms of $\mathbf{X}^T\mathbf{W}\mathbf{X}$ (after averaging over $\boldsymbol{\beta}$ and $\mathbf{e}$).

Remark 5. Solving (3.6) for $\mathbf{W}$ determines the sought optimal design. However, the binary constraint in (3.6) renders this quadratic program NP-hard. Thus, a proper relaxation becomes necessary, which is the theme of the ensuing section.

3.2 SDR-based algorithm

Changing variables $\mathbf{Z} = \mathbf{I} - \mathbf{W}$ transforms the cost in (3.6) to $\|\mathbf{X}^T\mathbf{Z}\mathbf{X}\|_F^2$. The latter can be put in the vector form $\|\text{vec}\left(\mathbf{X}^T\mathbf{Z}\mathbf{X}\right)\|_2^2$, where the operator vec$(\cdot)$ concatenates the columns of its argument matrix. This expression can be further simplified by using the following properties of the Khatri-Rao product $\odot$ [116]

$$\text{vec}\left(\mathbf{X}^T\mathbf{Z}\mathbf{X}\right) = (\mathbf{X}^T \odot \mathbf{X}^T)\mathbf{z} \tag{3.7}$$

$$(\mathbf{X}^T \odot \mathbf{X}^T)^T(\mathbf{X}^T \odot \mathbf{X}^T) = (\mathbf{X}\mathbf{X}^T) \star (\mathbf{X}\mathbf{X}^T) \tag{3.8}$$

where $\mathbf{z} \in \mathbb{R}^N$ denotes the diagonal of matrix $\mathbf{Z}$, and $\star$ the Hadamard (entry-wise) product. Using these two properties and defining $\mathbf{Q} := (\mathbf{X}\mathbf{X}^T) \star (\mathbf{X}\mathbf{X}^T)$, problem (3.6) can be equivalently recast as

$$\begin{aligned} \min_{\mathbf{z}\in\{0,1\}^{N_c}} \; & \mathbf{z}^T\mathbf{Q}\mathbf{z} \\ \text{s. to} \; & \sum_{j=1}^{N} z_j = N_c - N_w \; . \end{aligned} \tag{3.9}$$

Yet another equivalent problem results after dropping the explicit binary constraint and implicitly replacing it by the addition of a symmetric, positive, semi-definite matrix variable $\mathbf{Z} \in \mathcal{S}_+$, yielding

$$\begin{aligned}
\min_{\mathbf{Z}\in\mathcal{S}_+,\mathbf{z}\in\mathbb{R}^{N_c}} & \text{Tr}(\mathbf{Q}\mathbf{Z}) \\
\text{s. to } & \mathbf{Z}-\mathbf{z}\mathbf{z}^T \succeq \mathbf{0} \\
& \text{Rank}(\mathbf{Z}) = 1 \\
& \text{Diag}(\mathbf{Z}) = \mathbf{z} \\
& \text{Tr}(\mathbf{Z}) = N_c - N_w \\
& \mathbf{z} \in [0,1]^{N_c}.
\end{aligned} \tag{3.10}$$

The equivalence of (3.10) with (3.9) implies that (3.10) is still NP-hard, and thus justifies the following relaxation consisting in simply dropping the rank constraint

$$\begin{aligned}
\min_{\mathbf{Z}\in\mathcal{S}_+,\mathbf{z}\in\mathbb{R}^{N_c}} & \text{Tr}(\mathbf{Q}\mathbf{Z}) \\
\text{s. to } & \mathbf{Z}-\mathbf{z}\mathbf{z}^T \succeq \mathbf{0} \\
& \text{Diag}(\mathbf{Z}) = \mathbf{z} \\
& \text{Tr}(\mathbf{Z}) = N_c - N_w \\
& \mathbf{z} \in [0,1]^{N_c}.
\end{aligned} \tag{3.11}$$

Problem (3.11) is a semi-definite program that can be solved using standard optimization tools. Semi-definite relaxation provides theoretical guarantees, including a provably smaller gap between the relaxed and the original minimum costs, when compared to the alternative of relaxing (3.6) directly by dropping the binary constraint [118].

Once the solution of (3.11) is obtained, the result is compared against a threshold to obtain the vector $\mathbf{z} \in \{0,1\}^{N_c}$, whose *null* entries indicate the rows to be retained according to (3.4).

3.3 Results

The novel methods are applied to the $D = 82$ chemical-genetic profiles $\mathbf{y}_d$, $d = 1, \ldots, D$ in [138], which are compared to the double mutant fitness profiles $\mathbf{X}$ in [52]. The dimensions of $\mathbf{y}_d$ and $\mathbf{X}$ are $N_c \times 1$ and $N_c \times N_g$, respectively, with $N_c = 2,725$ and $N_g = 1,709$. These data are collected after a preprocessing step, where the number of rows in $\mathbf{y}_d$ and $\mathbf{X}$ is preliminarily reduced to N_c, to keep only those genes both in [138] and [52] data-sets.

Benchmark results are obtained using the full data set, before implementing the design of experiments (3.4). For each test drug $d \in \{1, \ldots, D\}$ in [138], the correlation rule (3.2) was applied to identify $N_\tau = 10$ target genes as those whose double mutant profiles present the 10-largest correlation coefficients with $\mathbf{y}_d$. The resulting target genes per test-drug are tabulated in Appendix B, in those rows of Tables 3.1-3.5 labeled as (100% data.) Names of the target drugs revealed are listed in Tables 3.1-3.5 together with the correlation coefficients r_{dg} between drug d and target gene g.

In order to test the design of experiments achieved through (3.4), its convex approximation (3.11) was applied repeatedly to $\mathbf{X}$ by varying the number of retained rows N_w. For each value of $N_w \in \{200, 500, 1200, 1700, 2200, 2700\}$, the procedure of discarding the noninformative rows specified by the solution of (3.11) resulted in a matrix $\mathbf{X}_w$ of reduced dimensions $N_w \times N_g$. For each of these matrices $\mathbf{X}_w$, and per chemical-genetic drug profile $\mathbf{y}_d$, $d = 1, \ldots, D$, the rule (3.2) was applied to identify the $N_\tau = 10$ target-gene profiles that exhibit the 10-largest correlation coefficients with y_d. The performance of (3.4) is quantified by the relative mismatch between the set of target genes identified from the N_w-long profiles and the benchmark results obtained from the full dataset. Fig. 3.2 depicts this relative mismatch as a function of the retained number of rows N_w, shown with a full red line.

It can be observed from Fig. 3.2, that if a 10% mismatch is affordable (corresponding to the average misclassification of one target gene), then the number of experiments can be reduced by 30% requiring only $N_w = 1,932 < N$ laboratory tests to obtain the shortened chemical-genetic profile $\mathbf{y}_w$. This systematic reduction is further appreciated in Fig. 1, by comparing the red line with the black dashed line, which represents the relative mismatch corresponding to a random design. Discarding 30% of the entries of $\mathbf{y}_d$ uniformly at random yields a mismatch of 45% (cf. 10% for the proposed design (3.11).) Accordingly, only a 4% reduction in the number of laboratory tests can be afforded if these are selected at random, and a 10% mismatch is to be guaranteed (cf. 30% for (3.11).)

Although the design of experiments is applied with the data $\mathbf{X}$ at hand only, the performance comparison in Fig. 3.2 makes use of the benchmark results for which the full profile $\mathbf{y}_d$ is assumed

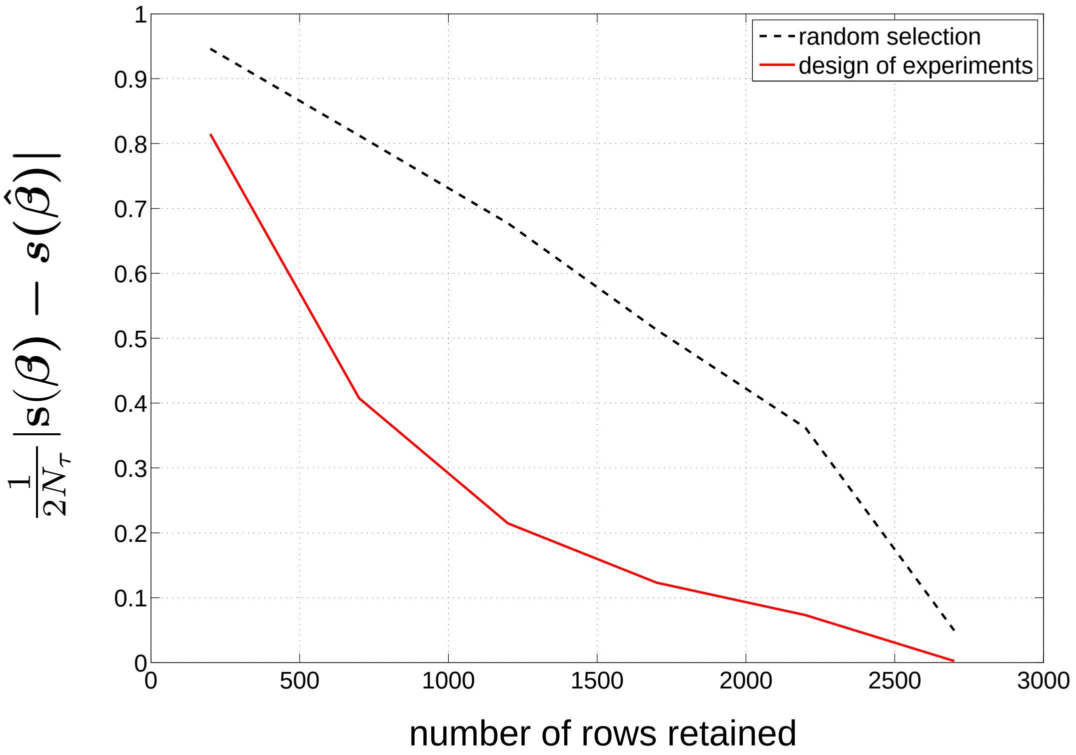


Figure 3.2: Target identification via correlation rule (3.2), support mismatch incurred by the optimal and randomized designs.

available. Alternatively, the expected distance between the benchmark correlation vector $\mathbf{r}$ and the reduced $\mathbf{r}_w$ in (3.5) can be computed in terms of $\mathbf{X}$ and $\mathbf{W}$ only, and serve as an indirect comparison between (3.4) and the randomized design. This expected distance is represented in Fig. 3.3 as a function of N_w, which replicates the pattern in Fig. 3.2.

3.4 Discussion

A sparse linear model was introduced for expressing chemical-genetic interaction profiles in terms of the phenotype expression of yeast double-deletion mutants. This relationship between genetic and chemical-genetic profiles enables identification of multiple target genes in the process of analyzing the mode of action of a new test drug. The main contribution of this work is a novel and systematic design for optimally reducing the number of experiments during acquisition of the

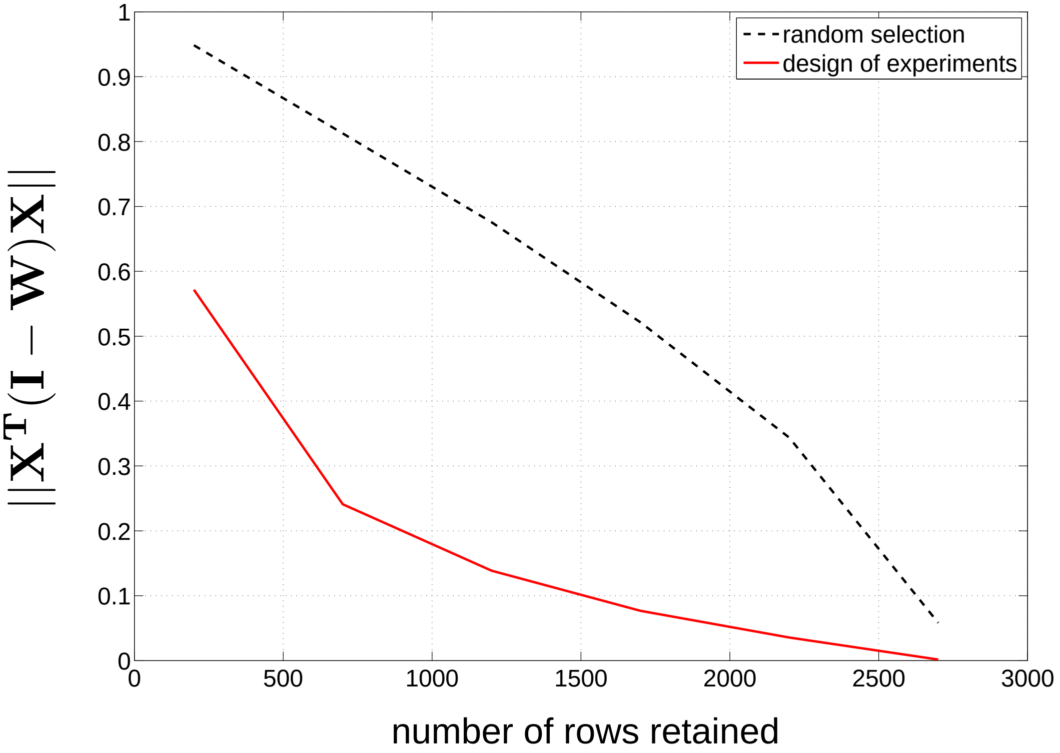


Figure 3.3: Expected distance between correlation vectors resulting from via the optimal randomized designs.

chemical-genetic profile. The proposed design of experiments was tested on real data, and the set of targets identified from the reduced profile were compared with the benchmark results obtained from the full data. It was observed that a 30% reduction in the number of experiments is possible if a single mismatch is admitted over the set of 10 identified target-genes, thus corroborating the performance gains over a randomized design, which can only afford a 4% data reduction at a 10% mismatch. A comprehensive list of 10 target-genes for 82 test-drugs is included, further establishing that the primary target-gene identification is unaltered by the 30% data reduction induced by the proposed design.

Chapter 4

Nonparametric Basis Pursuit via Sparse Kernel-based Learning[†]

Reproducing kernel Hilbert spaces (RKHSs) provide an orderly analytical framework for nonparametric regression, with the optimal kernel-based function estimate emerging as the solution of a regularized variational problem [191]. The pivotal role of RKHS is further appreciated through its connections to "workhorse" signal processing tasks, such as the Nyquist-Shannon sampling and reconstruction result that involves sinc kernels [132]. Alternatively, spline kernels replace sinc kernels, when smoothness rather than bandlimitedness is to be present in the underlying function space [182].

Kernel-based function estimation can be also seen from a Bayesian viewpoint. RKHS and linear minimum mean-square error (LMMSE) function estimators coincide when the pertinent covariance matrix equals the kernel Gram matrix. This equivalence has been leveraged in the context of field estimation, where spatial LMMSE estimation referred to as Kriging, is tantamount to two-dimensional RKHS interpolation [53]. Finally, RKHS based function estimators can linked with Gaussian processes (GPs) obtained upon defining their covariances via kernels [146].

Yet another seemingly unrelated, but increasingly popular theme in contemporary statistical learning and signal processing, is that of matrix completion [68], where data organized in a matrix can have missing entries due to e.g., limitations in the acquisition process. This article builds on the assertion that imputing missing entries amounts to interpolation, as in classical sampling theory, but with the low-rank constraint replacing that of bandlimitedness. From this point of view, RKHS interpolation emerges as the prudent framework for matrix completion that allows effective incorporation of a priori information via kernels [6], including sparsity attributes.

Recent advances in sparse signal recovery and regression motivate a sparse kernel-based learning (KBL) redux, which is the purpose and core of the present chapter. Building blocks of sparse signal processing include the (group) least-absolute shrinkage and selection operator (Lasso) and its weighted versions [88], compressive sampling [36], and nuclear norm regularization [68]. The common denominator behind these operators is the sparsity on a signal's support that the ℓ_1-norm regularizer induces. Exploiting sparsity for KBL leads to several innovations regarding the selection of multiple kernels [105, 127], additive modeling [114, 147], collaborative filtering [6], matrix and tensor completion via dictionary learning [24], as well as nonparametric basis selection [21]. In this context, the main contribution of this chapter is a *nonparametric* basis pursuit (NBP) tool, unifying and advancing a number of *sparse* KBL approaches.

Constrained by space limitations, a sample of applications stemming from such an encompassing analytical tool will be also delineated. Sparse KBL and its various forms contribute to computer vision [162, 185], cognitive radio sensing [21], management of user preferences [6], bioinformatics [167], econometrics [114, 147], and forecasting of electric prices, load, and renewables (e.g., wind speed) [102], to name a few.

The remainder of the chapter is organized as follows. Section II reviews the theory of RKHS in connection with GPs, describing the Representer Theorem and the kernel trick, and presenting the Nyquist-Shannon Theorem (NST) as an example of KBL. Section III deals with sparse KBL including sparse additive models (SpAMs) and multiple kernel learning (MKL) as examples of additive nonparametric models. NBP is introduced in Section IV, with a basis expansion model capturing the general framework for sparse KBL. Blind versions of NBP for matrix completion and dictionary learning are developed in Sections V and VI. Finally, Section VII presents numerical tests using real and simulated data, including RF spectrum measurements, expression levels in yeast, and network traffic loads. Conclusions are drawn in Section VIII, while most technical details are deferred to Appendix C.

4.1 KBL Preliminaries

In this section, basic tools and approaches are reviewed to place known schemes for nonparametric (function) estimation under a common denominator.

4.1.1 RKHS and the Representer Theorem

In the context of reproducing kernel Hilbert spaces (RKHS) [191], nonparametric estimation of a function $f : \mathcal{X} \to \mathbb{R}$ defined over a measurable space $\mathcal{X}$ is performed via interpolation of N training points $\{(x_1, z_1), \ldots, (x_N, z_N)\}$, where $x_n \in \mathcal{X}$, and $z_n = f(x_n) + e_n \in \mathbb{R}$. For this purpose, a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ selected to be *symmetric* and *positive definite,* specifies a linear space of interpolating functions $f(x)$ given by

$$\mathcal{H}_{\mathcal{X}} := \left\{ f(x) = \sum_{n=1}^{\infty} \alpha_n k(x_n, x) : \alpha_n \in \mathbb{R}, x_n \in \mathcal{X}, n \in \mathbb{N} \right\}.$$

For many choices of $k(\cdot,\cdot)$, $\mathcal{H}_{\mathcal{X}}$ is exhaustive with respect to (w.r.t) families of functions obeying certain regularity conditions. The spline kernel for example, generates the Sobolev space of all low-curvature functions [64]. Likewise, the sinc kernel gives rise to the space of bandlimited functions. Space $\mathcal{H}_{\mathcal{X}}$ becomes a Hilbert space when equipped with the inner product $< f, f' >_{\mathcal{H}_{\mathcal{X}}} := \sum_{n,n'=1}^{\infty} \alpha_n \alpha'_{n'} k(x_n, x'_{n'})$, and the associated norm is $\|f\|_{\mathcal{H}_{\mathcal{X}}} := \sqrt{< f, f >_{\mathcal{H}_{\mathcal{X}}}}$. A key result in this context is the so-termed Representer Theorem [191], which asserts that based on $\{(x_n, z_n)\}_{n=1}^{N}$, the optimal interpolator in $\mathcal{H}_{\mathcal{X}}$, in the sense of

$$\hat{f} = \arg \min_{f \in \mathcal{H}_{\mathcal{X}}} \sum_{n=1}^{N} (z_n - f(x_n))^2 + \mu \|f\|_{\mathcal{H}_{\mathcal{X}}}^2 \tag{4.1}$$

admits the finite-dimensional representation

$$\hat{f}(x) = \sum_{n=1}^{N} \alpha_n k(x_n, x). \tag{4.2}$$

This result is nice in its simplicity, since functions in space $\mathcal{H}_{\mathcal{X}}$ are compound by a numerable but arbitrarily large number of kernels, while $\hat{f}$ is a combination of just a *finite* number of kernels around the training points. In addition, the regularizing term $\mu \|f\|_{\mathcal{H}_{\mathcal{X}}}^2$ controls smoothness, and thus reduces overfitting. After substituting (4.2) into (4.1), the coefficients $\boldsymbol{\alpha}^T := [\alpha_1, \ldots, \alpha_N]$

minimizing the regularized least-squares (LS) cost in (4.1) are given by $\boldsymbol{\alpha} = (\mathbf{K} + \mu\mathbf{I})^{-1}\mathbf{z}$, upon recognizing that $\|f\|^2_{\mathcal{H}_\mathcal{X}} := \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$, and defining $\mathbf{z}^T := [z_1, \ldots, z_N]$ as well as the kernel dependent Gram matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ with entries $\mathbf{K}_{n,n'} := k(x_n, x_{n'})$ ($\cdot^T$ stands for transposition).

Remark 1. The finite-dimensional expansion (4.2) solves (4.1) for more general fitting costs and regularizing terms. In its general form, the Representer Theorem asserts that (4.2) is the solution

$$\hat{f} = \arg\min_{f\in\mathcal{H}_\mathcal{X}} \sum_{n=1}^{N} \ell(z_n, f(x_n)) + \mu\Omega(\|f\|_{\mathcal{H}_\mathcal{X}}) \tag{4.3}$$

where the loss function $\ell(z_n, f(x_n))$ replacing the LS cost in (4.1) can be selected to serve either robustness (e.g., using the absolute-value instead of the square error); or, application dependent objectives (e.g., the Hinge loss to serve classification applications); or, for accommodating non-Gaussian noise models when viewing (4.3) from a Bayesian angle. On the other hand, the regularization term can be chosen as any increasing function Ω of the norm $\|f\|_{\mathcal{H}_\mathcal{X}}$, which will turn out to be crucial for introducing the notion of sparsity, as described in the ensuing sections.

4.1.2 LMMSE, Kriging, and GPs

Instead of the deterministic treatment of the previous subsection, the unknown $f(x)$ can be considered as a random process. The KBL estimate (4.2) offered by the Representer Theorem has been linked with the LMMSE-based estimator of random fields $f(x)$, under the term Kriging [53]. To predict the value $\zeta = f(x)$ at an exploration point x via Kriging, the predictor $\hat{f}(x)$ is modeled as a linear combination of noisy samples $z_n := f(x_n) + \eta(x_n)$ at measurement points $\{x_n\}_{n=1}^N$; that is,

$$\hat{f}(x) = \sum_{n=1}^{N} \hat{\beta}_n z_n = \mathbf{z}^T\hat{\boldsymbol{\beta}} \tag{4.4}$$

where $\hat{\boldsymbol{\beta}}^T := [\hat{\beta}_1, \ldots, \hat{\beta}_N]$ are the expansion coefficients, and $\mathbf{z}^T := [z_1, \ldots, z_N]$ collects the data. The MSE criterion is adopted to find the optimal $\hat{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta}} E[f(x) - \mathbf{z}^T\boldsymbol{\beta}]^2$. Solving the latter yields $\hat{\boldsymbol{\beta}} = \mathbf{R}_{\mathbf{zz}}^{-1}\mathbf{r}_{\mathbf{z}\zeta}$, where $\mathbf{R}_{\mathbf{zz}} := E[\mathbf{z}\mathbf{z}^T]$ and $\mathbf{r}_{\mathbf{z}\zeta} := E[\mathbf{z}f(x)]$. If $\eta(x)$ is zero-mean white noise with power σ_η^2, then $\mathbf{R}_{\mathbf{zz}}$ and $\mathbf{r}_{\mathbf{z}\zeta}$ can be expressed in terms of the unobserved $\boldsymbol{\zeta}^T := [f(x_1), \ldots, f(x_N)]$ as $\mathbf{R}_{\mathbf{zz}} = \mathbf{R}_{\boldsymbol{\zeta}\boldsymbol{\zeta}} + \sigma_\eta^2\mathbf{I}$, where $\mathbf{R}_{\boldsymbol{\zeta}\boldsymbol{\zeta}} := E[\boldsymbol{\zeta}\boldsymbol{\zeta}^T]$, and $\mathbf{r}_{\mathbf{z}\zeta} = \mathbf{r}_{\boldsymbol{\zeta}\zeta}$, with $\mathbf{r}_{\boldsymbol{\zeta}\zeta} := E[\boldsymbol{\zeta}f(x)]$. Hence, the LMMSE estimate in (4.4) takes the form

$$\hat{f}(x) = \mathbf{z}^T(\mathbf{R}_{\boldsymbol{\zeta}\boldsymbol{\zeta}} + \sigma_\eta^2\mathbf{I})^{-1}\mathbf{r}_{\boldsymbol{\zeta}\zeta} = \sum_{n=1}^{N} \alpha_n r(x, x_n) \tag{4.5}$$

where $\boldsymbol{\alpha}^T := \mathbf{z}^T(\mathbf{R}_{\zeta\zeta} + \sigma_\eta^2\mathbf{I})^{-1}$, and the n-th entry of $\mathbf{r}_{\zeta\zeta}$, denoted by $r(x_n, x) := E[f(x)f(x_n)]$, is indeed a function of the exploration point x, and the measurement point x_n.

With the Kriging estimate given by (4.5), the RKHS and LMMSE estimates coincide when the kernel in (4.2) is chosen equal to the covariance function $r(x, x')$ in (4.5).

The linearity assumption in (4.4) is unnecessary when $f(x)$ and $e(x)$ are modeled as zero-mean GPs [146]. GPs are those in which instances of the field at arbitrary points are jointly Gaussian. Zero-mean GPs are specified by cov$(x, x') := E[f(x)f(x')]$, which determines the covariance matrix of any vector comprising instances of the field, and thus its specific zero-mean Gaussian distribution. In particular, the vector $\bar{\boldsymbol{\zeta}}^T := [f(x), f(x_1), \ldots, f(x_N)]$ collecting the field at the exploration and measurement points is Gaussian, and so is the vector $\bar{\mathbf{z}}^T := [f(x), f(x_1) + \eta(x_1), \ldots, f(x_N) + \eta(x_N)] = [\zeta, \mathbf{z}^T]$. Hence, the MMSE estimator, given by the expectation of $f(x)$ conditioned on $\mathbf{z}$, reduces to [100]

$$\hat{f}(x) = E(f(x)|\mathbf{z}) = \mathbf{z}^T\mathbf{R}_{\mathbf{z}\mathbf{z}}^{-1}\mathbf{r}_{\mathbf{z}\zeta}^T = \sum_{n=1}^{N} \alpha_n \text{cov}(x_n, x). \tag{4.6}$$

By comparing (4.6) with (4.5), one deduces that the MMSE estimator of a GP coincides with the LMMSE estimator, hence with the RKHS estimator, when cov$(x, x') = k(x, x')$.

4.1.3 The kernel trick

Analogous to the spectral decomposition of matrices, Mercer's Theorem establishes that if the symmetric positive definite kernel is square-integrable, it admits a possibly infinite eigenfunction decomposition $k(x, x') = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(x')$ [191], with $< e_i(x), e_{i'}(x) >_{\mathcal{H}_\mathcal{X}} = \delta_{i-i'}$ where δ_i stands for Kronecker's delta. Using the weighted eigenfunctions $\phi_i(x) := \sqrt{\lambda_i} e_i(x),\ i \in \mathbb{N}$, a point $x \in \mathcal{X}$ can be mapped to a vector (sequence) $\boldsymbol{\phi} \in \mathbb{R}^\infty$ such that $\phi_i = \phi_i(x),\ i \in \mathbb{N}$. This mapping interprets a kernel as an inner product in $\mathbb{R}^\infty$, since for two points $x, x' \in \mathcal{X}$, $k(x, x') = \sum_{i=1}^{\infty} \phi_i(x)\phi_i(x') := \boldsymbol{\phi}^T(x)\boldsymbol{\phi}(x')$. Such an inner product interpretation forms the basis for the *"kernel trick."*

The kernel trick allows for approaches that depend on inner products of functions (given by infinite kernel expansions) to be recast and implemented using finite dimensional covariance (kernel) matrices. A simple demonstration of this valuable property can be provided through kernel-based ridge regression. Starting from the standard ridge estimator $\hat{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^D} \sum_{n=1}^{N}(z_n - \boldsymbol{\phi}_n^T\boldsymbol{\beta})^2 + \mu\|\boldsymbol{\beta}\|^2$ for $\boldsymbol{\phi}_n \in \mathbb{R}^D$, and $\boldsymbol{\Phi} := [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$, it is possible to rewrite and solve $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^D} \|\mathbf{z} - \boldsymbol{\Phi}^T\boldsymbol{\beta}\|^2 + \mu\|\boldsymbol{\beta}\|^2 = (\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \mu\mathbf{I})^{-1}\boldsymbol{\Phi}\mathbf{z}$. After $\hat{\boldsymbol{\beta}}$ is obtained in the training phase, it can be used for prediction of an ensuing $\hat{z}_{N+1} = \boldsymbol{\phi}_{N+1}^T\hat{\boldsymbol{\beta}}$ given $\boldsymbol{\phi}_{N+1}$. By using the matrix inversion lemma, $\hat{z}_{N+1}$ can be written as $\hat{z}_{N+1} = (1/\mu)\boldsymbol{\phi}_{N+1}^T\boldsymbol{\Phi}\mathbf{z} - (1/\mu)\boldsymbol{\phi}_{N+1}^T\boldsymbol{\Phi}(\mu\boldsymbol{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{z}$.

Now, if $\boldsymbol{\phi}_n = \boldsymbol{\phi}(x_n)$ with $D = \infty$ is constructed from $x_n \in \mathcal{X}$ using eigenfunctions $\{\phi_i(x_n)\}_{i=1}^{\infty}$, then $\boldsymbol{\phi}_{N+1}^T\boldsymbol{\Phi} = \mathbf{k}^T(x_{N+1}) := [k(x_{N+1}, x_1), \ldots, k(x_{N+1}, x_N)]$, and $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \mathbf{K}$, which yields

$$\begin{aligned}\hat{z}_{N+1} &= (1/\mu)\mathbf{k}^T(x_{N+1})[\mathbf{I} - (\mu\boldsymbol{I} + \mathbf{K})^{-1}\mathbf{K}]\mathbf{z} \\ &= \mathbf{k}^T(x_{N+1})(\mu\boldsymbol{I} + \mathbf{K})^{-1}\mathbf{z} \qquad (4.7)\end{aligned}$$

coinciding with (4.6), (4.5), and with the solution of (4.1).

Expressing a linear predictor in terms of inner products only is instrumental for mapping it into its kernel-based version. Although the mapping entails the eigenfunctions $\{\phi_i(x)\}$, these are not explicitly present in (4.7), which is given solely in terms of $k(x, x')$. This is crucial since $\boldsymbol{\phi}$ can be infinite dimensional which would render the method computationally intractable, and more importantly the explicit form of $\phi_i(x)$ may not be available. Use of kernel trick was demonstrated in the context of ridge regression. However, the trick can be used in any vectorial regression or classification method whose result can be expressed in terms of inner products only. One such example is offered by support vector machines, which find a kernel-based version of the optimal linear classifier in the sense of minimizing Vapnik's ϵ-insensitive Hinge loss function, and can be shown equivalent to the Lasso [80].

In a nutshell, the kernel trick provides a means of designing KBL algorithms, both for nonparametric function estimation [cf. (4.1)], as well as for classification.

4.1.4 KBL vis à vis Nyquist-Shannon Theorem

Kernels can be clearly viewed as interpolating bases [cf. (4.2)]. This viewpoint can be further appreciated if one considers the family of bandlimited functions $\mathcal{B}_\pi := \{f \in \mathcal{L}^2(\mathcal{X}) : \int f(x)e^{-i\omega x}dx = 0, \ \forall|\omega| > \pi\}$, where $\mathcal{L}^2$ denotes the class of square-integrable functions defined over $\mathcal{X} = \mathbb{R}$ (e.g., continuous-time, finite-power signals). The family $\mathcal{B}_\pi$ constitutes a linear space. Moreover, any $f \in \mathcal{B}_\pi$ can be generated as the linear combination (span) of sinc functions; that is, $f(x) = \sum_{n\in\mathbb{Z}} f(n)\text{sinc}(x-n)$. This is the cornerstone of signal processing, namely the NST for sampling and reconstruction, but can be viewed also under the lens of RKHS with $k(x, x') = \text{sinc}(x - x')$ as a reproducing kernel [132]. The following properties (which are proved in Appendix C) elaborate further on this connection.

P1. The sinc-kernel Gram matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ satisfies $\mathbf{K} \succeq \mathbf{0}$.
P2. The sinc kernel decomposes over orthonormal eigenfunctions $\{\phi_n(x) = \text{sinc}(x - n),\ n \in \mathbb{Z}\}$.
P3. The RKHS norm is $\|f\|^2_{\mathcal{H}_\mathcal{X}} = \int f^2(x)dx$.

P1 states that $\text{sinc}(x - x')$ qualifies as a kernel, while P2 characterizes the eigenfunctions used in the kernel trick, and P3 shows that the RKHS norm is the restriction of the $\mathcal{L}^2$ norm to $\mathcal{B}_\pi$.

P1-P3 establish that the space of bandlimited functions $\mathcal{B}_\pi$ is indeed an RKHS. Any $f \in \mathcal{B}_\pi$ can thus be decomposed as a numerable combination of eigenfunctions, where the coefficients and eigenfunctions obey the NST. Consequently, existence of eigenfunctions $\{\phi_n(x)\}$ spanning $\mathcal{B}_\pi$ is a direct consequence of $\mathcal{B}_\pi$ being a RKHS, and does not require the NST unless an explicit form for $\phi_n(x)$ is desired. Finally, strict adherence to NST requires an infinite number of samples to reconstruct $f \in \mathcal{B}_\pi$. Alternatively, the Representer Theorem fits $f \in \mathcal{B}_\pi$ to a finite set of (possibly noisy) samples by regularizing the power of f.

4.2 Sparse additive nonparametric modeling

The account of sparse KBL methods begins with SpAMs and MKL approaches. Both model the function to be learned as a sparse sum of nonparametric components, and both rely on group Lasso to find it. The additive models considered in this section will naturally lend themselves to the general model for NBP introduced in Section IV, and used henceforth.

4.2.1 SpAMs for High-Dimensional Models

Additive function models offer a generalization of linear regression to the nonparametric setup, on the premise of dealing with *the curse of dimensionality,* which is inherent to learning from high dimensional data [88].

Consider learning a multivariate function $f : \mathcal{X} \to \mathbb{R}$ defined over the Cartesian product $\mathcal{X} := \mathcal{X}_1 \otimes \ldots \otimes \mathcal{X}_P$ of measurable spaces $\mathcal{X}_i$. Let $\mathbf{x}^T := [x_1, \ldots, x_P]$ denote a point in $\mathcal{X}$, k_i the kernel defined over $\mathcal{X}_i \times \mathcal{X}_i$, and $\mathcal{H}_i$ its associated RKHS. Although $f(\mathbf{x})$ can be interpolated from data via (4.1) after substituting $\mathbf{x}$ for x, the fidelity of (4.2) is severely degraded in high dimensions. Indeed, the accuracy of (4.2) depends on the availability of nearby points $\mathbf{x}_n$, where the function is fit to the (possibly noisy) data z_n. But proximity of points $\mathbf{x}_n$ in high dimensions is challenged by the curse of dimensionality, demanding an excessively large dataset. For instance, consider positioning N datapoints randomly in the hypercube $[0,1]^P$, repeatedly for P growing unbounded and N constant. Then $\lim_{P\to\infty} \min_{n \neq n'} \mathbf{E}\|\mathbf{x}_n - \mathbf{x}_{n'}\| = 1$; that is, the expected distance between any two points is equal to the side of the hypercube [88].

To overcome this problem, an additional modeling assumption is well motivated, namely constraining $f(\mathbf{x})$ to the family of separable functions of the form

$$f(\mathbf{x}) = \sum_{i=1}^{P} c_i(x_i) \tag{4.8}$$

with $c_i \in \mathcal{H}_i$ depending only on the i-th entry of $\mathbf{x}$, as in e.g., linear regression models $f_{\text{linear}}(\mathbf{x}) := \sum_{i=1}^{P} \beta_i x_i$. With $f(\mathbf{x})$ separable as in (4.8), the interpolation task is split into P one-dimensional problems that are not affected by the curse of dimensionality.

The additive form in (4.8) is also amenable to subsect selection, which yields a SpAM. As in sparse linear regression, SpAMs involve functions f in (4.8) that can be expressed using only a few entries of $\mathbf{x}$. Those can be learned using a variational version of the Lasso given by [147]

$$\hat{f} = \arg \min_{f \in \mathcal{F}_P} \frac{1}{2} \sum_{n=1}^{N} (z_n - f(\mathbf{x}_n))^2 + \mu \sum_{i=1}^{P} \|c_i\|_{\mathcal{H}_i} \tag{4.9}$$

where $\mathcal{F}_P := \{f : \mathcal{X} \to \mathbb{R} : \ f(\mathbf{x}) = \sum_{i=1}^{P} c_i(x_i)\}$.

With x_{ni} denoting the ith entry of $\mathbf{x}_n$, the Representer Theorem (4.3) can be applied per component $c_i(x_i)$ in (4.9), yielding kernel expansions $\hat{c}_i(x_i) = \sum_{n=1}^{N} \gamma_{ni} k_i(x_{ni}, x_i)$ with scalar coefficients $\{\gamma_{ni}, \ i = 1, \ldots, P, \ n = 1, \ldots, N\}$. The fact that (4.9) yields a SpAM is demonstrated by substituting these expansions back into (4.9) and solving for $\boldsymbol{\gamma}_i^T := [\gamma_{i1}, \ldots, \gamma_{iN}]$, to obtain

$$\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^{P} = \arg \min_{\{\boldsymbol{\gamma}_i\}_{i=1}^{P}} \frac{1}{2} \left\| \mathbf{z} - \textstyle\sum_{i=1}^{P} \mathbf{K}_i \boldsymbol{\gamma}_i \right\|_2^2 + \mu \sum_{i=1}^{P} \|\boldsymbol{\gamma}_i\|_{\mathbf{K}_i} \tag{4.10}$$

where $\mathbf{K}_i$ is the Gram matrix associated with kernel k_i, and $\|\cdot\|_{\mathbf{K}_i}$ denotes the weighted ℓ_2-norm $\|\gamma_i\|_{\mathbf{K}_i} := (\gamma_i^T \mathbf{K}_i \gamma_i)^{1/2}$.

4.2.2 Nonparametric Lasso

Problem (4.10) constitutes a weighted version of the group Lasso formulation for sparse linear regression. Its solution can be found either via block coordinate descent (BCD) [147], or by substituting $\gamma_i' = \mathbf{K}_i^{1/2}\gamma_i$ and applying the alternating-direction method of multipliers (ADMM) [21], with convergence guaranteed by its convexity and the separable structure of the its non-differentiable term [181]. In any case, group Lasso regularizes sub-vectors $\boldsymbol{\gamma}_i$ separately, effecting group-sparsity in the estimates; that is, some of the vectors $\hat{\gamma}_i$ in (4.10) end up being identically zero. To gain intuition on this, (4.10) can be rewritten using the change of variables $\mathbf{K}_i^{1/2}\boldsymbol{\gamma}_i = t_i\mathbf{u}_i$, with $t_i \geq 0$ and $\|\mathbf{u}_i\| = 1$. It will be argued that if μ exceeds a threshold, then the optimal t_i and thus $\hat{\gamma}_i$ will be null. Focusing on the minimization of (4.10) w.r.t. a particular sub-vector $\boldsymbol{\gamma}_i$, as in a BCD algorithm, the substitute variables t_i and $\mathbf{u}_i$ should minimize

$$\frac{1}{2}\left\|\mathbf{z}_i - \mathbf{K}_i^{1/2} t_i \mathbf{u}_i\right\|_2^2 + \mu t_i \tag{4.11}$$

where $\mathbf{z}_i := \mathbf{z} - \sum_{j\neq i}\mathbf{K}_j\boldsymbol{\gamma}_j$. Minimizing (4.11) over t_i is a convex univariate problem whose solution lies either at the border of the constraint, or, at a stationary point; that is,

$$t_i = \max\left\{0, \frac{\mathbf{z}_i^T\mathbf{K}_i^{1/2}\mathbf{u}_i - \mu}{\mathbf{u}_i^T\mathbf{K}_i\mathbf{u}_i}\right\}. \tag{4.12}$$

The Cauchy-Schwarz inequality implies that $\mathbf{z}_i^T\mathbf{K}_i^{1/2}\mathbf{u}_i \leq \|\mathbf{K}_i^{1/2}\mathbf{z}_i\|$ holds for any $\mathbf{u}_i$ with $\|\mathbf{u}_i\| = 1$. Hence, it follows from (4.12) that if $\mu \geq \|\mathbf{K}_i^{1/2}\mathbf{z}_i\|$, then $t_i = 0$, and thus $\boldsymbol{\gamma_i} = \mathbf{0}$.

The sparsifying effect of (4.9) on the additive model (4.8) is now revealed. If μ is selected large enough, some of the optimal sub-vectors $\hat{\gamma}_i$ will be null, and the corresponding functions $\hat{c}_i(x_i) = \sum_{n=1}^{N}\hat{\gamma}_{ni}k(x_{ni}, x_i)$ will be identically zero in (4.8). Thus, estimation via (4.9) provides a nonparametric counterpart of Lasso, offering the flexibility of selecting the most informative component-function regressors in the additive model.

The separable structure postulated in (4.8) facilitates subset selection in the nonparametric setup, and mitigates the problem of interpolating scattered data in high dimensions. However, such a model reduction may render (4.8) inaccurate, in which case extra components depending on two or more variables can be added, turning (4.8) into the ANOVA model [114].

4.2.3 Multi-Kernel Learning

Specifying the kernel that "shapes" $\mathcal{H}_{\mathcal{X}}$, and thus judiciously determines $\hat{f}$ in (4.1) is a prerequisite for KBL. Different candidate kernels $k_1, \ldots, k_P$ would produce different function estimates. Convex combinations can be also employed in (4.1), since elements of the convex hull $\mathcal{K} := \{k = \sum_{i=1}^{P} a_i k_i,\ a_i \geq 0,\ \sum_{i=1}^{P} a_i = 1\}$ conserve the defining properties of kernels.

A data-driven strategy to select "the best" $k \in \mathcal{K}$ is to incorporate the kernel as a variable in (4.3), that is [105]

$$\hat{f} = \arg \min_{k \in \mathcal{K}, f \in \mathcal{H}_{\mathcal{X}}^k} \sum_{n=1}^{N} (z_n - f(x_n))^2 + \mu \|f\|_{\mathcal{H}_{\mathcal{X}}^k} \tag{4.13}$$

where the notation $\mathcal{H}_{\mathcal{X}}^k$ emphasizes dependence on k.

Then, the following Lemma brings MKL to the ambit of sparse additive nonparametric models.

Lemma 4.1 ([127]) *Let $\{k_1, \ldots, k_P\}$ be a set of kernels and k an element of their convex hull $\mathcal{K}$. Denote by $\mathcal{H}_i$ and $\mathcal{H}_{\mathcal{X}}^k$ the RKHSs corresponding to k_i and k, respectively, and by $\mathcal{H}_{\mathcal{X}}$ the direct sum $\mathcal{H}_{\mathcal{X}} := \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_P$. It then holds that:*

a) $\mathcal{H}_{\mathcal{X}}^k = \mathcal{H}_{\mathcal{X}},\ \forall k \in \mathcal{K}$; and

b) $\forall\ f,\ \inf\{\|f\|_{\mathcal{H}_{\mathcal{X}}^k} :\ k \in \mathcal{K}\} = \min\{\sum_{i=1}^{P} \|c_i\|_{\mathcal{H}_i} :\ f = \sum_{i=1}^{P} c_i,\ c_i \in \mathcal{H}_i\}$.

According to Lemma 4.1, $\mathcal{H}_{\mathcal{X}}$ can replace $\mathcal{H}_{\mathcal{X}}^k$ in (4.13), rendering it equivalent to

$$\begin{gathered} \hat{f} = \arg \min_{f \in \mathcal{H}_{\mathcal{X}}} \sum_{n=1}^{N} (z_n - f(x_n))^2 + \mu \sum_{i=1}^{P} \|c_i\|_{\mathcal{H}_i} \\ \text{s. to } \{f = \sum_{i=1}^{P} c_i,\ c_i \in \mathcal{H}_i,\ \mathcal{H}_{\mathcal{X}} := \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_P\}. \end{gathered} \tag{4.14}$$

MKL as in (4.14) resembles (4.9), differing in that components $c_i(x)$ in (4.14) depend on the same variable x. Taking into account this difference, (4.14) is reducible to (4.10) and thus solvable via BCD or ADMoM, after substituting $k_i(x_n, x)$ for $k_i(x_{ni}, x_i)$. On the other hand, a more general case of MKL is presented in [127], where $\mathcal{K}$ is the convex hull of an infinite and possibly uncountable family of kernels.

An example of MKL applied to wireless communications is offered in Section 4.6, where two different kernels are employed for estimating path-loss and shadowing propagation effects in a cognitive radio sensing paradigm.

In the ensuing section, basis functions depending on a second variable y will be incorporated to broaden the scope of the additive models just described.

4.3 Nonparametric basis pursuit

Consider function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over the Cartesian product of spaces $\mathcal{X}$ and $\mathcal{Y}$ with associated RKHSs $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$, respectively. Let f abide to the bilinear expansion form

$$f(x,y) = \sum_{i=1}^{P} c_i(x) b_i(y) \tag{4.15}$$

where $b_i : \mathcal{Y} \to \mathbb{R}$ can be viewed as bases, and $c_i : \mathcal{X} \to \mathbb{R}$ as expansion coefficient functions. Given a finite number of training data, learning $\{c_i, b_i\}$ under sparsity constraints constitutes the goal of the NBP approaches developed in the following sections.

The first method for sparse KBL of f in (4.15) is related to a *nonparametric* counterpart of basis pursuit, with the goal of fitting the function $f(x,y)$ to data, where $\{b_i\}$ are prescribed and $\{c_i\}$s are to be learned. The designer's degree of confidence on the modeling assumptions is key to deciding whether $\{b_i\}$s should be prescribed or learned from data. If the prescribed $\{b_i\}$s are unreliable, model (4.15) will be inaccurate and the performance of KBL will suffer. But neglecting the prior knowledge conveyed by $\{b_i\}$s may be also damaging. Parametric basis pursuit [44] hints toward addressing this tradeoff by offering a compromising alternative.

A functional dependence $z = f(y) + e$ between input y and output z is modeled in [44] with an overcomplete set of bases $\{b_i(y)\}$ (a.k.a. regressors) as

$$z = \sum_{i=1}^{P} c_i b_i(y) + e, \quad e \sim \mathcal{N}(0, \sigma^2). \tag{4.16}$$

Certainly, leveraging an overcomplete set of bases $\{b_i(y)\}$ can accommodate uncertainty. Practical merits of basis pursuit however, hinge on its capability to learn the few $\{b_i\}$s that “best” explain the given data.

The crux of NBP on the other hand, is to fit $f(x,y)$ with a basis expansion over the y domain, but learn its dependence on x through nonparametric means. Model (4.15) comes handy for this purpose, when $\{b_i(y)\}_{i=1}^P$ is a generally overcomplete collection of prescribed bases.

With $\{b_i(y)\}_{i=1}^P$ known, $\{c_i(x)\}_{i=1}^P$ need to be estimated, and a kernel-based strategy can be adopted to this end. Accordingly, the optimal function $\hat{f}(x,y)$ is searched over the family $\mathcal{F}_b := \{f(x,y) = \sum_{i=1}^P c_i(x) b_i(y)\}$, which constitutes the feasible set for the NBP-tailored nonparametric Lasso [cf. (4.9)]

$$\hat{f} = \arg\min_{f \in \mathcal{F}_b} \sum_{n=1}^{N} (z_n - f(x_n, y_n))^2 + \mu \sum_{i=1}^{P} \|c_i\|_{\mathcal{H}_\mathcal{X}}. \tag{4.17}$$

The Representer Theorem in its general form (4.3) can be applied recursively to minimize (4.17) w.r.t. each $c_i(x)$ at a time, rendering $\hat{f}$ expressible in terms of the kernel expansion as $\hat{f}(x,y) = \sum_{i=1}^P \sum_{n=1}^N \gamma_{in} k(x_n, x) b_i(y)$, where coefficients $\boldsymbol{\gamma}_i^T := [\gamma_{i1}, \ldots, \gamma_{iN}]$ are learned from data $\mathbf{z}^T := [z_1, \ldots, z_N]$ via group Lasso [cf. (4.10)]

$$\min_{\{\boldsymbol{\gamma}_i \in \mathbb{R}^N\}_{i=1}^P} \left\| \mathbf{z} - \textstyle\sum_{i=1}^P \mathbf{K}_i \boldsymbol{\gamma}_i \right\|^2 + \mu \sum_{i=1}^{P} \|\boldsymbol{\gamma}_i\|_{\mathbf{K}} \tag{4.18}$$

with $\mathbf{K}_i :=$ Diag$[b_i(y_1), \ldots, b_i(y_N)]\mathbf{K}$.

As it was argued in Section III, group Lasso in (4.18) effects group-sparsity in the subvectors $\{\boldsymbol{\gamma}_i\}_{i=1}^P$. This property inherited by (4.17) is the capability of selecting bases in the nonparametric setup. Indeed, by zeroing $\boldsymbol{\gamma}_i$ the corresponding coefficient function $c_i(x) = \sum_{n=1}^N \gamma_{in} k(x_n, x)$ is driven to zero, and correspondingly $b_i(y)$ drops from the expansion (4.15).

Remark 2. A single kernel $k_\mathcal{X}$ and associated RKHS $\mathcal{H}_\mathcal{X}$ can be used for all components $c_i(x)$ in (4.17), since the summands in (4.15) are differentiated through the bases. Specifically, for a common $\mathbf{K}$, a different $b_i(y)$ per coefficient $c_i(x)$, yields a distinct diagonal matrix Diag$[b_i(y_1), \ldots, b_i(y_N)]$, defining an individual $\mathbf{K}_i$ in (4.18) that renders vector $\boldsymbol{\gamma}_i$ identifiable. This is a particular characteristic of (4.17), in contrast with (4.9) and Lemma 4.1 which are designed for, and require, multiple kernels.

Remark 3. The different sparse kernel-based approaches presented so far, namely SpAMs, MKL, and NBP, should not be viewed as competing but rather as complementary choices. Multiple kernels can be used in basis pursuit, and a separable model for $c_i(x)$ may be due in high dimensions. An NBP-MKL hybrid applied to spectrum cartography illustrates this point in Section 4.6, where bases are utilized for the frequency domain $\mathcal{Y}$.

4.4 Blind NBP for matrix and tensor completion

A kernel-based matrix completion scheme will be developed in this section using a *blind* version of NBP, in which bases $\{b_i\}$ will not be prescribed, but they will be learned together with coefficient functions $\{c_i\}$. The matrix completion task entails imputation of missing entries of a data matrix $\mathbf{Z} \in \mathbb{R}^{M \times N}$. Entries of an index matrix $\mathbf{W} \in \{0,1\}^{M \times N}$ specify whether datum z_{mn} is available ($w_{mn} = 1$), or missing ($w_{mn} = 0$). Low rank of $\mathbf{Z}$ is a popular attribute that relates missing with available data, thus granting feasibility to the imputation task. Low-rank matrix imputation is achieved by solving

$$\hat{\mathbf{Z}} = \arg \min_{\mathbf{A} \in \mathbb{R}^{M \times N}} \frac{1}{2} \|(\mathbf{Z} - \mathbf{A}) \odot \mathbf{W}\|_F^2 \text{ s. to rank}(\mathbf{A}) \leq P \tag{4.19}$$

where $\odot$ stands for the Hadamard (element-wise) product. The low-rank constraint corresponds to an upperbound on the number of nonzero singular values of matrix $\mathbf{A}$, as given by its ℓ_0-norm. Specifically, if $\mathbf{s}^T := [s_1, \ldots, s_{\min\{M,N\}}]$ denotes vector of singular values of $\mathbf{A}$, and the cardinality $|\{s_i \neq 0,\ i = 1, \ldots, \min\{M,N\}\}| := \|\mathbf{s}\|_0$ defines its ℓ_0-norm, then the ball of radius P, namely $\|\mathbf{s}\|_0 \leq P$, can replace rank$(\mathbf{A}) \leq P$ in (4.19). The feasible set $\|\mathbf{s}\|_0 \leq P$ is not convex because $\|\mathbf{s}\|_0$ is not a proper norm (it lacks linearity), and solving (4.19) requires a combinatorial search for the nonzero entries of $\mathbf{s}$. A convex relaxation is thus well motivated. If the ℓ_0-norm is surrogated by the ℓ_1-norm, the corresponding ball $\|\mathbf{s}\|_1 \leq P$ becomes the convex hull of the original feasible set. As the singular values of $\mathbf{A}$ are non-negative by definition, it follows that $\|\mathbf{s}\|_1 = \sum_{i=1}^{\min\{M,N\}} s_i$. Since the sum of singular values equals the dual norm of the ℓ_2-norm of $\mathbf{A}$ [30, p.637], $\|\mathbf{s}\|_1$ defines a norm over the matrix $\mathbf{A}$ itself, namely the nuclear norm of $\mathbf{A}$, denoted by $\|\mathbf{A}\|_*$.

Upon substituting $\|\mathbf{A}\|_*$ for the rank, (4.19) is further transformed to its Lagrangian form by placing the constraint in the objective as a regularization term, i.e.,

$$\hat{\mathbf{Z}} = \arg \min_{\mathbf{A} \in \mathbb{R}^{M \times N}} \frac{1}{2} \|(\mathbf{Z} - \mathbf{A}) \odot \mathbf{W}\|_F^2 + \mu \|\mathbf{A}\|_*. \tag{4.20}$$

The next step towards kernel-based matrix completion relies on an alternative definition of $\|\mathbf{A}\|_*$. Consider bilinear factorizations of matrix $\mathbf{A} = \mathbf{C}\mathbf{B}^T$ with $\mathbf{B} \in \mathbb{R}^{N\times P}$ and $\mathbf{C} \in \mathbb{R}^{M\times P}$, in which the constraint rank$(\mathbf{A}) \leq P$ is implicit. The nuclear norm of $\mathbf{A}$ can be redefined as (see e.g., [119])

$$\|\mathbf{A}\|_* = \inf_{\mathbf{A}=\mathbf{C}\mathbf{B}^T} \frac{1}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2). \tag{4.21}$$

Result (4.21) states that the infimum is attained by the singular value decomposition of $\mathbf{A}$. Specifically, if $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ with $\mathbf{U}$ and $\mathbf{V}$ unitary and $\mathbf{\Sigma} := \text{diag}(\mathbf{s})$, and if $\mathbf{B}$ and $\mathbf{C}$ are selected as $\mathbf{B} = \mathbf{V}\mathbf{\Sigma}^{1/2}$, and $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}^{1/2}$, then $\frac{1}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) = \sum_{i=1}^P s_i = \|\mathbf{A}\|_*$. Given (4.21), it is possible to rewrite (4.20) as

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{A}=\mathbf{C}\mathbf{B}^T} \frac{1}{2}\|(\mathbf{Z}-\mathbf{A})\odot\mathbf{W}\|_F^2 + \frac{\mu}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2). \tag{4.22}$$

A formal proof of the equivalence between (4.20) and (4.22) can be found in [119].

Matrix completion in its factorized form (4.22) can be reformulated in terms of (4.15) and RKHSs. Following [6], define spaces $\mathcal{X} := \{1,\ldots,M\}$ and $\mathcal{Y} := \{1,\ldots,N\}$ with associated kernels $k_{\mathcal{X}}(m,m')$ and $k_{\mathcal{Y}}(n,n')$, respectively. Let $f(m,n)$ represent the (m,n)-th entry of the approximant matrix $\mathbf{A}$ in (4.22), and P a prescribed overestimate of its rank. Consider estimating $f : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ in (4.15) over the family $\mathcal{F} := \{f(m,n) = \sum_{i=1}^P c_i(n)b_i(m),\ c_i \in \mathcal{H}_{\mathcal{X}},\ b_i \in \mathcal{H}_{\mathcal{Y}}\}$ via

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} w_{mn}(z_{mn} - f(m,n))^2 + \frac{\mu}{2}\sum_{i=1}^{P}\left(\|c_i\|_{\mathcal{H}_{\mathcal{X}}}^2 + \|b_i\|_{\mathcal{H}_{\mathcal{Y}}}^2\right). \tag{4.23}$$

If both kernels are selected as Kronecker delta functions, then (4.23) coincides with (4.22). This equivalence is stated in the following lemma.

Lemma 4.2 *Consider spaces $\mathcal{X} := \{1,\ldots,M\}$, $\mathcal{Y} := \{1,\ldots,N\}$ and kernels $k_{\mathcal{X}}(m,m') := \delta(m-m')$ and $k_{\mathcal{Y}}(n,n') := \delta(n-n')$ over the product spaces $\mathcal{X}\times\mathcal{X}$ and $\mathcal{Y}\times\mathcal{Y}$, respectively. Define functions $f : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$, $c_i : \mathcal{X} \to \mathbb{R}$, and $b_i : \mathcal{Y} \to \mathbb{R}$, $i = 1,\ldots,P$, and matrices $\mathbf{A} \in \mathbb{R}^{M\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times P}$, and $\mathbf{C} \in \mathbb{R}^{M\times P}$. It holds that:*

a) *RKHS $\mathcal{H}_{\mathcal{X}}$ ($\mathcal{H}_{\mathcal{Y}}$) of functions over $\mathcal{X}$ (correspondingly $\mathcal{Y}$), associated with $k_{\mathcal{X}}$ ($k_{\mathcal{Y}}$) reduce to $\mathcal{H}_{\mathcal{X}} = \mathbb{R}^M$ ($\mathcal{H}_{\mathcal{Y}} = \mathbb{R}^N$).*

b) Problems (4.23), (4.22), and (4.20) are equivalent upon identifying $f(m,n) = A_{mn}$*,* $b_i(n) = B_{ni}$*, and* $c_i(m) = C_{mi}$*.*

According to Lemma 4.2, the intricacy of rewriting (4.20) as in (4.23) does not introduce any benefit when the kernel is selected as the Kronecker delta. But as it will be argued next, the equivalence between these two estimators generalizes nicely the matrix completion problem to sparse KBL of missing data with arbitrary kernels.

The separable structure of the regularization term in (4.23) enables a finite dimensional representation of functions

$$\begin{aligned}\hat{c}_i(m) &= \sum_{m'=1}^{M} \gamma_{im'} k_{\mathcal{X}}(m', m),\ m = 1, \ldots, M,\\ \hat{b}_i(n) &= \sum_{n'=1}^{N} \beta_{in'} k_{\mathcal{Y}}(n', n),\ n = 1, \ldots, N.\end{aligned} \tag{4.24}$$

Optimal scalars $\{\gamma_{im}\}$ and $\{\beta_{in}\}$ are obtained by substituting (4.24) into (4.23), and solving

$$\min_{\substack{\tilde{\mathbf{C}} \in \mathbb{R}^{M \times P}\\ \tilde{\mathbf{B}} \in \mathbb{R}^{N \times P}}} \frac{1}{2} \|(\mathbf{Z} - \mathbf{K}_{\mathcal{X}} \tilde{\mathbf{C}} \tilde{\mathbf{B}}^T \mathbf{K}_{\mathcal{Y}}^T) \odot \mathbf{W}\|_F^2 + \frac{\mu}{2} \left[\text{trace}(\tilde{\mathbf{C}}^T \mathbf{K}_{\mathcal{X}} \tilde{\mathbf{C}}) + \text{trace}(\tilde{\mathbf{B}}^T \mathbf{K}_{\mathcal{Y}} \tilde{\mathbf{B}})\right] \tag{4.25}$$

where matrix $\tilde{\mathbf{C}}$ ($\tilde{\mathbf{B}}$) is formed with entries γ_{mi} (β_{ni}).

A Bayesian approach to kernel-based matrix completion is given next, followed by an algorithm to solve for $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$.

4.4.1 Bayesian Low-Rank Imputation and Prediction

To recast (4.23) in a Bayesian framework, suppose that the available entries of $\mathbf{Z}$ obey the additive white Gaussian noise (AWGN) model $\mathbf{Z} = \mathbf{A} + \mathbf{E}$, with $\mathbf{E}$ having entries independent identically distributed (i.i.d.) according to the zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

Matrix $\mathbf{A}$ is factorized as $\mathbf{A} = \mathbf{C}\mathbf{B}^T$ without loss of generality (w.l.o.g.). Then, a Gaussian prior is assumed for each of the columns $\mathbf{b}_i$ and $\mathbf{c}_i$ of $\mathbf{B}$ and $\mathbf{C}$, respectively,

$$\mathbf{b}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_B),\ \mathbf{c}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_C) \tag{4.26}$$

independent across i, and with $\text{trace}(\mathbf{R}_B) = \text{trace}(\mathbf{R}_C)$. Invariance across i is justifiable, since columns are a priori interchangeable, while $\text{trace}(\mathbf{R}_B) = \text{trace}(\mathbf{R}_C)$ is introduced w.l.o.g. to remove the scalar ambiguity in $\mathbf{A} = \mathbf{C}\mathbf{B}^T$.

Under the AWGN model, and with priors (4.26), the maximum a posteriori (MAP) estimator of $\mathbf{A}$ given $\mathbf{Z}$ at the entries indexed by $\mathbf{W}$ takes the form [cf. (4.25)]

$$\min_{\substack{\mathbf{C}\in\mathbb{R}^{M\times P}\\ \mathbf{B}\in\mathbb{R}^{N\times P}}} \frac{1}{2}\|(\mathbf{Z}-\mathbf{C}\mathbf{B}^T)\odot\mathbf{W}\|_F^2+\frac{\sigma^2}{2}\left[\text{trace}(\mathbf{C}^T\mathbf{R}_C^{-1}\mathbf{C})+\text{trace}(\mathbf{B}^T\mathbf{R}_B^{-1}\mathbf{B})\right]. \tag{4.27}$$

With $\mathbf{R}_C = \mathbf{K}_{\mathcal{X}}$ and $\mathbf{R}_B = \mathbf{K}_{\mathcal{Y}}$, and substituting $\mathbf{B} := \mathbf{K}_{\mathcal{Y}}\tilde{\mathbf{B}}$ and $\mathbf{C} := \mathbf{K}_{\mathcal{X}}\tilde{\mathbf{C}}$, the MAP estimator that solves (4.27) coincides with the estimator solving (4.25) for the coefficients of kernel-based matrix completion, provided that covariance and Gram matrices coincide. From this Bayesian perspective, the KBL matrix completion method (4.23) provides a generalization of (4.20), which can accommodate a priori knowledge in the form of correlation across rows and columns of the incomplete $\mathbf{Z}$.

With prescribed correlation matrices $\mathbf{R}_B$ and $\mathbf{R}_C$, (4.23) can even perform smoothing and prediction. Indeed, if a column (or row) of $\mathbf{Z}$ is completely missing, (4.23) can still find an estimate $\hat{\mathbf{Z}}$ relying on the covariance between the missing and available columns. This feature is not available with (4.20), since the latter relies only on rank-induced colinearities, so it cannot reconstruct a missing column. The prediction capability is useful for instance in collaborative filtering [6], where a group of users rates a collection of items, to enable inference of new-user preferences or items entering the system. Additionally, the Bayesian reformulation (4.27) provides an explicit interpretation for the regularization parameter $\mu = \sigma^2$ as the variance of the model error, which can thus be obtained from training data. The kernel-based matrix completion method (4.27) is summarized in Algorithm 2, which solves (4.27) upon identifying $\mathbf{R}_C = \mathbf{K}_{\mathcal{X}}$, $\mathbf{R}_B = \mathbf{K}_{\mathcal{Y}}$, and $\sigma^2 = \mu$, and solves (4.25) after changing variables $\mathbf{B} := \mathbf{K}_{\mathcal{Y}}\tilde{\mathbf{B}}$ and $\mathbf{C} := \mathbf{K}_{\mathcal{X}}\tilde{\mathbf{C}}$ (compare (4.25) with lines 13-14 in Algorithm 2).

Detailed derivations of the updates in Algorithm 2 are provided in Appendix C. For a high-level description, the columns of $\mathbf{B}$ and $\mathbf{C}$ are updated cyclically, solving (4.27) via BCD iterations. This procedure converges to a stationary point of (4.27), which in principle does not guarantee global optimality. Opportunely, it can be established that local minima of (4.27) are global minima, by transforming (4.27) into a convex problem through the same change of variables proposed in [119] for the analysis of (4.22). This observation implies that Algorithm 2 yields the global optimum of (4.25), and thus (4.23).

Algorithm 2 : Kernel Matrix Completion (KMC)

```
Initialize $\mathbf{B}$ and $\mathbf{C}$ randomly.
Set the identity matrix $\mathbf{I}_P$, with dimensions $P \times P$, and columns $\mathbf{e}_i,\ i = 1, \ldots, P$
while $|\text{cost} - \text{cost\_old}| < \epsilon$ do
    for $i = 1, \ldots, P$ do
        Set $\mathbf{Z}_i := \mathbf{Z} - \mathbf{C}(\mathbf{I}_P - \mathbf{e}_i\mathbf{e}_i^T)\mathbf{B}^T$
        Compute $\mathbf{H}_i := \text{Diag}[\mathbf{W}(\mathbf{B}\mathbf{e}_i \odot \mathbf{B}\mathbf{e}_i)] + \mu\mathbf{K}_{\mathcal{Y}}^{-1}$
        Update column $\mathbf{c}_i = \mathbf{H}_i^{-1}(\mathbf{W} \odot \mathbf{Z}_i)\mathbf{B}\mathbf{e}_i$
    end for
    for $i = 1,\ldots, P$ do
        Set $\mathbf{Z}_i := \mathbf{Z} - \mathbf{C}(\mathbf{I}_P - \mathbf{e}_i\mathbf{e}_i^T)\mathbf{B}^T$
        Compute $\bar{\mathbf{H}}_i := \text{Diag}[\mathbf{W}^T(\mathbf{C}\mathbf{e}_i \odot \mathbf{C}\mathbf{e}_i)] + \mu\mathbf{K}_{\mathcal{X}}^{-1}$
        Update column $\mathbf{b}_i = \bar{\mathbf{H}}_i^{-1}(\mathbf{W}^T \odot \mathbf{Z}_i^T)\mathbf{C}\mathbf{e}_i$
    end for
    Recalculate cost $= \frac{1}{2}\|(\mathbf{Z} - \mathbf{C}\mathbf{B}^T) \odot \mathbf{W}\|_F^2$
                $+\frac{\mu}{2}\left[\text{trace}(\mathbf{C}^T\mathbf{K}_{\mathcal{X}}^{-1}\mathbf{C}) + \text{trace}(\mathbf{B}^T\mathbf{K}_{\mathcal{Y}}^{-1}\mathbf{B})\right]$
end while
return $\tilde{\mathbf{B}} = \mathbf{K}_{\mathcal{Y}}^{-1}\mathbf{B}$, $\tilde{\mathbf{C}} = \mathbf{K}_{\mathcal{X}}^{-1}\mathbf{C}$, and $\hat{\mathbf{Z}} = \mathbf{C}\mathbf{B}^T$
```

The kernel-based matrix completion method here offers an alternative to [6], where the low-rank constraint is introduced indirectly through the kernel trick. Furthermore, bypassing the nuclear norm and using (4.21) instead, renders (4.23) generalizable to tensor imputation [24].

4.5 Kernel-based dictionary learning

Basis pursuit approaches advocate an overcomplete set of bases to cope with model uncertainty, thus learning from data the most concise subset of bases that represents the signal of interest. But the extensive set of candidate bases (a.k.a. dictionary) still needs to be prescribed. The next step towards model-agnostic KBL is to learn the dictionary from data, along with the sparse regression coefficients. Under the sparse linear model

$$\mathbf{z}_m = \mathbf{B}\boldsymbol{\gamma}_m + \mathbf{e}_m,\ m = 1, \ldots, M \tag{4.28}$$

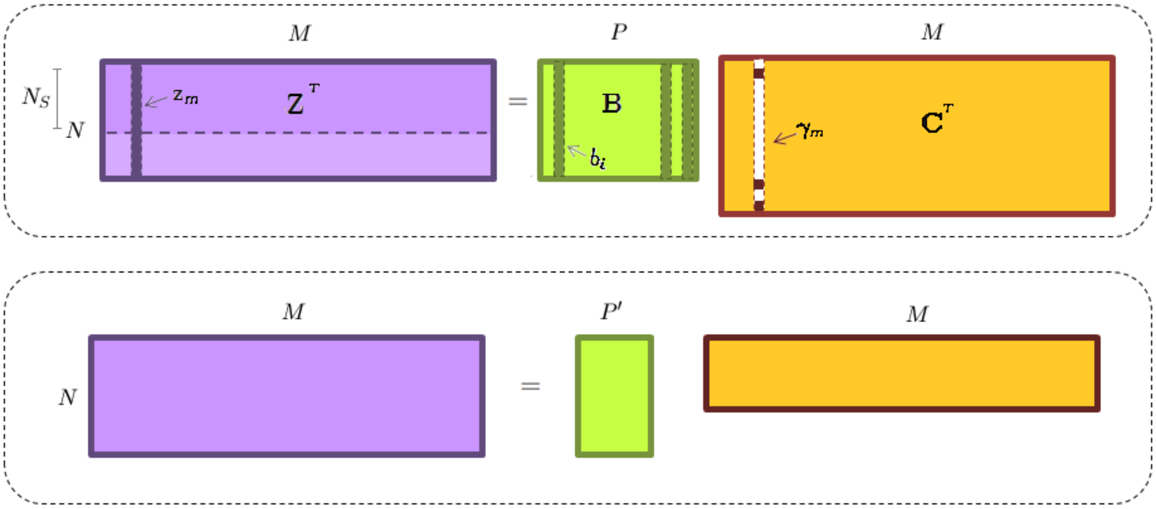


Figure 4.1: Comparison between KDL and NBP; (top) dictionary $\mathbf{B}$ and sparse coefficients γ_m for KDL, where MN_S equations are sufficient to recover $\mathbf{C}$; (bottom) low-rank structure $\mathbf{A} = \mathbf{C}\mathbf{B}^T$ presumed in KMC.

with dictionary of bases $\mathbf{B} \in \mathbb{R}^{N\times P}$, and vector of coefficients $\gamma_m \in \mathbb{R}^P$, the goal of dictionary learning is to obtain $\mathbf{B}$ and $\mathbf{C} := [\gamma_1, \ldots, \gamma_M]^T$ from data $\mathbf{Z} := [\mathbf{z}_1, \ldots, \mathbf{z}_M]^T$. A swift count of equations and unknowns yields $NP + MP$ scalar variables to be learned from MN data (see Fig. 4.1). This goal is not plausible for an overcomplete design ($P > N$) unless sparsity of $\{\gamma_m\}_{m=1}^M$ is exploited. Under proper conditions, it is possible to recover a sparse γ_m containing at most S nonzero entries from a reduced number $N_s := \theta S \log P \leq N$ of equations [36], where θ is a proportionality constant. Hence, the number of equations needed to specify $\mathbf{C}$ reduces to MN_s, as represented by the darkened region of $\mathbf{Z}^T$ in Fig. 4.1. With $N_s < N$, it is then possible and crucial to collect a sufficiently large number M of data vectors in order to ensure that $MN \geq NP + MN_S$, thus accommodating the additional NP equations needed to determine $\mathbf{B}$, and enable learning of the dictionary.

Having collected sufficient training data, one possible approach to find $\mathbf{B}$ and $\mathbf{C}$ is to fit the data via the LS cost $\|\mathbf{Z} - \mathbf{C}\mathbf{B}^T\|_F^2$ regularized by the ℓ_1-norm of $\mathbf{C}$ in order to effect sparsity in the coefficients [106]. This dictionary leaning approach can be recast into the form of blind NBP (4.23) by introducing the additional regularizing term $\lambda \sum_{i=1}^P \|c_i\|_1$, with $\|c_i\|_1 := \sum_{m=1}^M |c_i(m)|$. The new regularizer on functions $c_i : \mathcal{X} \to \mathbb{R}$ depends on their values at the measurement points m only, and can be absorbed in the loss part of (4.3). Thus, the optimal $\{c_i\}$ and $\{b_i\}$ conserve their finite expansion representations dictated by the Representer Theorem. Coefficients $\{\gamma_{mp}, \beta_{np}\}$ must be adapted according to the new cost, and (4.27) becomes

$$\min_{\substack{\mathbf{C}\in\mathbb{R}^{M\times P}\\ \mathbf{B}\in\mathbb{R}^{N\times P}}} \frac{1}{2}\|(\mathbf{Z}-\mathbf{C}\mathbf{B}^T)\odot\mathbf{W}\|_F^2 + \lambda\|\mathbf{C}\|_1 + \frac{\sigma^2}{2}\left[\text{trace}(\mathbf{B}^T\mathbf{R}_B^{-1}\mathbf{B}) + \text{trace}(\mathbf{C}^T\mathbf{R}_C^{-1}\mathbf{C})\right]. \tag{4.29}$$

Remark 4. Kernel-based dictionary learning (KDL) via (4.29) inherits two attractive properties of kernel matrix completion (KMC), that is blind NBP, namely its flexibility to introduce a priori information through $\mathbf{R}_B$ and $\mathbf{R}_C$, as well as the capability to cope with missing data. While both KDL and KMC estimate bases $\{b_i\}$ and coefficients $\{c_i\}$ jointly, their difference lies in the size of the dictionary. As in principal component analysis, KMC presumes a low-rank model for the approximant $\mathbf{A}=\mathbf{C}\mathbf{B}^T$, compressing signals $\{\mathbf{z}_m\}$ with $P' < M$ components (Fig. 4.1 (bottom)). Low rank of $\mathbf{A}$ is not required by the dictionary learning approach, where signals $\{\mathbf{z}_m\}$ are spanned by $P \geq M$ dictionary atoms $\{b_i\}$ (Fig. 4.1 (top)), provided that each $\mathbf{z}_m$ is composed by a few atoms only.

Algorithm 2 can be modified to solve (4.29) by replacing the update for column $\mathbf{c}_i$ in line 7 with the Lasso estimate

$$\mathbf{c}_i := \arg\min_{\mathbf{c}\in\mathbb{R}^M} \frac{1}{2}\mathbf{c}^T\mathbf{H}_i\mathbf{c} + \mathbf{c}^T(\mathbf{W}\odot\mathbf{Z}_i)\mathbf{B}\mathbf{e}_i + \lambda\|\mathbf{c}\|_1. \tag{4.30}$$

The Bayesian interpretation of (4.29) brings KDL close to [186], where a Bernoulli-Gaussian model for $\mathbf{C}$ accounts for its sparsity, and a Beta distribution is introduced for learning the distribution of $\mathbf{C}$ through hyperparameters. Although [186] assumes independent Gaussian variables across "time" samples in the underlying model for $\mathbf{C}$, generalization to correlated variables is straightforward. Bernoulli parameters controlling the sparsity of c_{mp} are assumed invariant across m in [186], which amounts to stationarity over c_{mp}.

Sparse learning of temporally correlated data is studied also in [197], although the time-invariant model for the support of $\mathbf{c}_m$ does not lend itself to dictionary learning.

Although dictionary learning can indeed be viewed as a blind counterpart of compressive sampling, its capability of recovering $\mathbf{B}$ and $\mathbf{C}$ from data is typically illustrated by examples rather than theoretical guarantees. Recent efforts on establishing identifiability and local optimality of dictionary learning can be found in [76] and [85]. A related KDL strategy has been proposed in [162], where data and dictionary atoms are organized in classes, and the regularized learning criterion is designed to promote cohesion of atoms within a class.

4.6 Applications

4.6.1 Spectrum cartography via NBP and MKL

Consider the setup in [21] with $N_c = 100$ radios distributed over an area $\mathcal{X}$ of $100 \times 100\text{m}^2$ to measure the ambient RF power spectral density (PSD) at $N_f = 24$ frequencies equally spaced in the band from $2,400$MHz to $2,496$MHz, as specified by IEEE 802.11 wireless LAN standard [3]. The radios collaborate by sharing their $N = N_c N_f$ measurements with the goal of obtaining a map of the PSD across space and frequency, while specifying at the same time which of the $P = 14$ frequency sub-bands are occupied. The wireless propagation is simulated according to the pathloss model affected by shadowing described in [7], with parameters $n_p = 3$, $\Delta_0 = 60$m, $\delta = 25$m , $\sigma_X^2 = 25dB$, and with AWGN variance $\sigma_n^2 = -10dB$. Fig. 4.2 depicts the distribution of power across space generated by two sources transmitting over bands $i = 5$ and $i = 8$ with center frequencies $2,432$MHz and $2,447$MHz, respectively. Fig. 4.3 shows the PSD as seen by a representative radio located at the center of $\mathcal{X}$.

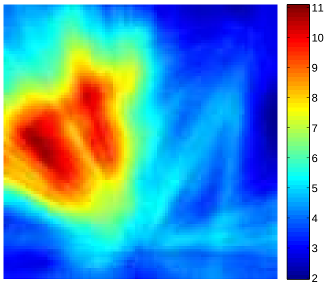


Figure 4.2: Aggregate power distribution across space.

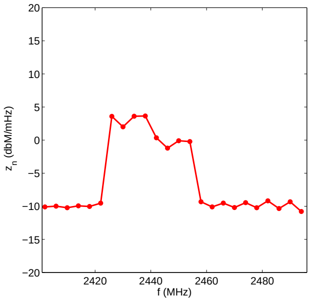


Figure 4.3: PSD measurements at a representative location x_n.

Model (4.15) is adopted for collaborative PSD sensing, with x and y representing the spatial and frequency variables, respectively. Bases $\{b_i\}$ are prescribed as Hann-windowed pulses in accordance with [3], and the distribution of power across space per sub-band is given by $\{c_i(x)\}$ after interpolating the measurements obtained by the radios via (4.17). Two exponential kernels $k_r(x, x') = \exp(-\|x - x'\|^2/\theta_r^2),\ r = 1, 2$ with $\theta_1 = 10$m and $\theta_2 = 20$m are selected, and convex combinations of the two are considered as candidate interpolators $k(x, x')$. This MKL strategy is intended for capturing two different levels of resolution as produced by pathloss and shadowing. Correspondingly, each $c_i(x)$ is decomposed into two functions $c_{i1}(x)$ and $c_{i2}(x)$ which are regularized separately in (4.17).

Solving (4.17) generates the PSD maps of Fig. 4.4. Only $\boldsymbol{\gamma}_5$ and $\boldsymbol{\gamma}_8$ in the solution to (4.18) take nonzero values (more precisely $\boldsymbol{\gamma}_{5r}$ and $\boldsymbol{\gamma}_{8r},\ r = 1, 2$ in the MKL adaptation of (4.18)), which correctly reveals which frequency bands are occupied as shown in Fig. 4.4(a). The estimated PSD across space is depicted in Fig. 4.4 (b,first row) for each band respectively, and compared to the ground truth depicted in Fig. 4.4 (b,second row). The multi-resolution components $c_{5r}(x)$ and $c_{8r}(x)$ are depicted in Fig. 4.4 (b,last two rows), demonstrating how kernel k_1 captures the coarse pathloss distribution, while k_2 refines the map by revealing locations affected by shadowing.

These results demonstrate the usefulness of model (4.15) for collaborative spectrum sensing, with bases abiding to [3] and multi-resolution kernels. The sparse nonparametric estimator (4.17) serves the purpose of revealing the occupied frequency bands, and capturing the PSD map across space per source. Compared to the spline-based approach in [21], the MKL adaptation of (4.17) here provides the appropriate multi-resolution capability to capture pathloss and shadowing effects when interpolating the data across space.

4.6.2 Completion of Gene Expression Data via Blind NBP

The imputation method (4.23) is tested here on microarray data described in [160]. Expression levels of yeast across $N_g = 4,772$ genes sampled at $N = 13$ time points during the cell cycle are considered. A subset of $M = 100$ genes is extracted and their expression levels are organized in the matrix $\mathbf{Z} \in \mathbb{R}^{M \times N}$ depicted in Fig. 4.5 (left). Severe data losses are simulated by discarding 90% of the entries of $\mathbf{Z}$, including the nearly 5% actually missing data.

According to the Bayesian model (4.26), it follows that

$$E[\mathbf{Z}\mathbf{Z}^T] = \theta\mathbf{R}_C + \sigma_e^2\mathbf{I}, \quad E[\mathbf{Z}^T\mathbf{Z}] = \theta\mathbf{R}_B + \sigma_e^2\mathbf{I} \,. \tag{4.31}$$

To study the effect of hydrogen peroxide on the cell cycle arrest, two extra microarray datasets $\mathbf{Z}^{(1)},\ \mathbf{Z}^{(2)} \in \mathbb{R}^{M\times N}$, synchronized with $\mathbf{Z}$, are collected in [160]. These two matrices are employed to form an estimate of $E[\mathbf{Z}\mathbf{Z}^T]$, which is used instead of $\mathbf{R}_C$ in (4.27) after neglecting the noise term in (4.31). Since the presence of hydrogen peroxide in samples $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ induces cell cycle arrest, the correlation between samples across time in $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ is altered, and thus these samples are not appropriate for estimating $E[\mathbf{Z}^T\mathbf{Z}]$. Alternatively, the sample estimate of $E[\mathbf{Z}^T\mathbf{Z}]$ is formed with the microarray data of the $(N_g - M) \times N$ genes set aside, and then used in place of $\mathbf{R}_B$ in (4.27).

Solving (4.27) with the available data (10% of the total) as shown in Fig. 4.5 (second left) results in the matrix $\hat{\mathbf{Z}}$ depicted in Fig. 4.5 (second right), where the imputed missing data introduce an average recovery error of -8dB [cf. Fig. 4.6]. In producing $\hat{\mathbf{Z}}$, the smoothing capability of (4.23) to recover completely missing rows of $\mathbf{Z}$ (amounting to 25 in this example) is corroborated. Missing rows cannot be recovered by nuclear norm regularization alone [cf. (4.20)], even if $\mathbf{Z}$ is padded with expression levels of the discarded $N_g - M$ genes. Fig. 4.5 (right) presents this case confirming that its performance dagrades w.r.t. NBP; while Fig. 4.6 illustrates the sensitivity of the estimation error to the cross-validated regularization parameter μ for both estimators. Similar degraded results are observed when imputing missing entries of $\mathbf{Z}$ using the impute.knn() and svdImpute() methods, as implemented in the R packages pcaMethods and BioConductor-impute. These two methods were applied to the padded $\mathbf{Z}$, after the requisite discarding of the 25 missing rows, resulting in recovery errors on the remaining missing entries at -3.84dB and -0.12dB (with parameter nPcs= 12), respectively.

4.6.3 Network Flow Prediction via Blind NBP

The Abilene network in Fig. 4.7, a.k.a. Internet 2, comprising 11 nodes and $M = 30$ links [1], is utilized as a testbed for traffic load prediction. Aggregate link loads z_{mn} are recorded every 5 minute intervals in the morning of December 22, 2008, between 12:00am and 11:55pm, and are collected in the first $N/2 = 144$ columns of matrix $\mathbf{Z} \in \mathbb{R}^{M \times N}$. These samples are then used to predict link loads hours ahead, by capitalizing on their mutual cross-correlation, the periodic correlation across days, and their interdependence across links as dictated by the network topology.

The correlation matrix $E(\mathbf{Z}\mathbf{Z}^T)$ represented in Fig. 4.8 is estimated with training samples collected during the two previous weeks, from December 8 to December 21, 2008, and substituted for $\mathbf{R}_C$ in (4.27) according to (4.31). A singular point at 11:00am in the traffic curve, as depicted in black in Fig. 4.9, is reflected in the sharp transition noticed in Fig. 4.8. On the other hand, $\mathbf{R}_B$ is not estimated but derived from the network structure. Supposing i.i.d. flows across the network, it holds that $E(\mathbf{Z}^T\mathbf{Z}) = \sigma_f^2\mathbf{R}^T\mathbf{R}$, where $\mathbf{R}$ represents the network routing matrix and σ_f^2 the flow variance. Thus, $\sigma_f^2\mathbf{R}^T\mathbf{R}$, was used instead of $\mathbf{R}_B$ in (4.27), with σ_f^2 adjusted to satisfy tr$(E[\mathbf{Z}^T\mathbf{Z}]) =$ tr$(E[\mathbf{Z}\mathbf{Z}^T])$.

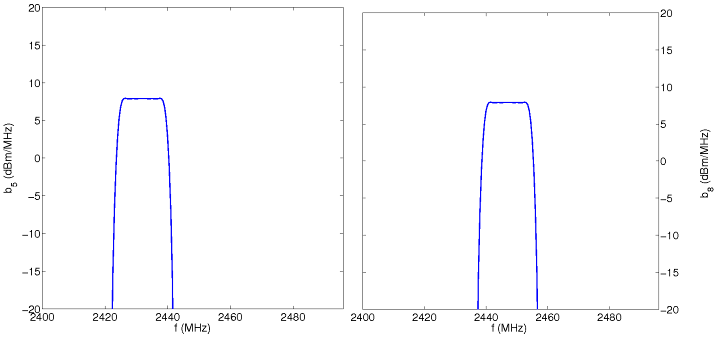
Fig. 4.9 shows link loads predicted by (4.27) on December 22, 2008, for a representative link, along with the actually recorded samples for that day. Prediction accuracy is compared in Fig. 4.9 to a base strategy comprising independent LMMSE estimators per link, which yield a relative prediction error $e_p = 0.22$ aggregated across links, against $e_p = 0.15$ that results from (4.27). Strong correlation among samples from 12:00am to 2:00pm [cf. Fig. 4.8] renders LMMSE prediction accurate in this interval, relying on single-link data only. The benefit of considering the links jointly is appreciated in the subsequent interval from 2:00pm to 11:55pm, where the traffic correlation with morning samples fades away and the network structure comes to add valuable information, in the form of $\mathbf{R}_B$, to stabilize prediction.

4.7 Summary

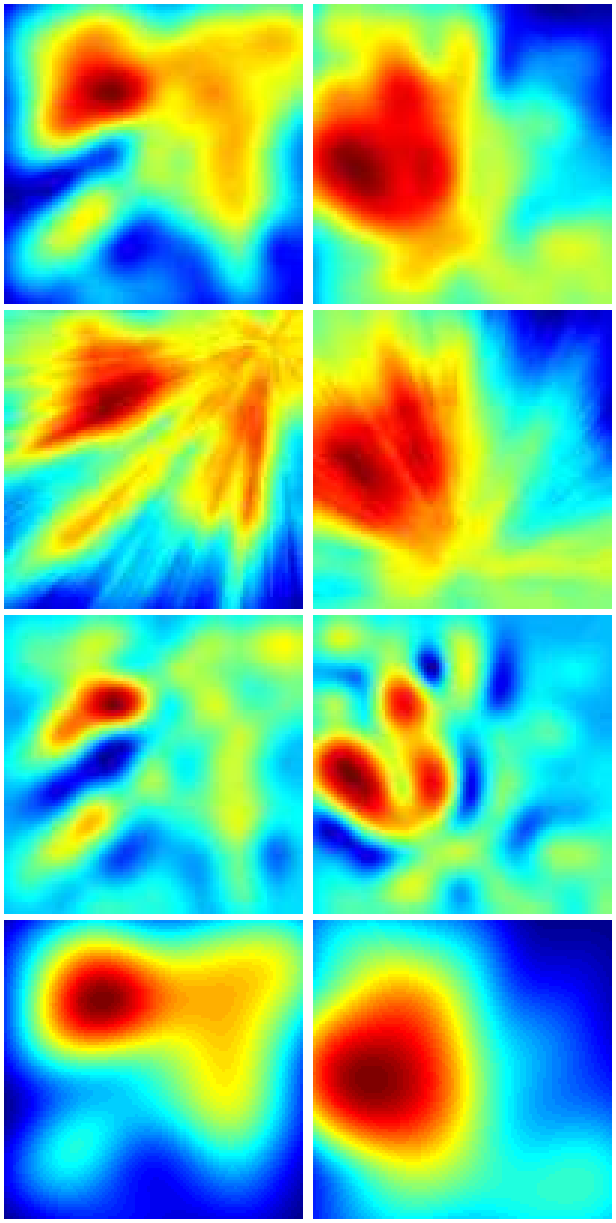
A new methodology was outlined in this chapter by cross fertilizing sparsity-aware signal processing tools with kernel-based learning. It goes well beyond translating sparse vector regression techniques into their nonparametric counterparts, to generate a series of unique possibilities such as kernel selection or kernel-based matrix completion. The present article contributes to these efforts by advancing NBP as the cornerstone of sparse KBL, including blind versions that emerge as nonparametric nuclear norm regularization and dictionary learning.

KBL was connected with GP analysis, promoting a Bayesian viewpoint where kernels convey prior information. Alternatively, KBL can be regarded as an interpolation toolset though its connection with the NST, suggesting that the impact of the prior model choice is attenuated when the size of the dataset is large, especially when kernel selection is also incorporated. Further insights on parallel sparse KBL can also be found in Appendix C.

All in all, sparse KBL was envisioned as a fruitful research direction. Its impact on signal processing practice was illustrated through a diverse set of application paradigms.



(a)



(b)

Figure 4.4: NBP for spectrum cartography using multiple kernels

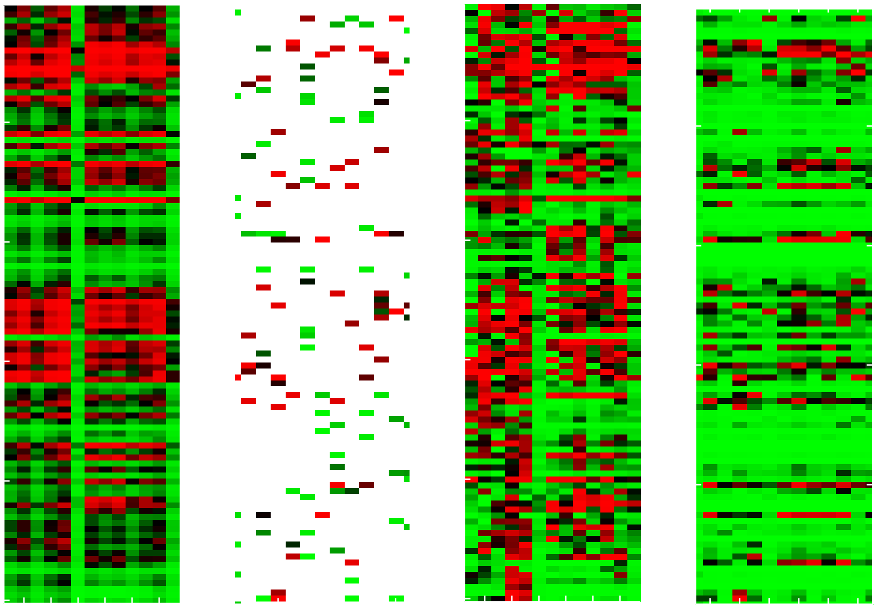


Figure 4.5: Microarray data completion; from left to right: original sample; 10% available data; recovery via NBP; and recovery via nuclear-norm regularized LS.

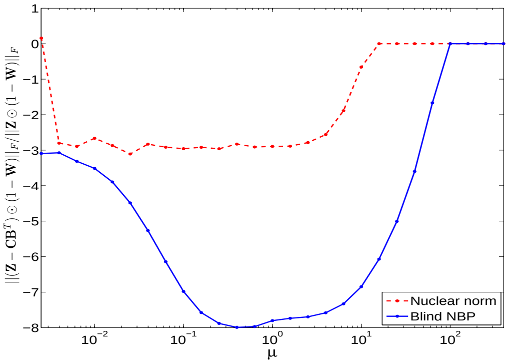


Figure 4.6: Relative recovery error in dB with 90% missing data; comparison between blind NBP (KMC) and nuclear norm regularization.

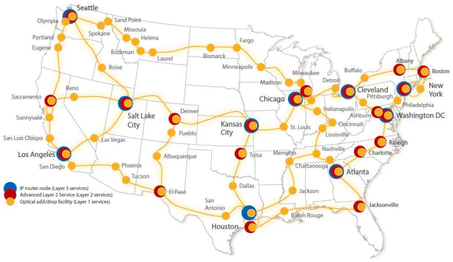


Figure 4.7: Internet 2 network topology graph [1].

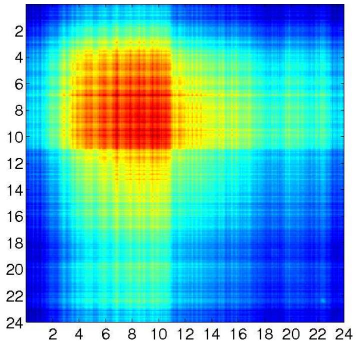


Figure 4.8: Sample estimates of $E(\mathbf{ZZ}^T)$ for link loads across time, are used to replace $\mathbf{R}_C$ and $\mathbf{K}_\mathcal{Y}$.

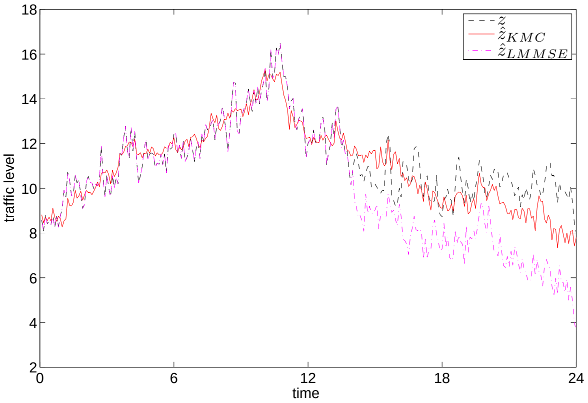


Figure 4.9: Network prediction via KMC (blind NBP). Measured and predicted traffic on link $m = 21$.

Chapter 5

Rank regularization and Bayesian inference for tensor completion and extrapolation[†]

5.1 Preliminaries

5.1.1 Nuclear-norm minimization for matrix completion

Low-rank approximation is a popular method for estimating missing values of a matrix $\mathbf{Z} \in \mathbb{R}^{N \times M}$, which capitalizes on "regularities" across the data [68]. For the imputation to be feasible, a binding assumption that relates the available entries with the missing ones is required. An alternative is to postulate that $\mathbf{Z}$ has low rank $R \ll \min(N, M)$. The problem of finding matrix $\hat{\mathbf{Z}}$ with rank not exceeding R, which approximates $\mathbf{Z}$ in the given entries specified by a binary matrix $\boldsymbol{\Delta} \in \{0,1\}^{N \times M}$, can be formulated as

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{X}} \|(\mathbf{Z} - \mathbf{X}) \circledast \boldsymbol{\Delta}\|_F^2 \quad \mathrm{s.\,to}\ \ \mathrm{rank}(\mathbf{X}) \leq R\,. \tag{5.1}$$

The low-rank property of matrix $\mathbf{X}$ implies that the vector $\mathbf{s}(\mathbf{X})$ of its singular values is sparse. Hence, the rank constraint is equivalent to $\|\mathbf{s}(\mathbf{X})\|_0 \leq R$, where the ℓ_0-(pseudo)norm $\|\cdot\|_0$ equals the number of nonzero entries of its vector argument.

Aiming at a convex relaxation of the NP-hard problem (5.1), one can leverage recent advances in compressive sampling [68] and surrogate the ℓ_0-norm with the ℓ_1-norm, which here equals the nuclear norm of $\mathbf{X}$ defined as $\|\mathbf{X}\|_* := \|\mathbf{s}(\mathbf{X})\|_1$. With this relaxation, the Lagrangian counterpart of (5.1) is

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|(\mathbf{Z}-\mathbf{X})\circledast\mathbf{\Delta}\|_F^2 + \mu\|\mathbf{X}\|_* \tag{5.2}$$

where $\mu \geq 0$ is a rank-controlling parameter. Problem (2) can be further transformed by considering the following characterization of the nuclear norm [171]

$$\|\mathbf{X}\|_* = \min_{\{\mathbf{B},\mathbf{C}\}} \frac{1}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \quad \text{s. to } \ \mathbf{X} = \mathbf{B}\mathbf{C}^T. \tag{5.3}$$

For an arbitrary matrix $\mathbf{X}$ with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the minimum in (5.3) is attained for $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}^{1/2}$ and $\mathbf{C} = \mathbf{V}\mathbf{\Sigma}^{1/2}$. The optimization in (5.3) is over all possible bilinear factorizations of $\mathbf{X}$, so that the number of columns of $\mathbf{B}$ and $\mathbf{C}$ is also a variable.

For given R, note that the factorization $\mathbf{X} = \mathbf{B}\mathbf{C}^T$ with $\mathbf{B} \in \mathbb{R}^{N\times R}$ and $\mathbf{C} \in \mathbb{R}^{M\times R}$ implies rank$(\mathbf{X}) \leq R$. Introducing the aforementioned bilinear factorization of $\mathbf{X}$, and replacing $\|\mathbf{X}\|_*$ in (5.2) with the Frobenius-norm regularization dictated by (5.3), one arrives at the following reformulation of (5.2) [119]

$$\begin{aligned}\hat{\mathbf{Z}}' = &\arg\min_{\{\mathbf{X},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\|(\mathbf{Z}-\mathbf{X})\circledast\mathbf{\Delta}\|_F^2 + \frac{\mu}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2)\\ &\text{s. to } \mathbf{X} = \mathbf{B}\mathbf{C}^T.\end{aligned} \tag{5.4}$$

Problems (5.2) and (5.4) can be readily proved equivalent [cf. Proposition 1-a)], in the sense that by finding the global minimum of (5.4), one can recover the optimal solution of (5.2). However, since (5.4) is *nonconvex*, it may have multiple stationary points that need not be globally optimal. Interestingly, the next result provides global optimality conditions for these stationary points [parts a) and b) are proved in Appendix D, while the proof for c) can be found in [119].]

Proposition 5.1 *If $R \geq$ rank$(\hat{\mathbf{Z}})$, then problems (5.2) and (5.4) are equivalent, in the sense that:*

a) global minima coincide: $\hat{\mathbf{Z}} = \hat{\mathbf{Z}}'$;

b) all local minima of (5.4) are globally optimal; and, *c) stationary points* $\mathbf{X}$ *of (5.4) satisfying* $\|(\mathbf{X}-\mathbf{Z})\circledast\boldsymbol{\Delta}\|_2 \leq \mu$ *are globally optimal.*

This result plays a critical role in this paper, as the Frobenius-norm regularization for controlling the rank in (5.4) will be useful to obtain its tensor counterparts in Section 5.2.

Remark 5.1 *Without missing data, all entries of* $\boldsymbol{\Delta}$ *are equal to one, and (5.1) boils down to principal component analysis. In this case, (5.1) can be solved by truncating the SVD of* $\mathbf{Z}$*, so that only its* R *largest singular values are retained. The presence of missing entries changes the problem profoundly, as (5.1) becomes NP-hard [184]. This highlights the importance of the nuclear norm regularizer (5.2) as a clever alternative to rank minimization in the presence of missing data. Reduced complexity alternatives to SVD are also available; e.g., the truncated multi-stage Wiener filter (MSWF) [83]. MSWF offers an attractive alternative to (5.1) for matrix (and even tensor) dimensionality reduction. This approach is not pursued here however, since redesigning the MSWF to cope with* missing data *may prove challenging [cf. (5.1) with and without missing data.] Conversely, exploring variants of (5.2) for reduced-rank Wiener filtering in the presence of missing data, constitutes an interesting direction for future research.*

5.1.2 PARAFAC decomposition

The PARAFAC decomposition of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{M\times N\times P}$ is at the heart of the proposed imputation method, since it offers a means to define its rank [107, 176]. Given $R \in \mathbb{N}$, consider matrices $\mathbf{A} \in \mathbb{R}^{N\times R}$, $\mathbf{B} \in \mathbb{R}^{M\times R}$, and $\mathbf{C} \in \mathbb{R}^{P\times R}$, such that

$$\underline{\mathbf{X}}(m,n,p) = \sum_{r=1}^{R} \mathbf{A}(m,r)\mathbf{B}(n,r)\mathbf{C}(p,r). \tag{5.5}$$

The rank of $\underline{\mathbf{X}}$ is the minimum value of R for which this decomposition is possible. For $R^* :=$ rank$(\underline{\mathbf{X}})$, the PARAFAC decomposition is given by the corresponding factor matrices $\{\mathbf{A},\mathbf{B},\mathbf{C}\}$ (all with R^* columns), so that (5.5) holds with $R = R^*$.

To appreciate why the aforementioned rank definition is natural, rewrite (5.5) as $\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, where $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ represent the r-th columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively; and the outer products $\underline{\mathbf{O}}_r := \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \in \mathbb{R}^{M\times N\times P}$ have entries $\underline{\mathbf{O}}_r(m,n,p) := \mathbf{A}(m,r)\mathbf{B}(n,r)\mathbf{C}(p,r)$. The rank of a tensor is thus the minimum number of outer products (rank one factors) required to

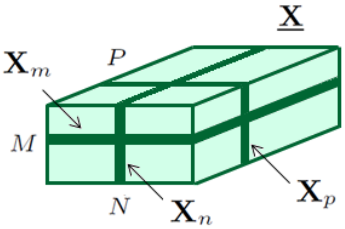


Figure 5.1: Tensor slices along the row, column, and tube dimensions.

represent the tensor. It is not uncommon to adopt an equivalent normalized representation

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = \sum_{r=1}^{R} \gamma_r (\mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r) \tag{5.6}$$

by defining unit-norm vectors $\mathbf{u}_r := \mathbf{a}_r/\|\mathbf{a}_r\|$, $\mathbf{v}_r := \mathbf{b}_r/\|\mathbf{b}_r\|$, $\mathbf{w}_r := \mathbf{c}_r/\|\mathbf{c}_r\|$, and weights $\gamma_r := \|\mathbf{a}_r\|\|\mathbf{b}_r\|\|\mathbf{c}_r\|$, $r = 1, \ldots, R$.

Let $\mathbf{X}_p,\ p = 1, \ldots, P$ denote the p-th slice of $\underline{\mathbf{X}}$ along its third (tube) dimension, such that $\mathbf{X}_p(m,n) := \underline{\mathbf{X}}(m,n,p)$; see Fig. 5.1. The following compact form of the PARAFAC decomposition in terms of slice factorizations will be used in the sequel

$$\mathbf{X}_p = \mathbf{A}\text{diag}\left[\mathbf{e}_p^T \mathbf{C}\right] \mathbf{B}^T, \quad p = 1, \ldots, P \tag{5.7}$$

where the diagonal matrix diag $[\mathbf{u}]$ has the vector $\mathbf{u}$ on its diagonal, and $\mathbf{e}_p^T$ is the p-th row of the $P \times P$ identity matrix. The PARAFAC decomposition is symmetric [cf. (5.5)], and one can also write $\mathbf{X}_m = \mathbf{B}\text{diag}\left[\mathbf{e}_m^T \mathbf{A}\right] \mathbf{C}^T$, or, $\mathbf{X}_n = \mathbf{C}\text{diag}\left[\mathbf{e}_n^T \mathbf{B}\right] \mathbf{A}^T$ in terms of slices along the first (row), or, second (column) dimensions. Given $\underline{\mathbf{X}}$, under some technical conditions then $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ are unique up to a common column permutation and scaling (meaning PARAFAC is identifiable); see e.g. [107, 163, 172, 176].

5.2 Rank regularization for tensors

Generalizing the nuclear-norm regularization technique (5.2) from low-rank matrix to tensor completion is not straightforward, since singular values of a tensor (given by the Tucker decomposition) are not related to the rank [104]. Fortunately, the Frobenius-norm regularization outlined in Section

5.1.1 offers a viable option for low-rank tensor completion under the PARAFAC model, by solving

$$\hat{\underline{\mathbf{Z}}} := \arg\min_{\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\|(\underline{\mathbf{Z}}-\underline{\mathbf{X}})\circledast\underline{\mathbf{\Delta}}\|_F^2 + \frac{\mu}{2}\left(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2+\|\mathbf{C}\|_F^2\right)$$

$$\text{s. to}\quad \mathbf{X}_p = \mathbf{A}\text{diag}\left[\mathbf{e}_p^T\mathbf{C}\right]\mathbf{B}^T, \quad p=1,\ldots,P \tag{5.8}$$

where the Frobenius norm of a tensor is defined as $\|\underline{\mathbf{X}}\|_F^2 := \sum_m\sum_n\sum_p \underline{\mathbf{X}}^2(m,n,p)$, and the Hadamard product as $(\underline{\mathbf{X}}\circledast\underline{\mathbf{\Delta}})(m,n,p) := \underline{\mathbf{X}}(m,n,p)\underline{\mathbf{\Delta}}(m,n,p)$.

Different from the matrix case, it is unclear whether the regularization in (5.8) bears any relation with the tensor rank. Interestingly, the following analysis corroborates the capability of (5.8) to produce a low-rank tensor $\hat{\underline{\mathbf{Z}}}$, for sufficiently large μ. In this direction, consider an alternative completion problem stated in terms of the normalized tensor representation (5.6)

$$\hat{\underline{\mathbf{Z}}}' := \arg\min_{\{\underline{\mathbf{X}},\boldsymbol{\gamma},\{\mathbf{u}_r\},\{\mathbf{v}_r\},\{\mathbf{w}_r\}\}} \frac{1}{2}\|\left(\underline{\mathbf{Z}}-\underline{\mathbf{X}}\right)\circledast\underline{\mathbf{\Delta}}\|_F^2 + \frac{\mu}{2}\|\boldsymbol{\gamma}\|_{2/3}^{2/3}$$

$$\text{s. to}\ \underline{\mathbf{X}} = \sum_{r=1}^{R}\gamma_r(\mathbf{u}_r\circ\mathbf{v}_r\circ\mathbf{w}_r) \tag{5.9}$$

where $\boldsymbol{\gamma} := [\gamma_1,\ldots,\gamma_R]^T$; the nonconvex $\ell_{2/3}$ (pseudo)-norm is given by $\|\boldsymbol{\gamma}\|_{2/3} := (\sum_{r=1}^R|\gamma_r|^{2/3})^{3/2}$; and the unit-norm constraint on the factors' columns is left implicit. Problems (5.8) and (5.9) are equivalent as established by the following proposition (see Appendix D for a proof.)

Proposition 5.2 *The solutions of (5.8) and (5.9) coincide, i.e.,* $\hat{\underline{\mathbf{Z}}}' = \hat{\underline{\mathbf{Z}}}$, *with optimal factors related by* $\hat{\mathbf{a}}_r = \sqrt[3]{\hat{\gamma}_r}\hat{\mathbf{u}}_r$, $\hat{\mathbf{b}}_r = \sqrt[3]{\hat{\gamma}_r}\hat{\mathbf{v}}_r$, *and* $\hat{\mathbf{c}}_r = \sqrt[3]{\hat{\gamma}_r}\hat{\mathbf{w}}_r$, $r=1,\ldots,R$.

To further stress the capability of (5.8) to produce a low-rank approximant tensor $\underline{\mathbf{X}}$, consider transforming (5.9) once more by rewriting it in the constrained-error form

$$\hat{\underline{\mathbf{Z}}}'' := \arg\min_{\{\underline{\mathbf{X}},\boldsymbol{\gamma},\{\mathbf{u}_r\},\{\mathbf{v}_r\},\{\mathbf{w}_r\}\}} \|\boldsymbol{\gamma}\|_{2/3} \tag{5.10}$$

$$\text{s. to}\ ||\left(\underline{\mathbf{Z}}-\underline{\mathbf{X}}\right)\circledast\underline{\mathbf{\Delta}}||_F^2 \leq \sigma^2, \quad \underline{\mathbf{X}} = \sum_{r=1}^{R}\gamma_r(\mathbf{u}_r\circ\mathbf{v}_r\circ\mathbf{w}_r).$$

For any value of σ^2 there exists a corresponding Lagrange multiplier λ such that (5.9) and (5.10) yield the same solution, under the identity $\mu = 2/\lambda$. [Since $f(x) = x^{2/3}$ is an increasing function,

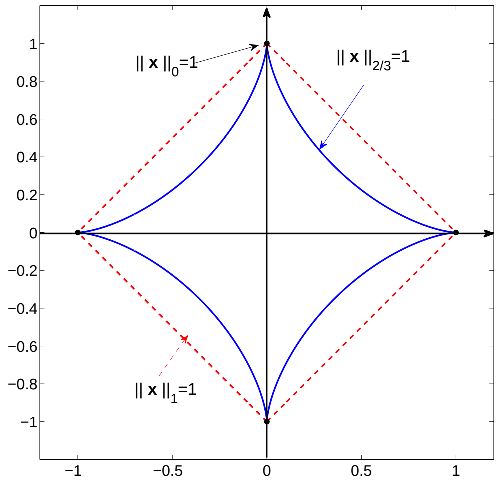


Figure 5.2: The unit $\ell_{2/3}$-norm ball compared to its ℓ_0- and ℓ_1-norm counterparts.

the exponent of $\|\boldsymbol{\gamma}\|_{2/3}$ can be safely eliminated without affecting the minimizer of (5.10).] The key observation is that minimizing $\|\boldsymbol{\gamma}\|_{2/3}$ in (5.10) yields a sparse vector $\boldsymbol{\gamma}$ [42]. As with the well-known sparsity-promoting ℓ_1-norm, the unit $\ell_{2/3}$-norm ball exhibits a “pointy geometry” at the axes responsible for inducing sparsity; see Fig. 5.2.

With (5.8) equivalently rewritten as in (5.10), its low-rank inducing property is now revealed. As $\boldsymbol{\gamma}$ in (5.10) becomes sparse, some of its entries γ_r are nulled, and the corresponding outer-products $\gamma_r(\mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r)$ drop from the sum in (5.6), thus lowering the rank of $\underline{\mathbf{X}}$.

The next property is a direct consequence of the low-rank promoting property of (5.8) as established in Proposition 5.2; see Appendix D for a proof.

Corollary 5.1 *Let $\hat{\underline{\mathbf{Z}}}$ denote the solution of (5.8). If $\mu \geq \mu_{\max} := \|\underline{\boldsymbol{\Delta}} \circledast \underline{\mathbf{Z}}\|_F^{4/3}$, then $\hat{\underline{\mathbf{Z}}} = \mathbf{0}_{M\times N\times P}$.*

Corollary 5.1 asserts that if μ is chosen large enough, the rank is reduced to the extreme case $\text{rank}(\hat{\underline{\mathbf{Z}}}) = 0$. To see why this is a non-trivial property, it is prudent to think of linear models and ridge-regression estimates which entail similar quadratic regularizers, but an analogous property does not hold. In ridge regression one needs to let $\mu \to \infty$ in order to obtain an all-zero solution. Characterization of $\mu_{\max}$ is also of practical relevance as it provides a frame of reference for tuning the regularization parameter.

Using (5.10), it is also possible to relate (5.8) with the atomic norm in [40]. Indeed, the infimum ℓ_1-norm of $\boldsymbol{\gamma}$ is a proper norm for $\underline{\mathbf{X}}$, named atomic norm, and denoted by $\|\underline{\mathbf{X}}\|_{\mathcal{A}} := \|\boldsymbol{\gamma}\|_1$ [40]. Thus, by replacing $\|\boldsymbol{\gamma}\|_{2/3}$ with $\|\underline{\mathbf{X}}\|_{\mathcal{A}}$, (5.10) becomes convex in $\underline{\mathbf{X}}$. Still, the complexity of solving such a variant of (5.10) resides in that $\|\underline{\mathbf{X}}\|_{\mathcal{A}}$ is generally intractable to compute [40]. In this regard, it is remarkable that arriving to (5.10) had the sole purpose of demonstrating the low-rank inducing property, and that (5.8) is to be solved by the algorithm developed in the ensuing section. Such an algorithm will neither require computing the atomic norm or PARAFAC decomposition of $\underline{\mathbf{X}}$, nor knowing its rank. The number of columns in $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be set to an overestimate of the rank of $\underline{\mathbf{Z}}$, such as the upper bound $\bar{R} := \min\{MN, NP, PM\} \geq$ rank($\underline{\mathbf{Z}}$), and the low-rank of $\underline{\mathbf{X}}$ will be induced by regularization as argued earlier. It is also fair to say that only convergence to a stationary point of (5.8) will be established in this paper.

Remark 5.2 These insights foster future research directions for the design of a convex regularizer of the tensor rank. Specifically, substituting $\rho(\mathbf{A}, \mathbf{B}, \mathbf{C}) := \sum_{r=1}^{R}(\|\mathbf{a}_r\|^3 + \|\mathbf{b}_r\|^3 + \|\mathbf{c}_r\|^3)$ for the regularization term in (5.8), turns $\|\boldsymbol{\gamma}\|_{2/3}$ into $\|\boldsymbol{\gamma}\|_1 = \|\underline{\mathbf{X}}\|_{\mathcal{A}}$ in the equivalent (5.10). It is envisioned that with such a modification in place, the acquired convexity of (5.10) would enable a reformulation of Proposition 5.1 for the tensor case, providing conditions for global optimality of the stationary points of (5.8).

Remark 5.3 *Feasibility of the imputation task relies fundamentally on assuming a low-dimensional data model, to couple the available and missing entries as well as reduce the effective degrees of freedom in the problem. Different low-dimensional models would lead to alternative imputation methods, as the unfolded tensor regularization in [73], or the truncated MSWF [83] discussed in Remark 5.1. The comparative performance of these methods would depend on the accuracy of their modeling assumptions. This paper focuses on low-rank tensors, hence (5.8) is expected to outperform its competitors. This intuition is corroborated by numerical tests in Section 5.5.*

Still, a limitation of (5.8) is that it does not allow for incorporating side information that could be available in addition to the given entries $\underline{\boldsymbol{\Delta}} \circledast \underline{\mathbf{Z}}$.

Remark 5.4 In the context of recommender systems, a description of the users and/or products through attributes (e.g., gender, age) or measures of similarity, is typically available. It is thus meaningful to exploit both known preferences and descriptions to model the preferences of users [6]. In three-way (samples, genes, conditions) microarray data analysis, the relative position of single-nucleotide polymorphisms in the DNA molecule implies degrees of correlation among genotypes [157]. These correlations could be available either through a prescribed model, or, through estimates obtained using a reference tensor $\check{\underline{\mathbf{Z}}}$. A probabilistic approach to tensor completion capable of incorporating such types of extra information is the subject of the ensuing section.

5.3 Bayesian low-rank tensor approximation

5.3.1 Bayesian PARAFAC model

A probabilistic approach is developed in this section in order to integrate the available statistical information into the tensor imputation setup. To this end, suppose that the observation noise is zero-mean, white, Gaussian; that is the noisy tensor measurements $z_{mnp} := \underline{\mathbf{Z}}(m,n,p)$ are given by

$$z_{mnp} = x_{mnp} + e_{mnp}, \quad e_{mnp} \sim \mathcal{N}(0,\sigma^2), \text{ i.i.d..} \tag{5.11}$$

Since vectors $\mathbf{a}_r$ in (5.6) are interchangeable, identical distributions are assigned across $r = 1,\ldots,R$, and they are modeled as independent from each other, zero-mean Gaussian distributed with covariance matrix $\mathbf{R}_A \in \mathbb{R}^{M\times M}$. Similarly, vectors $\mathbf{b}_r$ and $\mathbf{c}_r$ are uncorrelated and zero-mean, Gaussian, with covariance matrix $\mathbf{R}_B$ and $\mathbf{R}_C$, respectively. In addition $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ are assumed mutually uncorrelated. Since scale ambiguity is inherently present in the PARAFAC model, vectors $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ are set to have equal power; that is,

$$\theta := \text{Tr}(\mathbf{R}_A) = \text{Tr}(\mathbf{R}_B) = \text{Tr}(\mathbf{R}_C) \tag{5.12}$$

where Tr$(\cdot)$ denotes the matrix trace operator.

Under these assumptions, the posterior distribution $p(\mathbf{A},\mathbf{B},\mathbf{C}|\underline{\mathbf{Z}})$ can be factorized as $p(\underline{\mathbf{Z}}|\mathbf{A},\mathbf{B},\mathbf{C})p(\mathbf{A})p(\mathbf{B})p(\mathbf{C})/p(\underline{\mathbf{Z}})$ and is thus proportional to $\exp(-L(\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}))$, where

$$\begin{aligned} L(\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}) &= \frac{1}{2\sigma^2}\|(\underline{\mathbf{Z}}-\underline{\mathbf{X}})\circledast\underline{\mathbf{\Delta}}\|_F^2 + \frac{1}{2}\sum_{r=1}^{R}\left(\mathbf{a}_r^T\mathbf{R}_A^{-1}\mathbf{a}_r + \mathbf{b}_r^T\mathbf{R}_B^{-1}\mathbf{b}_r + c_r^T\mathbf{R}_C^{-1}\mathbf{c}_r\right) \\ &= \frac{1}{2\sigma^2}\|(\underline{\mathbf{Z}}-\underline{\mathbf{X}})\circledast\underline{\mathbf{\Delta}}\|_F^2 + \frac{1}{2}\left[\text{Tr}\left(\mathbf{A}^T\mathbf{R}_A^{-1}\mathbf{A}\right) + \text{Tr}\left(\mathbf{B}^T\mathbf{R}_B^{-1}\mathbf{B}\right) + \text{Tr}\left(\mathbf{C}^T\mathbf{R}_C^{-1}\mathbf{C}\right)\right]. \end{aligned}$$

and with $\underline{\mathbf{X}} := \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ as in (5.5).

The MAP estimator of $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is defined as the maximizer of $p(\mathbf{A}, \mathbf{B}, \mathbf{C}|\underline{\mathbf{Z}})$ [100, p. 350]. Equivalently, the MAP estimator of $\underline{\mathbf{X}}$ follows from minimizing $L(\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ w.r.t. $\underline{\mathbf{X}}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, with (5.5) as a constraint; i.e.,

$$\hat{\underline{\mathbf{Z}}} := \arg\min_{\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2\sigma^2}\|(\underline{\mathbf{Z}} - \underline{\mathbf{X}})\circledast\underline{\mathbf{\Delta}}\|_F^2 + \frac{1}{2}\left[\mathrm{Tr}\left(\mathbf{A}^T\mathbf{R}_A^{-1}\mathbf{A}\right) + \mathrm{Tr}\left(\mathbf{B}^T\mathbf{R}_B^{-1}\mathbf{B}\right) + \mathrm{Tr}\left(\mathbf{C}^T\mathbf{R}_C^{-1}\mathbf{C}\right)\right]$$
$$\text{s. to } \mathbf{X}_p = \mathbf{A}\mathrm{diag}\left[\mathbf{e}_p^T\mathbf{C}\right]\mathbf{B}^T,\ p = 1, \ldots, P \qquad (5.13)$$

reducing to (5.8) when $\mathbf{R}_A = \mathbf{I}_M$, $\mathbf{R}_B = \mathbf{I}_N$, and $\mathbf{R}_C = \mathbf{I}_P$.

Remark 5.5 *From this Bayesian vantage point, the regularization parameter μ [cf. (5.8)] can be interpreted as the noise variance, which is useful in practice to select μ. This parameter choice is complemented by the guidelines to obtain the prior covariances which are outlined in Section IV-C.*

First, the ensuing section explores the advantages of incorporating prior information to the imputation method.

5.3.2 Nonparametric tensor decomposition

Incorporating the information conveyed by $\mathbf{R}_A$, $\mathbf{R}_B$, and $\mathbf{R}_C$, together with a practical means of finding these matrices can be facilitated by interpreting (5.13) in the context of RKHS [191]. In particular, the analysis presented next will use the Representer Theorem, interpreted as an instrument for finding the best interpolating function in a Hilbert space spanned by kernels, just as interpolation with sinc-kernels is carried out in the space of bandlimited functions for the purpose of reconstructing a signal from its samples [132].

In this context, it is instructive to look at a tensor $f : \mathcal{M} \times \mathcal{N} \times \mathcal{P} \to \mathbb{R}$ as a function of three variables $m, n,$ and p, living in measurable spaces $\mathcal{M}, \mathcal{N}$, and $\mathcal{P}$, respectively. Generalizing (5.8) to this nonparametric framework, low-rank functions f are formally defined to belong to the following family

$$\mathcal{F}_R := \{f : \mathcal{M}\times\mathcal{N}\times\mathcal{P}\to\mathbb{R} :\ f(m,n,p) = \sum_{r=1}^{R} a_r(m)b_r(n)c_r(p)$$
$$\text{such that } a_r(m) \in \mathcal{H}_{\mathcal{M}},\ b_r(n) \in \mathcal{H}_{\mathcal{N}},\ c_r(p) \in \mathcal{H}_{\mathcal{P}}\}$$

where $\mathcal{H}_{\mathcal{M}}$, $\mathcal{H}_{\mathcal{N}}$, and $\mathcal{H}_{\mathcal{P}}$ are Hilbert spaces constructed from specified kernels $k_{\mathcal{M}}$, $k_{\mathcal{N}}$ and $k_{\mathcal{P}}$, defined over $\mathcal{M}$, $\mathcal{N}$, and $\mathcal{P}$, while R is an initial overestimate of the rank of f.

The following nonparametric fitting criterion is adopted for finding the best $\hat{f}_R$ interpolating data $\{z_{mnp} :\ \delta_{mnp} = 1\}$

$$\hat{f}_R := \arg \min_{f\in\mathcal{F}_R} \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{i=1}^{P} \delta_{mnp}(z_{mnp} - f(m,n,p))^2 + \frac{\mu}{2}\sum_{r=1}^{R}\left(\|a_r\|^2_{\mathcal{H}_{\mathcal{M}}} + \|b_r\|^2_{\mathcal{H}_{\mathcal{N}}} + \|c_r\|^2_{\mathcal{H}_{\mathcal{P}}}\right) . \tag{5.14}$$

It is shown in Appendix D that leveraging the Representer Theorem, the minimizer of (5.14) admits a finite dimensional representation in terms of $k_{\mathcal{M}}$, $k_{\mathcal{N}}$ and $k_{\mathcal{P}}$,

$$\hat{f}_R(m,n,p) = \boldsymbol{k}^T_{\mathcal{M}}(m)\mathbf{K}^{-1}_{\mathcal{M}}\mathbf{A}\text{diag}\left[\boldsymbol{k}^T_{\mathcal{P}}(p)\mathbf{K}^{-1}_{\mathcal{P}}\mathbf{C}\right]\mathbf{B}^T\mathbf{K}^{-1}_{\mathcal{N}}\boldsymbol{k}_{\mathcal{N}}(n) \tag{5.15}$$

where vector $\mathbf{k}^T_{\mathcal{M}}(m) := [k_{\mathcal{M}}(m,1),\ldots,k_{\mathcal{M}}(m,M)]$, $m \in \mathcal{M}$, and matrix $\mathbf{K}_{\mathcal{M}}$ has entries $k_{\mathcal{M}}(m,m')$, $m,m' = 1,\ldots,M$. Likewise, $\mathbf{k}_{\mathcal{N}}(n)$, $\mathbf{K}_{\mathcal{N}}$, $\mathbf{k}_{\mathcal{P}}(p)$, and $\mathbf{K}_{\mathcal{P}}$ are correspondingly defined in terms of $k_{\mathcal{N}}$ and $k_{\mathcal{P}}$. It is also shown in Appendix D that the coefficient matrices $\mathbf{A} \in \mathbb{R}^{M\times R}$, $\mathbf{B} \in \mathbb{R}^{N\times R}$, and $\mathbf{C} \in \mathbb{R}^{P\times R}$ in (5.15) can be found by solving

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \sum_{i=1}^{P} \left\|\left(\mathbf{Z}_p - \mathbf{A}\text{diag}\left[\mathbf{e}^T_p\mathbf{C}\right]\mathbf{B}^T\right)\circledast\boldsymbol{\Delta}_p\right\|^2_F + \frac{\mu}{2}\left(\text{Tr}(\mathbf{A}^T\mathbf{K}^{-1}_{\mathcal{M}}\mathbf{A}) + \text{Tr}(\mathbf{B}^T\mathbf{K}^{-1}_{\mathcal{N}}\mathbf{B}) + \text{Tr}(\mathbf{C}^T\mathbf{K}^{-1}_{\mathcal{P}}\mathbf{C})\right) . \tag{5.16}$$

Problem (5.16) reduces to (5.8) when the side information is discarded by selecting $k_{\mathcal{M}}$, $k_{\mathcal{N}}$ and $k_{\mathcal{P}}$ as Kronecker deltas, in which case $\mathbf{K}_{\mathcal{M}}$, $\mathbf{K}_{\mathcal{N}}$, and $\mathbf{K}_{\mathcal{P}}$ are identity matrices. In the general case, (5.16) yields the sought nonlinear low-rank approximation method for $f(m,n,p)$ when combined with (5.15), evidencing the equivalence between (5.14) and (5.13).

Interpreting (5.14) as an interpolator renders (5.13) a natural choice for tensor completion, where in general, missing entries are to be imputed by connecting them to surrounding points on the three-dimensional arrangement. Relative to (5.8), this RKHS perspective also highlights (5.13)'s extra smoothing and extrapolation capabilities. Indeed, by capitalizing on the similarities captured by $\mathbf{K}_{\mathcal{M}}$, $\mathbf{K}_{\mathcal{N}}$ and $\mathbf{K}_{\mathcal{P}}$, (5.16) can recover completely missing slices. This feature is not shared by imputation methods that leverage low-rank only, since these require at least one point in the slice to build on colinearities. Extrapolation is also possible in this sense. If for instance $\mathbf{K}_{\mathcal{M}}$ can be expanded to capture a further point $M+1$ not in the original set, then a new slice of data can be predicted by (5.15) based on its correlation $k_{\mathcal{M}}(M+1)$ with the available entries. These extra capabilities will be exploited in Section 5.5.3, where correlations are leveraged for the imputation of MRI data. The method described by (5.13) and (5.16) can be applied to matrix completion by just setting entries of $\mathbf{C}$ to one, and can be extended to higher-order dimensions with a straightforward alteration of the algorithms and propositions throughout this paper.

Identification of covariance matrices $\mathbf{R}_A$, $\mathbf{R}_B$, and $\mathbf{R}_C$ with kernel matrices $\mathbf{K}_{\mathcal{M}}$, $\mathbf{K}_{\mathcal{N}}$ and $\mathbf{K}_{\mathcal{P}}$ is the remaining aspect to clarify in the connection between (5.13) and (5.16). It is apparent from (5.13) and (5.16) that correlations between columns of the factors are reflected in similarities between the tensor slices, giving rise to the opportunity of obtaining one from the other. This aspect is explored next.

5.3.3 Covariance estimation

To implement (5.13), matrices $\mathbf{R}_A$, $\mathbf{R}_B$, and $\mathbf{R}_C$ must be postulated a priori, or alternatively replaced by their sample estimates. Such estimates need a training set of vectors $\{\mathbf{a}\}$, $\{\mathbf{b}\}$, and $\{\mathbf{c}\}$ abiding to the Bayesian model described in Section 5.3.1, and this requires PARAFAC decomposition of training data. In order to abridge this procedure, it is convenient to inspect how $\mathbf{R}_A$, $\mathbf{R}_B$, and $\mathbf{R}_C$ are related to their kernel counterparts.

Based on the equivalence between the standard RKHS interpolator and the linear mean-square error estimator [146], it is useful to re-visit the probabilistic framework and identify kernel similarities between slices of $\underline{\mathbf{X}}$, with their mutual covariances. Focusing on the tube dimension of $\underline{\mathbf{X}}$, one can write $\mathbf{K}_{\mathcal{P}}(p',p) := \mathbb{E}[\text{Tr}(\mathbf{X}_{p'}^T\mathbf{X}_p)]$, that is, the covariance between slices $\mathbf{X}_{p'}$ and $\mathbf{X}_p$ taking $\langle \mathbf{X}, \mathbf{Y} \rangle := \text{Tr}(\mathbf{X}^T\mathbf{Y})$ as the standard inner product in the matrix space. Under this alternative definition for $\mathbf{K}_{\mathcal{P}}$, and corresponding definitions for $\mathbf{K}_{\mathcal{N}}$, and $\mathbf{K}_{\mathcal{M}}$, it is shown in Appendix D that

$$\mathbf{K}_{\mathcal{M}} = \theta^2 \mathbf{R}_A, \quad \mathbf{K}_{\mathcal{N}} = \theta^2 \mathbf{R}_B, \quad \mathbf{K}_{\mathcal{P}} = \theta^2 \mathbf{R}_C \tag{5.17}$$

and that θ is related to the second-order moment of $\underline{\mathbf{X}}$ by

$$\mathbb{E}[\|\underline{\mathbf{X}}\|_F^2] = R\theta^3. \tag{5.18}$$

Since sample estimates for $\mathbf{K}_{\mathcal{M}}$, $\mathbf{K}_{\mathcal{N}}$, $\mathbf{K}_{\mathcal{P}}$, and $\mathbb{E}[\|\underline{\mathbf{X}}\|_F]$ can be readily obtained from the tensor data, (5.17) and (5.18) provide an agile means of estimating $\mathbf{R}_A$, $\mathbf{R}_B$, and $\mathbf{R}_C$ without requiring PARAFAC decompositions over the set of training tensors; see also the numerical tests in Section 5.5.3.

This strategy remains valid when kernels are not estimated from data. One such case emerges in collaborative filtering of user preferences, where the similarity of two users is modeled as a prescribed function of a few attributes, such as age or income [6].

5.3.4 Block successive upper-bound minimization algorithm

An iterative algorithm is developed here for solving (5.13), by cyclically minimizing the cost over $\mathbf{A} \rightarrow \mathbf{B} \rightarrow \mathbf{C}$. This alternating-minimization procedure is typically adopted to fit PARAFAC models, and is also known as block-coordinate descent (BCD) in the optimization parlance [148]. In the first step of the cycle the cost in (5.13) is minimized with respect to (w.r.t.) $\mathbf{A}$, considering $\mathbf{B}$ and $\mathbf{C}$ as fixed parameters taking on their previous iteration values. Accordingly, the partial cost to minimize reduces to the convex function

$$f(\mathbf{A}) := \frac{1}{2}\| (\underline{\mathbf{Z}} - \underline{\mathbf{X}}) \circledast \underline{\mathbf{\Delta}}\|_F^2 + \frac{\mu}{2}\text{Tr}\left(\mathbf{A}^T \mathbf{R}_A^{-1} \mathbf{A}\right) \tag{5.19}$$

where μ was identified with and substituted for σ^2. Function (5.19) is quadratic in $\mathbf{A}$ and can be readily minimized after re-writing it in terms of $\mathbf{a} := \text{vec}(\mathbf{A})$. However, such an approach becomes computationally infeasible for other than small datasets, since it involves storing P matrices of dimensions $NM \times MR$, and solving a square linear system of MR equations. The alternative pursued here relies on the so-called block successive upper-bound minimization (BSUM) algorithm [148]. As it will become clear later on, this way the computational complexity in updating $\mathbf{A}$ is reduced from $\mathcal{O}((MR)^3)$ to $\mathcal{O}(MR^3)$ per iteration, and likewise for $\mathbf{B}$ and $\mathbf{C}$.

BSUM follows the same cyclic architecture as BCD, but one instead minimizes a judiciously chosen upper-bound $g(\mathbf{A}, \bar{\mathbf{A}})$ of $f(\mathbf{A})$. As such, it blends the properties of BCD and majorization-minimization algorithms. The majorizing function $g(\mathbf{A}, \bar{\mathbf{A}})$ depends on the current iterate $\bar{\mathbf{A}}$, and should be crafted such that it: i)it is simpler to optimize than $f(\mathbf{A})$; and ii) satisfies certain local-tightness conditions; see also [148] and properties i)-iii) in Lemma 5.1.

For given $\bar{\mathbf{A}}$, consider the function

$$g(\mathbf{A}, \bar{\mathbf{A}}) := \frac{1}{2} \| \left(\underline{\mathbf{Z}} - \underline{\mathbf{X}} \right) \circledast \underline{\boldsymbol{\Delta}} \|_F^2 + \mu \left(\frac{\lambda}{2} \mathrm{Tr}\left(\mathbf{A}^T \mathbf{A} \right) - \mathrm{Tr}(\boldsymbol{\Theta}^T \mathbf{A}) + \frac{1}{2} \mathrm{Tr}(\boldsymbol{\Theta}^T \bar{\mathbf{A}}) \right) \tag{5.20}$$

where $\lambda := \lambda_{\max}(\mathbf{R}_A^{-1})$ is the maximum eigenvalue of $\mathbf{R}_A^{-1}$, and $\boldsymbol{\Theta} := (\lambda \mathbf{I} - \mathbf{R}_A^{-1})\bar{\mathbf{A}}$. The following properties of $g(\mathbf{A}, \bar{\mathbf{A}})$ imply that it majorizes $f(\mathbf{A})$ at $\bar{\mathbf{A}}$, satisfying the technical conditions required for the convergence of BSUM (see Appendix D for a proof).

Lemma 5.1 *Function $g(\mathbf{A}, \bar{\mathbf{A}})$ in (5.20) satisfies the following properties*

i) $f(\bar{\mathbf{A}}) = g(\bar{\mathbf{A}}, \bar{\mathbf{A}})$*;*

ii) $\frac{d}{d\mathbf{A}} f(\mathbf{A})|_{\mathbf{A}=\bar{\mathbf{A}}} = \frac{d}{d\mathbf{A}} g(\mathbf{A}, \bar{\mathbf{A}})|_{\mathbf{A}=\bar{\mathbf{A}}}$*; and,*

iii) $f(\mathbf{A}) \leq g(\mathbf{A}, \bar{\mathbf{A}}), \ \forall \mathbf{A}$.

The computational advantage of minimizing $g(\mathbf{A}, \bar{\mathbf{A}})$ instead of $f(\mathbf{A})$ comes from the separability of $g(\mathbf{A}, \bar{\mathbf{A}})$ across rows of $\mathbf{A}$. To appreciate this, consider the Khatri-Rao product $\boldsymbol{\Pi} := \mathbf{C} \odot \mathbf{B} := [\mathbf{c}_1 \otimes \mathbf{b}_1, \ldots \mathbf{c}_R \otimes \mathbf{b}_R]$, defined by the column-wise Kronecker products $\mathbf{c}_r \otimes \mathbf{b}_r$. Let also matrix $\mathbf{Z} := [\mathbf{Z}_1, \ldots, \mathbf{Z}_P] \in \mathbb{N}^{M \times NP}$ denote the mode-1 unfolding of $\underline{\mathbf{Z}}$ (along its tube dimension; see e.g., [50, p.30],) and likewise for $\boldsymbol{\Delta} := [\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_P] \in \{0, 1\}^{M \times NP}$ and $\mathbf{X} := [\mathbf{X}_1, \ldots, \mathbf{X}_P] \in \mathbb{R}_+^{M \times NP}$. Using the following identity that relates the unfolded tensor with its factors [48]

$$\mathbf{X} := [\mathbf{X}_1, \ldots, \mathbf{X}_P] = \mathbf{A} \boldsymbol{\Pi}^T \tag{5.21}$$

it is possible to rewrite (5.20) as

$$g(\mathbf{A}, \bar{\mathbf{A}}) := \frac{1}{2} \| \left(\mathbf{Z} - \mathbf{A} \boldsymbol{\Pi}^T \right) \circledast \boldsymbol{\Delta} \|_F^2 + \mu \left(\frac{\lambda}{2} \mathrm{Tr}\left(\mathbf{A}^T \mathbf{A} \right) - \mathrm{Tr}(\boldsymbol{\Theta}^T \mathbf{A}) + \frac{1}{2} \mathrm{Tr}(\boldsymbol{\Theta}^T \bar{\mathbf{A}}) \right)$$

which can be decomposed as

$$g(\mathbf{A}, \bar{\mathbf{A}}) = \sum_{m=1}^{M} \left[\frac{1}{2} \|\text{diag}(\boldsymbol{\delta}_m)(\mathbf{z}_m - \boldsymbol{\Pi}\mathbf{a}_m)\|_2^2 + \mu \left((\lambda/2) \|\mathbf{a}_m\|^2 - \boldsymbol{\theta}_m^T \mathbf{a}_m + \boldsymbol{\theta}_m^T \bar{\mathbf{a}}_m \right) \right] \qquad (5.22)$$

where $\mathbf{z}_m^T$, $\mathbf{a}_m^T$, $\boldsymbol{\delta}_m^T$, $\boldsymbol{\theta}_m^T$, and $\bar{\mathbf{a}}_{\mathbf{m}}^T$, represent the m-th rows of matrices $\mathbf{Z}$, $\mathbf{A}$, $\boldsymbol{\Delta}$, $\boldsymbol{\Theta}$, and $\bar{\mathbf{A}}$, respectively. Not only (5.22) evidences the separability of (5.20) across rows of $\mathbf{A}$, but it also presents each of its summands in a standardized quadratic form that can be readily minimized by equating their gradients to zero, namely (define $\mathbf{D}_m := \text{diag}(\boldsymbol{\delta}_m)$ for convenience)

$$(\boldsymbol{\Pi}^T \mathbf{D}_m \boldsymbol{\Pi} + \lambda\mu \mathbf{I})\mathbf{a}_m - \boldsymbol{\Pi}^T \mathbf{D}_m \mathbf{z}_m - \mu\boldsymbol{\theta} = \mathbf{0}, \; m = 1, \ldots, M.$$

Accordingly, the majorization strategy reduces the computational load to M systems of R equations that can be solved in parallel, where R is typically small (cf. the low tensor rank assumption). Collecting the solution of such quadratic programs into the rows of a matrix $\mathbf{A}^*$ yields the minimizer of (5.20), and the update $\mathbf{A} \leftarrow \mathbf{A}^*$ for the BSUM cycle. Such a procedure is presented in Algorithm 3, where analogous updates for $\mathbf{B}$ and $\mathbf{C}$ are carried out cyclically per iteration.

Remark 5.6 A different algorithm for solving (5.13) was put forth in the conference precursor of this paper [23], which cyclically minimizes the columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Distinct from Algorithm 3 that entails *parallel* row-wise updates per factor, iterates in [23] involve *sequential* updates across columns and factors, thus incurring a per iteration complexity of $\mathcal{O}(R(M^3+N^3+P^3))$. Because the factor matrices are tall [$\min(M, N, P) \gg R$], the aforementioned computational load is markedly higher than the one incurred by Algorithm 3, namely $\mathcal{O}((M + N + P)R^3)$.

By virtue of properties i)-iii) in Lemma 5.1, convergence of Algorithm 3 follows readily from that of the BSUM algorithm [148].

Proposition 5.3 *The iterates for* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{C}$ *generated by Algorithm 3 converge to a stationary point of (5.13).*

5.4 Inference for low-rank Poisson tensors

Adoption of the LS criterion in (5.8) assumes in a Bayesian setting that the random $\underline{\mathbf{Z}}$ is Gaussian distributed conditioned on $\underline{\mathbf{X}}$. This section deals with a Poisson-distributed tensor $\underline{\mathbf{Z}}$, a natural

Algorithm 3 : Low-rank tensor imputation (LRTI)

```
function UPDATE_FACTOR($(\mathbf{A}, \mathbf{R}, \mathbf{\Pi}, \underline{\mathbf{\Delta}}, \underline{\mathbf{Z}}, \mu)$)
    Set $\lambda = \lambda_{\max}(\mathbf{R}^{-1})$
    Unfold $\underline{\mathbf{\Delta}}$ and $\underline{\mathbf{Z}}$ over dimension of $\mathbf{A}$ into $\mathbf{\Delta}$ and $\mathbf{Z}$
    Set $\mathbf{\Theta} = (\lambda \mathbf{I} - \mathbf{R}^{-1})\mathbf{A}$
    for $m = 1, \ldots, M$ do
        Select rows $\mathbf{z}_m^T$, $\boldsymbol{\delta}_m^T$, and $\boldsymbol{\theta}_m^T$, and set $\mathbf{D}_m = \text{diag}(\boldsymbol{\delta}_m)$
        Compute $\mathbf{a}_m = (\mathbf{\Pi}^T \mathbf{D}_m \mathbf{\Pi} + \lambda \mu \mathbf{I})^{-1} (\mathbf{\Pi}^T \mathbf{D}_m \mathbf{z}_m + \mu \boldsymbol{\theta}_m)$
        Update $\mathbf{A}$ with row $\mathbf{a}_m^T$
    end for
    return $\mathbf{A}$
end function
Initialize $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ randomly.
while $|\text{cost} - \text{cost\_old}| < \epsilon$ do
    $\mathbf{A} =$ UPDATE_FACTOR$(\mathbf{A}, \mathbf{R}_A, (\mathbf{C} \odot \mathbf{B}), \underline{\mathbf{\Delta}}, \underline{\mathbf{Z}}, \mu)$
    $\mathbf{B} =$ UPDATE_FACTOR$(\mathbf{B}, \mathbf{R}_B, (\mathbf{A} \odot \mathbf{C}), \underline{\mathbf{\Delta}}, \underline{\mathbf{Z}}, \mu)$
    $\mathbf{C} =$ UPDATE_FACTOR$(\mathbf{C}, \mathbf{R}_C, (\mathbf{B} \odot \mathbf{A}), \underline{\mathbf{\Delta}}, \underline{\mathbf{Z}}, \mu)$
    Recalculate cost in $(5.13)$
end while
return $\underline{\hat{\mathbf{X}}}$ with slices $\hat{\mathbf{X}}_{\mathbf{p}} = \mathbf{A}\text{diag}[\mathbf{e}_p^T \mathbf{C}]\mathbf{B}^T$
```

alternative to the Gaussian model when integer-valued data are obtained by counting independent events [48]. Such a model is also well-suited for sparse tensor data, since the Poisson distribution has mass at the origin.

Suppose that the entries z_{mnp} of $\underline{\mathbf{Z}}$ are Poisson distributed, with probability mass function

$$P(z_{mnp} = k) = \frac{x_{mnp}^k e^{-x_{mnp}}}{k!} \tag{5.23}$$

and means given by the corresponding entries in tensor $\underline{\mathbf{X}}$. For mutually-independent $\{z_{mnp}\}$, the log-likelihood $l_{\underline{\mathbf{\Delta}}}(\underline{\mathbf{Z}}; \underline{\mathbf{X}})$ of $\underline{\mathbf{X}}$ given data $\underline{\mathbf{Z}}$ only on the entries specified by $\underline{\mathbf{\Delta}}$, takes the form

$$l_{\underline{\mathbf{\Delta}}}(\underline{\mathbf{Z}}; \underline{\mathbf{X}}) = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{i=1}^{P} \delta_{mnp}[z_{mnp} \log(x_{mnp}) - x_{mnp}] \tag{5.24}$$

after dropping terms $\log(z_{mnp}!)$ that do not depend on $\underline{\mathbf{X}}$.

The choice of the Poisson distribution in (5.23) over a Gaussian one for counting data, prompts minimization of the K-L divergence (5.24) instead of LS [cf. (5.8)] as a more suitable criterion [48]. Still, the entries of $\underline{\mathbf{X}}$ are not coupled in (5.24), and a binding PARAFAC modeling assumption is natural for feasibility of the tensor approximation task under missing data. Mimicking the method for Gaussian data, (nonnegative) Gaussian priors are assumed for the factors of the PARAFAC decomposition. Accordingly, the MAP estimator of $\underline{\mathbf{X}}$ given Poisson-distributed data (entries of $\underline{\mathbf{Z}}$ indexed by $\underline{\boldsymbol{\Delta}}$) becomes

$$\begin{aligned}\hat{\underline{\mathbf{Z}}} := \arg\min_{\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}\}\in\mathcal{T}} \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{i=1}^{P} \delta_{mnp}(x_{mnp} - z_{mnp}\log(x_{mnp}))\\ + \frac{\mu}{2}\left[\text{Tr}\left(\mathbf{A}^T\mathbf{R}_A^{-1}\mathbf{A}\right)+\text{Tr}\left(\mathbf{B}^T\mathbf{R}_B^{-1}\mathbf{B}\right)+\text{Tr}\left(\mathbf{C}^T\mathbf{R}_C^{-1}\mathbf{C}\right)\right] \qquad (5.25)\end{aligned}$$

over the feasible set $\mathcal{T}:=\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C} : \mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0},\mathbf{C}\geq\mathbf{0},\mathbf{X}_p = \mathbf{A}\text{diag}\left[\mathbf{e}_p^T\mathbf{C}\right]\mathbf{B}^T,\ \ p = 1,\ldots,P\}$, where the symbol $\geq$ should be understood to imply entry-wise nonnegativity.

Remark 5.7 *The parameter μ in (5.25) was introduced to add flexibility in varying the sparsity level of $\hat{\underline{\mathbf{Z}}}$. However, derivation of the Poisson MAP estimator with Gaussian priors leads to $\mu = 1$, which is used as the default value in the applications of Section VI and is corroborated to be a reasonable choice in Fig. 4. The reason behind μ taking a specific value is that in the Poisson distribution (5.23) the mean and variance are related (in fact they are equal). This should be contrasted with the MAP estimator in Section IV, where μ equals σ^2 which is a free parameter under the Gaussian data model (5.11).*

With the aid of the Representer Theorem, it is also possible to interpret (5.25) as a variational estimator in RKHS, with K-L analogues to (5.14)-(5.16), so that the conclusions thereby regarding smoothing, prediction and prior covariance estimation carry over to the low-rank Poisson imputation method (5.25).

5.4.1 BSUM algorithm

A K-L counterpart of the LRTI algorithm is developed in this section, that provably converges to a stationary point of (5.25). An alternating-minimization scheme is adopted once again, which

Algorithm 4 : Low-rank Poisson-tensor imputation (LRPTI)

```
function UPDATE_FACTOR($\mathbf{A}, \mathbf{R}, \boldsymbol{\Pi}, \underline{\boldsymbol{\Delta}}, \underline{\mathbf{Z}}, \mu$)
    Set $\lambda = \lambda_{\max}(\mathbf{R}^{-1})$
    Unfold $\underline{\boldsymbol{\Delta}}$ and $\underline{\mathbf{Z}}$ over dimension of $\mathbf{A}$ into $\boldsymbol{\Delta}$ and $\mathbf{Z}$
    Compute $\mathbf{S} = \frac{\mathbf{A}}{\lambda\mu} \circledast \left(\frac{\boldsymbol{\Delta} \circledast \mathbf{Z}}{\mathbf{A}\boldsymbol{\Pi}^T}\boldsymbol{\Pi}\right)$ (element-wise division)
    Compute $\mathbf{T} = \frac{1}{2\lambda\mu}\left(\mu(\lambda\mathbf{I} - \mathbf{R}^{-1})\mathbf{A} - \boldsymbol{\Delta}\boldsymbol{\Pi}\right)$
    Update $\mathbf{A}$ with entries $a_{mr} = t_{mr} + \sqrt{t_{mr}^2 + s_{mr}}$
    return $\mathbf{A}$
end function
Initialize $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ randomly.
while $|\text{cost} - \text{cost\_old}| < \epsilon$ do
    $\mathbf{A} =$ UPDATE_FACTOR($\mathbf{A}, \mathbf{R}_A, (\mathbf{C} \odot \mathbf{B}), \underline{\boldsymbol{\Delta}}, \underline{\mathbf{Z}}, \mu$)
    $\mathbf{B} =$ UPDATE_FACTOR($\mathbf{B}, \mathbf{R}_B, (\mathbf{A} \odot \mathbf{C}), \underline{\boldsymbol{\Delta}}, \underline{\mathbf{Z}}, \mu$)
    $\mathbf{C} =$ UPDATE_FACTOR($\mathbf{C}, \mathbf{R}_C, (\mathbf{B} \odot \mathbf{A}), \underline{\boldsymbol{\Delta}}, \underline{\mathbf{Z}}, \mu$)
    Recalculate cost in $(5.25)$
end while
return $\underline{\hat{\mathbf{X}}}$ with slices $\hat{\mathbf{X}}_{\mathbf{p}} = \mathbf{A}\text{diag}(\mathbf{e}_p^T\mathbf{C})\mathbf{B}^T$
```

optimizes (a suitable upper-bound of) (5.25) cyclically w.r.t. one factor matrix, while holding the others fixed.

In the sequel, the goal is to arrive at a suitable expression for the cost in (5.25), when viewed only as a function of e.g., $\mathbf{A}$. To this end, let matrix $\mathbf{Z} := [\mathbf{Z}_1, \ldots, \mathbf{Z}_P] \in \mathbb{N}^{M \times NP}$ denote the mode-1 unfolding of $\underline{\mathbf{Z}}$, and likewise for $\boldsymbol{\Delta} := [\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_P] \in \{0,1\}^{M \times NP}$ and $\mathbf{X} := [\mathbf{X}_1, \ldots, \mathbf{X}_P] \in \mathbb{R}_+^{M \times NP}$. Based on these definitions, (5.24) can be written as

$$l_{\boldsymbol{\Delta}}(\mathbf{Z}; \mathbf{X}) = \mathbf{1}_M^T(\boldsymbol{\Delta} \circledast [\mathbf{X} - \mathbf{Z} \circledast \log(\mathbf{X})])\mathbf{1}_{NP} \tag{5.26}$$

where $\mathbf{1}_M$, $\mathbf{1}_{NP}$ are all-one vectors of dimensions M and NP respectively, and $\log(\cdot)$ should be understood entry-wise. The log-likelihood in (5.26) can be expressed in terms of $\mathbf{A}$, and the Khatri-Rao product $\boldsymbol{\Pi} := \mathbf{B} \odot \mathbf{C}$ by resorting again to (5.21). Substituting (5.21) into (5.26) one arrives at the desired expression for the cost in (5.25) as a function of $\mathbf{A}$, namely

$$f(\mathbf{A}) := \mathbf{1}_M^T(\boldsymbol{\Delta} \circledast [\mathbf{A}\boldsymbol{\Pi} - \mathbf{Z} \circledast \log(\mathbf{A}\boldsymbol{\Pi}^T)])\mathbf{1}_{NP} + \frac{\mu}{2}\text{Tr}\left(\mathbf{A}^T\mathbf{R}_A^{-1}\mathbf{A}\right).$$

A closed-form minimizer $\mathbf{A}^\star$ for $f(\mathbf{A})$ is not available, but since $f(\mathbf{A})$ is convex one could in principle resort to an iterative procedure to obtain $\mathbf{A}^\star$. To avoid extra inner iterations, the approach here relies again on the BSUM algorithm [148].

For $\bar{\mathbf{A}}$ given, consider the *separable* function

$$g(\mathbf{A}, \bar{\mathbf{A}}) := \mu\lambda \sum_{m,r=1}^{M,R} \Big(\frac{a_{mr}^2}{2} - 2t_{mr}a_{mr} - s_{mr}\log(a_{mr}) + u_{mr} \Big) \tag{5.27}$$

where $\lambda := \lambda_{\max}(\mathbf{R}_A^{-1})$, and parameters s_{mr}, t_{mr}, and u_{mr} are defined in terms of $\bar{\mathbf{A}}$, $\mathbf{Z}$, $\boldsymbol{\Delta}$, $\boldsymbol{\Pi}$, and $\boldsymbol{\Theta} := \left(\lambda\mathbf{I} - \mathbf{R}_A^{-1}\right)\bar{\mathbf{A}}$ by

$$s_{mr} := \frac{1}{\lambda\mu}\sum_{k=1}^{NP}\frac{\delta_{mk}z_{mk}\bar{a}_{mr}\pi_{kr}}{\sum_{r'=1}^{R}\bar{a}_{mr'}\pi_{kr'}},$$

$$t_{mr} := \frac{1}{2\lambda\mu}\Big(\mu\theta_{mr} - \sum_{k=1}^{NP}\pi_{kr}\delta_{mk}\Big)$$

and $u_{mr} := \frac{1}{\lambda\mu}\left(\theta_{mr}\bar{a}_{mr} + \sum_{k=1}^{NP}\delta_{mk}z_{mk}\bar{a}_{mr}\pi_{kr}\upsilon_{mrk}\right)$, with $\upsilon_{mrk} := \log(\bar{a}_{mr}\pi_{kr}/\sum_{r'=1}^{R}\bar{a}_{mr'}\pi_{kr'})$ $/\sum_{r'=1}^{R}\bar{a}_{mr'}\pi_{kr'}$. As asserted in the following lemma, $g(\mathbf{A}, \bar{\mathbf{A}})$ majorizes $f(\mathbf{A})$ at $\bar{\mathbf{A}}$ and satisfies the technical conditions required for the convergence of BSUM (see the D D.0.19 for a proof.)

Lemma 5.2 *Function $g(\mathbf{A}, \bar{\mathbf{A}})$ in (5.27) satisfies the following properties*

i) $f(\bar{\mathbf{A}}) = g(\bar{\mathbf{A}}, \bar{\mathbf{A}})$*;*

ii) $\frac{d}{d\mathbf{A}}f(\mathbf{A})|_{\mathbf{A}=\bar{\mathbf{A}}} = \frac{d}{d\mathbf{A}}g(\mathbf{A}, \bar{\mathbf{A}})|_{\mathbf{A}=\bar{\mathbf{A}}}$*; and,*

iii) $f(\mathbf{A}) \leq g(\mathbf{A}, \bar{\mathbf{A}}),\ \forall\mathbf{A}$.

Moreover, $g(\mathbf{A}, \bar{\mathbf{A}})$ *is minimized at* $\mathbf{A} = \mathbf{A}_g^\star$ *with entries* $a_{g,mr}^\star := t_{mr} + \sqrt{t_{mr}^2 + s_{mr}}$.

Lemma 5.2 highlights the reason behind adopting the majorizing function $g(\mathbf{A}, \bar{\mathbf{A}})$ in the proposed BSUM algorithm: (5.27) is separable across the entries of its matrix argument, and hence it admits a closed-form minimizer given by the MR scalars $a_{g,mr}^\star$. The resulting updates $\mathbf{A} \leftarrow \mathbf{A}_g^*$ are tabulated under Algorithm 4, where analogous updates for $\mathbf{B}$ and $\mathbf{C}$ are carried out cyclically per iteration.

By virtue of properties i)-iii) in Lemma 5.2, convergence of Algorithm 4 follows readily from the general convergence theory available for the BSUM algorithm [148].

Proposition 5.4 *The iterates for* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{C}$ *generated by Algorithm 4 converge to a stationary point of (5.25).*

A related algorithm, abbreviated as CP-APR can be found in [48], where the objective is to find the tensor's low-rank factors per se. The LRPTI algorithm here generalizes CP-APR by focusing on recovering missing data, and incorporating prior information through rank regularization. In terms of convergence to a stationary point, the added regularization allows for lifting the assumption on the linear independence of the rows of $\mathbf{\Pi}$, as required by CP-APR [48] - an assumption without a straightforward validation since iterates $\mathbf{\Pi}$ are not accessible beforehand.

5.5 Numerical Tests

5.5.1 Simulated Gaussian data

Synthetic tensor-data of dimensions $M \times N \times P = 16 \times 4 \times 4$ were generated according to the Bayesian tensor model described in Section 5.3. Specifically, entries of $\underline{\mathbf{Z}}$ consist of realizations of Gaussian random variables generated according to (5.11), with means specified by entries of $\underline{\mathbf{X}}$ and variance scaled to yield an SNR of -20dB . Tensor $\underline{\mathbf{X}}$ is constructed from factors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, as in (5.7). Matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ have $R = 6$ columns containing realizations of independent zero-mean, unit-variance, Gaussian random variables.

A quarter of the entries of $\underline{\mathbf{Z}}$ were removed at random and reserved to evaluate performance. The remaining seventy five percent of the data were used to recover $\underline{\mathbf{Z}}$ considering the removed data as missing entries. Method (5.8) was employed for recovery, as implemented by the LRTI Algorithm, with regularization $\frac{\mu}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2)$ resulting from setting $\mathbf{R}_A = \mathbf{I}_M$, $\mathbf{R}_B = \mathbf{I}_N$, and $\mathbf{R}_C = \mathbf{I}_P$.

The relative recovery error between $\hat{\underline{\mathbf{Z}}}$ and data $\underline{\mathbf{Z}}$ was computed, along with the rank of the recovered tensor, as a measure of performance. Fig. 5.3 depicts these figures of merit averaged over 100 repetitions of the experiment, across values of μ varying on the interval $10^{-5}\mu_{\max}$ to $\mu_{\max}$, which is computed as in Corollary 5.1. Fig 5.3 (bottom) shows that the LRTI algorithm is successful in recovering the missing entries of $\underline{\mathbf{Z}}$ up to -10dB for a wide range of values of μ, presenting a minimum at $\mu = 10^{-2}\mu_{\max}$. This result is consistent with Fig. 5.3 (top), which shows that rank

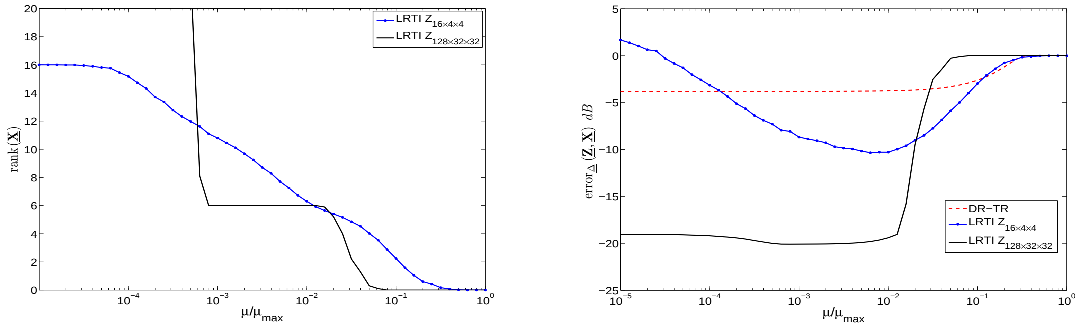


Figure 5.3: Performance of the low-rank tensor imputation method as a function of μ; (top) rank of the tensor as recovered by (5.8) averaged over 100 test repetitions, compared to the DR-TR algorithm in [73]; (bottom) relative recovery error.

$R^* = 6$ is approximately recovered at the minimum error. Fig. 5.3 (top) also corroborates the low-rank inducing effect of (5.8), with the recovered rank varying from the upper bound $\bar{R} = NP = 16$ to $R = 0$, as μ is increased, and confirms that the recovered tensor is null at $\mu_{\max}$ as asserted by Corollary 5.1.

Fig. 5.3 (bottom) also depicts the imputation error that results from applying the Douglas-Rachford (DR-TR) method for tensor recovery in [73]. Since the DR-TR method is not designed to capture the PARAFAC rank, the LRTI offers better performance in terms of recovery error when $\underline{\mathbf{Z}}$ indeed abides to a low-rank model. In addition, Fig. 5.3 depicts the LRTI results obtained for a larger tensor $\underline{\mathbf{Z}}$ of dimensions $M = 128$, $N = 32$, and $P = 32$, and rank $R = 6$. Similar to the prior simulation setting where $M = 16$, $N = 4$, and $P = 4$, the minimum error is again attained at a similar value of $\mu/\mu_{\max}$, where the true rank is recovered.

5.5.2 Simulated Poisson data

The synthetic example just described was repeated for the low-rank Poisson-tensor model described in Section 5.4. Specifically, tensor data of dimensions $M \times N \times P = 16 \times 4 \times 4$ were generated according to the low-rank Poisson-tensor model of Section 5.4. Entries of $\underline{\mathbf{Z}}$ consist of realizations of Poisson random variables generated according to (5.23), with means specified by entries of $\underline{\mathbf{X}}$. Tensor $\underline{\mathbf{X}}$ is again constructed as in (5.7) from factors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ having $R = 6$ columns, containing the absolute value of realizations of independent Gaussian random variables scaled to yield

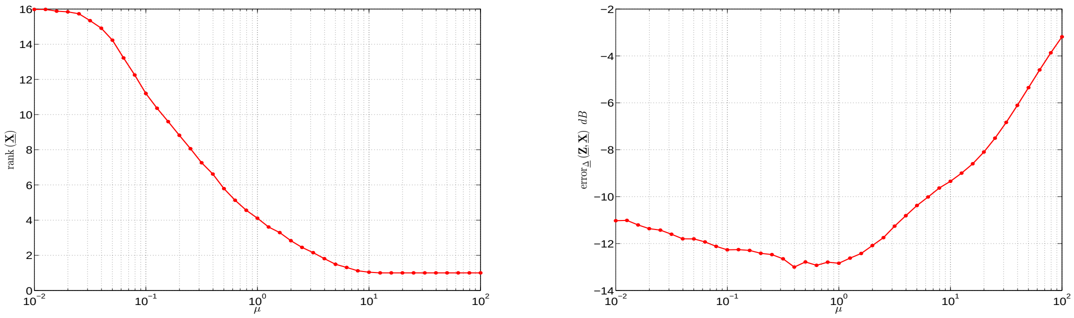


Figure 5.4: Performance of the low-rank Poisson imputation method as function of the regularizing parameter μ; (top) rank of the recovered tensor averaged over 100 test repetitions, (bottom) relative recovery error.

$\mathbb{E}[x_{mnp}] = 100$. Half of the entries of $\underline{\mathbf{Z}}$ were considered missing to be recovered from the remaining half. Method (5.25) was employed for recovery, as implemented by the LRPTI Algorithm, with regularization $\frac{\mu}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2)$.

Fig. 5.4 shows the estimated rank and recovery error over 100 realizations of the experiment, for μ in the interval 0.01 to 100. The recovery error in Fig. 5.4 (bottom) exhibits a minimum of -15dB at $\mu = 1$, where the rank $R^* = 6$ is recovered [cf. Fig. 5.4 (top).] The low-rank inducing effect of (5.8) is again corroborated by the decreasing trend in Fig. 5.4 (top), but in this case the rank is lower bounded by $R = 1$, because the K-L fitting criterion prevents (5.25) from yielding a null estimate $\hat{\underline{\mathbf{Z}}}$.

5.5.3 MRI data

Estimator (5.14) was tested against a corrupted version of the MRI brain data set 657 from the Internet brain segmentation repository [4]. The tensor $\underline{\mathbf{Z}}$ to be estimated corresponds to a three-dimensional MRI scan of the brain comprising a set of $P = 18$ images, each of $M \times N = 256 \times 196$ pixels. Fifty percent of the data is removed uniformly at random together with the whole slice $\mathbf{Z}_n,\ n = 50$. Fig. 5.6 depicts the results of applying estimator (5.14) to the remaining data, which yields a reconstruction error of -11.49dB. The original slice $\mathbf{Z}_p,\ p = 5$, its corrupted counterpart, and the resulting estimate are shown on top and center left.

Parameter μ is set equal to σ^2 as per Remark 5.5. The noise variance is estimated from 150

entries at each corner of $\mathbf{Z}_p,\ p = 1, \ldots, P$, which are assumed to contain background noise only. Covariance matrices $\mathbf{K}_\mathcal{M}$, $\mathbf{K}_\mathcal{N}$ and $\mathbf{K}_\mathcal{P}$ are estimated from six additional tensor samples containing complementary scans of the brain also available at [4]. Fig. 5.6 (center right) represents the covariance matrix $\mathbf{K}_\mathcal{N}$ for column slices perpendicular to $\mathbf{Z}_p$, showing a structure that reflects symmetries of the brain. This correlation is the key enabler for the method to recover the missing slice up to -9.60dB (see Fig. 5.6 (bottom)) by interpolating its a priori similar parallel counterparts.

For $\mu = \sigma^2$, rank($\underline{\hat{\mathbf{X}}}$) $= R = 100$, i.e., the rank is not reduced but remains equal to the number of columns R set for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The results are weakly dependent on the selection of R, with a reconstruction error in the interval $[-10.50, -12.66]$dB for R between 50 and 200. If μ is increased the rank of the estimated tensor is reduced, but the recovery error is increased. For instance, selecting $\mu = 0.1\mu_{\max}$, results in rank($\underline{\hat{\mathbf{X}}}$) $= 14 < R$, but the recovery error increases to -7.8dB. It is thus noticed that (5.14) is able to regularize the tensor taking into account correlations, but without necessarily forcing a reduced rank.

These properties are further appreciated when comparing the performance of LRTI with state-of-the-art methods for tensor completion. The missing entries of $\underline{\mathbf{Z}}$ were imputed via the CP-WOPT algorithm in the Tensor Toolbox 2.5 [5]. CP-WOPT was run 100 times with candidate values for the rank between 1 and 100, yielding higher reconstruction errors in the interval $[0, -5.98]$dB.

All in all, the experiment evidences the merits of low-rank PARAFAC decomposition for modeling a tensor, the ability of the Bayesian estimator (5.13) in recovering missing data, and the usefulness of incorporating correlations as side information.

On account of the comprehensive analysis of three-way MRI data arrays in [50], and the nonnegative PARAFAC decomposition advanced thereby, inference of tensors with nonnegative continuous entries will be pursued as future research, combining methods and algorithms in Sections 5.3 and 5.4 of this paper.

5.5.4 RNA sequencing data

The RNA-Seq method described in [131] exhaustively counts the number of RNA transcripts from yeast cells. The reverse transcription of RNA molecules into cDNA is achieved by $P = 2$ alternative methods, differentiated by the use of oligo-dT, or random-hexonucleotide primers. These cDNA

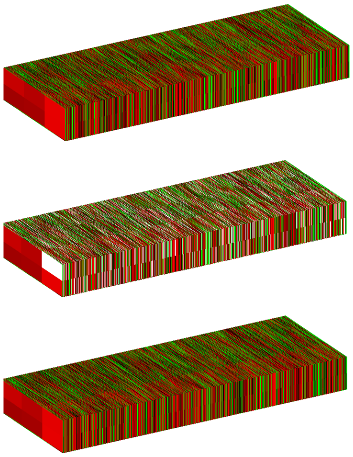


Figure 5.5: Imputation of sequencing count data via LRPTI; (top) original data; (center) data with missing entries; (bottom) recovered tensor.

molecules are sequenced to obtain counts of RNA molecules across $M = 6,604$ genes on the yeast genome. The experiment was repeated in [131] for a biological and a technological replicate of the original sample totalling $N = 3$ instances per primer selection. The data are thus organized in a tensor of dimensions $6,604 \times 3 \times 2$ as shown in Fig. 5.5 (top), with integer data that are modeled as Poisson counts. Fifteen percent of the data is removed and reserved for assessing performance. The missing data are represented in white in Fig. 5.5 (center).

The LRPTI algorithm is run with the data available in Fig. 5.5 (center) producing the recovered tensor depicted in Fig. 5.5 (bottom). The parameter μ is set equal to 1 as per Remark 5.7, resulting in rank($\underline{\hat{\mathbf{X}}}$) $= NP = 6$ and a recovery error of -15dB.

5.6 Concluding summary

It was shown in this paper that regularizing with the Frobenius-norm square of the PARAFAC decomposition factors, controls the tensor's rank by inducing sparsity in the vector of amplitudes of its rank-one components. A Bayesian method for tensor completion was developed based on this property, introducing priors on the tensor factors. It was argued, and corroborated numerically, that this prior information endows the completion method with extra capabilities in terms of smoothing and extrapolation. It was also suggested through a parallelism between Bayesian and RKHS inference, that the prior covariance matrices can be obtained from (sample) correlations among the tensor's slices. In such a probabilistic context, generic distribution models for the data lead to multiple fitting criteria. Gaussian and Poisson processes were especially considered by developing algorithms that minimize the regularized LS and K-L divergence, respectively.

Numerical tests on synthetic data corroborated the low-rank inducing property, and the ability of the completion method to recover the "ground-truth" rank, while experiments with brain images and gene expression levels in yeast served to evaluate the method's performance on real datasets.

Although the results and algorithms in this paper were presented for three-way arrays, they are readily extendible to higher-order tensors or reducible to the matrix case.

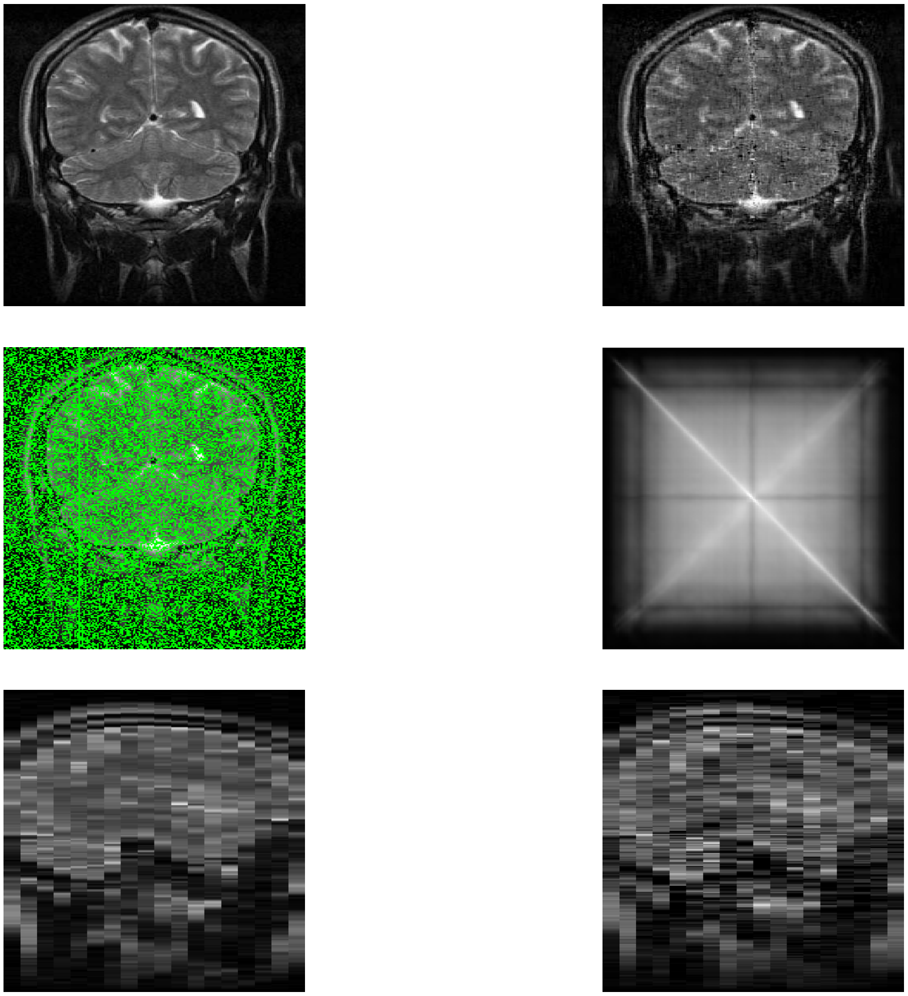


Figure 5.6: Results of applying (5.14) to the MRI brain data set 657 [4]. (top) Original and recovered fibers $\mathbf{Z}_p$ and $\hat{\mathbf{Z}}_p$ for $p = 5$. (center) Input fiber $\mathbf{Z}_p,\ p = 5$ with missing data, and covariance matrix $\mathbf{K}_{\mathcal{N}}$. (bottom) Original and recovered columns $\mathbf{Z}_n$ and $\hat{\mathbf{Z}}_n$ for the position $n = 50$ in which the whole input slice is missing.)

Chapter 6

Future Work

This dissertation dealt with sparse models for gene-regulatory and wireless-cognitive networks, reporting novel results on topology inference, design of experiments, nonparametric basis pursuit, and tensor completion. This final chapter is intended to outline future research directions that are envisioned to emerge from these results.

6.1 Topology changes and cell differentiation

With the long term objective of devising a thorough model of the cell that includes stochastic biological processes and impulse responses, the next quest on SEMs is to understand the universality of GRNs. It is accepted that DNA sequences are identical for all cells of an individual, and that the differentiation of a blood cell from a neuron, for instance, is given by the cell's ability to activate different processes [9]. The question is whether the GRN is invariant across cells with different regions activated, or, if the differentiation includes the GRN itself. Specifically, recent results on topology change detection could be explored in the context of cyclic and directed networks [11]. Detecting sparse changes in the topology of GRNs across time could lead to the detection of cancer triggering effects, or, to the statistical description of the developmental stages of an *embryo* [174].

6.2 Network abridgement for big data processing

Assessing causality is a challenging task, especially if not all nodes of a network are accessible. But considering the huge number of network nodes could lead to computationally infeasible problems when dealing with big data, as is the case of SEMs for human GRN where the number of variables is on the order of 4×10^8. Usually, regulatory pathways are studied in the laboratory by focusing on small subset of genes or proteins. This motivates looking for a reduced-size equivalent of a large network. Norton and Thévenin equivalent circuits attest that this goal could be feasible, and shed light on how to tackle this challenge using linear algebra tools [63]. A key research issue is to specify additional perturbation data needed for identifying the surrogate network. Addressing this issue in a general context would facilitate universal applicability to big-data networked scenarios.

6.3 Active link prediction in communication networks

Network prediction was studied in Chapter 4 to infer rates over network links, treated as multi-variable stochastic processes, assuming a static topology. The next step is to allow the topology to change dynamically imposing time-evolving sparse constraints to capture (dis)appearing edges. This is envisioned to be possible by jointly leveraging the stochastic SEMs and network reduction proposed in the previous sections, together with the blind NBP approaches of Chapter 4. Application of active link prediction will impact not only communication networks but also biological systems where GRN anomalies would be anticipated, and social networks where critical node interactions could be foreseen.

6.4 Joint prediction and energy sharing over the Smart Grid

The next future research direction pertains to the distinct generation-consumption interactions inherent to the Smart Grid. Renewable sources; eg., wind or solar power, will be installed at end-user premises or at remote locations. The main challenge for the use of these technologies is that they are intermittent, posing the risk of a significant outage probability [142]. This aggravates the basic challenge in energy management, where power consumption have a cyclostationary component that is typically not aligned with generation patterns. To cope with this generation-consumption mismatch and provide stable power delivery, reservoirs are being installed to enhance system stability [59]. All in all, the power grid can be modeled as an interconnected three-layer network of generators, reservoirs, and consumers. The power flows from generators to reservoirs, and from reservoirs to consumers can be viewed as decision variables to be designed for controlling the generation cost, under a prescribed limit on a desirable outage probability. A key enabler for such a design is a predictor of the renewable power availability and power consumption. The envisioned forecasting approach will jointly leverage the network flow predictor of Chapter 4, and the tensor extrapolation algorithms of Chapter 5, taking into account the correlations between generation and consumption patterns.

6.5 Spatially-aware communications in cognitive networks

The spatial awareness obtained through spectrum cartography in Chapter 4 will positively affect the transmission opportunities of CRNs, especially when they are spread across a wide area where the space-invariant assumption on spectrum occupancy is not valid. Neighbors in a multi-hop architecture will transmit over a frequency band that is unoccupied within their coverage range. Cognizant of the spectrum maps, the next relay-CR in the path will be selected by analyzing the candidate's interference temperature. The frequency band and transmission power will be stochastically adapted to optimally trade off three objectives: i) procure the maximum possible signal-to-interference-plus-noise ratio at the receiver; ii) minimize interference to primary users by estimating the channel-gain across space [56]; and, iii) maintain orthogonality to neighboring cognitive radio transmissions. The corresponding optimization problem could even incorporate higher communication layers if CRs are to withhold their transmissions, which will increase transmission backlogs, when all frequency bands are momentarily in use [150].

6.6 Performance analysis of sparse regression in drug targeting

When the sparse regression vector in Chapter 3 has multiple nonzero entries, the proposed correlation rule is suboptimal. This motivates exploring advanced subset selection methods using e.g., the least-absolute shrinkage and selection operator (Lasso), which offers finite-sample as well as asymptotic performance guarantees [35]. However, the so-termed restricted isometry assumptions required for the aforementioned guarantees are not satisfied by the matrix of double mutant profiles, since genes present a certain degree of redundancy in their functionality to protect the cell [110]. The Elastic-Net augments the Lasso cost with a quadratic regularizer, which is particularly useful for sparse estimation when the regression matrix entails highly correlated columns [202]. In particular, it has been proved that Elastic-Net assigns equal regression coefficients to identical regressors, thus implicitly performing subset selection jointly with clustering. This attribute promotes the Elastic-net as a viable alternative for drug targeting. An optimal design of experiments based on the Elastic-net is thus an additional path for future research, involving on the way metric of estimation performance of Elastic-net in the finite case, which entails an open problem.

6.7 Atomic norm regularization for tensor completion

The atomic norm of a tensor presented in Chapter 5 is a proper norm; thus it is convex and using it as a regularizer for tensor completion would lead to a convex problem. Although common wisdom suggests that convex problems are “easy” to solve, this assumes that the cost can be computed in polynomial time. But this is not the case for the atomic norm which is defined through the PARAFAC decomposition. In general, finding global optima in tensor problems is challenging, but worth investigating given the importance of tensor algebra, as demonstrated in this thesis through multiple applications. New results on the convergence of coordinate descent algorithms for non-convex problems [119], [148], together with the methods for tensor completion developed in Chapter 5, prompts deeper convergence analysis of low-rank tensor completion algorithms.

6.8 Signal detection via local linear embedding

The Nyquist-Shannon Theorem establishes that if a generic bandlimited signal is to be reconstructed from samples, the sampling rate should be at least twice the signal’s bandwidth. But this is not the case if extra information is available. On the other extreme, reconstructing signals formed by convolving a known pulse shaper with points in a binary-phase-key-shifting constellation, requires only a single bit. Recent connections between DL and local linear embedding [168], together with the advances in nonparametric DL in Chapter 4, open new possibilities to blindly reconstruct signals belonging to a manifold. This is of practical relevance in image and video processing, as well as in cognitive radio applications, where signals exhibit limited variability and can thus be modeled as lying on an unknown manifold, which can be blindly learned using data acquired at a reduced rate.

Bibliography

[1] [Online]. Available: http://internet2.edu/observatory/archive/data-collections.html.

[2] FCC Spectrum Policy Task Force, 2002

[3] *IEEE Standard for Info. Tech.-Telecomms. and Info. Exchchange between Systems-Local and Metropolitan Area Nets., Part 11: Wir. LAN MAC and PHY Specifications,* IEEE Standard 802.11-2012, pp. 1-1184, Mar. 2012.

[4] Internet brain segmentation repository, "MR brain data set 657," *Center for Morphometric Analysis at Massachusetts General Hospital,* available at *http://www.cma.mgh.harvard.edu/ibsr/.*

[5] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mrup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems,* vol. 106, no. 1, pp. 41-56, 2011.

[6] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *J. Machine Learning Res.,* vol. 10, pp. 803-826, Mar. 2009.

[7] P. Agrawal and N. Patwari, "Correlated link shadow fading in multihop wireless network," *IEEE Trans. on Wireless Comm.,* vol. 8, no. 8, pp. 4024-4036, Aug. 2009.

[8] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, "Informed spectrum usage in cognitive radio networks: Interference cartography," in *Proc. of 19th Intl. Symp. on Personal, Indoor and Mobile Radio Comms.*, Cannes, France, Aug./Sep. 2008, pp. 1–5.

[9] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell,* 5th edition, Garland Science, NY, 2007.

[10] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proc. of the Natl. Academy of Science,* vol. 97, no. 18, pp. 10101-10106, 2000.

[11] D. Angelosante, G. B. Giannakis, and N. D. Sidiropoulos, “Estimating multiple frequency-hopping signal parameters via sparse linear regression,” *IEEE Trans. Signal Process.*, 2010.

[12] B. D. Anson, J. Ma, J.-Q. He, “Identifying Cardiotoxic Compounds,” *Genetic Engineering & Biotechnology News*, 29, no. 9 pp. 3435, May 2009.

[13] J. E. Aten, T. F. Fuller, A. J. Lusis, and S. Horvath S, “Using genetic markers to orient the edges in quantitative trait networks: The NEO software,” emphBMC Syst Biol, vol. 2 no. 34, pp. 1-21, 2008.

[14] L. Balzano, R. Nowak, and B. Recht, “Online identification and tracking of subspaces from highly incomplete information,” in *Proc. of Allerton Conference on Communication, Control, and Computing*, Monticello, USA, Jun. 2010.

[15] A. Baryshnikova, M. Costanzo, Y. Kim, et. al, ”Quantitative analysis of fitness and genetic interactions in yeast on a genome-wide scale,” *Nature Methods*, vol. 7, pp. 10171024, Nov. 2010.

[16] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, et al., “Reverse engineering of regulatory networks in human B cells,” *Nat Genet* vol. 37, pp. 382-90, 2005.

[17] J. A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol 58, pp. 1847-1862, 2010.

[18] J. A. Bazerque and G. B. Giannakis, ”Nonparametric Basis Pursuit via Sparse Kernel-based Learning,” IEEE Signal Processing Magazine, July 2013 (to appear).

[19] J. A. Bazerque, G. Mateos, and G. B. Giannakis, “Basis pursuit for spectrum cartography,” *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 22-27, 2011.

[20] J. A. Bazerque, G. Mateos, and G. B. Giannakis, “Distributed Lasso for In-Network Linear Regression,” *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Dallas, Texas, March 14-19, 2010.

[21] J. A. Bazerque, G. Mateos, and G. B. Giannakis, “Group-Lasso on splines for spectrum cartography,” *IEEE Trans. on Signal Proc.*, vol. 59, no. 10, pp. 4648-4663, Oct. 2011.

[22] J. A. Bazerque, G. Mateos, and G. B. Giannakis, “Inference of Poisson Count Processes using Low-Rank Tensor Data,” *Proc. of Intl. Conf. on Acoust., Speech, and Signal Processing*, Vancouver, Canada, May 26-31, 2013.

[23] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Nonparametric Low-Rank Tensor Imputation," *Proc. of IEEE Workshop on Statistical Signal Processing,* pp. 888-891, Ann Arbor, USA, August 5-8, 2012.

[24] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank Regularization in Bayesian Inference for Tensor Completion and Extrapolation," IEEE Transactions on Signal Processing," Apr. 2013 (in review); also available online arXiv:1301.7619v1 [cs.IT].

[25] D. P. Bertsekas and J. N. Tsitsiklis, "Parallel and Distributed Computation: Numerical Methods," *Athena-Scientific,* 1999.

[26] G. Bollag, P. Hirth, J. Tsai, et al., "Clinical efficacy of a RAF inhibitor needs broad target blockade in BRAF-mutant melanoma", *Nature,* vol. 467, no.7315, pp. 596599 Sep. 2010. doi:10.1038/nature09454. PMC 2948082. PMID 20823850.

[27] K. A. Bollen KA *Structural Equations with Latent Variables* Wiley-Interscience, 1989.

[28] R. Bonneau, D. Reiss, P. Shannon, M. Facciotti, L. Hood, et al., "The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*," *Genome Biol,* vol 7, no. 5, , 2006.

[29] M. L. Boulland,j. Marquet, V. Molinier-Frenkel, P. M oller, C. Guiter C, et al., "Human IL4I1 is a secreted l-phenylalanine oxidase expressed by mature dendritic cells that inhibits T-lymphocyte proliferation," *Blood* vol. 110, pp. 220-227, 2007.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization,* Cambridge University Press, 2004.

[31] L. Brouwers, M. Iskar, G. Zeller, V. van Noort, and P. Bork, "Network Neighbors of Drug Targets Contribute to Drug Side-Effect Similarity," *PLoS ONE,* vol 6, no. 7, pp. 1-7, 2011.

[32] A. j. Butte, P. Tamayo, D. Slonim, T. R. Golub, I. S. Kohane, "Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks," *Proc Natl Acad Sci,* vol. 97, no. 12, pp. 182-186, 2000.

[33] J. F. Cai, E. J. Candes and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimization,* vol. 20, pp. 1956–1982, Jan. 2010.

[34] E. J. Candés and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE,* vol. 98, pp. 925–936, June 2010.

[35] E. J. Candès and Y. Plan, "Near-ideal model selection by $\ell - 1$ minimization," *Annals of Statistics,* vol. 37, no. 5A, pp. 2145-2177, 2009.

[36] E. J. Candés, and T. Tao, "Decoding by linear programming," *IEEE Trans. on Info. Theory,* vol. 51, no. 12, pp. 4203-4215, Dec. 2005.

[37] B. P. Carlin, T. A. Louis, *Bayesian Methods for Data Analysis*, 3rd edition, Chapman and Hall/CRC, 2008.

[38] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Gene Network Inference via Sparse Structural Equation Modeling with Genetic Perturbations," *Proc. of IEEE Intl. Workshop on Genomic Signal Proc. and Statistics,* San Antonio, TX, December 4-6, 2011.

[39] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Sparse Structural Equation Modeling for Inference of Gene Regulatory Networks Exploiting Genetic Perturbations," *PLoS, Computational Biology,* 2013 (to appear).

[40] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, vol. 12, no. 6, Dec. 2012.

[41] Y. M. Chang, "Inferring relationships between somatic cell score and milk yield using simultaneous and recursive models," *J Dairy Sci,* vol. 90, pp. 3508-3521, 2007.

[42] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters,* vol. 14, no. 10, pp. 707-710, Oct. 2007.

[43] S. S. Chavana, W. Tiana, K. Hsueha, D. Jawaheerd, P.K. Gregersend, et al., "Characterization of the humanhomolog of the IL-4 induced gene-1," *Proc Natl Acad Sci*, vol. 1576, pp. 7080, 2002.

[44] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Computing*, vol. 20, no. 1, pp. 33-61, Dec. 1998.

[45] L. S. Chen, F. Emmert-Streib, J. D. Storey, "Harnessing naturally randomized transcription to infer regulatory relationships among genes," *Genome Biol* vol. 8, 2007.

[46] J. Chen, and Y. Saad, "On the tensor SVD and the optimal low-rank othogonal approximation of tensors," *SIAM Journal on Matrix Analysis and Applications (SIMAX),* vol. 30, no. 4, pp. 1709-1734, 2009.

[47] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, Mar. 2012.

[48] E. C. Chi and T. G. Kolda, “On tensors, sparsity, and nonnegative factorizations,” *SIAM Journal on Matrix Analysis and Applications,* Dec. 2012 (to appear; see also arXiv:1112.2414v3 [math.NA]).

[49] C. C. Chu and W. E. Paul, “An interleukin 4-induced mouse B cell gene isolated by cDNA representational difference analysis,” *Proc Natl Acad Sci* vol. 94, pp. 2507-2512, 1997

[50] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis*, John Wiley, 2009.

[51] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, “Tools for privacy preserving distributed data mining,” *ACM SIGKDD Explorations*, vol. 4, 2002.

[52] M. Costanzo, A. Baryshnikova, et al., “The Genetic Landscape of a Cell,” *Science* vol. 327, pp. 425-431, Jan. 2010.

[53] N. Cressie, *Statistics for Spatial Data,* Wiley, 1991.

[54] P. Csermely, V. ǵoston, and S. Pongor, “The efficiency of multi-target drugs: the network approach might help drug design,” *TRENDS in Pharmacological Sciences* vol. 26, pp. 178-182, 2005.

[55] E. Dall’Anese, J. A. Bazerque, and G. B. Giannakis, “Group Sparse Lasso for Cognitive Network Sensing Robust to Model Uncertainties and Outliers,” *Physical Communication,* vol. 5, no. 2, pp. 161-172, Elsevier, June 2012.

[56] S.-J. Kim, E. Dall’Anese, J. A. Bazerque, K. Rajawat, and G. B. Giannakis, “Advances in Spectrum Sensing and Cross-Layer Design for Cognitive Radio Networks,” *EURASIP, E-Reference Signal Processing,* November 2012.

[57] E. Dall’Anese, J. A. Bazerque, H. Zhu, and G. B. Giannakis, “Group Sparse Total Least-Squares for Cognitive Spectrum Sensing,” *Proc. of 12th Wrkshp. on Signal Processing Advances in Wireless Communications,* San Francisco, California, USA, June 26-29, 2011.

[58] G. de los Campos, D. Gianola D, and B. Heringstad, “A structural equation model for describing relationships between somatic cell score and milk yield in first-lactation dairy cows,” *J Dairy Sci* vol. 89, pp. 4445-4455, 2006.

[59] P. Denholm, E. Ela, B. Kirby, and M. Milligan, “The Role of Energy Storage with Renewable Electricity Generation,” *Technical Report, National Renewable Energy Laboratory,* 2010.

[60] D. di Bernardo, M. J. Thompson, T. S. Gardner, S. E. Chobot, E. L. Eastwood, et al., “Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks,” *Nature Biotechnology* vol. 23, pp. 377-383, 2005.

[61] F. Diella, N. Haslam, C. Chica, A. Budd, S. Michael, et al., “Understanding eukaryotic linear motifs and their role in cell signaling and regulation,” *Frontiers in Bioscience,* vol. 13, pp. 6580-6603, 2008.

[62] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, et al., “Sparse graphical models for exploring gene expression data,” *J Multivar Anal* vol. 90, pp. 196-212, 2004.

[63] R. C. Dorf, and J. A. Svoboda, *Introduction to Electric Circuits*, 8th edition, John Wiley & Sons, NJ, 2010.

[64] J. Duchon, *Splines Minimizing Rotation-Invariant Semi-norms in Sobolev Spaces,* New York: Springer-Verlag, 1977.

[65] L. El Ghaoui, V. Viallon, and T. Rabbani, “Safe feature elimination in sparse supervised learning,” *Technical Report UC/EECS-2010-126, EECS Dept., University of California at Berkeley,* 2010.

[66] E. Esser, “Applications of Lagrangian-based alternating direction methods and connections to split Bregman,” *Technical Report,* 2009.

[67] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *J. Amer. Statist. Assoc.* vol. 96, pp. 1348-1360, 2001.

[68] M. Fazel, “Matrix rank minimization with applications” *PhD Thesis*, Electrical Engineering Dept., Stanford University, vol. 54, pp. 1-130, 2002.

[69] K. A. Frazer, D. G. Ballinger, D. R. Cox, D. A. Hinds, and L. L. Stuve,“A second generation human haplotype map of over 3.1 million SNPs,” *Nature* vol. 449, pp. 851-861, 2007.

[70] J. Friedman and T. Hastie and R. Tibshirani”, “A note on the group lasso and sparse group lasso”, *Technical Report,* 2010.

[71] J. Friedman and T. Hastie and R. Tibshirani”, “Regularized paths for generalized linear models via coordinate descent,” *Journal of Statistical Software,* vol. 33, 2010.

[72] N. Friedman, M. Linial, I. Nachman, and Pe’er D, “Using Bayesian network to analyze expression data,” *J Comput Biol* vol. 7, pp. 601-620, 2000.

[73] S. Gandy, B. Recht, and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” *Inverse Problems,* vol. 27, no. 2, pp. 1-19, 2011.

[74] G. Ganesan, Y. Li, B. Bing, and S. Li, “Spatiotemporal sensing in cognitive radio networks,” *IEEE Jrnl. on Selected Areas in Communications*, vol. 26, pp. 5–12, Jan. 2006.

[75] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, “Inferring genetic networks and identifying compound mode of action via expression profiling,” *Science* vol. 301, pp. 102-105, 2003.

[76] Q. Geng and J. Wright, “On the local correctness of ℓ_1-minimization for dictionary learning,” *IEEE Trans. on Info. Theory*, 2011 (submitted); see arXiv:1101.5672v1 [cs.IT].

[77] G. Giaever, D. D. Shoemaker, T. W. Jones, H. Liang, E. A. Winzeler, A. Astromoff, and R,. W. Davis, “Genomic profiling of drug sensitivities via induced haploinsufficiency,” *Nature Genetics*, vol. 21, no. 3, pp. 278-283, 1999.

[78] G. Giannakis, G. Mateos, S. Farahmand, V. Kekatos, and H. Zhu, “Uspacor: Universal sparsity-controlling outlier rejection. *In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 1952-1955, Prague, Czech Republic, 2011.

[79] D. Gianola, and D. Sorensen, “Quantitative genetic models for describing simultaneous and recursive relationships between phenotypes,” *Genetics* vol. 167, pp. 1407-1424, 2004.

[80] F. Girosi, “An equivalence between sparse approximation and support vector machines,” *Neural Computation* vol. 10, no. 6, pp. 1455-1480, Aug. 1998.

[81] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, MA, 1989.

[82] T. Goldstein and S. Osher, “The split Bregman method for L1 regularized problems,” *SIAM Journal on Imaging Sciences*, vol. 2, pp. 323-343, 2009.

[83] J. S. Goldstein, I. S. Reed, and L. L. Scharf, “A multistage representation of the Wiener filter based on orthogonal projections,” *IEEE Transactions on Information Theory,* vol. 44, no.7 pp. 2943-2959, 1998.

[84] G. Golub and C. F. Van Loan, *Matrix Computations,* Johns Hopkins University Press, 3rd edition, Oct. 1996.

[85] R. Gribonval and K. Schnass, “Dictionary identification - sparse matrix factorization via ℓ_1-minimization” *IEEE Trans. on Info. Theory,* vol. 56, no. 7, pp. 3523 - 3539, July 2010.

[86] P. Hall, E. R. Lee, and B. U. Park, “Bootstrap-based penalty choice for the Lasso, achieving oracle performance,” *Statistica Sinica,* vol. 19, pp. 449-471, 2009.

[87] J. Håstad, “Tensor rank is NP-complete,” *J. Algorithms,* vol. 11, no. 4, pp. 644-654, 1990.

[88] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,* 2nd ed, Springer, New York, 2009.

[89] Mitola III and Maguire, Jr., 1999; Haykin, 2005

[90] M. E. Hillenmeyer, E. Fung, J. Wildenhain, et al., “The Chemical Genomic Portrait of Yeast: Uncovering a Phenotype for All Genes,” *Science,* vol. 320, no. 5874, pp. 362-365, 2008.

[91] J. H. Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, MI, 1972.

[92] D. Hosmer and S. Lemeshow, *Applied logistic regression,* Wiley, NY, 1989.

[93] B. N. Howie, P. Donnelly, and J. Marchini, “A flexible and accurate genotype imputation method for the next generation of genome-wide association studies,” *PLoS Genet,* vol. 5, 2009.

[94] D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources,” *Nat Protoc,* vol. 4, pp. 44-57, 2009.

[95] T. R. Hughes, M. J. Matthew, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett et al., “Functional discovery via a compendium of expression profiles,” *Cell* vol. 102, no. 1, pp. 109-126, 2000.

[96] J. Jamrozik, J. Bohmanova, and L. R. Schaeffer, “Relationships between milk yield and somatic cell score in canadian holsteins from simultaneous and recursive random regression models,” *J Dairy Sci,* vol. 93, pp 1216-1233, 2010.

[97] H. Jeong, S. P. Mason, A. L. Barabássi, and Z. N. Oltvai, “Lethality and centrality in protein networks,” *Nature* vol. 411, pp. 41-42, 2001.

[98] M. Kalisch, and P. Bühlmann, “Estimating high-dimensional directed acyclic graphs with the PC-algorithm,” *J Mach Learn Res,* vol. 8, pp. 613-636, 2007.

[99] D. Kaplan*Structural Equation Modeling: Foundations and Extensions,* 2nd edition, Sage Publications, 2009.

[100] S. Kay, *Fundamentals of Statistical Signal Processing,* vol. 1, Prentice Hall, 2001.

[101] V. Kekatos and G. B. Giannakis", "Selecting reliable sensors via convex optimization", *Proc. of Intl. Wkshp. on Signal Proc. Adv. in Wireless Comm.,* Marrakech, Morroco, June 2010".

[102] V. Kekatos, S. Veeramachaneni, M. Light, and G. B. Giannakis, "Day-ahead electricity market forecasting using kernels," *Proc. of IEEE-PES on Innovative Smart Grid Technologies*, Washington, DC, Feb. 24-27, 2013.

[103] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models,* New York: Springer, 2009.

[104] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455-500, 2009.

[105] V. Koltchinskii and M. Yuan, "Sparsity in multiple kernel learning," *Annals of Statistics* vol. 38, no. 6, pp. 3660-3695, Apr. 2010.

[106] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349-396, Feb. 2003.

[107] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Lin. Alg. Applicat.*, vol. 18, no. 2, pp. 95138, 1977.

[108] D. C. Kulp and M. Jagalur "Causal inference of regulator-target pairs by gene mapping of expression phenotypes," *BMC Genet*, vol. 7, 2006.

[109] S. Y. Lee, *Structural Equation Modeling: A Bayesian Approach,* Wiley, 2007

[110] K. S. Lee, L. K. Hines, and D. E. Levin, "A pair of functionally redundant yeast genes (PPZ1 and PPZ2) encoding type 1-related protein phosphatases function within the PKC1-mediated pathway," *Molecular and Cellular Biology,* vol. 13, no. 9, pp. 58435853, 1993.

[111] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, and Z. Bar-Joseph, "Transcriptional regulatory networks in *Saccharomyces cerevisiae*," *Science*, vol. 298, pp. 799-804, 2002.

[112] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[113] R. Li, S. W. Tsaih, K. Shockley, "Structural model analysis of multiple quantitative traits," *PLoS Genet.*, vol. 2, 2006.

[114] Y. Lin and H. H. Zhang, "Component selection and smoothing in multivariate nonparametric regression," *Annals of Statistics,* vol. 34, no. 5, pp. 2272-2297, May 2006.

[115] B. Liu, de la Fuente A, Hoeschele I08) Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics,* vol. 178, pp. 1763-1776, 2008.

[116] S. Liu and G. Trenkler, "Hadamard, Khatri-Rao, Kronecker and other matrix products," *Int. J. Inform. Syst. Sci,* vol. 4, pp. 160177, 2008.

[117] B. A. Logsdon and J. Mezey, "Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations," *PLoS Comput Biol,* vol. 6, 2010.

[118] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang "Semidefinite Relaxation of Quadratic Optimization Problems," *IEEE Signal Processing Magazine,* vol.27, no.3, pp.20-34, May 2010

[119] M. Mardani, G. Mateos, and G. B. Giannakis, "In-network sparsity-regularized rank minimization: Algorithms and applications," *IEEE Trans. on Signal Proc.*, 2012; see also arXiv:1203.1507v1 [cs.MA].

[120] M. Mardani, G. Mateos, and G. B. Giannakis, "Rank minimization for subspace tracking from incomplete data," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013.

[121] J. Martinez, R. Carroll, S. Müller, J. Sampson, and N. Chatterjee, "Empirical performance of cross-validation with oracle methods in a genomics context," *The American Statistician,* vol. 65, pp. 223-228, 2011.

[122] G. Mateos and J. A. Bazerque and G. B. Giannakis", "Distributed sparse linear regression", *IEEE Trans. Signal Process.,* 2010.

[123] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Spline-based Spectrum Cartography for Cognitive Radios," *Proc. of 43rd Asilomar Conf. on Signals, Systems, and Computers,* Pacific Grove, CA, November 1-4, 2009.

[124] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Parallelizable Algorithms for the Selection of Grouped Variables," *Proc. of 14th DSP Wkshp.,* Sedona, AZ, January 4-7, 2011.

[125] N. Meinshausen, P. Bhlmann, "Stability selection," *J R Statist Soc B* vol. 72, pp. 417-473, 2010.

[126] X .J. Mi, K. Eskridge, and D. Wang, "Regression-based multi-trait QTL mapping using a structural equation model," *Stat Appl Genet Mol Biol,* vol. 9, 2010.

[127] C. Micchelli and M. Pontil, “Learning the kernel function via regularization,” *J. Machine Learning Res.*, vol. 6, pp. 1099-1125, Sep. 2005.

[128] J. Millstein, B. Zhang, J. Zhu, E. E. Schadt, “Disentangling molecular relationships with a causal inference test,” *BMC Genet* vol. 10, 2009.

[129] S. M. Mishra, A. Sahai, and R. W. Brodersen, “Cooperative sensing among cognitive radios,” in *Proc. of 42nd Intl. Conf. on Communications*, Istanbul, Turkey, Jun. 2006, pp. 1658–1663.

[130] Mitola III and Maguire, Jr., 1999; Haykin, 2005

[131] U. Nagalakshmi et al., “The transcriptional landscape of the yeast genome defined by RNA sequencing” *Science,* vol. 320, no. 5881, pp. 1344-1349, June 2008.

[132] M. Z. Nashed and Q. Sun, “Function spaces for sampling expansions,” *Multiscale Signal Analysis and Modelling,* edited by X. Shen and A. Zayed, Lecture Notes in EE, Springer, pp. 81-104, 2012.

[133] A. Nedic and A. Ozdaglar”, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control,* vol. 54, pp. 48-61, 2009.

[134] E. C. Neto, C. T. Ferrara, A. D. Attie, and B. S. Yandell, “Inferring causal phenotype networks from segregating populations,” *Genetics,* vol. 179, pp. 1089-1100, 2008.

[135] E. C. Neto, M. P. Keller, A. D. Attie, and B. S. Yandell, “Causal graphical models in systems genetics: A unified framework for joint inference of causal network and genetic architecture for correlated phenotypes,” *Ann Appl Stat* vol. 4, pp. 320-339, 2010.

[136] K. Nishimori, R. D. Taranto, H. Yomo, P. Popovski, Y. Takatori, R. Prasad, and S. Kubota, “Spatial opportunity for cognitive radio systems with heterogeneous path loss conditions,” in *Proc. of 65th Vehicular Technology Conference*, Dublin, Ireland, Apr. 2007, pp. 2631–2635.

[137] A.B. Parsons, R. L. Brost, H. Ding, et al., “Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways,” *Nature Biotech.*, vol. 22, no. 1, pp. 62-69, 2004.

[138] A. B. Parsons, A. Lopez, I. E. Givoni, et al., “Exploring the Mode-of-Action of Bioactive Compounds by Chemical-Genetic Profiling in Yeast,” *Cell,* vol. 126, no. 3, pp. 611-625, August 2006.

[139] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht, “How to improve R&D productivity: the pharmaceutical industry’s grand challenge,” *Nature Reviews Drug Discovery* vol. 9, no. 3, pp. 203214, 2010.

[140] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2 edition, Cambridge University Press, 2009.

[141] C. A. Penfold and D. L. Wild, “How to infer gene networks from expression profiles, revisited” *Interface Focus* vol. 1, pp. 857-870, 2011.

[142] I. J. Pérez-Arriaga, “Managing large scale penetration of intermittent renewables,” *Thechnical report, Massachusetts Institute of Technology,* 2011.

[143] Pickrell JK, Marioni JC, Pai AA, Degner JF, Engelhardt BE, et al., “Understanding mechanisms underlying human gene expression variation with RNA sequencing,” *Nautre*, vol. 464, pp. 768-772, 2010.

[144] A. T. Puig, A. Wiesel and A. O. Hero, “A multidimensional shrinkage-thresholding operator,” *Proc. of Wkshp. on Stat. Signal. Proc.*, Cardiff, Wales, Aug-Sep 2009.

[145] Z. Quan, S. Cui, V. H. Poor, and A. H. Sayed, “Collaborative wideband sensing for cognitive radios,” *IEEE Signal Processing Magazine*, vol. 25, pp. 60–73, Nov. 2008.

[146] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning,* the MIT Press, 2006.

[147] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, “Sparse additive models,” *J. Roy. Stat. Soc. B*, vol. 71, no. 5, pp. 1009-1030, Oct. 2009.

[148] M. Razaviyayn, M. Hong, and Z.-Q. Luo, “A unified convergence analysis of block successive minimization methods for nonsmooth optimization,” *SIAM J. Opt.*, 2012; see also arXiv:1209.2385v1 [math.OC]).

[149] B. Recht and C. Re, “Parallel stochastic gradient algorithms for large-scale matrix completion,” 2011 (submitted).

[150] A. Ribeiro and G. B. Giannakis, “Separation Principles in Wireless Networking,” *IEEE Transactions on Information Theory,* vol. 56, no. 9, pp. 4488-4505, September 2010.

[151] C. P. Robert and G. Casella G, *Monte Carlo statistical method,* 2 edition, Springer, 2004.

[152] M. V. Rockman, “Reverse engineering the genotype-phenotype map with natural genetic variation,” *Nature*, vol. 456, pp. 738-744, 2008.

[153] G. J. Rosa , B. D. Valente, and G. de los Campos, “Inferring causal phenotype networks using structural equation models,” *Genet Sel Evol*, vol. 43, 2011.

[154] T. Santiago, S. V. Kulemzin, E. S. Reshetnikova, N. A. Chikaev, O. Y. Volkova, et al., “Fcrla is a resident endoplasmic reticulum protein that associates with intracellular igs, igm, igg and iga,” *Int Immunol,* vol. 23, pp. 43-53, 2011.

[155] J. Schäfer and K. Strimmer K, “An empirical Bayes approach to inferring large-scale gene association networks,” *Bioinform,* vol. 21, pp. 754-764, 2005.

[156] j. Schäfer and Strimmer K, “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics,” *Stat Appl Genet Mol Biol,* vol. 4, no. 32, 2005.

[157] P. Scheet and M. Stephens, “A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase,” *American Journal of Human Genetics,* vol. 78, pp. 629-644, 2006.

[158] M. Seeger “Bayesian Inference and Optimal Design in the Sparse Linear Model,” *Journal of Machine Learning Research,* vol. 9, pp. 759-813, 2008.

[159] E. Segal, M. Shapira, A. Regev, D. Pe’er, D. Botstein, et al., “Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data,” *Nat Genet,* vol. 34, pp. 166-178, 2003.

[160] M. Shapira, M. E. Segal, and D. Botstein, “Disruption of yeast forkhead-associated cell cycle transcription by oxidative stress,” *Molecular Biology of the Cell,* vol. 15, no. 12, pp. 5659-5669, Dec. 2004.

[161] B. Shipley, *Cause and Correlation in Biology: A User’s Guide to Path Analysis, Structural Equations and Causal Inference,* Cambridge University Press, 2002.

[162] A. Shrivastava, H. V. Nguyen, V. M. Patel, and R. Chellappa, “Design of non-linear discriminative dictionaries for image classification,” *Proc. of Asian Conf. on Computer Vision,* Daejeon, Korea, 2012.

[163] N. D. Sidiropoulos, R. Bro “On the uniqueness of multilinear decomposition of N-way arrays,” *Journal of chemometrics,* vol. 14, no. 3, pp. 229-239, 2000.

[164] N. D. Sidiropoulos, R. Bro, and G. B Giannakis, “Parallel factor analysis in sensor array processing,” *IEEE Transactions on Signal Processing,* vol. 48 no. 8 pp. 2377-2388 2000.

[165] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, “Blind PARAFAC receivers for DS-CDMA systems,” *IEEE Transactions on Signal Processing,* vol. 48, no. 3, pp. 810-823, 2000.

[166] C. Sima, J. Hua, and S. Jung, “Inference of gene regulatory networks using time-series data: a survey,” *Curr Genomics,* vol. 10, pp. 416-429, 2009.

[167] V. Sindhwani and A. C. Lozano, “Non-parametric group orthogonal matching pursuit for sparse learning with multiple kernels,” *Advances in Neural Information Processing Systems*, pp. 2519-2527, Granada, Spain, 2011.

[168] K. Slavakis, G. B. Giannakis, and G. Leus, “Nonlinear Compression and Reconstruction via Robust Sparse Embeddings,” *Proc. of Conf. of Info. Sciences, and Systems*, Johns Hopkins Univ., Baltimore, March 20-22, 2013.

[169] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd edition, MIT Press, Cambridge, MA 2000.

[170] P. Sprechmann and I. Ramirez and G. Sapiro and Y. C. Eldar, “Collaborative Hierarchical Sparse Modeling,” *Proc. of 44th Conf. on Info. Sciences and Systems*, Princeton, NJ, March 2010.

[171] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, “Maximum-margin matrix factorization,” *Advances in Neural Information Processing Systems*, vol. 17, pp. = 1329-1336, 2005.

[172] A. Stegeman and N. D. Sidiropoulos “On Kruskals uniqueness condition for the Candecomp/Parafac decomposition,” *Linear Algebra and its applications*, vol. 420, no. 2, pp. 540-552, 2007.

[173] J. F. Stürm, “Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones,” *Optimization Methods and Software*, vol. = 12, pp. 625-653, 1999.

[174] T. S. Tanaka, T. Kunath, W. L. Kimber, S. A. Jaradat, C. A. Stagg, M. Usuda, T. Yokota, H. Niwa, J. Rossant, and M. S. Ko, “Gene expression profiling of embryo-derived stem cells reveals candidate genes associated with pluripotency and lineage specificity,” *Genome Research*, vol . 12, no. 12, pp. 1921-1928, 2002.

[175] J. Tegner, M. K . Yeung, J. Hasty, and J. J. Collins, “Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling,” *Proc Natl Acad Sci*, vol. 100, pp. 5944-5949, 2003.

[176] J. M. F. ten Berge and N. D. Sidiropoulos, “On uniqueness in CANDECOMP/PARAFAC,” *Psychometrika*, vol. 67, no. 3, pp. 399-409, 2002.

[177] D. Thieffry, A. M. Huerta, E. Pérez-Rueda, and J. Collado-Vides, “From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*,” *Bioessays*, vol. 20, pp. 433-440, 1998.

[178] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Royal. Statist. Soc. B,* vol. 58, pp. 267-288, 1996.

[179] Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, et al., “Strong rules for discarding predictors in lasso-type problems. *J R Statist Soc Series B,* vol. 74, pp. 245-266, 2012.

[180] R. Tomioka, K. Hayashi, and H. Kashima, “Estimation of low-rank tensors via convex optimization,” submited 2011, also available at *ArXiv:1010.0789v2 [stat.ML].*

[181] P. Tseng and S. Yun, “A coordinate gradient descent method for nonsmooth separable minimization,” *J. Mathematical Programming,* vol. 117, no. 1-2, pp. 387-423, Mar. 2009.

[182] M. Unser, “Splines: A perfect fit for signal and image processing,” *IEEE Signal Proc. Magazine,* vol. 16, no. 6, pp. 22-38, Nov. 1999.

[183] B. D. Valente, G. J. Rosa, G. de los Campos, “Searching for recursive causal structures in multivariate quantitative genetics mixed models,” *Genetics,* vol. 185, pp. 633-644, 2010.

[184] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM Review,* vol. 38, no. 1, pp. 49-95, 1996.

[185] P. Vincent and Y. Bengio, “Kernel matching pursuit,” *Machine Learning,* vol. 48, pp. 169-191, 2002.

[186] Z. Xing, M. Zhou, A. Castrodad, G. Sapiro and L. Carin, “Dictionary learning for noisy and incomplete hyperspectral images,” *SIAM Journal on Imaging Sciences,* vol. 5, no. 1, pp. 33-56, 2012.

[187] M. Xiong, J. Li, and X. Fang, “Identification of genetic networks,” *Genetics,* vol. 166, pp. 1037-1052, 2004.

[188] J. Yang and Y. Zhang and W. Yin, “A Fast Alternating Direction Method for TVL1-L2 Signal Reconstruction From Partial Fourier Data,” *IEEE Jrnl. Sel. Topics in Signal Process.,* vol. 4, pp. 288-297, 2010.

[189] K. Y. Yip, R. P. Alexander, K. K. Yan, and M. Gerstein, “Improved reconstruction of *in silico* gene regulatory networks by integrating knockout and perturbation data,” *PLoS ONE,* vol. 5, 2010.

[190] M. Yuan and Y.Lin, “Model selection and estimation in regression with grouped variables,” *J. Royal Statistic. Soc. B,* vol. 68, pp. 49-67, 2006.

[191] G. Wahba, *Spline Models for Observational Data,* SIAM, PA 1990.

[192] H. Wang and C. Leng", "A note on adaptive group Lasso," "*Computational Statistics and Data Analysis*, vol. 52, pp. 5277-5286, 2008.

[193] T. J. Wilson, S. Gilfillan, and M. Colonna, "Fc receptor-like a associates with intracellular igg and igm but is dispensable for antigen-specific immune responses," *J Immunol*, vol. 185, pp. 2960-2967, 2010.

[194] S. J. Wright and R. D. Nowak and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, pp. 2479-2493, 2009.

[195] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recognition Letters*, vol. 22, pp. 1255-1261, 2001.

[196] X. L. Wu, B. Heringstad, and D. Gianola, "Bayesian structural equation models for inferring relationships between phenotypes: a review of methodology, identifiability, and applications," *J Anim Breed Genet*, vol. 127, pp. 3-15, 2010.

[197] Z. Zhang, and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics in Signal Proc.*, vol. 5, no. 5, pp. 912-926, Sep. 2011.

[198] J. Zhu, P.Y. Lum, J. Lamb, D. GuhaThakurta, S. Edwardsa S, et al., "An integrative genomics approach to the reconstruction of gene networks in segregating populations," *Cytogenet Genome Res*, vol. 105, pp. 363-374, 2004.

[199] J. Zhu, M. C. Wiener, C. Zhang, A. Fridman, E. Minch, et al., "Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations," *PLoS Comput Biol*, vol. 3, 2007.

[200] J. Zhu, B. Zhang, E. N. Smith, B. Drees, R. B. Brem, et al., "Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks," *Nat Genet*, vol. 40, pp. 854-61, 2008.

[201] H. Zou, "The adaptive Lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, pp. 1418-1429, 2006.

[202] H. Zou, and H. H. Zhang, "On The Adaptive Elastic-Net With A Diverging Number of Parameters," *Annals of Statistics*, vol. 37, no. 4, pp. 1733-1751, 2009.

Appendix A

Inference of gene regulatory networks

A.0.1 Cross-validation

Cross-validation for ridge regression

The solution of (2.4) requires specifying the parameter ρ. A $K-$fold cross-validation (CV) scheme is adopted for this purpose with typical choices of $K = 5$ or 10, as suggested in [88]. For $\kappa = 1, \ldots, K$ the dataset is divided in two parts, namely $(\tilde{\mathbf{Y}}_\kappa, \tilde{\mathbf{X}}_\kappa)$ with N_s/K samples and $(\tilde{\mathbf{Y}}_{(-\kappa)}, \tilde{\mathbf{X}}_{(-\kappa)})$ with the remaining $(K-1)N_s/K$ samples. For each value of ρ on a grid of $R = 30$ points regularly spaced in logarithmic scale between 10^{-6} and 1, the solution to (2.4) computed using $(\tilde{\mathbf{Y}}_{(-\kappa)}, \tilde{\mathbf{X}}_{(-\kappa)})$ is denoted as $(\tilde{\mathbf{B}}_{(\rho,\kappa)}, \tilde{\mathbf{F}}_{(\rho,\kappa)})$. The error $e_\kappa(\rho) := \|\tilde{\mathbf{Y}}_\kappa - \tilde{\mathbf{B}}_{(\rho,\kappa)}\tilde{\mathbf{Y}}_\kappa - \tilde{\mathbf{F}}_{(\rho,\kappa)}\tilde{\mathbf{X}}_\kappa\|_F^2$ is obtained and averaged across folds to obtain the error estimate $e(\rho)$. The value of ρ that attains a minimum $e(\rho)$ is selected as the optimal value. In order to save computations, the grid of ρ_r values is scanned progressively for $r = 1, \ldots, R$. The procedure is stopped when $e(\rho_{r-1}) < e(\rho_r)$, and ρ_{r-1} is chosen as the optimal value.

Cross-validation for ℓ_1-regularized ML estimation

The CV procedure for selecting λ follows the steps used to select ρ in ridge regression. The sample is divided into K folds, and for $\kappa = 1, \ldots, K$ the κ-th fold is set aside for validation. For L values of λ between $\lambda_{\min} = 10^{-4}\lambda_{\max}$ and $\lambda_{\max}$, the solution to (2.3) computed using $(\mathbf{Y}_{(-\kappa)}, \mathbf{X}_{(-\kappa)})$ is denoted as $(\hat{\mathbf{B}}(\lambda,\kappa), \hat{\mathbf{F}}(\lambda,\kappa))$, and the validation error is computed for each κ using $(\hat{\mathbf{B}}(\lambda,\kappa), \hat{\mathbf{F}}(\lambda,\kappa))$ and $(\mathbf{Y}_\kappa, \mathbf{X}_\kappa)$. Upon averaging the validation errors across κ, an optimal λ is selected as the largest parameter that minimizes this mean-CV error within one standard deviation.

Stability of model selection under CV perturbations

A set of simulations were run to test robustness of the SML algorithm. First, the fold number of CV was changed from $k = 5$ to $k = 10$ for the DAGs of 30 genes in Figures 2(c) and 2(d) and the DCGs of 30 genes in Figures 3(c) and 3(d) with an expected number of edges $N_e = 3$. As shown in Figure A.1, $k = 5$ and $k = 10$ offer almost identical performance. Simulations with a suboptimal λ that is 10% less than the optimal λ obtained from 5-fold CV were then run for the networks used in Figure A.1. As expected, the suboptimal λ yielded slightly worse performance as shown in Figure A.2; the performance degradation is very small for the DAGs but relatively large for the DCGs, which implies that it is important to choose the optimal value of λ.

A.0.2 Discarding rules

In Lasso regression, it is known that for a given λ some predictors can be set to zero *a priori* without solving the Lasso inference problem [65, 179]. Hence, these predictors can be discarded when inferring other predictors. Rules for discarding predictors were also derived in [65, 179]. In particular, the strong rules in [179] can discard a large number of predictors, which significantly reduces the computation needed to solve the Lasso problem. To reduce computational burden and improve the speed of the SML algorithm, the technique in [179] is employed next to derive strong rules for setting some entries of matrix $\mathbf{B}$ to zero *a priori*, before running the coordinate-ascent algorithm.

Let $\hat{\mathbf{B}}(\lambda)$ and $\hat{\mathbf{F}}(\lambda)$ denote the optima of (2.3) for a given λ. Let also $Q_{ij}(\lambda)$ stand for the derivative of the differentiable part of (2.3); i.e., $N\sigma^2 \log|\det(\mathbf{I} - \mathbf{B})| - \frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2$, w.r.t. B_{ij} evaluated at $\hat{\mathbf{B}}(\lambda)$ and $\hat{\mathbf{F}}(\lambda)$. Then, $Q_{ij}(\lambda)$ can be found as

$$Q_{ij}(\lambda) = -\frac{N\sigma^2 c_{ij}(\lambda)}{\det(\mathbf{I} - \hat{\mathbf{B}}(\lambda))} + \left[\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T - \hat{\mathbf{B}}(\lambda)\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T - \hat{\mathbf{F}}(\lambda)\tilde{\mathbf{X}}\tilde{\mathbf{Y}}^T\right]_{ij} \tag{A.1}$$

where $c_{ij}(\lambda)$ is the (i, j)th co-factor of $\mathbf{I} - \hat{\mathbf{B}}(\lambda)$, and σ^2 can be estimated as $\hat{\sigma}^2 = \frac{1}{NN_g}\|\tilde{\mathbf{Y}} - \hat{\mathbf{B}}(\lambda)\tilde{\mathbf{Y}} - \hat{\mathbf{F}}(\lambda)\tilde{\mathbf{X}}\|_F^2$. Let $\lambda_{\max}$ denote the smallest value of λ that yields $\hat{B}_{ij} = 0$, $\forall i, j$ (an expression for $\lambda_{\max}$ will be given later). After recognizing that $c_{ij}(\lambda_{\max}) = 1$, it follows that

$$Q_{ij}(\lambda_{\max}) = -N\sigma^2 + \left[\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T - \hat{\mathbf{F}}(\lambda_{\max})\tilde{\mathbf{X}}\tilde{\mathbf{Y}}^T\right]_{ij} \tag{A.2}$$

where $\hat{\mathbf{F}}(\lambda_{\max})$ is obtained by substituting $\mathbf{B} = \mathbf{0}$ into (2.10). Note that $Q_{ij}(\lambda_{\max})$ can be computed without knowledge of $\lambda_{\max}$.

For $\lambda < \lambda_{\max}$, the discarding rule is given by

$$|Q_{ij}(\lambda_{\max})| < w_{ij}(2\lambda - \lambda_{\max}) \;\Rightarrow\; \hat{B}_{ij}(\lambda) = 0. \tag{A.3}$$

When trying to find solutions of (2.3) along a path of λ defined with a decreasing set of values $\lambda_0 = \lambda_{\max} > \lambda_1 > \ldots > \lambda_{\min}$, which are needed in CV, the following alternative rule is possible:

$$|Q_{ij}(\lambda_{l-1})| < w_{ij}(2\lambda_l - \lambda_{l-1}) \;\Rightarrow \hat{B}_{ij}(\lambda_l) = 0. \tag{A.4}$$

Let $\mathcal{S}_B(\lambda_l)$ denote the set of $\hat{B}_{ij}(\lambda_l) = 0$ obtained from (A.4) or (A.3).

The rationale behind rules (A.3) and (A.4) is described in the following. By the optimality of $\hat{\mathbf{B}}(\lambda)$, the KKT condition implies that

$$Q_{ij}(\lambda) = \lambda w_{ij} s_{ij} \tag{A.5}$$

where s_{ij} is the subgradient of $|B_{ij}(\lambda)|$, and $s_{ij} = 1$ if $\hat{B}_{ij}(\lambda) > 0$, $s_{ij} = -1$ if $\hat{B}_{ij}(\lambda) < 0$, or, $s_{ij} \in [-1, 1]$ if $\hat{B}_{ij}(\lambda) = 0$. Taking the derivative w.r.t. λ on both sides of (A.5) results in $\frac{dQ_{ij}(\lambda)}{d\lambda} = \left(s_{ij} + \lambda \frac{ds_{ij}}{d\lambda}\right) w_{ij}$. Thus, under the assumption that $\left|s_{ij} + \lambda \frac{ds_{ij}}{d\lambda}\right| \leq 1$ (see [179] for a discussion on this assumption), it follows that

$$\left|\frac{dQ}{d\lambda}\right| \leq w_{ij}. \tag{A.6}$$

Applying the mean-value theorem between λ_l and λ_{l-1} yields

$$|Q(\lambda_{l-1}) - Q(\lambda_l)| \leq w_{ij}(\lambda_{l-1} - \lambda_l). \tag{A.7}$$

If the inequality in (A.4) holds, then (A.7) implies $|Q(\lambda_l)| < \lambda_l w_{ij}$, which in accordance with (A.5) yields $|s_{ij}| < 1$ and thus $\hat{B}_{ij}(\lambda_l) = 0$, as specified by rule (A.4). Similarly, one can justify rule (A.3).

Computation of $\lambda_{\max}$

When λ is sufficiently large such that $\hat{\mathbf{B}} = \mathbf{0}$, (A.5) and the definition of s_{ij} imply that

$$\left|\frac{Q_{ij}(\lambda)}{w_{ij}}\right| \leq \lambda, \; \forall i, j = 1, \ldots, N_g. \tag{A.8}$$

Since $Q_{ij}(\lambda) = Q_{ij}(\lambda_{\max})$ for $\lambda > \lambda_{\max}$ as indicated in (A.2), we obtain

$$\lambda_{\max} = \max_{i,j=1,\ldots,N_g} \left|\frac{Q_{ij}(\lambda_{\max})}{w_{ij}}\right|, \tag{A.9}$$

being the minimum possible value satisfying (A.8) and thereby giving rise to a λ yielding $\hat{\mathbf{B}} = \mathbf{0}$ in (2.3). Substituting $Q_{ij}(\lambda_{\max})$ from (A.2) into (A.9) yields $\lambda_{\max}$. Recall from (A.2) that $Q_{ij}(\lambda_{\max})$ can be computed without knowledge of $\lambda_{\max}$.

A.0.3 Extensions of the SML algorithm

Stability selection

In Algorithm 1, CV is used to select the optimal value of λ that determines the level of sparsity in $\hat{\mathbf{B}}$. However, it was observed that a single run of CV may not yield a consistent estimate of variables [86, 121]. An alternative approach to choosing appropriate variables is stability selection (STS) [125] that offers a theoretical upper bound on the FDR. We next describe the procedure for applying STS to our SML algorithm. Upon drawing N_{STS} random data subsamples of size $N_s = \lfloor N/2 \rfloor$, where $\lfloor N/2 \rfloor$ stands for the largest integer less than $N/2$, (2.3) is solved per subsample and per λ, yielding a collection of estimates $\hat{\mathbf{B}}_\nu(\lambda)$, $\nu = 1, \dots N_{STS},\ \lambda = \lambda_{\min}, \dots, \lambda_{\max}$. Defining an $N_g \times N_g$ matrix $\mathbf{T}(\lambda) := \sum_{\nu=1}^{N_{STS}}$ abs(sgn($\hat{\mathbf{B}}_\nu(\lambda)$)) whose (i,j)th entry counts the nonzero $[\hat{B}_\nu(\lambda)]_{ij}$'s across $\nu = 1, \dots, N_{STS}$ estimates, edge (i,j) is declared as stably identified at level λ, if $T_{i,j}(\lambda)$ exceeds a threshold δN_{STS} with $\delta \in (0.6, 0.9)$. For a given λ, an upper bound on the FDR resulting from the STS procedure is given by $\overline{\mathrm{FDR}}(\lambda) := \frac{q^2}{(2\pi-1)N_g^2 q_s}$ [125], where q denotes the average number of nonzeros in $\hat{\mathbf{B}}_\nu(\lambda)$ across $\nu = 1, \cdots, N_{STS}$ estimates, and q_s the average number of stably identified edges. Both q and q_s, and thus $\overline{\mathrm{FDR}}(\lambda)$, increase as the sparsity-controlling parameter λ decreases. Therefore, the optimal λ denoted as λ_{STS} for a target $\overline{\mathrm{FDR}}$ is selected as the smallest λ satisfying $\overline{\mathrm{FDR}}(\lambda) \leq \overline{\mathrm{FDR}}$. The overall result presents low sensitivity on frequency δ, since a higher and more restrictive π is automatically compensated for by a lower more permissive λ. Note that the original STS procedure [125] employs the random LASSO where the weights are randomly selected from some specified values. In our case, we do not use random weights but still use the weights obtained from ridge regression, since our simulations show that those weights yield improved performance.

Heteroscedasticity

Removing the assumption that the residual error $\boldsymbol{\epsilon}_i$ in (2.1) has covariance matrix $\sigma^2\mathbf{I}$, the SML algorithm can be extended to the more general case where the covariance of $\boldsymbol{\epsilon}_i$ is a diagonal matrix $\mathbf{R} = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_{N_g}^2)$ with unequal diagonal entries $\sigma_i^2, i = 1, \cdots, N_g$. In this case, the log-likelihood function in (2.2) becomes $\log p(\mathbf{Y}|\mathbf{X}; \mathbf{B}, \mathbf{F}, \boldsymbol{\mu}) = \frac{N}{2} \log|\det(\mathbf{I} - \mathbf{B})|^2 - \frac{N}{2} \log[\det(R)] - \frac{NN_g}{2} \log(2\pi) - \frac{1}{2}\mathrm{Tr}[(\mathbf{Y} - \mathbf{BY} - \mathbf{FX} - \boldsymbol{\mu}\mathbf{1}^T)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{BY} - \mathbf{FX} - \boldsymbol{\mu}\mathbf{1}^T)$, where $\mathrm{Tr}(\cdot)$ denotes the trace of the matrix in parentheses. It is easy to show that maximizing this likelihood function w.r.t. $\boldsymbol{\mu}$ yields the same expression for $\boldsymbol{\mu}$ as the one obtained earlier by maximizing the likelihood function in (2.2). Then the objective function in ridge regression problem (2.4) becomes $J_{\text{ridge}} = \frac{1}{2}\mathrm{Tr}[(\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}})^T \mathbf{R}^{-1} (\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}})] + \rho\|\mathbf{B}\|_F^2 = \sum_{i=1}^{N_g} \left[\frac{1}{2\sigma_i^2}\|\check{\mathbf{y}}_i^T - \mathbf{b}_i^T\tilde{\mathbf{Y}} - \mathbf{f}_i^T\tilde{\mathbf{X}}\|_2^2 + \rho\|\mathbf{b}_i\|_2^2\right]$. Therefore, it is again possible to solve (2.4) row by row separately, but replace the objective function in (2.8) with $\sum_{i=1}^{N_g} \left[\frac{1}{2}\|\check{\mathbf{y}}_i^T - \mathbf{b}_i^T\tilde{\mathbf{Y}} - \mathbf{f}_i^T\tilde{\mathbf{X}}\|_2^2 + \rho_i\|\mathbf{b}_i\|_2^2\right]$, where $\rho_i = \rho\sigma_i^2$. Specifically, problem

(2.8) can be solved with this new objective function and a specific value ρ_i that is obtained from CV performed separately for each row. Variance σ_i^2 is then estimated as the residual error for the ith row obtained with estimated $\mathbf{b}_i$ and $\mathbf{f}_i$. The ℓ_1-regularized ML problem (2.3) can also be reformulated by replacing the objective function with the following one: $J_{\mathrm{ML}} = N \log|\det(\mathbf{I}-\mathbf{B})| - \frac{1}{2}\mathrm{Tr}\big[(\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}})^T \mathbf{R}^{-1}(\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}})\big] - \lambda\|\mathbf{B}\|_{1,W}$. With this new objective function, (2.14) becomes $g_{ij}(B_{ij}) = N\hat{\sigma}_i^2 \log|\alpha_0 - c_{ij}B_{ij}| + \alpha_1 B_{ij} - \frac{1}{2}\alpha_2 B_{ij}^2 - \lambda\hat{\sigma}_i^2 w_{ij}|B_{ij}|$, where $\hat{\sigma}_i^2$ is the estimate of σ_i^2. Therefore, the coordinate-ascent algorithm can be easily modified by replacing $\hat{\sigma}^2$ with $\hat{\sigma}_i^2$ and w_{ij} with $w_{ij}\hat{\sigma}_i^2$ in $g_{ij}(B_{ij})$ to estimate B_{ij}.

Identification of eQTLs

The SML algorithm can be extended to handle the case where some or all phenotypes have unidentified *cis*-eQTLs, if a new penalty term, that involves the weighed ℓ_1-norm of the entries of $\mathbf{F}$ excluding those corresponding to the identified *cis*-eQTL, is added to the objective function in (3). In this case, it is only necessary to modify line 13 of the SML algorithm as follows. Consider redefining $\check{\mathbf{f}}_i$ as the one that contains the entries of $\mathbf{f}_i$ corresponding to the known *cis*-eQTLs and let $\check{\mathbf{f}}_i'$ contain the remaining entries of $\mathbf{f}_i$. Similarly, let $\check{\mathbf{X}}_i$ collect rows of $\tilde{\mathbf{X}}$ corresponding to the known *cis*-eQTLs and $\check{\mathbf{X}}_i'$ contain the remaining rows of $\tilde{\mathbf{X}}$. Then on line 13 of the SML algorithm, (7) is replaced by $\check{\mathbf{f}}_i = \left(\check{\mathbf{X}}_i\check{\mathbf{X}}_i^T\right)^{-1}\check{\mathbf{X}}_i\left(\check{\mathbf{y}}_i - \check{\mathbf{Y}}_i\check{\mathbf{b}}_i - \check{\mathbf{X}}_i'^T\check{\mathbf{f}}_i'\right)$ with $\mathbf{f}_i'$ taking values obtained in the previous iteration. The entries of $\mathbf{f}_i'$ can be updated using the coordinate ascent method in the glmnet algorithm [71] for Lasso based linear regression.

A.0.4 State-of-the-art algorithms

Adaptive Lasso-based algorithm

The AL-based algorithm [117] involves three basic steps: the first one performs standard eQTL mapping to identify a *cis*-eQTL per gene; the second one applies the adaptive Lasso [201] to infer the SEM; and the third step performs a permutation test to ensure that edges in the network obtained from the second step correspond to correct dependencies in the directed graph. Since the core of the AL-based algorithm is the adaptive Lasso in step 2, it is described here briefly for completeness. The adaptive lasso estimates $\mathbf{B}$ and $\mathbf{F}$ as follows

$$(\hat{\mathbf{B}}, \hat{\mathbf{F}}) = \arg\max_{\mathbf{B},\mathbf{F}} -\frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2 - \lambda\psi_W(\mathbf{B},\mathbf{F})\} \tag{A.10}$$
$$\text{subject to } \; B_{ii} = 0, \forall i = 1, \ldots, N_g, \; F_{jk} = 0, \; \forall (j,k) \in \mathcal{S}_q$$

where $\psi_W(\mathbf{B},\mathbf{F}) := \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} w_{ij}|B_{ij}| + \sum_{i=1}^{N_g}\sum_{j=1}^{N_q} v_{ij}|F_{ij}|$. Weights w_{ij} and v_{ij} are given by $w_{ij} := |\tilde{B}_{ij}|^{-1/2}$ and $v_{ij} := |\tilde{F}_{ij}|^{-1/2}$, where $\tilde{B}_{ij}$ and $\tilde{F}_{ij}$ are obtained by solving the following

Lasso problem

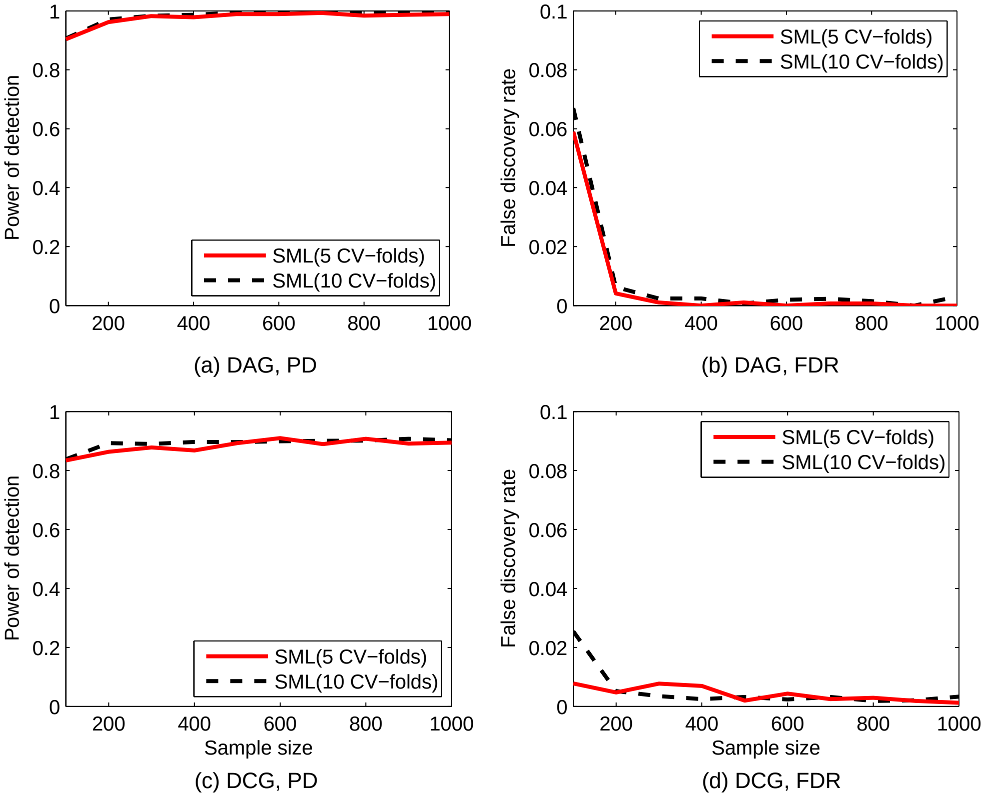
$$(\tilde{\mathbf{B}}, \tilde{\mathbf{F}}) = \arg\max_{\mathbf{B},\mathbf{F}} -\frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2 - \rho\psi(\mathbf{B}, \mathbf{F})\} \tag{A.11}$$

$$\text{subject to } \; B_{ii} = 0, \forall i = 1, \ldots, N_g, \; F_{jk} = 0, \forall (j,k) \in \mathcal{S}_q$$

with $\psi(\mathbf{B}, \mathbf{F}) := \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} |B_{ij}| + \sum_{i=1}^{N_g} \sum_{j=1}^{N_q} |F_{ij}|$. Constants λ and ρ are obtained via CV. We obtained the program implementing the AL-based algorithm from the authors of [117] and used it in our simulation studies. In this program, the glmnet algorithm [71] is employed to solve Lasso problems (26) and (27).

QDG Algorithm

The QDG algorithm [134] first builds an undirected graph for the phenotypes under consideration, using an undirected dependency graph [161] or a skeleton derived from the PC algorithm [169]. It then orients edges in the undirected graph by using a score calculated from the likelihood of the data for different edge directions. The edge orientation process is performed iteratively for each edge until no edge changes its direction. We obtained the program implementing the QDG algorithm from the authors [134] and used the default settings of the program in our simulations.



(a) DAG, PD

(b) DAG, FDR

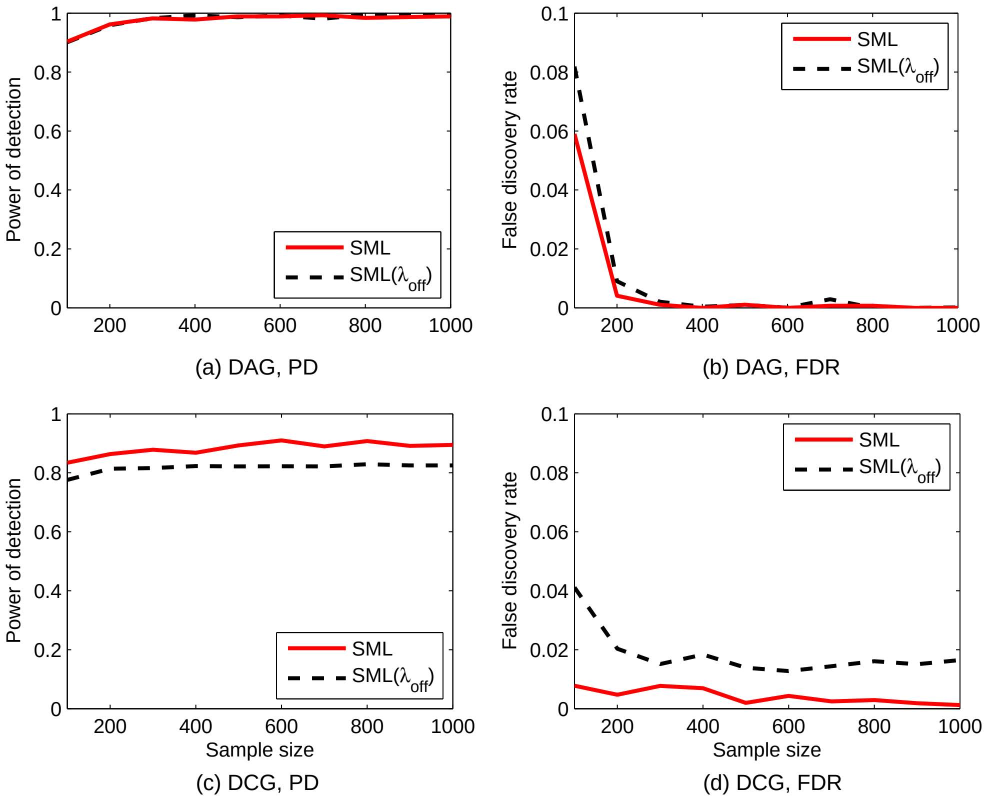
(c) DCG, PD

(d) DCG, FDR

Figure A.1: Performance of the SML algorithm for DAGs [(a) and (b)] or DCGs [(c) and (d)] of N_g=30 genes obtained with 5 (solid line) or 10 (dashed line) fold cross validation. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes from 100 to 1,000.

Table A.1: Thirty nine immune-related human genes used to infer a network. (see file TableS1.xlsx)

Table A.2: Edges of the gene network in Figure 2.7 inferred with the SML algorithm and edges detected with AL and QDG algorithms. (see file TableS2.xlsx)



(a) DAG, PD

(b) DAG, FDR

(c) DCG, PD

(d) DCG, FDR

Figure A.2: Performance of the SML algorithm for DAGs [(a) and (b)] or DCGs [(c) and (d)] of N_g=30 genes obtained with the optimal λ (solid line) or an λ 10% less than the optimal λ (dashed line). Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes from 100 to 1,000.

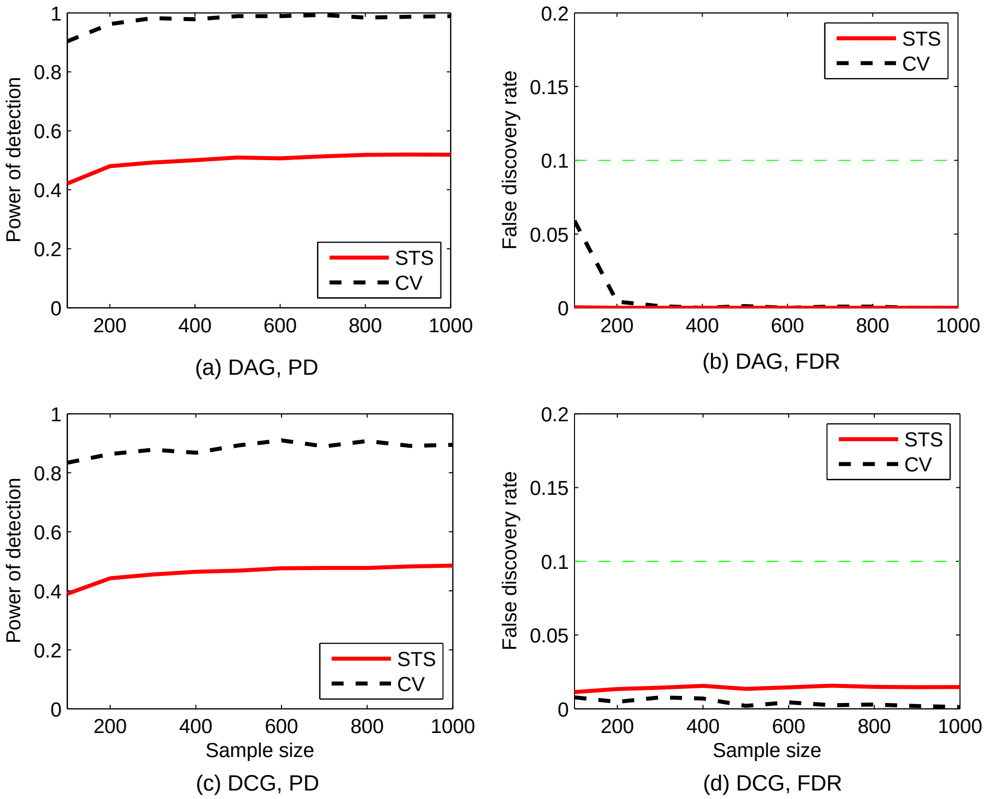


Figure A.3: Performance of the SML algorithm with stability selection (STS) or cross validation for DAGs [(a) and (b)] or DCGs [(c) and (d)] of N_g=30 genes. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes from 100 to 1,000.

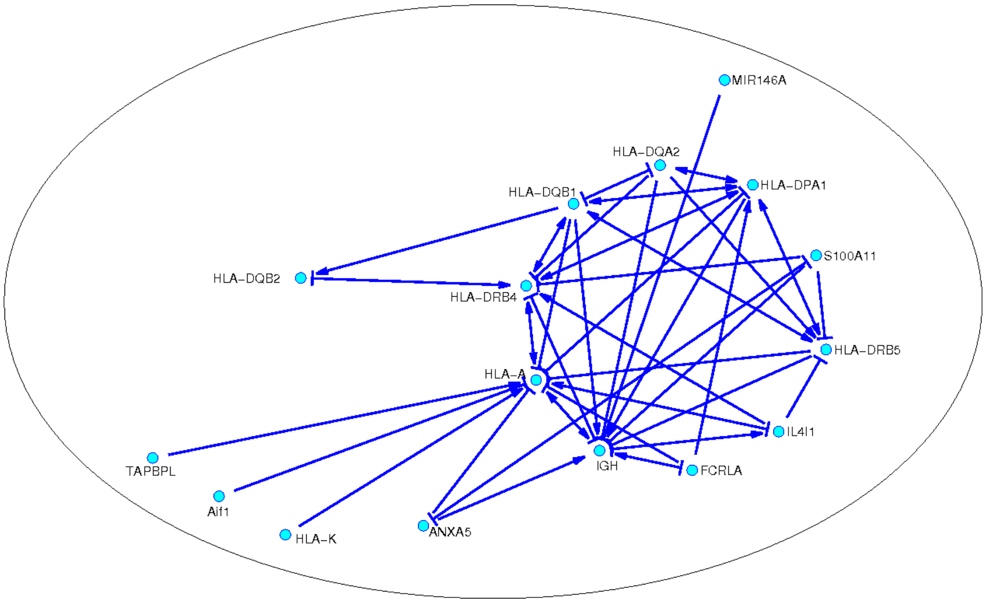


Figure A.4: The network of 39 human genes inferred from gene expression and eQTL data with the SML algorithm. The 39 immune-related genes were chosen from [143] to have a reliable eQTL per gene. The SML algorithm was run with stability selection and edges were detected at an FDR < 0.3. See Table A.1 for the IDs and description of 39 genes. IGH in this figure corresponds to gene ID ENSG00000211897. Edge ends ⊣ and → represent inhibitory or activating effects, respectively.

Appendix B

List of inferred chemical-genetic interactions

70% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
100% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
Sulfometuron methyl	YGL195W	YGR244C	YIL017C	YBR104W	YDR033W	YKR026C	YCR053W	YER073W	YCL044C	YBL015W
	0.129	0.120	0.109	0.104	0.104	0.102	0.100	0.099	0.096	0.087
	YGL195W	YGR244C	YIL017C	YBR104W	YDR033W	YKR026C	YCR053W	YER073W	YCL044C	YBL015W
	0.129	0.120	0.109	0.104	0.104	0.102	0.100	0.099	0.096	0.087
MMS	YDR457W	YKL056C	YER133W_damp	YMR269W	YAL025C_damp	YBL050W_tsq48	YOL068C	YLR298C_tsq844	YLR166C_tsq66	YPR101W
	0.122	0.102	0.093	0.086	0.084	0.084	0.082	0.081	0.081	0.081
	YDR457W	YKL056C	YER133W_damp	YMR269W	YAL025C_damp	YBL050W_tsq48	YOL068C	YLR298C_tsq844	YLR166C_tsq66	YHR208W
	0.122	0.102	0.093	0.086	0.084	0.084	0.082	0.081	0.081	0.080
Clotrimazole	YDR457W	YGL179C	YGL137W_tsq546	YDL245C	YPL270W	YLL040C	YMR272C	YLR079W	YDR217C	YDR505C
	0.125	0.103	0.090	0.087	0.080	0.078	0.077	0.077	0.075	0.074
	YDR457W	YGL179C	YGL137W_tsq546	YDL245C	YPL270W	YLL040C	YMR272C	YLR079W	YDR217C	YJL085W
	0.125	0.103	0.090	0.087	0.080	0.078	0.077	0.077	0.075	0.064
Benomyl	YDL192W	YFL025C	YER017C	YMR233W	YLR314C_tsq130	YGL006W	YKL150W	YPL003W	YNL061W_tsq630	YGR034W
	0.096	0.086	0.073	0.068	0.068	0.067	0.064	0.064	0.062	0.062
	YDL192W	YFL025C	YER017C	YMR233W	YLR314C_tsq130	YGL006W	YPL003W	YNL061W_tsq630	YOR298W	YDL245C
	0.096	0.086	0.073	0.068	0.068	0.067	0.064	0.062	0.060	0.058
Plumbagin	YDR457W	YML102W	YKL173W_tsq621	YDR037W_damp	YNL021W	YBR082C	YDL192W	YDR174W	YKL056C	YNL299W
	0.174	0.116	0.106	0.106	0.105	0.100	0.096	0.086	0.086	0.084
	YDR457W	YML102W	YKL173W_tsq621	YDR037W_damp	YNL021W	YBR082C	YDL192W	YNL299W	YNL061W_tsq624	YAL021C_damp
	0.174	0.116	0.106	0.106	0.105	0.100	0.096	0.084	0.079	0.079
Hydroxyurea	YLR292C	YFR002W_damp	YKL056C	YLR018C	YOR326W_tsq337	YPR145W	YFR010W	YPL239W	YDL245C	YLR017W
	0.089	0.089	0.081	0.080	0.079	0.077	0.075	0.075	0.071	0.069
	YLR292C	YFR002W_damp	YKL056C	YLR018C	YOR326W_tsq337	YPR145W	YFR010W	YDL245C	YLR017W	YPL086C
	0.089	0.089	0.081	0.080	0.079	0.077	0.075	0.071	0.069	0.068
Artemisinin	YDR457W	YER180C-A	YKL173W_tsq621	YDR037W_damp	YDL006W	YGL105W	YML102W	YDL192W	YDL051W	YBR200W
	0.154	0.136	0.130	0.124	0.121	0.121	0.120	0.108	0.106	0.103
	YDR457W	YER180C-A	YKL173W_tsq621	YDR037W_damp	YDL006W	YGL105W	YML102W	YDL192W	YDL051W	YBR200W
	0.154	0.136	0.130	0.124	0.121	0.121	0.120	0.108	0.106	0.103
Amantadine hydrochloride	YLR314C_tsq130	YLR452C	YKL184W	YML069W_tsq846	YGR119C_tsq957	YKR062W_tsq692	YLR298C_tsq844	YDR457W	YHR174W	YNL328C
	0.123	0.121	0.114	0.100	0.090	0.085	0.081	0.079	0.075	0.073
	YLR314C_tsq130	YLR452C	YKL184W	YML069W_tsq846	YGR119C_tsq957	YKR062W_tsq692	YLR298C_tsq844	YDR457W	YNL328C	YBR193C_tsq741
	0.123	0.121	0.114	0.100	0.090	0.085	0.081	0.079	0.073	0.072
4-Hydroxytamoxifen	YKL184W	YKR062W_tsq692	YLR078C_tsq199	YMR272C	YHR194W	YNL128W	YDL245C	YKR082W	YKL068W	YGL179C
	0.126	0.113	0.097	0.082	0.081	0.076	0.073	0.066	0.065	0.064
	YKL184W	YKR062W_tsq692	YLR078C_tsq199	YMR272C	YHR194W	YNL128W	YDL245C	YKR082W	YKL068W	YDR253C
	0.126	0.113	0.097	0.082	0.081	0.076	0.073	0.066	0.065	0.059
Usnic acid	YDR457W	YML102W	YDL192W	YER083C	YGL173C	YNL299W	YDL006W	YPL158C	YJL124C	YHR208W
	0.202	0.141	0.131	0.120	0.119	0.111	0.109	0.107	0.106	0.106
	YDR457W	YML102W	YDL192W	YER083C	YGL173C	YNL299W	YDL006W	YJL124C	YHR208W	YGL020C
	0.202	0.141	0.131	0.120	0.119	0.111	0.109	0.106	0.106	0.102
Sodium Azide	YHR167W	YDR457W	YNL199C	YJR140C	YIL084C	YOR038C	YLR268W_tsq121	YDR363W-A	YDR037W_damp	YPR070W
	0.158	0.142	0.140	0.135	0.133	0.131	0.121	0.120	0.116	0.115
	YHR167W	YDR457W	YNL199C	YJR140C	YIL084C	YOR038C	YLR268W_tsq121	YDR363W-A	YDR037W_damp	YPR070W
	0.158	0.142	0.140	0.135	0.133	0.131	0.121	0.120	0.116	0.115

Table B.1: Target genes for $D = 82$ test chemical compounds. The list of $N_\tau = 10$ target genes (columns) for each test drug d in [138] (rows) is obtained by comparing the chemical-genetic fitness profile $\mathbf{y}_d$, with the double-mutant fitness profiles $\{\mathbf{x}_g\}_{g=1}^{N_g}$ of $N_g = 1,709$ candidate gene targets, and selecting the 10-largest correlation coefficients r_{dg}. The procedure is repeated with 70% of the data after reducing the number of equations in (3.1) from $N_c = 2,725$ to $N_w = 1,932$ via (3.11). Consistently for all drugs, the primary target is not altered when discarding data, whereas there is a 10% mismatch between the entire collections of N_τ target genes obtained with the full and reduced ensembles. Table 3.1 is continued as Tables 3.2-3.5 in the following pages.

70% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
100% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
Nystatin	YDR457W	YFR010W	YLR342W	YFR004W_tsq534	YCR077C	YOR329C_damp	YFR052W_tsq405	YLR251W	YGR027C	YDL007W_damp
	0.173	0.122	0.114	0.112	0.091	0.090	0.080	0.079	0.076	0.075
	YDR457W	YFR010W	YLR342W	YFR004W_tsq534	YCR077C	YOR329C_damp	YFR052W_tsq405	YLR251W	YGR027C	YGR244C
	0.173	0.122	0.114	0.112	0.091	0.090	0.080	0.079	0.076	0.073
Neomycin sulfate	YDL120W_damp	YNL197C	YDR500C	YDL245C	YMR145C	YFL039C_tsq142	YMR233W	YDL126C_tsq209	YGR119C_tsq957	YNL328C
	0.110	0.097	0.094	0.088	0.086	0.083	0.079	0.075	0.075	0.075
	YDL120W_damp	YNL197C	YDR500C	YDL245C	YMR145C	YFL039C_tsq142	YMR233W	YDL126C_tsq209	YGR119C_tsq957	YNL302C
	0.110	0.097	0.094	0.088	0.086	0.083	0.079	0.075	0.075	0.074
Caffeine	YDL245C	YML069W_tsq846	YDL192W	YMR083W	YPR124W	YER017C	YDR228C_tsq686	YGL179C	YGR072W	YBR042C
	0.128	0.093	0.089	0.089	0.085	0.083	0.081	0.078	0.077	0.076
	YDL245C	YML069W_tsq846	YDL192W	YMR083W	YPR124W	YER017C	YDR228C_tsq686	YGL179C	YBR042C	YPR068C
	0.128	0.093	0.089	0.089	0.085	0.083	0.081	0.078	0.076	0.075
Menthol	YOR038C	YAR002W	YJR140C	YLR452C	YPR070W	YMR233W	YBR060C_tsq295	YER006W_tsq786	YGR006W_tsq434	YBR193C_tsq741
	0.138	0.133	0.130	0.123	0.121	0.119	0.117	0.114	0.110	0.109
	YOR038C	YAR002W	YJR140C	YLR452C	YPR070W	YBR060C_tsq295	YER006W_tsq786	YGR006W_tsq434	YBR193C_tsq741	YKL173W_tsq621
	0.138	0.133	0.130	0.123	0.121	0.117	0.114	0.110	0.109	0.108
Verrucarin	YFL001W	YNL299W	YHR066W	YOR001W	YGL078C	YDL082W	YDL192W	YPL158C	YEL055C_tsq595	YAL059W
	0.133	0.102	0.097	0.093	0.090	0.088	0.084	0.084	0.082	0.079
	YFL001W	YNL299W	YHR066W	YOR001W	YGL078C	YDL082W	YDL192W	YPL158C	YEL055C_tsq595	YAL059W
	0.133	0.102	0.097	0.093	0.090	0.088	0.084	0.084	0.082	0.079
Valinomycin	YDR457W	YML114C_tsq695	YGL173C	YFR004W_tsq534	YNL299W	YGR027C	YDL020C	YCR077C	YBR010W	YMR224C
	0.245	0.154	0.136	0.130	0.128	0.121	0.111	0.109	0.106	0.106
	YDR457W	YML114C_tsq695	YGL173C	YFR004W_tsq534	YNL299W	YGR027C	YDL020C	YCR077C	YBR010W	YMR224C
	0.245	0.154	0.136	0.130	0.128	0.121	0.111	0.109	0.106	0.106
Trifluoroperazine	YDR457W	YFR010W	YKL184W	YPR028W	YER073W	YPR179C	YLR314C_tsq130	YIL071C	YML069W_tsq846	YDR495C
	0.094	0.093	0.089	0.086	0.079	0.077	0.075	0.075	0.074	0.072
	YDR457W	YFR010W	YKL184W	YPR028W	YER073W	YPR179C	YLR314C_tsq130	YDR495C	YMR048W	YDR113C_tsq862
	0.094	0.093	0.089	0.086	0.079	0.077	0.075	0.072	0.071	0.066
Tamoxifen	YDR457W	YFR010W	YKR062W_tsq692	YLR298C_tsq844	YNL061W_tsq630	YLR314C_tsq130	YLR262C	YLR298C_tsq840	YPR028W	YOL104C
	0.121	0.103	0.100	0.085	0.084	0.077	0.076	0.074	0.070	0.070
	YDR457W	YFR010W	YKR062W_tsq692	YLR298C_tsq844	YNL061W_tsq630	YLR314C_tsq130	YLR262C	YLR298C_tsq840	YPR028W	YBR098W
	0.121	0.103	0.100	0.085	0.084	0.077	0.076	0.074	0.070	0.068
Raloxifene	YMR056C	YJL105W	YDL245C	YFR003C_damp	YCR077C	YDR363W-A	YDR457W	YDL120W_damp	YKL184W	YOL104C
	0.105	0.091	0.084	0.084	0.076	0.062	0.058	0.058	0.053	0.053
	YMR056C	YJL105W	YDL245C	YFR003C_damp	YCR077C	YDR363W-A	YDL120W_damp	YOL104C	YBR042C	YOL062C
	0.105	0.091	0.084	0.084	0.076	0.062	0.058	0.053	0.052	0.052
Pentamidine	YDR457W	YDL006W	YGL173C	YNL230C	YML069W_tsq846	YGR245C_tsq523	YKR086W_tsq644	YFR052W_tsq405	YCR077C	YML087C
	0.195	0.119	0.118	0.107	0.107	0.106	0.104	0.102	0.101	0.101
	YDR457W	YDL006W	YGL173C	YML069W_tsq846	YGR245C_tsq523	YKR086W_tsq644	YFR052W_tsq405	YCR077C	YML087C	YJL124C
	0.195	0.119	0.118	0.107	0.106	0.104	0.102	0.101	0.101	0.098
Nigericin	YDR457W	YGL173C	YJL176C	YFR004W_tsq534	YDL192W	YLR314C_tsq130	YLR314C_tsq885	YMR078C	YJR043C	YHR166C_tsq89
	0.111	0.103	0.092	0.092	0.091	0.089	0.080	0.079	0.074	0.071
	YDR457W	YGL173C	YJL176C	YFR004W_tsq534	YDL192W	YLR314C_tsq130	YLR314C_tsq885	YMR078C	YJR043C	YHR166C_tsq89
	0.111	0.103	0.092	0.092	0.091	0.089	0.080	0.079	0.074	0.071
LY-294,002	YJR073C	YLR314C_tsq130	YBR009C	YAL060W	YDL120W_damp	YMR069W	YNL061W_tsq630	YBR016W	YDL192W	YML102W
	0.076	0.075	0.074	0.066	0.065	0.065	0.063	0.061	0.059	0.058
	YJR073C	YLR314C_tsq130	YBR009C	YAL060W	YDL120W_damp	YMR069W	YNL061W_tsq630	YDL192W	YML102W	YKR019C
	0.076	0.075	0.074	0.066	0.065	0.065	0.063	0.059	0.058	0.055
Latrunculin B	YDL245C	YLR452C	YPR119W	YBL031W	YMR083W	YLR165C	YOL077W-A	YGL019W	YDL192W	YER017C
	0.120	0.104	0.096	0.094	0.094	0.091	0.087	0.078	0.078	0.077
	YDL245C	YLR452C	YPR119W	YBL031W	YMR083W	YLR165C	YOL077W-A	YGL019W	YDL192W	YER017C
	0.120	0.104	0.096	0.094	0.094	0.091	0.087	0.078	0.078	0.077
Hydroxyethilhidrazine	YDR457W	YPR178W_tsq819	YML103C	YLR350W	YHR167W	YOR116C_tsq831	YER111C	YDR335W	YLR268W_tsq121	YNL299W
	0.121	0.120	0.119	0.104	0.097	0.093	0.091	0.086	0.085	0.084
	YDR457W	YPR178W_tsq819	YML103C	YLR350W	YHR167W	YOR116C_tsq831	YER111C	YDR335W	YLR268W_tsq121	YNL299W
	0.121	0.120	0.119	0.104	0.097	0.093	0.091	0.086	0.085	0.084
Hydrogen peroxide	YDR168W_tsq315	YHR208W	YPR070W	YKL055C	YOL054W	YOL012C	YHR194W	YOR070C	YJR049C	YDR298C
	0.106	0.102	0.099	0.097	0.090	0.083	0.081	0.081	0.081	0.080
	YDR168W_tsq315	YHR208W	YPR070W	YKL055C	YOL054W	YHR194W	YOR070C	YJR049C	YDR298C	YPR167C
	0.106	0.102	0.099	0.097	0.090	0.081	0.081	0.081	0.080	0.079
Hoechst	YDR162C	YDL006W	YGL233W_tsq43	YNL113W_damp	YHR187W	YNL215W	YHR107C_tsq33	YLR166C_tsq66	YJL095W	YPL211W_tsq114
	0.146	0.134	0.114	0.113	0.110	0.106	0.105	0.104	0.101	0.098
	YDR162C	YDL006W	YGL233W_tsq43	YNL113W_damp	YHR187W	YNL215W	YHR107C_tsq33	YLR166C_tsq66	YJL095W	YDL225W
	0.146	0.134	0.114	0.113	0.110	0.106	0.105	0.104	0.101	0.096
Harmine	YPL256C	YGR274C_tsq524	YGL105W	YDR505C	YNL199C	YGL213C	YOR038C	YDL174C	YDR457W	YDR037W_damp
	0.171	0.138	0.128	0.127	0.121	0.120	0.118	0.117	0.115	0.114
	YPL256C	YGR274C_tsq524	YGL105W	YDR505C	YNL199C	YGL213C	YOR038C	YDR037W_damp	YOL006C	YJR140C
	0.171	0.138	0.128	0.127	0.121	0.120	0.118	0.114	0.113	0.113

Table B.2: Target genes for $D = 82$ test chemical compounds (continuation); see description in Table 3.1.

70% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
100% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
Haloperidol	YFR002W_damp	YGL116W_tsq368	YER105C	YFL009W_tsq334	YKL173W_tsq621	YDL051W	YOR025W	YFR052W_tsq405	YKR082W	YER107C
	0.092	0.092	0.090	0.078	0.077	0.077	0.074	0.074	0.073	0.072
	YFR002W_damp	YGL116W_tsq368	YER105C	YFL009W_tsq334	YKL173W_tsq621	YDL051W	YOR025W	YFR052W_tsq405	YKR082W	YFL008W_tsq71
	0.092	0.092	0.090	0.078	0.077	0.077	0.074	0.074	0.073	0.072
Fenpropimorph	YDR315C	YBR060C_tsq295	YEL065W	YDR511W	YOR136W	YCR086W	YLR452C	YNL199C	YEL061C	YDL245C
	0.080	0.074	0.065	0.064	0.062	0.062	0.062	0.060	0.059	0.059
	YDR315C	YBR060C_tsq295	YEL065W	YOR136W	YCR086W	YLR452C	YNL199C	YDL245C	YKL057C	YIR019C
	0.080	0.074	0.065	0.062	0.062	0.062	0.060	0.059	0.058	0.053
Emetine	YBR211C_tsq716	YKL004W_damp	YJL187C	YGR059W	YBL105C_tsq535	YDR507C	YER157W_tsq44	YOR068C	YDL051W	YOL068C
	0.133	0.103	0.102	0.102	0.101	0.100	0.098	0.095	0.094	0.093
	YBR211C_tsq716	YKL004W_damp	YJL187C	YGR059W	YBL105C_tsq535	YDR507C	YER157W_tsq44	YOR068C	YDL051W	YOL068C
	0.133	0.103	0.102	0.102	0.101	0.100	0.098	0.095	0.094	0.093
Dyclonine	YGL213C	YHL047C	YJL105W	YHR087W	YNR051C	YDL226C	YOR310C_damp	YHR111W	YIL011W	YLL004W_tsq351
	0.085	0.083	0.083	0.081	0.081	0.079	0.076	0.074	0.074	0.073
	YGL213C	YHL047C	YJL105W	YNR051C	YDL226C	YHR111W	YIL011W	YLL004W_tsq351	YLL050C_tsq234	YGL127C
	0.085	0.083	0.083	0.081	0.079	0.074	0.074	0.073	0.072	0.071
Doxycycline	YDR457W	YGL020C	YLR268W_tsq121	YER083C	YML102W	YER180C-A	YFR004W_tsq534	YFL001W	YMR048W	YBR087W_tsq887
	0.151	0.131	0.129	0.127	0.124	0.116	0.115	0.115	0.114	0.108
	YDR457W	YGL020C	YLR268W_tsq121	YER083C	YML102W	YER180C-A	YFR004W_tsq534	YFL001W	YMR048W	YGR009C_tsq60
	0.151	0.131	0.129	0.127	0.124	0.116	0.115	0.115	0.114	0.107
Cyclopiazonic acid	YMR233W	YNL300W	YGL252C	YHR194W	YLR239C	YLL039C	YLR378C_tsq213	YCL069W	YLR314C_tsq130	YIL048W_tsq188
	0.087	0.077	0.073	0.073	0.068	0.067	0.065	0.062	0.060	0.060
	YMR233W	YNL300W	YGL252C	YHR194W	YLR239C	YLL039C	YLR378C_tsq213	YCL069W	YLR314C_tsq130	YHR064C
	0.087	0.077	0.073	0.073	0.068	0.067	0.065	0.062	0.060	0.052
Clomiphene	YFR010W	YDR457W	YDR159W	YLR298C_tsq844	YLR438C-A_damp	YDR511W	YPR043W	YDL192W	YGR009C_tsq60	YPR028W
	0.140	0.108	0.098	0.090	0.089	0.088	0.087	0.083	0.083	0.080
	YFR010W	YDR457W	YDR159W	YLR298C_tsq844	YLR438C-A_damp	YDR511W	YPR043W	YDL192W	YGR009C_tsq60	YPR028W
	0.140	0.108	0.098	0.090	0.089	0.088	0.087	0.083	0.083	0.080
Cisplatin	YDR457W	YLR298C_tsq844	YPR101W	YKL173W_tsq621	YIL137C	YKR046C	YGR124W	YGL097W_tsq958	YKL056C	YDR243C_tsq574
	0.120	0.115	0.109	0.098	0.090	0.090	0.086	0.086	0.085	0.085
	YDR457W	YLR298C_tsq844	YPR101W	YKL173W_tsq621	YIL137C	YKR046C	YGL097W_tsq958	YDR243C_tsq574	YER006W_tsq786	YDR330W
	0.120	0.115	0.109	0.098	0.090	0.090	0.086	0.085	0.084	0.083
Chlorpromazine	YPR028W	YER017C	YKL184W	YHR174W	YLR017W	YKR062W_tsq692	YHR064C	YHR194W	YMR083W	YMR282C
	0.095	0.087	0.087	0.086	0.084	0.083	0.082	0.082	0.077	0.076
	YPR028W	YER017C	YKL184W	YHR174W	YLR017W	YKR062W_tsq692	YHR064C	YHR194W	YMR083W	YNL061W_tsq624
	0.095	0.087	0.087	0.086	0.084	0.083	0.082	0.082	0.077	0.069
Cerulenin	YNL270C	YMR319C	YGL009C	YNL291C	YGR027C	YKR026C	YDR283C	YDR329C	YDL217C_tsq714	YDR312W
	0.111	0.093	0.090	0.090	0.082	0.078	0.078	0.076	0.075	0.074
	YNL270C	YMR319C	YGL009C	YNL291C	YGR027C	YKR026C	YDR283C	YDR329C	YDL217C_tsq714	YGR081C
	0.111	0.093	0.090	0.090	0.082	0.078	0.078	0.076	0.075	0.072
Calcium ionophore	YOL151W	YFR051C_tsq823	YHR179W	YHR176W	YDR122W	YDR168W_tsq315	YOR124C	YPL141C	YDR002W_tsq582	YDR457W
	0.075	0.075	0.068	0.066	0.065	0.065	0.064	0.064	0.062	0.061
	YOL151W	YFR051C_tsq823	YHR179W	YHR176W	YOR124C	YPL141C	YDR002W_tsq582	YDR457W	YEL018W	YDR037W_damp
	0.075	0.075	0.068	0.066	0.064	0.064	0.062	0.061	0.059	0.054
Anisomycin	YPL017C	YLR314C_tsq885	YBR211C_tsq716	YOR329C_damp	YDL150W_damp	YPL204W_damp	YPR020W	YKR062W_tsq692	YBR080C_tsq57	YNL051W
	0.075	0.075	0.074	0.071	0.069	0.068	0.068	0.068	0.067	0.067
	YPL017C	YLR314C_tsq885	YBR211C_tsq716	YOR329C_damp	YDL150W_damp	YPL204W_damp	YPR020W	YNL051W	YBR037C	YDL107W
	0.075	0.075	0.074	0.071	0.069	0.068	0.068	0.067	0.064	0.058
Amphotericin	YDR457W	YDL192W	YFR010W	YCR077C	YDL245C	YGL173C	YKL078W_damp	YJL124C	YDR052C_tsq163	YGR244C
	0.179	0.120	0.110	0.108	0.103	0.094	0.093	0.093	0.092	0.092
	YDR457W	YDL192W	YFR010W	YCR077C	YDL245C	YGL173C	YKL078W_damp	YJL124C	YDR052C_tsq163	YGR244C
	0.179	0.120	0.110	0.108	0.103	0.094	0.093	0.093	0.092	0.092
Amiodarone	YDR457W	YDL192W	YFR010W	YLR262C	YNL299W	YLR298C_tsq844	YBR082C	YGL173C	YDR174W	YGL195W
	0.195	0.098	0.093	0.093	0.091	0.088	0.083	0.081	0.079	0.076
	YDR457W	YDL192W	YFR010W	YLR262C	YNL299W	YLR298C_tsq844	YBR082C	YGL173C	YGL195W	YJL124C
	0.195	0.098	0.093	0.093	0.091	0.088	0.083	0.081	0.076	0.074
Alamethicin	YGL127C	YLR398C	YPR070W	YNR010W	YOR038C	YHR167W	YNL199C	YBR278W	YOR076C	YOL054W
	0.102	0.095	0.094	0.092	0.090	0.086	0.085	0.084	0.083	0.079
	YGL127C	YLR398C	YPR070W	YNR010W	YOR038C	YHR167W	YNL199C	YBR278W	YOL054W	YPR167C
	0.102	0.095	0.094	0.092	0.090	0.086	0.085	0.084	0.079	0.077
Actinomycin	YDR457W	YDL245C	YMR319C	YPR178W_tsq500	YLR268W_tsq121	YPR178W_tsq819	YOL062C	YCR077C	YMR092C	YDR243C_tsq574
	0.128	0.126	0.120	0.110	0.105	0.096	0.093	0.087	0.085	0.083
	YDR457W	YDL245C	YMR319C	YPR178W_tsq500	YLR268W_tsq121	YPR178W_tsq819	YOL062C	YMR092C	YDR243C_tsq574	YOR116C_tsq831
	0.128	0.126	0.120	0.110	0.105	0.096	0.093	0.085	0.083	0.081
Abietic acid	YBR249C	YFR010W	YDR148C	YJL105W	YCR077C	YDR283C	YIL023C	YLR410W	YPL109C	YLR418C
	0.086	0.083	0.082	0.081	0.079	0.079	0.077	0.075	0.074	0.073
	YBR249C	YFR010W	YDR148C	YJL105W	YCR077C	YDR283C	YIL023C	YLR410W	YKL218C	YDR322C-A
	0.086	0.083	0.082	0.081	0.079	0.079	0.077	0.075	0.070	0.067
Wortmannin	YDL192W	YGR028W	YDR457W	YDR033W	YKR075C	YOR285W	YMR302C	YER173W	YKR092C	YLL046C
	0.139	0.116	0.111	0.102	0.095	0.092	0.091	0.089	0.089	0.087
	YDL192W	YGR028W	YDR457W	YDR033W	YKR075C	YOR285W	YMR302C	YER173W	YKR092C	YMR083W
	0.139	0.116	0.111	0.102	0.095	0.092	0.091	0.089	0.089	0.085

Table B.3: Target genes for $D = 82$ test chemical compounds (continuation); see description in Table 3.1.

70% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
100% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
Staurosporine	YDL245C	YOL062C	YPL037C	YPL270W	YOL077W-A	YNL061W_tsq624	YBR042C	YER017C	YMR056C	YLR298C_tsq840
	0.175	0.107	0.107	0.104	0.104	0.095	0.093	0.087	0.086	0.083
	YDL245C	YOL062C	YPL037C	YPL270W	YOL077W-A	YNL061W_tsq624	YBR042C	YER017C	YMR056C	YLR298C_tsq840
	0.175	0.107	0.107	0.104	0.104	0.095	0.093	0.087	0.086	0.083
Conine	YMR071C	YJL187C	YBR034C	YFL009W_tsq415	YER133W_damp	YDR200C	YKL211C	YOR076C	YLL049W	YER086W
	0.090	0.087	0.081	0.080	0.080	0.080	0.074	0.073	0.071	0.071
	YMR071C	YJL187C	YBR034C	YFL009W_tsq415	YER133W_damp	YDR200C	YKL211C	YLL049W	YER086W	YMR025W
	0.090	0.087	0.081	0.080	0.080	0.080	0.074	0.071	0.071	0.063
Parthenolide	YNL299W	YDL120W_damp	YKL056C	YKR062W_tsq692	YHR194W	YKL184W	YFR002W_damp	YOR077W_damp	YFL009W_tsq334	YLR452C
	0.094	0.076	0.075	0.074	0.073	0.072	0.066	0.064	0.062	0.062
	YNL299W	YDL120W_damp	YKL056C	YKR062W_tsq692	YHR194W	YKL184W	YFR002W_damp	YOR077W_damp	YFL009W_tsq334	YLR452C
	0.094	0.076	0.075	0.074	0.073	0.072	0.066	0.064	0.062	0.062
Radicicol	YMR272C	YDR457W	YGL020C	YFL001W	YDR037W_damp	YER083C	YLR268W_tsq121	YGR027C	YOR116C_tsq831	YMR235C_tsq172
	0.130	0.124	0.113	0.112	0.112	0.111	0.109	0.101	0.097	0.096
	YMR272C	YDR457W	YGL020C	YFL001W	YDR037W_damp	YER083C	YLR268W_tsq121	YGR027C	YOR116C_tsq831	YMR235C_tsq172
	0.130	0.124	0.113	0.112	0.112	0.111	0.109	0.101	0.097	0.096
Mitomycin C	YCR077C	YLR059C	YLR160C	YDL061C	YER177W	YBR045C	YPR155C	YJL105W	YGL200C	YBR093C
	0.081	0.078	0.077	0.072	0.069	0.062	0.062	0.062	0.059	0.059
	YCR077C	YLR059C	YLR160C	YER177W	YBR045C	YPR155C	YJL105W	YGL200C	YBR077C	YNL216W_damp
	0.081	0.078	0.077	0.069	0.062	0.062	0.062	0.059	0.058	0.055
Trichostatin A	YLR452C	YKL051W	YER006W_tsq786	YOL077W-A	YMR308C_tsq683	YLR314C_tsq130	YOL006C	YOR310C_damp	YFL001W	YNR022C
	0.149	0.124	0.109	0.108	0.104	0.103	0.103	0.101	0.100	0.099
	YLR452C	YKL051W	YER006W_tsq786	YOL077W-A	YMR308C_tsq683	YLR314C_tsq130	YOL006C	YOR310C_damp	YFL001W	YDR037W_damp
	0.149	0.124	0.109	0.108	0.104	0.103	0.103	0.101	0.100	0.099
FK506	YPL207W	YLR410W	YDR033W	YML114C_tsq695	YOL092W	YBR087W_tsq887	YIL017C	YBR042C	YDR315C	YGR244C
	0.129	0.122	0.114	0.095	0.093	0.093	0.089	0.087	0.085	0.083
	YPL207W	YLR410W	YDR033W	YML114C_tsq695	YOL092W	YBR087W_tsq887	YIL017C	YBR042C	YDR315C	YGR244C
	0.129	0.122	0.114	0.095	0.093	0.093	0.089	0.087	0.085	0.083
Brefeldin A	YPR179C	YLR314C_tsq130	YMR223W	YMR179W	YNL021W	YHR174W	YPL047W	YMR078C	YDL192W	YNL061W_tsq624
	0.107	0.106	0.104	0.101	0.098	0.096	0.093	0.091	0.087	0.083
	YPR179C	YLR314C_tsq130	YMR223W	YMR179W	YNL021W	YHR174W	YPL047W	YMR078C	YDL192W	YNL061W_tsq624
	0.107	0.106	0.104	0.101	0.098	0.096	0.093	0.091	0.087	0.083
U73122	YDR457W	YDL192W	YNL156C	YDR253C	YLR017W	YLR298C_tsq844	YLR314C_tsq130	YPR134W	YDL226C	YBL047C
	0.152	0.109	0.102	0.090	0.085	0.084	0.081	0.080	0.078	0.077
	YDR457W	YDL192W	YNL156C	YLR017W	YLR298C_tsq844	YLR314C_tsq130	YPR134W	YDL226C	YBL047C	YDL102W_tsq135
	0.152	0.109	0.102	0.085	0.084	0.081	0.080	0.078	0.077	0.074
Tunicamycin	YKL048C	YKL101W	YDR457W	YCR077C	YNL298W	YDR382W	YGL173C	YPL270W	YNL328C	YHR166C_tsq89
	0.119	0.115	0.103	0.092	0.091	0.089	0.079	0.077	0.074	0.073
	YKL048C	YKL101W	YDR457W	YCR077C	YNL298W	YDR382W	YGL173C	YPL270W	YNL328C	YHR166C_tsq89
	0.119	0.115	0.103	0.092	0.091	0.089	0.079	0.077	0.074	0.073
Thialysine	YMR319C	YBL105C_tsq535	YBL015W	YER007C-A	YDR457W	YGL098W_tsq592	YBR104W	YLR384C	YNL016W	YBL003C
	0.164	0.127	0.126	0.119	0.110	0.107	0.104	0.103	0.103	0.102
	YMR319C	YBL105C_tsq535	YBL015W	YER007C-A	YDR457W	YGL098W_tsq592	YBR104W	YBL003C	YGL020C	YOR246C
	0.164	0.127	0.126	0.119	0.110	0.107	0.104	0.102	0.102	0.101
Rapamycin	YLR314C_tsq130	YDL192W	YDR153C	YPL167C	YIL078W_damp	YGL055W_tsq506	YDL245C	YGR072W	YDL181W	YDR228C_tsq685
	0.089	0.085	0.074	0.073	0.070	0.070	0.066	0.066	0.065	0.061
	YLR314C_tsq130	YDL192W	YDR153C	YPL167C	YIL078W_damp	YGL055W_tsq506	YDL245C	YGR072W	YDL181W	YOR361C_tsq30
	0.089	0.085	0.074	0.073	0.070	0.070	0.066	0.066	0.065	0.057
Phenylarsine oxide	YKR082W	YKR062W_tsq692	YKL184W	YFR010W	YJR140C	YKL055C	YHR167W	YDR457W	YKL211C	YLR452C
	0.140	0.119	0.111	0.110	0.106	0.097	0.091	0.089	0.088	0.086
	YKR082W	YKR062W_tsq692	YKL184W	YFR010W	YJR140C	YKL055C	YHR167W	YDR457W	YKL211C	YLR452C
	0.140	0.119	0.111	0.110	0.106	0.097	0.091	0.089	0.088	0.086
Phenantroline	YML069W_tsq846	YJL091C_tsq655	YMR149W_damp	YFR052W_tsq405	YHR200W	YMR198W	YOR123C	YDR162C	YLR350W	YDR472W_damp
	0.155	0.126	0.119	0.115	0.111	0.109	0.107	0.105	0.103	0.102
	YML069W_tsq846	YJL091C_tsq655	YMR149W_damp	YFR052W_tsq405	YHR200W	YMR198W	YOR123C	YDR162C	YLR350W	YDR472W_damp
	0.155	0.126	0.119	0.115	0.111	0.109	0.107	0.105	0.103	0.102
Oligomycin	YDR457W	YKR019C	YGL116W_tsq368	YIL106W_damp	YFR004W_tsq534	YDR435C	YBR082C	YNL299W	YDR121W	YPR141C
	0.102	0.095	0.094	0.092	0.091	0.087	0.085	0.080	0.076	0.076
	YDR457W	YKR019C	YGL116W_tsq368	YIL106W_damp	YFR004W_tsq534	YDR435C	YBR082C	YNL299W	YPR141C	YEL027W
	0.102	0.095	0.094	0.092	0.091	0.087	0.085	0.080	0.076	0.074
Nocodazole	YFR052W_tsq405	YER111C	YMR198W	YDR448W	YKL101W	YER083C	YDR457W	YFR004W_tsq534	YGL020C	YML103C
	0.156	0.146	0.140	0.126	0.124	0.123	0.119	0.118	0.111	0.108
	YFR052W_tsq405	YER111C	YMR198W	YDR448W	YKL101W	YER083C	YDR457W	YFR004W_tsq534	YGL020C	YML103C
	0.156	0.146	0.140	0.126	0.124	0.123	0.119	0.118	0.111	0.108
Hygromycin B	YDR293C	YDL192W	YDR512C	YMR023C	YMR078C	YPL130W	YIL017C	YER073W	YEL050C	YJR043C
	0.098	0.094	0.091	0.086	0.081	0.076	0.075	0.073	0.071	0.071
	YDR293C	YDL192W	YDR512C	YMR023C	YMR078C	YPL130W	YIL017C	YER073W	YOR361C_tsq30	YBL103C
	0.098	0.094	0.091	0.086	0.081	0.076	0.075	0.073	0.070	0.070
Extract 95-57	YKL184W	YPL003W	YER017C	YHR194W	YLR133W	YLR061W	YKR062W_tsq692	YOR361C_tsq30	YJL101C	YIL134W
	0.099	0.073	0.071	0.065	0.063	0.060	0.059	0.056	0.056	0.055
	YKL184W	YPL003W	YER017C	YHR194W	YLR133W	YLR061W	YKR062W_tsq692	YJL101C	YLR078C_tsq199	YLR388W
	0.099	0.073	0.071	0.065	0.063	0.060	0.059	0.056	0.049	0.046

Table B.4: Target genes for $D = 82$ test chemical compounds (continuation); see description in Table 3.1.

70% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
100% data	target gene 1	target gene 2	target gene 3	target gene 4	target gene 5	target gene 6	target gene 7	target gene 8	target gene 9	target gene 10
	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}	r_{dj}
Extract 6592	YDR457W	YJL124C	YDR037W_damp	YBR278W	YML102W	YOR038C	YGR027C	YBR009C	YDR174W	YJL013C
	0.190	0.109	0.104	0.102	0.101	0.098	0.097	0.097	0.096	0.095
	YDR457W	YJL124C	YDR037W_damp	YBR278W	YML102W	YOR038C	YGR027C	YBR009C	YDR174W	YHR167W
	0.190	0.109	0.104	0.102	0.101	0.098	0.097	0.097	0.096	0.093
Extract 00-89	YKL184W	YKR062W_tsq692	YGL233W_tsq43	YGR118W	YBR048W	YDR472W_damp	YPL029W	YPL110C	YJR040W	YLR388W
	0.122	0.096	0.089	0.080	0.073	0.073	0.072	0.071	0.068	0.066
	YKL184W	YKR062W_tsq692	YGL233W_tsq43	YGR118W	YBR048W	YDR472W_damp	YPL029W	YPL110C	YJR040W	YBR236C_tsq78
	0.122	0.096	0.089	0.080	0.073	0.073	0.072	0.071	0.068	0.056
Extract 00-303C	YML114C_tsq695	YDR037W_damp	YDR457W	YGL127C	YPR178W_tsq500	YGL195W	YOR329C_damp	YDL192W	YPR101W	YFR010W
	0.158	0.111	0.111	0.108	0.108	0.096	0.096	0.094	0.092	0.091
	YML114C_tsq695	YDR037W_damp	YDR457W	YGL127C	YPR178W_tsq500	YGL195W	YOR329C_damp	YDL192W	YPR101W	YFR010W
	0.158	0.111	0.111	0.108	0.108	0.096	0.096	0.094	0.092	0.091
Extract 00-243	YDR457W	YML114C_tsq695	YLR410W	YGL213C	YDL192W	YJL074C_tsq63	YCR077C	YGL195W	YER013W_tsq237	YDR174W
	0.130	0.120	0.111	0.106	0.103	0.093	0.092	0.091	0.091	0.091
	YDR457W	YML114C_tsq695	YLR410W	YGL213C	YDL192W	YJL074C_tsq63	YCR077C	YGL195W	YER013W_tsq237	YDR315C
	0.130	0.120	0.111	0.106	0.103	0.093	0.092	0.091	0.091	0.085
Extract 00-192	YDR457W	YCR077C	YGL195W	YMR179W	YDR283C	YDL020C	YML114C_tsq695	YNL299W	YJL124C	YLR438C-A_damp
	0.225	0.117	0.109	0.099	0.098	0.095	0.094	0.093	0.092	0.091
	YDR457W	YCR077C	YGL195W	YMR179W	YDR283C	YDL020C	YML114C_tsq695	YNL299W	YJL124C	YLR438C-A_damp
	0.225	0.117	0.109	0.099	0.098	0.095	0.094	0.093	0.092	0.091
Extract 00-132	YDR457W	YDL192W	YGL195W	YML008C	YDR283C	YPL085W_tsq119	YPR101W	YPL158C	YPR062W	YGL179C
	0.158	0.095	0.086	0.083	0.077	0.075	0.075	0.074	0.071	0.070
	YDR457W	YDL192W	YGL195W	YML008C	YDR283C	YPL085W_tsq119	YPR101W	YPL158C	YPR062W	YPL213W
	0.158	0.095	0.086	0.083	0.077	0.075	0.075	0.074	0.071	0.069
Emodin	YDR180W_tsq69	YKL173W_tsq621	YFL008W_tsq71	YDL174C	YDR457W	YBL078C	YMR233W	YKR019C	YOL006C	YNL223W
	0.092	0.085	0.084	0.081	0.079	0.076	0.075	0.074	0.073	0.073
	YDR180W_tsq69	YKL173W_tsq621	YFL008W_tsq71	YDL174C	YDR457W	YBL078C	YMR233W	YKR019C	YOL006C	YNL223W
	0.092	0.085	0.084	0.081	0.079	0.076	0.075	0.074	0.073	0.073
Desipramine	YKL184W	YLR314C_tsq130	YHR194W	YHR064C	YNL061W_tsq624	YDR253C	YHR174W	YKR026C	YGL195W	YKR062W_tsq692
	0.103	0.102	0.091	0.084	0.083	0.082	0.079	0.074	0.072	0.071
	YKL184W	YLR314C_tsq130	YHR194W	YHR064C	YNL061W_tsq624	YDR253C	YHR174W	YKR026C	YKR062W_tsq692	YGR127W
	0.103	0.102	0.091	0.084	0.083	0.082	0.079	0.074	0.071	0.070
Cytochalasin A	YDL245C	YER017C	YDL192W	YOL062C	YDR228C_tsq686	YOL077W-A	YKL055C	YHR194W	YDR228C_tsq685	YGL019W
	0.145	0.108	0.101	0.098	0.088	0.080	0.079	0.075	0.073	0.073
	YDL245C	YER017C	YDL192W	YOL062C	YDR228C_tsq686	YOL077W-A	YKL055C	YHR194W	YDR228C_tsq685	YLR298C_tsq840
	0.145	0.108	0.101	0.098	0.088	0.080	0.079	0.075	0.073	0.072
CG4-Theopalauamide	YDR457W	YPR124W	YDL192W	YMR179W	YER061C	YPL085W_tsq119	YGR244C	YDR283C	YOR306C	YKL214C
	0.121	0.092	0.074	0.073	0.069	0.068	0.066	0.065	0.064	0.061
	YDR457W	YPR124W	YDL192W	YMR179W	YER061C	YPL085W_tsq119	YGR244C	YDR283C	YGL179C	YPL158C
	0.121	0.092	0.074	0.073	0.069	0.068	0.066	0.065	0.060	0.058
Caspofungin	YDL245C	YDR207C	YDR457W	YPL177C	YNL061W_tsq624	YKL184W	YCR073W-A	YKR062W_tsq692	YGL009C	YNL315C
	0.097	0.081	0.076	0.075	0.073	0.073	0.068	0.068	0.068	0.066
	YDL245C	YDR207C	YDR457W	YPL177C	YNL061W_tsq624	YCR073W-A	YKR062W_tsq692	YGL009C	YOR329C_damp	YKL055C
	0.097	0.081	0.076	0.075	0.073	0.068	0.068	0.068	0.065	0.064
Camptothecin	YOL068C	YDL245C	YPR101W	YLR298C_tsq840	YMR056C	YCR077C	YLR298C_tsq844	YKL056C	YBL050W_tsq48	YDL192W
	0.120	0.110	0.107	0.106	0.103	0.095	0.093	0.093	0.087	0.085
	YOL068C	YDL245C	YPR101W	YLR298C_tsq840	YMR056C	YCR077C	YLR298C_tsq844	YKL056C	YDL192W	YKR092C
	0.120	0.110	0.107	0.106	0.103	0.095	0.093	0.093	0.085	0.084
Basiliskamide	YDR457W	YAL021C_damp	YOR080W	YDR335W	YBR042C	YKL062W	YNL299W	YDR168W_tsq315	YFR004W_tsq534	YNL061W_tsq630
	0.101	0.080	0.080	0.075	0.072	0.072	0.070	0.069	0.064	0.064
	YDR457W	YAL021C_damp	YOR080W	YDR335W	YBR042C	YKL062W	YNL299W	YDR168W_tsq315	YMR078C	YOR348C
	0.101	0.080	0.080	0.075	0.072	0.072	0.070	0.069	0.063	0.060
192A4-Stichloroside	YDR457W	YPL158C	YCR077C	YDR448W	YNL229C	YDR283C	YGL173C	YOL062C	YKR077W	YGR252W
	0.188	0.114	0.103	0.100	0.092	0.090	0.084	0.083	0.083	0.077
	YDR457W	YPL158C	YCR077C	YDR448W	YNL229C	YDR283C	YGL173C	YOL062C	YKR077W	YGR252W
	0.188	0.114	0.103	0.100	0.092	0.090	0.084	0.083	0.083	0.077
Papuamide B	YDL245C	YCR077C	YOL062C	YKL061W	YMR056C	YDR457W	YDR283C	YGL009C	YNL245C_damp	YEL038W
	0.085	0.082	0.068	0.065	0.065	0.064	0.059	0.059	0.058	0.058
	YDL245C	YCR077C	YOL062C	YKL061W	YMR056C	YDR457W	YGL009C	YEL038W	YER114C	YBL032W
	0.085	0.082	0.068	0.065	0.065	0.064	0.059	0.058	0.056	0.055
Agelasine E	YKL184W	YKR062W_tsq692	YDR472W_damp	YLR166C_tsq66	YGL116W_tsq368	YLR078C_tsq199	YDR121W	YDL058W_tsq441	YMR198W	YJR140C
	0.149	0.127	0.107	0.094	0.093	0.080	0.079	0.078	0.075	0.074
	YKL184W	YKR062W_tsq692	YDR472W_damp	YLR166C_tsq66	YGL116W_tsq368	YLR078C_tsq199	YDR121W	YDL058W_tsq441	YMR198W	YML069W_tsq846
	0.149	0.127	0.107	0.094	0.093	0.080	0.079	0.078	0.075	0.069
Fluconazole	YDR457W	YDL192W	YDR037W_damp	YGR009C_tsq60	YLR314C_tsq885	YPR018W	YGL020C	YER008C	YDR033W	YGR185C_tsq290
	0.146	0.107	0.105	0.102	0.102	0.099	0.092	0.092	0.091	0.091
	YDR457W	YDL192W	YDR037W_damp	YGR009C_tsq60	YLR314C_tsq885	YPR018W	YGL020C	YER008C	YGR185C_tsq290	YAL056W
	0.146	0.107	0.105	0.102	0.102	0.099	0.092	0.092	0.091	0.086
Geldanamycin	YMR272C	YML128C	YGR028W	YNL156C	YMR083W	YDR228C_tsq686	YPL183W-A	YLR268W_tsq121	YBR087W_tsq887	YJL092W
	0.094	0.091	0.088	0.087	0.087	0.087	0.086	0.086	0.085	0.084
	YMR272C	YML128C	YGR028W	YNL156C	YMR083W	YDR228C_tsq686	YLR268W_tsq121	YBR087W_tsq887	YNL231C	YJR099W
	0.094	0.091	0.088	0.087	0.087	0.087	0.086	0.085	0.082	0.069

Table B.5: Target genes for $D = 82$ test chemical compounds (continuation); see description in Table 3.1.

Appendix C

Nonparametric basis pursuit

C.0.5 Proofs of Properties P1-P3

Proof: 1) If white noise $n(x) : x \in \mathbb{R}$ is fed to an ideal low-pass filter with cutoff frequency $\omega_{\max} = \pi$, then $r(\xi) := E(z(x)z(x+\xi)) = \text{sinc}(\xi)$ is the autocorrelation of the output $z(x)$. Hence, $\mathbf{K}$ equals the covariance matrix of $\mathbf{z}^T := [z(x_1), \ldots, z(x_N)]$, and as such $\mathbf{K} \succeq \mathbf{0}$.

Proof: 2) Rewrite the kernel $f_{x'}(x) := \text{sinc}(x - x')$ as a function parameterized by x'. Then, the NST applied to the bandlimited $f_{x'}(x)$ yields $f_{x'}(x) = \sum_{n \in \mathbb{Z}} f_{x'}(n)\text{sinc}(x - n) = \sum_{n \in \mathbb{Z}} \phi_n(x')\phi_n(x)$.

Proof: 3) Upon defining $\alpha_n := f(x_n)$, the reconstruction formula $f(x) := \sum_{n \in \mathbb{Z}} f(n)\text{sinc}(x-n)$ gives the kernel expansion of $f \in \mathcal{B}_\pi$. Hence, by definition of the RKHS norm $\|f\|^2_{\mathcal{H}_\mathcal{X}} = \sum_{n \in \mathbb{Z}} \sum_{n' \in \mathbb{Z}} f(n)\text{sinc}(n - n')f(n')$. Substituting the reconstructed $f(n) = \sum_{n' \in \mathbb{Z}} \text{sinc}(n - n')f(n')$ into the last equation yields $\|f\|^2_{\mathcal{H}_\mathcal{X}} = \sum_{n \in \mathbb{Z}} f^2(n)$.

C.0.6 Design of Algorithm 1

In order to rewrite the cost $\frac{1}{2}\|(\mathbf{Z} - \mathbf{C}\mathbf{B}^T) \odot \mathbf{W}\|_F^2 + \frac{\mu}{2}\left[\text{Tr}(\mathbf{C}^T\mathbf{K}_\mathcal{X}^{-1}\mathbf{C}) + \text{Tr}(\mathbf{B}^T\mathbf{K}_\mathcal{Y}^{-1}\mathbf{B})\right]$ in terms $\mathbf{c}_i = \mathbf{C}\mathbf{e}_i$ and $\mathbf{b}_i = \mathbf{B}\mathbf{e}_i$, representing the i-th columns of matrix $\mathbf{B}$ and $\mathbf{C}$, respectively, define $\bar{\mathbf{C}}_i = \mathbf{C} - \mathbf{c}_i{\mathbf{e}_i}^T$ and decompose $\mathbf{C}\mathbf{B}^T = \bar{\mathbf{C}}_i\mathbf{B}^T + \mathbf{c}_i{\mathbf{b}_i}^T$. Then rewrite the cost as

$$\frac{1}{2}\|(\mathbf{Z}_i - \mathbf{c}_i{\mathbf{b}_i}^T) \odot \mathbf{W}\|_F^2 + \frac{\mu}{2}{\mathbf{c}_i}^T\mathbf{K}_\mathcal{X}^{-1}\mathbf{c}_i \tag{C.1}$$

after defining $\mathbf{Z}_i := \mathbf{Z} - \bar{\mathbf{C}}_i\mathbf{B}^T$ and discarding regularization terms not depending on $\mathbf{c}_i$.

Let vec$(\mathbf{W})$ denote the vector operator that concatenates columns of $\mathbf{W}$, and $\mathbf{D} :=$ Diag$[\mathbf{x}]$ the diagonal matrix operator such that $d_{ii} = x_i$. The Hadamard product can be bypassed by defining

$\mathbf{D}_W := \text{Diag}[\text{vec}(\mathbf{W})]$, substituting $\|\mathbf{X}\|_F = \|\text{vec}(\mathbf{X})\|_2$, and using the following identities

$$\begin{aligned} \text{vec}(\mathbf{W} \odot \mathbf{X}) &= \mathbf{D}_W \text{vec}(\mathbf{X}), \\ \text{vec}(\mathbf{X}_i {\mathbf{b}_i}^T) &= (\mathbf{b}_i \otimes \mathbf{I}_M)\text{vec}(\mathbf{X}_i) \end{aligned} \tag{C.2}$$

with $\otimes$ representing the Kroneker product. Applying (C.2) to (C.1) yields

$$\frac{1}{2}\|\mathbf{D}_W \text{vec}(\mathbf{Z}_i) - \mathbf{D}_W(\mathbf{b}_i \otimes \mathbf{I}_M)\mathbf{c}_i\|_2^2 + \frac{\mu}{2}{\mathbf{c}_i}^T \mathbf{K}_{\mathcal{X}}^{-1}\mathbf{c}_i \tag{C.3}$$

Equating the gradient of (C.3) w.r.t. $\mathbf{c}_i$ to zero, and solving for $\mathbf{c}_i$ it results

$$\begin{aligned} \mathbf{c}_i &= \mathbf{H}_i^{-1}({\mathbf{b}_i}^T \otimes \mathbf{I}_M)\mathbf{D}_W \text{vec}(\mathbf{Z}_i) \\ \mathbf{H}_i &:= {\mathbf{b}_i}^T \otimes \mathbf{I}_M)\mathbf{D}_W \mathbf{D}_W({\mathbf{b}_i}^T \otimes \mathbf{I}_M) + \mu \mathbf{K}_{\mathcal{X}}^{-1} \end{aligned} \tag{C.4}$$

It follows from (C.2) that $({\mathbf{b}_i}^T \otimes \mathbf{I}_M)\mathbf{D}_W \text{vec}(\mathbf{Z}_i) = (\mathbf{W} \odot \mathbf{Z}_i)$, and it can be established by inspection that $({\mathbf{b}_i}^T \otimes \mathbf{I}_M)\mathbf{D}_W \mathbf{D}_W({\mathbf{b}_i}^T \otimes \mathbf{I}_M) = \sum_{n=1}^{N} b_{in}^2 \text{Diag}[\mathbf{w}_n] = \text{Diag}\left[\mathbf{W}(\mathbf{b}_i \odot \mathbf{b}_i)\right]$, so that (C.4) reduces to $\mathbf{c}_i = \left(\text{Diag}\left[\mathbf{W}(\mathbf{b}_i \odot \mathbf{b}_i)\right] + \mu \mathbf{K}_{\mathcal{X}}^{-1}\right)^{-1}(\mathbf{W} \odot \mathbf{Z}_i)\mathbf{b}_i$, coinciding with the update for $\mathbf{c}_i$ in Algorithm 1. The corresponding update for $\mathbf{b}_i$ follows from parallel derivations.

C.0.7 Parallelizable Algorithms for the Selection of Grouped Variables

Consider the classical problem of linear regression, where a vector $\mathbf{y} \in \mathbb{R}^n$ of observations is given along with a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of inputs. Suppose the p features are split into N_f disjoint *factors* (groups of features) such that the coefficient vector is $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_{N_f}']' \in \mathbb{R}^p$, where $'$ denotes transposition and $\boldsymbol{\zeta}_f$ corresponds to the coefficients of factor f. The *group* least-absolute shrinkage and selection operator (Lasso) [190] is a model selection and estimation technique used to select relevant factors in linear regression, and yields

$$\hat{\boldsymbol{\zeta}}_{\text{glasso}} := \arg\min_{\boldsymbol{\zeta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\zeta}\|_2^2 + \mu \sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|_2 \tag{C.5}$$

where $\mu \geq 0$ is a tuning parameter typically chosen via model selection techniques such as cross-validation (CV); see e.g., [88, 190]. If $\mu = 0$, no sparsity is enforced since (C.5) reduces to LS. As μ increases, more sub-vector estimates $\boldsymbol{\zeta}_f$ become zero due to the effect of the group sparsity-encouraging penalty, and the corresponding factors drop out of the model. When $N_f = p$, (C.5) becomes the Lasso [178] that performs variable – rather than factor – selection.

Finding $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$ requires solving (iteratively) for any given value of μ a second-order cone program (SOCP). While standard SOCP solvers can be invoked to this end, an increasing amount of effort has been put recently into developing fast algorithms that capitalize on the unique properties of the group-Lasso; see e.g. [190], [144], [82], [70], [194].

Typically, the training set is assumed to be centrally available, so that it can be jointly processed to obtain $\hat{\zeta}_{\text{glasso}}$. However, collecting all data in a central location may be prohibitive in timely applications of interest. In-network-based (group-)Lasso estimators find application in e.g., robust layered sensing [101], and in the sensing task of cognitive radio networks [17, 18]. In other cases such as the Internet or collaborative inter-laboratory studies, agents providing private data for the purpose of e.g., fitting a sparse model, may not be willing to share their training data but only the learning results. Distributed subgradient methods are applicable to sparse linear regression [133] as well, but are typically slow.

Having this context in mind, the present chapter develops a consensus-based distributed algorithm for the group-Lasso, which can be specialized for the Lasso as well. Problem (C.5) is recast as a convex *constrained* minimization in Section C.0.8, and is iteratively optimized using the alternating-direction method of multipliers (AD-MoM) [25, p. 253]. This way, provably convergent parallel recursions are derived to update each agent's local estimate, that entail simple vector soft-thresholding operations (Section C.0.9). This is possible by capitalizing on the closed form solution that (C.5) admits in the orthonormal case [194], [144], and evidences the factor-level sparsity encouraging property of group-Lasso. On a per iteration basis, agents only exchange their current local estimate with their neighbors. By specializing to a dummy single agent network, a novel centralized group-Lasso solver is obtained in Section C.0.10 as a byproduct. Different from [190] and [144], the algorithm here can handle non orthonormal matrix $\mathbf{X}$, and does not require an inner Newton-Raphson recursion per iteration. By comparing the centralized algorithm with its distributed counterparts of Section C.0.9, it is shown that the latter effectively split the computational burden across agents.

C.0.8 Problem Statement and Preliminaries

Consider J networked agents that are capable of performing some local computations, as well as exchanging messages among neighbors. An agent should be understood as an abstract entity, possibly representing a sensor node in a WSN, a router monitoring Internet traffic, a hospital, insurance company or laboratory involved in e.g., a medical study; or a sensing radio from a next-generation mobile communications technology. The network is naturally modeled as an undirected graph $\mathcal{G}(\mathcal{J}, \mathcal{E})$, where the vertex set $\mathcal{J} := \{1, \ldots, J\}$ corresponds to the agents, and the edges in $\mathcal{E}$ represent pairs of agents that can communicate. Agent $j \in \mathcal{J}$ communicates with its single-hop neighboring agents in $\mathcal{N}_j$, and the size of the neighborhood is denoted by $|\mathcal{N}_j|$. The graph $\mathcal{G}$ is assumed connected, i.e., there exists a (possibly multihop) path that joins any pair of agents in the network.

For the purpose of estimating an unknown vector $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_{N_f}']' \in \mathbb{R}^p$, each agent $j \in \mathcal{J}$ has available a local vector of observations $\mathbf{y}_j \in \mathbb{R}^{n_j}$ as well as its own matrix of inputs $\mathbf{X}_j \in \mathbb{R}^{n_j \times p}$. Agents collaborate to form the wanted group-Lasso estimator (C.5) in a distributed fashion, which can be rewritten as

$$\hat{\boldsymbol{\zeta}}_{\text{glasso}} := \arg \min_{\boldsymbol{\zeta}} \frac{1}{2}\sum_{j=1}^{J} \|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\zeta}\|_2^2 + \mu \sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|_2 \tag{C.6}$$

with $\mathbf{y} := [\mathbf{y}_1', \ldots, \mathbf{y}_J']' \in \mathbb{R}^n$ with $n := \sum_{j=1}^{J} n_j$, and $\mathbf{X} := [\mathbf{X}_1', \ldots, \mathbf{X}_J']' \in \mathbb{R}^{n \times p}$. In lieu of a central controller, the goal of this chapter is to develop a distributed solver of (C.6) based on in-network processing of the local training sets $\{\mathbf{y}_j, \mathbf{X}_j\}_{j \in \mathcal{J}}$. On a per iteration basis the algorithm should comprise: (i) a communication step where agents exchange messages with their neighbors; and (ii) a simple update step where each agent uses this information to refine its local estimate. An additional desirable property is that the collection of local estimates should eventually *consent* to the global solution $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$, namely, the estimate that would be obtained if the entire training data set were centrally available.

Consensus-based reformulation of group-Lasso

To distribute the cost in (C.6), replace the *global* variable $\boldsymbol{\zeta}$ which couples the per-agent summands, with *local* variables $\{\boldsymbol{\zeta}_j \in \mathbb{R}^p\}_{j=1}^{J}$ representing candidate estimates of $\boldsymbol{\zeta}$ per agent. It is now possible to reformulate (C.6) as a convex *constrained* minimization problem

$$\begin{aligned} \left\{\hat{\boldsymbol{\zeta}}_j\right\}_{j=1}^{J} := \arg \min_{\{\boldsymbol{\zeta}_j\}} \frac{1}{2}\sum_{j=1}^{J} \left[\|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\zeta}_j\|_2^2 + \frac{2\mu}{J}\sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_{jf}\|_2\right] \\ \text{s. t.} \qquad \boldsymbol{\zeta}_j = \boldsymbol{\zeta}_{j'},\ j \in \mathcal{J},\ j' \in \mathcal{N}_j \end{aligned} \tag{C.7}$$

where $\boldsymbol{\zeta}_j = \left[\boldsymbol{\zeta}_{j1}', \ldots, \boldsymbol{\zeta}_{jN_f}'\right]'$, $j \in \mathcal{J}$. The equality constraints directly effect local agreement between neighboring CRs. Since the communication graph $\mathcal{G}$ is assumed connected, these constraints also ensure *global* consensus a fortiori, meaning that $\boldsymbol{\zeta}_j = \boldsymbol{\zeta}_{j'}, \forall j, j' \in \mathcal{J}$. As a direct consequence of this observation, it follows that problems (C.6) and (C.7) are equivalent, i.e., $\hat{\boldsymbol{\zeta}}_{\text{glasso}} = \hat{\boldsymbol{\zeta}}_j,\ \forall\, j \in \mathcal{J}$.

Problem (C.7) will be modified further for the purpose of reducing the computational complexity of the resulting algorithm. To this end, for a given $\mathbf{a} \in \mathbb{R}^p$ consider the problem

$$\min_{\boldsymbol{\zeta}} \left[\frac{1}{2}||\boldsymbol{\zeta}||_2^2 - \mathbf{a}'\boldsymbol{\zeta} + \mu\sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|_2\right], \quad \boldsymbol{\zeta} := [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_{N_f}']' \tag{C.8}$$

and notice that it is separable in the N_f subproblems

$$\min_{\boldsymbol{\zeta}_f} \left[\frac{1}{2}||\boldsymbol{\zeta}_f||_2^2 - \mathbf{a}_f'\boldsymbol{\zeta}_f + \mu\|\boldsymbol{\zeta}_f\|_2\right], \quad \mathbf{a} := [\mathbf{a}_1', \ldots, \mathbf{a}_{N_f}']'. \tag{C.9}$$

Interestingly, each of these subproblems admits a closed-form solution as given in the following lemma.

Lemma C.1 *The minimizer* $\boldsymbol{\zeta}_f^\star$ *of (C.9) is obtained via the vector soft-thresholding operator* $\mathcal{T}_\mu(\cdot)$ *defined by*

$$\boldsymbol{\zeta}_f^\star = \mathcal{T}_\mu(\mathbf{a}_f) := (\|\mathbf{a}_f\|_2 - \mu)_+ \frac{\mathbf{a}_f}{\|\mathbf{a}_f\|_2} \tag{C.10}$$

where $(\cdot)_+ := \max\{\cdot, 0\}$ *.*

Problem (C.8) is an instance of group-Lasso (C.5) when $\mathbf{X}'\mathbf{X} = \mathbf{I}_p$, and $\mathbf{a} := \mathbf{X}'\mathbf{y}$. As such, result (C.10) can be viewed as a particular case of the operators in [144] and [194]. However it is worth to prove Lemma C.1 directly, since in this case the special form of (C.9) renders the proof neat in its simplicity.

Proof: It will be argued that the solver of (C.9) takes the form $\boldsymbol{\zeta}_f^\star = t\mathbf{a}_f$ for some scalar $t \geq 0$. This is because among all $\boldsymbol{\zeta}_f$ with the same ℓ_2-norm, the Cauchy-Schwarz inequality implies that the maximizer of $\mathbf{a}_f'\boldsymbol{\zeta}_f$ is colinear with (and in the same direction of) $\mathbf{a}_f$. Substituting $\boldsymbol{\zeta}_f = t\mathbf{a}_f$ into (C.9) renders the problem scalar in t, with solution $t^\star = (\|\mathbf{a}_f\| - \mu)_+ / (\|\mathbf{a}_f\|)$ completing the proof. □

In order to take advantage of the result in Lemma C.1, auxiliary variables $\boldsymbol{\gamma}_j,\ j \in \mathcal{J}$ are introduced as copies of $\boldsymbol{\zeta}_j$. Upon introducing appropriate constraints $\boldsymbol{\gamma}_j = \boldsymbol{\zeta}_j$ that guarantee the equivalence of the formulations, problem (C.7) can be recast as

$$\begin{aligned} \min_{\{\boldsymbol{\zeta}_j, \boldsymbol{\gamma}_j, \boldsymbol{\gamma}_j^{j'}\}} \ & \frac{1}{2}\sum_{j=1}^{J}\left[\|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\gamma}_j\|_2^2 + \frac{2\mu}{J}\sum_{f=1}^{N_f}\|\boldsymbol{\zeta}_{jf}\|_2\right] \\ \text{s. t.} \quad & \boldsymbol{\zeta}_j = \boldsymbol{\gamma}_j^{j'} = \boldsymbol{\zeta}_{j'},\ j \in \mathcal{J},\ j' \in \mathcal{N}_j \\ & \boldsymbol{\gamma}_j = \boldsymbol{\zeta}_j,\ j \in \mathcal{J}. \end{aligned} \tag{C.11}$$

The additional set of dummy variables $\{\boldsymbol{\gamma}_j^{j'}\}$ is inserted for technical reasons that will become apparent in the ensuing section, and will be eventually eliminated.

C.0.9 Distributed Group-Lasso Algorithm

The distributed group-Lasso algorithm is constructed by optimizing (C.11) using the alternating direction method of multipliers (AD-MoM) [25]. In this direction, associate Lagrange multipliers $\mathbf{v}_j, \bar{\mathbf{v}}_j^{j'}$ and $\check{\mathbf{v}}_j^{j'}$ with the constraints $\boldsymbol{\gamma}_j = \boldsymbol{\zeta}_j$, $\boldsymbol{\zeta}_j = \boldsymbol{\gamma}_j^{j'}$ and $\boldsymbol{\zeta}_{j'} = \boldsymbol{\gamma}_j^{j'}$ respectively, and consider the augmented Lagrangian with parameter $c > 0$

$$
\begin{aligned}
\mathcal{L}_c\left[\{\boldsymbol{\zeta}_r\}, \boldsymbol{\gamma}, \boldsymbol{v}\right] =& \frac{1}{2}\sum_{j=1}^{J}\left[\|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\gamma}_r\|_2^2 + \frac{2\mu}{J}\sum_{f=1}^{N_f}\|\boldsymbol{\zeta}_{jf}\|_2\right] \\
&+ \sum_{j=1}^{J}\left[\mathbf{v}_j'(\boldsymbol{\zeta}_j - \boldsymbol{\gamma}_j) + \frac{c}{2}\|\boldsymbol{\zeta}_j - \boldsymbol{\gamma}_j\|_2^2\right] \\
&+ \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left[(\check{\mathbf{v}}_j^{j'})'(\boldsymbol{\zeta}_j - \boldsymbol{\gamma}_j^{j'}) + \frac{c}{2}\|\boldsymbol{\zeta}_j - \boldsymbol{\gamma}_j^{j'}\|_2^2\right] \\
&+ \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left[(\bar{\mathbf{v}}_j^{j'})'(\boldsymbol{\zeta}_{j'} - \boldsymbol{\gamma}_j^{j'}) + \frac{c}{2}\|\boldsymbol{\zeta}_{j'} - \boldsymbol{\gamma}_j^{j'}\|_2^2\right]
\end{aligned}
\tag{C.12}
$$

where variables are grouped as $\boldsymbol{\gamma} := \{\boldsymbol{\gamma}_j, \{\boldsymbol{\gamma}_j^{j'}\}_{j'\in\mathcal{N}_j}\}_{j\in\mathcal{J}}$ and multipliers $\boldsymbol{v} := \{\mathbf{v}_j, \{\check{\mathbf{v}}_j^{j'}, \bar{\mathbf{v}}_j^{j'}\}_{j'\in\mathcal{N}_j}\}_{j\in\mathcal{J}}$.

Application of the AD-MoM to the problem at hand consists of a cycle of $\mathcal{L}_c$ minimizations in block-coordinate descent fashion w.r.t. $\{\boldsymbol{\zeta}_j\}$ firstly, and $\boldsymbol{\gamma}$ secondly, together with an update of the multipliers per iteration $k = 0, 1, 2, \ldots$. Omitting the details that can be found in [21, Appendix D], the four main properties of this procedure that are instrumental to the resulting algorithm can be highlighted as:

[P1] Thanks to the introduction of the local copies $\boldsymbol{\zeta}_j$ and the dummy variables $\boldsymbol{\gamma}_j^{j'}$, the minimizations of $\mathcal{L}_c$ w.r.t. both $\{\boldsymbol{\zeta}_j\}$ and $\boldsymbol{\gamma}$ decouple per agent j, thus enabling distribution of the algorithm. Moreover, the constraints in (C.11) involve variables of neighboring agents only, which allows the required communications to be local within each agent's neighborhood.

[P2] Introduction of the variables $\boldsymbol{\gamma}_j$ separates the quadratic cost $\|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\gamma}_j\|_2^2$ from the group-Lasso penalty $\sum_{f=1}^{N_f}\|\boldsymbol{\zeta}_{jf}\|_2$. As a result, minimization of (C.12) w.r.t. $\boldsymbol{\zeta}_j$ takes the form of (C.8), which admits a closed-form solution via the vector soft-thresholding operator $\mathcal{T}_\mu(\cdot)$ in (C.10).

[P3] Minimization of (C.12) w.r.t. $\boldsymbol{\gamma}$ consists of an unconstrained quadratic problem, which can also be solved in closed form. In particular, the optimal $\boldsymbol{\gamma}_j^{j'}$ at iteration k takes the value $\boldsymbol{\gamma}_j^{j'}(k) = \left(\boldsymbol{\zeta}_j(k) + \boldsymbol{\zeta}_{j'}(k)\right)/2$, and thus can be eliminated.

[P4] It turns out that it is not necessary to carry out updates of the Lagrange multipliers $\{\bar{\mathbf{v}}_j^{j'}, \check{\mathbf{v}}_j^{j'}\}_{j'\in\mathcal{N}_j}$ separately, but only of their sums which are henceforth denoted by $\mathbf{p}_j := \sum_{j'\in\mathcal{N}_j}(\bar{\mathbf{v}}_j^{j'} + \check{\mathbf{v}}_j^{j'})$. Hence, there is one price $\mathbf{p}_j$ per agent $j = 1, \ldots, J$, which can be updated locally.

Algorithm 5 : DGLasso

```
Initialize to zero $\{\boldsymbol{\zeta}_j(0), \boldsymbol{\gamma}_j(0), \mathbf{p}_j(-1), \mathbf{v}_j(-1)\}_{j\in\mathcal{J}}$, and locally run:
for $k = 0, 1, \ldots$ do
    Transmit $\boldsymbol{\zeta}_j(k)$ to neighbors in $\mathcal{N}_j$.
    Update $\mathbf{p}_j(k)$ using (C.13).
    Update $\mathbf{v}_j(k)$ using (C.14).
    Update $\boldsymbol{\zeta}_j(k+1)$ using (C.15).
    Update $\boldsymbol{\gamma}_j(k+1)$ using (C.16).
end for
```

Building on these four features, it is established in [21, Appendix D] that the proposed ADMoM scheme boils down to four parallel recursions run locally per agent, where $f = 1, \ldots, N_f$ in (C.15) and $\mathbf{M}_j := c\mathbf{I}_p + \mathbf{X}_j'\mathbf{X}_j$ in (C.16)

$$\mathbf{p}_j(k) = \mathbf{p}_j(k-1) + c \sum_{j' \in \mathcal{N}_j} [\boldsymbol{\zeta}_j(k) - \boldsymbol{\zeta}_{j'}(k)] \tag{C.13}$$

$$\mathbf{v}_j(k) = \mathbf{v}_j(k-1) + c[\boldsymbol{\zeta}_j(k) - \boldsymbol{\gamma}_j(k)] \tag{C.14}$$

$$\begin{aligned}\boldsymbol{\zeta}_{jf}(k+1) = \mathcal{T}_{\mu/J} \Bigg(& c\boldsymbol{\gamma}_{jf}(k) - \mathbf{p}_{jf}(k) - \mathbf{v}_{jf}(k) \\ & + c \sum_{j' \in \mathcal{N}_j} \left[\boldsymbol{\zeta}_{jf}(k) + \boldsymbol{\zeta}_{j'f}(k)\right] \Bigg) / [c(2|\mathcal{N}_j| + 1)],\end{aligned} \tag{C.15}$$

$$\boldsymbol{\gamma}_j(k+1) = \mathbf{M}_j^{-1} \left(\mathbf{X}_j'\mathbf{y}_j + c\boldsymbol{\zeta}_j(k+1) + \mathbf{v}_j(k)\right). \tag{C.16}$$

Recursions (C.13)-(C.16) comprise the novel DGLasso algorithm, tabulated as Algorithm 5.

The algorithm entails the following steps. During iteration $k+1$, agent j receives the local estimates $\{\boldsymbol{\zeta}_{j'}(k)\}_{j' \in \mathcal{N}_j}$ from the neighboring agents and plugs them into (C.13) to evaluate the dual price vector $\mathbf{p}_j(k)$. The new multiplier $\mathbf{v}_j(k)$ is then obtained using the locally available vectors $\{\boldsymbol{\gamma}_j(k), \boldsymbol{\zeta}_j(k)\}$. Subsequently, vectors $\{\mathbf{p}_j(k), \mathbf{v}_j(k)\}$ are jointly used along with $\{\boldsymbol{\zeta}_{j'}(k)\}_{j' \in \mathcal{N}_j}$ to obtain $\boldsymbol{\zeta}_j(k+1)$ via N_f parallel vector soft-thresholding operations $\mathcal{T}_{\mu/J}(\cdot)$ defined in (C.10). Finally, the updated $\boldsymbol{\gamma}_j(k+1)$ is obtained from (C.16), and requires the previously updated quantities along with the vector of local observations $\mathbf{y}_j$ and regression matrix $\mathbf{X}_j$. The $(k+1)$st iteration is concluded after agent j broadcasts $\boldsymbol{\zeta}_j(k+1)$ to its neighbors. The distributed $K-$fold CV protocol in [122] can be utilized to tune μ.

DGLasso algorithm does not require nested iterations, since all local updates are given in closed form. Even if an arbitrary initialization is allowed, the sparse nature of the estimator sought suggest the all-zero vectors as a natural choice. With regards to communication cost, only the p scalars in

$\boldsymbol{\zeta}_j$ have to be broadcasted per iteration. When p is large, major savings can be attained by only exchanging the set of nonzero entries. Further, the inter-agent communication cost does not depend on the size of the local training sets. A computational cost analysis will be deferred to the ensuing section.

Remark C.1 *(Reduction to distributed Lasso algorithm)* When $N_f = p$ and there are as many groups as entries of $\boldsymbol{\zeta}$, then the sum $\sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|$ becomes the ℓ_1-norm of $\boldsymbol{\zeta}$, and group-Lasso reduces to Lasso. In this case, DGLasso offers a distributed algorithm to solve Lasso that coincides with the one in [122].

To close this section, it is useful to mention that convergence of Algorithm 5 is ensured by the convergence of the AD-MoM [25]. This result is formally stated next.

Proposition C.1 *Let $\mathcal{G}$ be a connected graph, and consider recursions (C.13)-(C.16) that comprise the DGLasso algorithm. Then, for any value of the step-size $c > 0$, the iterates $\boldsymbol{\zeta}_j(k)$ converge to the group-Lasso solution [cf. (C.6)] as $k \to \infty$, i.e.,*

$$\lim_{k\to\infty} \boldsymbol{\zeta}_j(k) = \hat{\boldsymbol{\zeta}}_{glasso},\ \forall\, j \in \mathcal{J}. \tag{C.17}$$

In words, all local estimates $\boldsymbol{\zeta}_j(k)$ achieve consensus asymptotically, converging to a common vector that coincides with the desired estimator $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$. Formally, if the number of parameters p exceeds the number of data n, then a unique solution of (C.5) is not guaranteed for a general design matrix $\mathbf{X}$. Proposition C.1 remains valid however, if the right-hand side of (C.17) is replaced by the set of minima; i.e., $\lim_{k\to\infty} \boldsymbol{\zeta}_j(k) \in \text{arg } \min_{\boldsymbol{\zeta}} \frac{1}{J}\sum_{j=1}^{J} \|\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\zeta}\|_2^2 + \mu \sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|_2$.

From (C.17), all asymptotic (as n grows large) properties of centralized (group-)Lasso carry over to its distributed counterpart developed here. Those include not only the bias, but also weak support consistency as well as estimation consistency, which for the centralized (group-)Lasso have been studied in e.g., [192,201]. One can for instance borrow the weighted versions of the sparsifying penalties in [201] and [192], with weights provided by the (distributed) LS estimates in order to ensure the estimators enjoy (asymptotically) the aforementioned *oracle properties*.

C.0.10 Parallel Processing

The algorithmic framework developed so far for distributed sparse estimation, can also be applied to obtain efficient *centralized* (group-)Lasso solvers as special cases of Algorithm 5. These will be briefly described next, since they are important on their own right as standalone sparse linear regression tools. Moreover, they will serve as a baseline for comparison with the distributed algorithms of Section C.0.9, for the purpose of establishing that DGLasso has the property of parallelizing computations in multiprocessor architectures.

Algorithm 6 : GLasso

```
Initialize to zero $\{\boldsymbol{\zeta}(0), \boldsymbol{\gamma}(0), \mathbf{v}(-1)\}$, and run:
**for** $k = 0, 1,\ldots$ **do**
    Update $\mathbf{v}(k) = \mathbf{v}(k-1) + c[\boldsymbol{\zeta}(k) - \boldsymbol{\gamma}(k)]$.
    Update $\boldsymbol{\zeta}_f(k+1) = (1/c)\mathcal{T}_\mu\left(c\boldsymbol{\gamma}_f(k) - \mathbf{v}_f(k)\right), \; \forall\, f$.
    Update $\boldsymbol{\gamma}(k+1) = \mathbf{M}^{-1}\left(\mathbf{X}'\mathbf{y} + c\boldsymbol{\zeta}(k+1) + \mathbf{v}(k)\right)$.
**end for**
```

Recalling the network setup described in Section C.0.8, let $J = 1$ so that the network collapses to a single agent, and suppose that this central processing unit has available the training data set $\{\mathbf{y}, \mathbf{X}\}$, say. In this case DGLasso yields a novel algorithm for the standard (centralized) group-Lasso estimator (C.6), termed GLasso and tabulated as Algorithm 6. To arrive at this result, start from the DGLasso recursions (C.13)-(C.16) and note that: (i) index j can be dropped since there is a single agent; (ii) summations across neighborhoods disappear for the same reason; and (iii) $\mathbf{p}(k) = 0, \forall\, k$ since (C.13) simplifies to $\mathbf{p}(k) = \mathbf{p}(k-1)$ and $\mathbf{p}(-1) = 0$. Because there are no consensus constraints to be enforced, it is reasonable that $\mathbf{p}(k)$ is no longer needed. Alternatively, one can directly arrive at Algorithm 6 after applying AD-MoM iterations to solve the problem

$$\min_{\{\boldsymbol{\zeta}, \boldsymbol{\gamma}\}} \left[\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\gamma}\|_2^2 + \mu \sum_{f=1}^{N_f} \|\boldsymbol{\zeta}_f\|_2\right], \quad \text{s. t. } \boldsymbol{\gamma} = \boldsymbol{\zeta} \tag{C.18}$$

which is equivalent to (C.5). The sequence of iterates $\boldsymbol{\zeta}(k)$ generated by Algorithm 6 is thus provably convergent to $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$ as $k \to \infty$, for any value of $c > 0$ [25].

Notice that the thresholding operator $\mathcal{T}_\mu$ in GLasso sets the entire sub-vector $\boldsymbol{\zeta}_f(k+1)$ to zero whenever $\|c\boldsymbol{\gamma}_f(k) - \mathbf{v}_f(k)\|_2$ does not exceed μ, in par with the group sparsifying property of group-Lasso. After the thresholding, a proportional shrinkage typical of ridge (ℓ_2-penalized) estimators is performed [88]. In this case however, the shrinkage by a factor of c is due to quadratic term in the augmented Lagrangian. Not surprisingly, thresholding/proportional shrinkage type of updates have been also obtained in cyclic coordinate descent (CD) algorithms for the elastic net [71]. Different from [190], GLasso can handle a general (not orthonormal) regression matrix $\mathbf{X}$. Compared to the block-CD method proposed in [144], GLasso does not require an inner Newton-Raphson recursion per iteration.

As discussed in Remark C.1, if $N_f = p$ so that the factors coincide with the scalar entries of $\boldsymbol{\zeta}$, then GLasso yields the Lasso estimator. In particular, the vector soft-thresholding operator simplifies to its well-known scalar counterpart $\mathcal{S}_\mu(z) := (|z| - \mu)_+\text{sign}(z)$. The Lasso estimator is expressible in terms of $\mathcal{S}_\mu$ whenever the problem is orthonormal or scalar, and thus $\mathcal{S}_\mu$ typically characterizes the updates of cyclic CD solvers for Lasso [88, p. 93]. When specialized to Lasso, Algorithm 6 coincides with the split Bregman method in [82], provided a single iteration is carried out in the minimization of a suitable “energy” introduced in [82, eq. (1.1)]. Such inexact minimization heuristic is suggested in [82] for algorithmic efficiency reasons, without recognizing its relation to AD-MoM and lacking formal convergence guarantees. This connection between AD-MoM and the split Bregman method was also pointed out in [66]. To estimate hierarchical sparse models or reconstruct signals based on incomplete Fourier data, related ideas based on cost decoupling to capitalize on alternating minimization methods were applied in [170] and [188].

Computational load balancing

Consider the DGLasso recursions (C.13)-(C.16). Update (C.16) involves inversion of the $p \times p$ matrix $\mathbf{M}_j := c\mathbf{I}_p + \mathbf{X}_j'\mathbf{X}_j$ per agent, that may be computationally demanding for sufficiently large p. Fortunately, this operation as well as the evaluation of the local “ridge” estimate $[c\mathbf{I}_p + \mathbf{X}_j'\mathbf{X}_j]^{-1}\mathbf{X}_j'\mathbf{y}_j$ can be carried out offline before running the algorithm. Other than that, the updates comprising DGLasso are simple and solely involve scaling/addition and thresholding of (eventually sparse) p-dimensional vectors. As pointed out in [82], in several applications of interest there is specific structure that can be exploited to efficiently invert the aforementioned matrix. Circulant structure has been shown to arise in compressive sampling for magnetic resonance imaging [82], hence the inversion can be carried out through a suitable DFT. Sparsity is another characteristic of the matrix that can be capitalized upon in, e.g., multiple frequency-hopping signal estimation [11].

In any case, the matrix inversion lemma can be invoked to obtain

$$\mathbf{M}_j^{-1} = (1/c)\left[\mathbf{I}_p - \mathbf{X}_j'\left(c\mathbf{I}_{n_j} + \mathbf{X}_j\mathbf{X}_j'\right)^{-1}\mathbf{X}_j\right].$$

In this new form, the dimensionality of the matrix to invert becomes $n_j \times n_j$, where n_j is the number of locally acquired data. For highly underdetermined regression problems $(n_j \ll p)$ typically arising in genomics or computational biology [88, Ch. 18], (D)GLasso enjoys considerable computational savings through the aforementioned matrix inversion identity. More importantly, one also recognizes that the distributed operation parallelizes the numerical computation across agents: if GLasso is run centrally with all network-wide data $\mathbf{y} := [\mathbf{y}_1', \ldots, \mathbf{y}_J']'$ and $\mathbf{X} := [\mathbf{X}_1', \ldots, \mathbf{X}_J']'$ at hand, then the matrix to invert has dimension $n = \sum_{j \in \mathcal{J}} n_j$, which increases linearly with the network size J. Beyond a networked scenario as described in Section C.0.8, DGLasso provides an attractive alternative for computational load balancing in timely multi-processor architectures.

C.0.11 Numerical example

A simulated test is now presented to corroborate the convergence of DGLasso. The example will rely on the birthweight dataset considered in the seminal group-Lasso work of [190]. The objective is to predict the human birthweight from $N_f = 8$ factors including the mother’s `age`, `weight`, `race`, `smoke` habits, number of previous `premature` labors, history of `hypertension`, uterine `irritability`, and number of physician `visits` during the first trimester of pregnancy. Third-order polynomials were considered to model nonlinear effects of the `age` and `weight` on the response, augmenting the model size to $p = 12$ by grouping the polynomial coefficients in two subsets of three variables. The network of $J = 10$ agents is simulated as a random geometric graph on $[0, 1]^2$, with communication range $r = 0.4$. The $n = 189$ data samples are randomly split across agents, so that $n_j = 18$ for $j \in [1, 9]$, and $n_{10} = 27$.

By running Algorithm 5 and using "warm starts" [71], the path of group-Lasso solutions is computed at 100 different values of the regularization parameter μ. The penalty coefficient is set to $c = 8$, since several experiments suggested this value leads to fastest convergence. Fig. C.1 (top) shows the regularization path for agent $j = 2$, where for diminishing values of μ more factors enter the model. The dashed vertical line indicates the model for $\mu_{\text{CV}} = 8.513$, obtained via the 10-fold distributed CV procedure in [122]. Consensus is achieved after few iterations, as observed from Fig. C.1 (bottom) which depicts the evolution of the factors' strength measured by $\|\boldsymbol{\zeta}_{jf}\|_2$, for two representative agents with $j = 2, 7$. DGLasso converges to the same prediction model as in [190], and determines that `visits` is not significant even from the first iterations, allowing for early model selection.

We also compare DGLasso with the distributed subgradient method in [133], for which equal neighbor combining weights, zero initial conditions, and a diminishing stepsize $\alpha(k) = 10^{-2}/k$ are adopted. As figure of merit, the global error metric $\epsilon(k) := J^{-1}\sum_{j=1}^{J}\|\hat{\boldsymbol{\zeta}}_j(k) - \hat{\boldsymbol{\zeta}}_{\text{glasso}}\|_2^2$ is evaluated for all schemes. Algorithm 6 was utilized to obtain $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$, and the resulting errors are depicted in Fig. C.2. The decreasing trend of $\epsilon(k)$ confirms that all local estimates converge to $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$, as stated in Proposition C.1. All-zero initial vectors speed up DGLasso. With regards to the subgradient method, the speed of convergence is extremely slow since a descent along a subgradient direction is not effective in nulling factors of the local estimates.

C.0.12 Concluding Remarks

An in-network processing-based algorithm for fitting a group-Lasso model is developed in this chapter, based on AD-MoM iterations. Apart from an offline matrix inversion, the resulting per agent DGLasso updates are simple and given in closed form. In a nutshell, the DGLasso recursions entail linear combinations of vectors and a soft-thresholding operation. Interestingly, DGLasso has the property of parallelizing computations across agents, and requires affordable communications of sparse messages within the neighborhood. The sequences of local estimates generated by DGLasso are provably convergent to $\hat{\boldsymbol{\zeta}}_{\text{glasso}}$, and the algorithm can outperform alternatives based on distributed subgradient descent.

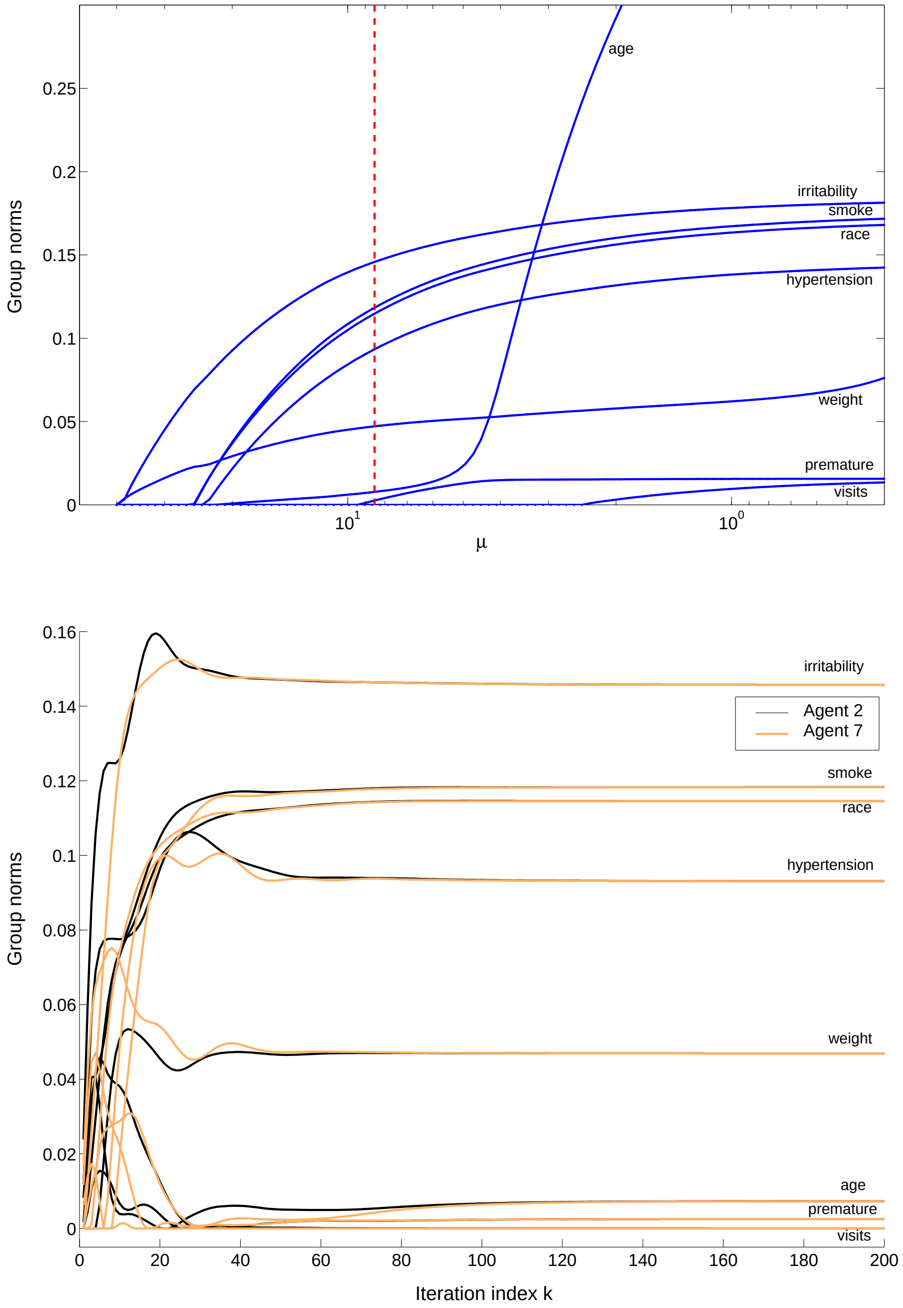


Figure C.1: (top) Group-Lasso regularization path; (bottom) evolution of the per factor norms for agents 2 and 7.

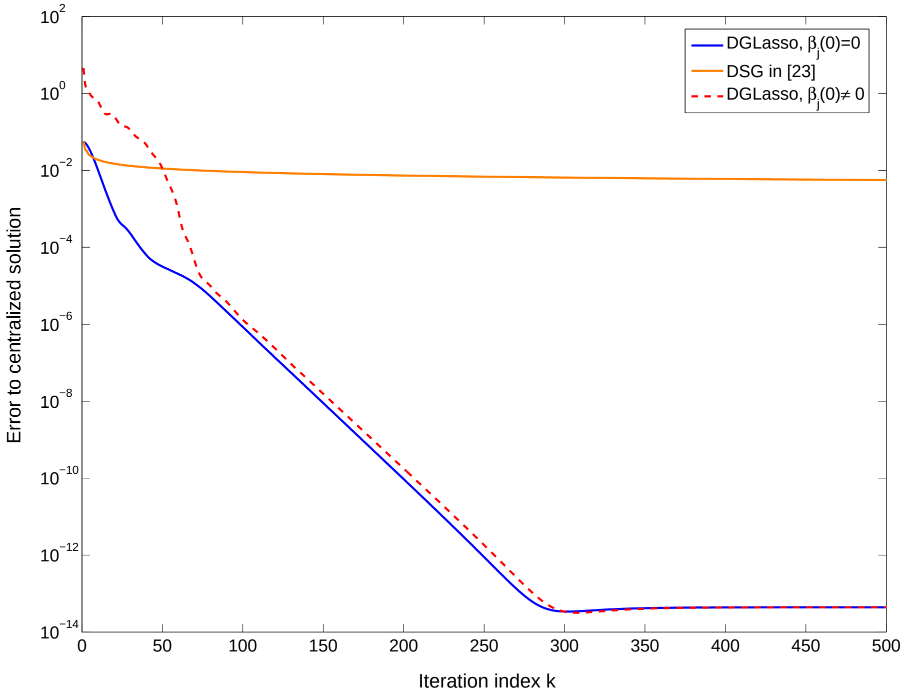


Figure C.2: Global estimation error evolution.

Appendix D

Rank regularization for tensor completion

D.0.13 Proof of Proposition 5.1

The equivalence between (5.2) and (5.4) stated in a) follows immediately from (5.3). Indeed, if (5.4) is minimized in two steps

$$\min_{\mathbf{X}} \left\{ \min_{\substack{\mathbf{B},\mathbf{C} \\ \text{s. to } \mathbf{B}\mathbf{C}^T = \mathbf{X}}} \frac{1}{2}\|(\mathbf{Z}-\mathbf{X})\circledast\mathbf{\Delta}\|_F^2 + \frac{\mu}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \right\} \tag{D.1}$$

it is apparent that the LS part of the cost does not depend on the inner minimization variables. Hence, (D.1) can be rewritten as

$$\min_{\mathbf{X}} \left\{ \frac{1}{2}\|(\mathbf{Z}-\mathbf{X})\circledast\mathbf{\Delta}\|_F^2 + \mu \left[\min_{\substack{\mathbf{B},\mathbf{C} \\ \text{s. to } \mathbf{B}\mathbf{C}^T = \mathbf{X}}} \frac{1}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \right] \right\} \tag{D.2}$$

and by recognizing (5.3) as the problem within the square brackets in (D.2), the equivalence follows.

To establish b), consider the cost in (5.4) at the local minimum $(\bar{\mathbf{B}}, \bar{\mathbf{C}})$

$$U(\bar{\mathbf{B}}, \bar{\mathbf{C}}) := \frac{1}{2}\|(\mathbf{Z}-\bar{\mathbf{X}})\circledast\mathbf{\Delta}\|_F^2 + \frac{\mu}{2}(\|\bar{\mathbf{B}}\|_F^2 + \|\bar{\mathbf{C}}\|_F^2)$$

where $\bar{\mathbf{X}} := \bar{\mathbf{B}}\bar{\mathbf{C}}^T$. Arguing by contradiction, suppose that there is a different local minimum $(\mathbf{B}, \mathbf{C})$ such that $U(\mathbf{B}, \mathbf{C}) \neq U(\bar{\mathbf{B}}, \bar{\mathbf{C}})$. Without loss of generality set $U(\mathbf{B}, \mathbf{C}) < U(\bar{\mathbf{B}}, \bar{\mathbf{C}})$ so that $dU := U(\mathbf{B}, \mathbf{C}) - U(\bar{\mathbf{B}}, \bar{\mathbf{C}}) < 0$, which can be expanded to

$$\begin{aligned} dU &= \operatorname{Tr}\left[\left(\mathbf{\Delta}\circledast(\mathbf{Z}-\bar{\mathbf{X}})\right)\left(\mathbf{\Delta}\circledast(\bar{\mathbf{X}}-\mathbf{X})\right)\right] + \frac{1}{2}\|\mathbf{\Delta}\circledast(\bar{\mathbf{X}}-\mathbf{X})\|_F^2 \\ &+ \frac{\mu}{2}\left(\|\mathbf{B}\|_F^2 - \|\bar{\mathbf{B}}\|_F^2 + \|\mathbf{C}\|_F^2 - \|\bar{\mathbf{C}}\|_F^2\right) < 0. \end{aligned} \tag{D.3}$$

Setting this inequality aside for now, consider the augmented matrix $\mathbf{Q}$ in terms of generic matrices $\mathbf{B}$ and $\mathbf{C}$:

$$\mathbf{Q} := \begin{bmatrix} \mathbf{B} \\ \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{B}^T & \mathbf{C}^T \end{bmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{B}^T & \mathbf{X} \\ \mathbf{X}^T & \mathbf{C}\mathbf{C}^T \end{pmatrix} \tag{D.4}$$

and the corresponding $\bar{\mathbf{Q}}$ defined in terms of $\bar{\mathbf{B}}$ and $\bar{\mathbf{C}}$. For each value of $\theta \in (0,1)$ consider the convex combination

$$\mathbf{Q}_\theta := \bar{\mathbf{Q}} + \theta(\mathbf{Q} - \bar{\mathbf{Q}}). \tag{D.5}$$

As both $\mathbf{Q}$ and $\bar{\mathbf{Q}}$ are positive semi-definite, so is $\mathbf{Q}_\theta$ and by means of the Choleski factorization one obtains

$$\mathbf{Q}_\theta := \begin{bmatrix} \mathbf{B}_\theta \\ \mathbf{C}_\theta \end{bmatrix} \begin{bmatrix} \mathbf{B}'_\theta & \mathbf{C}'_\theta \end{bmatrix} = \begin{pmatrix} \mathbf{B}_\theta\mathbf{B}'_\theta & \mathbf{X}_\theta \\ \mathbf{X}'_\theta & \mathbf{C}_\theta\mathbf{C}'_\theta \end{pmatrix} \tag{D.6}$$

which defines $\mathbf{B}_\theta$, $\mathbf{C}_\theta$ and $\mathbf{X}_\theta$.

Expanding the cost difference dU_θ as in (D.3) results in

$$\begin{aligned} dU_\theta &:= U(\mathbf{B}_\theta, \mathbf{C}_\theta) - U(\bar{\mathbf{B}}, \bar{\mathbf{C}}) \\ &= \mathrm{Tr}\left[\left(\mathbf{\Delta} \circledast (\mathbf{Z} - \bar{\mathbf{X}})\right)\left(\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X}_\theta)\right)\right] \\ &+ \frac{\mu}{2}\left(\|\mathbf{B}_\theta\|_F^2 - \|\bar{\mathbf{B}}\|_F^2 + \|\mathbf{C}_\theta\|_F^2 - \|\bar{\mathbf{C}}\|_F^2\right) + \frac{1}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X}_\theta)\|_F^2. \end{aligned}$$

From the definitions (D.4)-(D.6) it follows that $\bar{\mathbf{X}} - \mathbf{X}_\theta = \theta(\bar{\mathbf{X}} - \mathbf{X})$, $\|\mathbf{B}_\theta\|_F^2 - \|\bar{\mathbf{B}}\|_F^2 = \theta(\|\mathbf{B}\|_F^2 - \|\bar{\mathbf{B}}\|_F^2)$, and $\|\mathbf{C}_\theta\|_F^2 - \|\bar{\mathbf{C}}\|_F^2 = \theta(\|\mathbf{C}\|_F^2 - \|\bar{\mathbf{C}}\|_F^2)$, so that

$$\begin{aligned} dU_\theta &:= \theta\mathrm{Tr}\left[\left(\mathbf{\Delta} \circledast (\mathbf{Z} - \bar{\mathbf{X}})\right)\left(\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\right)\right] \\ &+ \frac{\mu\theta}{2}\left(\|\mathbf{B}\|_F^2 - \|\bar{\mathbf{B}}\|_F^2 + \|\mathbf{C}\|_F^2 - \|\bar{\mathbf{C}}\|_F^2\right) + \frac{\theta^2}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\|_F^2. \end{aligned}$$

Using (D.3), dU_θ can be expressed in terms of dU as

$$dU_\theta := \theta\left(dU - \frac{1}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\|_F^2\right) + \frac{\theta^2}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\|_F^2.$$

Since dU is strictly negative, so is $dU - \frac{1}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\|_F^2$, and hence

$$\lim_{\theta \to 0} \frac{1}{\theta} dU_\theta = \left(dU - \frac{1}{2}\|\mathbf{\Delta} \circledast (\bar{\mathbf{X}} - \mathbf{X})\|_F^2\right) < 0.$$

But then in every neighborhood of $(\bar{\mathbf{B}}, \bar{\mathbf{C}})$ there is a point $(\mathbf{B}_\theta, \mathbf{C}_\theta)$ such that $U(\mathbf{B}_\theta, \mathbf{C}_\theta) < U(\bar{\mathbf{B}}, \bar{\mathbf{C}})$, meaning $(\bar{\mathbf{B}}, \bar{\mathbf{C}})$ cannot be a local minimum. This contradiction implies that $U(\mathbf{B}, \mathbf{C}) = U(\bar{\mathbf{B}}, \bar{\mathbf{C}})$ for any pair of local minima, which completes the proof. ■

D.0.14 Proof of Proposition 5.2

The Frobenius norms squared of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are separable across columns; hence, the penalty in (5.8) can be rewritten as

$$\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2 = \sum_{r=1}^{R} \|\mathbf{a}_r\|^2 + \|\mathbf{b}_r\|^2 + \|\mathbf{c}_r\|^2 = \sum_{r=1}^{R} a_r^2 + b_r^2 + c_r^2 \tag{D.7}$$

where $a_r := \|\mathbf{a}_r\|$, $b_r := \|\mathbf{b}_r\|$, $c_r := \|\mathbf{c}_r\|$, $r = 1, \ldots, R$.

Without loss of generality, $\underline{\mathbf{X}}$ can be expressed in terms of the normalized outer products (5.6) with $\gamma_r := a_r b_r c_r$. Substituting (5.6) and (D.7) for the tensor and the penalty respectively, (5.8) reduces to

$$\min_{\{\mathbf{u}\},\{\mathbf{v}\},\{\mathbf{w}\}} \min_{\gamma} \min_{\{a_r\},\{b_r\},\{c_r\}} \frac{1}{2}\|\,(\underline{\mathbf{Z}} - \underline{\mathbf{X}}) \circledast \underline{\mathbf{\Delta}}\|_F^2 + \frac{\mu}{2}\sum_{r=1}^{R} a_r^2 + b_r^2 + c_r^2$$
$$\text{s. to } \underline{\mathbf{X}} = \sum_{r=1}^{R} \gamma_r (\mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r), \quad \gamma_r = a_r b_r c_r. \tag{D.8}$$

Focus first on the inner minimization w.r.t. norms a_r, b_r, and c_r, for arbitrary fixed directions $\{\mathbf{u}_r\}$, $\{\mathbf{v}_r\}$, and $\{\mathbf{w}_r\}$, as well as for fixed products $\gamma_r := a_r b_r c_r$. The constraints and hence the LS part of the cost depend on γ_r only, and not on their particular factorizations $a_r b_r c_r$. Thus, the penalty is the only term that varies when γ_r is constant, rendering the inner-most minimization in (D.8) equivalent to

$$\min_{a_r, b_r, c_r} a_r^2 + b_r^2 + c_r^2 \quad \text{s. to } \gamma_r = a_r b_r c_r,\ r = 1, \ldots, R. \tag{D.9}$$

The arithmetic-mean geometric-mean inequality yields the solution to (D.9), since for scalars a_r^2, b_r^2, and c_r^2 it holds that

$$\sqrt[3]{a_r^2 b_r^2 c_r^2} \leq (a_r^2 + b_r^2 + c_r^2)/3$$

with equality when $a_r^2 = b_r^2 = c_r^2$. This implies that the minimum of (D.9) is attained at $a_r^2 = b_r^2 = c_r^2 = \gamma_r^{2/3}$.

Substituting the corresponding $\sum_{r=1}^{R}(a_r^2 + b_r^2 + c_r^2) = 3\sum_{r=1}^{R} \gamma_r^{2/3} = 3\|\boldsymbol{\gamma}\|_{2/3}^{2/3}$ into (D.8) yields (5.9). Equivalence of optimization problems is transitive; hence, showing that both (5.9) and (5.8) are equivalent to (D.8) proves them equivalent to each other, as desired. ■

D.0.15 Proof of Corollary 5.1

The following result on the norm of the matrix inverse will be used in the proof of the corollary.

Lemma D.1 *[84, p.58] If* $\mathbf{E} \in \mathbb{R}^{m\times m}$ *satisfies* $\|\mathbf{E}\|_2 \leq 1$*, then* $\mathbf{I}+\mathbf{E}$ *is invertible, and* $\left\|(\mathbf{I}+\mathbf{E})^{-1}\right\|_2 \leq (1-\|\mathbf{E}\|_2)^{-1}$.

For any value of μ, and with $(\mathbf{A},\mathbf{B},\mathbf{C})$ being the minimizers of (5.8), the useful inequality

$$\mu\left(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2+\|\mathbf{C}\|_F^2\right) \leq \|\underline{\mathbf{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^2 \tag{D.10}$$

follows by comparing the cost at the minimum and at the origin

A second characterization of the minimum of (5.8) can be obtained from the first-order optimality condition. Upon vectorizing $\mathbf{A}$, the cost in (5.8) can be rewritten as

$$\sum_{p=1}^{P}\frac{1}{2}\left\|\text{diag}[\boldsymbol{\delta}_p]\left(\mathbf{z}_p-(\mathbf{B}\text{diag}[\mathbf{e}_p^T\mathbf{C}]\otimes\mathbf{I}))\mathbf{a}\right)\right\|_2^2+\frac{\mu}{2}\|\mathbf{a}\|_2^2 \tag{D.11}$$

where $\mathbf{z}_p$, $\boldsymbol{\delta}_p$, and $\mathbf{a}$ denote the vectorizations of matrices $\mathbf{Z}_p$, $\mathbf{D}_p$, and $\mathbf{A}$, respectively. Additional regularization terms that vanish when taking derivatives w.r.t. $\mathbf{A}$ were removed from (D.11). Nulling the gradient of (D.11) w.r.t. $\mathbf{a}$ yields

$$\mathbf{a} = (\mathbf{I}+\mathbf{E})^{-1}\boldsymbol{\zeta}$$

with

$$\mathbf{E} := \frac{1}{\mu}\sum_{p=1}^{P}\left(\mathbf{B}^T\text{diag}[\mathbf{e}_p^T\mathbf{C}]\otimes\mathbf{I}\right)\text{diag}[\boldsymbol{\delta}_p]\left(\mathbf{B}\text{diag}[\mathbf{e}_p^T\mathbf{C}]\otimes\mathbf{I}\right)$$
$$\boldsymbol{\zeta} := \frac{1}{\mu}\sum_{p=1}^{P}\left(\mathbf{B}^T\text{diag}[\mathbf{e}_p^T\mathbf{C}]\otimes\mathbf{I}\right)\text{diag}[\boldsymbol{\delta}_p]\mathbf{z}_p.$$

The norms of $\mathbf{E}$ and $\boldsymbol{\zeta}$ can be bounded by using the sub-multiplicative property of the norm, and the Cauchy-Schwarz inequality, which results in

$$\|\mathbf{E}\|_2 \leq \frac{1}{\mu}\|\mathbf{B}\|_F^2\|\mathbf{C}\|_F^2$$
$$\|\boldsymbol{\zeta}\|_2 \leq \frac{1}{\mu}\|\underline{\mathbf{\Delta}}\circledast\underline{\mathbf{Z}}\|_F\|\mathbf{B}\|_F\|\mathbf{C}\|_F.$$

According to Lemma D.1, if μ is chosen large enough so that $\|\mathbf{E}\|_2 \leq 1$, then the norm of $\mathbf{A}$ is bounded by

$$\|\mathbf{A}\|_F = \|\mathbf{a}\|_2 \leq (\mu-\|\mathbf{B}\|_F^2\|\mathbf{C}\|_F^2)^{-1}\|\mathbf{B}\|_F\|\mathbf{C}\|_F\|\underline{\mathbf{\Delta}}\circledast\underline{\mathbf{Z}}\|_F \tag{D.12}$$

which constitutes the sought second characterization of the minimum of (5.8).

Yet a third characterization was obtained in Appendix D.0.14, where the norm of the factor columns were shown equal to each other, so that

$$\|\mathbf{A}\|_F = \|\mathbf{B}\|_F = \|\mathbf{C}\|_F. \tag{D.13}$$

Substituting (D.13) into (D.10) and (D.12) yields

$$\|\mathbf{A}\|_F^2 \leq \|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^2/3\mu \tag{D.14}$$

$$\|\mathbf{A}\|_F \leq (\mu - \|\mathbf{A}\|_F^4)^{-1}\|\mathbf{A}\|_F^2\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F. \tag{D.15}$$

Form (D.15), two complementary cases arise:

c1) $\|\mathbf{A}\|_F = 0$; and

c2) $1 \leq (1 - \|\mathbf{A}\|_F^4/\mu)^{-1}\|\mathbf{A}\|_F\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F/\mu.$ (D.16)

To argue that c2) is impossible, substitute (D.14) into (D.16) and square the result to obtain

$$1 \leq (1 - \|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^4/9\mu^3)^{-2}\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^4/3\mu^3. \tag{D.17}$$

But by hypothesis $\mu \geq \|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^{4/3}$ so that $\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^4/\mu^3 \leq 1$, and the right-hand side of (D.17) is bounded by 0.43, so that (D.17) does not hold. This implies that c1); i.e., $\|\mathbf{A}\|_F = \|\mathbf{B}\|_F = \|\mathbf{C}\|_F = 0$, must hold, which completes the proof.

Still, the bound at 0.43 can be pushed to one by further reducing μ, and the proof remains valid under the slightly relaxed condition $\mu > (18/(5+\sqrt{21}))^{-1/3}\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^{4/3} \simeq 0.81\|\underline{\boldsymbol{\Delta}}\circledast\underline{\mathbf{Z}}\|_F^{4/3}$. ■

D.0.16 RKHS imputation

Recursive application of the Representer Theorem yields finitely-parameterized minimizers $\hat{a}_r$, $\hat{b}_r$, and $\hat{c}_r$ of (5.14), given by

$$\hat{a}_r(m) = \textstyle\sum_{m'=1}^{M}\alpha_{rm'}k_{\mathcal{M}}(m',m)$$
$$\hat{b}_r(n) = \textstyle\sum_{n'=1}^{N}\beta_{rn'}k_{\mathcal{N}}(n',n)$$
$$\hat{c}_r(p) = \textstyle\sum_{p'=1}^{P}\gamma_{rp'}k_{\mathcal{P}}(p',p).$$

Defining vectors $\boldsymbol{k}_{\mathcal{M}}^T(m) := [k_{\mathcal{M}}(1,m),\ldots,k_{\mathcal{M}}(M,m)]$, and correspondingly $\boldsymbol{k}_{\mathcal{N}}^T(n) := [k_{\mathcal{N}}(1,n),\ldots\ldots,k_{\mathcal{N}}(N,n)]$, and $\boldsymbol{k}_{\mathcal{P}}^T(p) := [k_{\mathcal{P}}(1,p),\ldots,k_{\mathcal{P}}(P,p)]$, along with matrices $\hat{\mathbf{A}} \in \mathbb{R}^{M\times R} : \hat{A}(m,r) := \alpha_{mr}$, $\hat{\mathbf{B}} \in \mathbb{R}^{N\times R} : \hat{B}(n,r) := \beta_{nr}$, and $\hat{\mathbf{C}} \in \mathbb{R}^{P\times R} : \hat{C}(p,r) := \gamma_{pr}$, it follows that

$$\hat{f}_R(m,n,p) = \sum_{r=1}^{R}\hat{a}_r(m)\hat{b}_r(n)\hat{c}_r(p) = \boldsymbol{k}_{\mathcal{M}}^T(m)\hat{\mathbf{A}}\mathrm{diag}\left[\boldsymbol{k}_{\mathcal{P}}^T(p)\hat{\mathbf{C}}\right]\hat{\mathbf{B}}^T\boldsymbol{k}_{\mathcal{N}}(n). \tag{D.18}$$

Matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$ are further obtained by solving

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{C}}} \sum_{i=1}^{P} \left\| \left(\mathbf{Z}_p - \mathbf{K}_{\mathcal{M}} \hat{\mathbf{A}} \text{diag} \left[\mathbf{e}_p^T \mathbf{K}_{\mathcal{P}} \hat{\mathbf{C}} \right] \hat{\mathbf{B}}^T \mathbf{K}_{\mathcal{N}} \right) \circledast \boldsymbol{\Delta}_p \right\|_F^2$$
$$+ \frac{\mu}{2} \left(\text{Tr}(\hat{\mathbf{A}}^{\text{T}} \mathbf{K}_{\mathcal{M}} \hat{\mathbf{A}}) + \text{Tr}(\hat{\mathbf{B}}^{\text{T}} \mathbf{K}_{\mathcal{N}} \hat{\mathbf{B}}) + \text{Tr}(\hat{\mathbf{C}}^{\text{T}} \mathbf{K}_{\mathcal{P}} \hat{\mathbf{C}}) \right)$$

which is equivalent to (5.16) after changing variables $\mathbf{A} := \mathbf{K}_{\mathcal{M}} \hat{\mathbf{A}}$, $\mathbf{B} := \mathbf{K}_{\mathcal{N}} \hat{\mathbf{B}}$, and $\mathbf{C} = \mathbf{K}_{\mathcal{P}} \hat{\mathbf{C}}$, just as (D.18) becomes (5.15).

D.0.17 Covariance estimation

Inspection of the entries of $\mathbf{K}_{\mathcal{P}}(p,p') := \mathbb{E}\left[\text{Tr}\left(\mathbf{X}_p^T \mathbf{X}_{p'}\right)\right]$ under the PARAFAC model, yields

$$\begin{aligned}
\mathbf{K}_{\mathcal{P}}(p,p') &:= \mathbb{E}\left[\text{Tr} \left(\sum_{r=1}^{R} \mathbf{b}_r \mathbf{c}_r(p) \mathbf{a}_r^T \sum_{r'=1}^{R} \mathbf{a}_{r'} \mathbf{c}_{r'}(p') \mathbf{b}_{r'}^T \right) \right] \\
&= \sum_{r=1}^{R} \sum_{r'=1}^{R} \mathbb{E}\left[\mathbf{c}_r^T(p) \mathbf{c}_{r'}(p') \right] \mathbb{E}\left[\mathbf{b}_{r'}^T \mathbf{b}_r \right] \mathbb{E}\left[\mathbf{a}_r^T \mathbf{a}_{r'} \right] \\
&= \sum_{r=1}^{R} \mathbb{E}\left[\mathbf{c}_r(p) \mathbf{c}_r(p') \right] \mathbb{E}[\|\mathbf{b}_r\|^2] \mathbb{E}[\|\mathbf{a}_r\|^2] \\
&= \sum_{r=1}^{R} \mathbf{R}_C(p,p') \text{Tr}(\mathbf{R}_B) \text{Tr}(\mathbf{R}_A) \\
&= R\, \mathbf{R}_C(p,p') \text{Tr}(\mathbf{R}_B) \text{Tr}(\mathbf{R}_A).
\end{aligned}$$

After summing over $p' = p$, one obtains

$$\mathbb{E}[\|\underline{\mathbf{X}}\|_F^2]\& = \sum_{p=1}^{P} \mathbb{E}[\|\mathbf{X}_p\|_F^2] = \sum_{p=1}^{P} \mathbf{R}_{\mathcal{P}}(p,p)$$
$$= R \text{Tr}(\mathbf{R}_C) \text{Tr}(\mathbf{R}_B) \text{Tr}(\mathbf{R}_A). \qquad \text{(D.19)}$$

In addition, by incorporating the equal power assumption (5.12), (D.19) further simplifies to

$$\mathbb{E}[\|\underline{\mathbf{X}}\|_F^2] = R\theta^3$$

as stated in (5.18).

D.0.18 Proof of Lemma 5.1

Towards establishing properties i)-iii) in Lemma 5.1, consider expanding the difference between $g(\mathbf{A},\bar{\mathbf{A}})$ and $f(\mathbf{A})$. One obtains

$$\begin{aligned} g(\mathbf{A},\bar{\mathbf{A}})-f(\mathbf{A}) &= \sum_{r=1}^{R}[\lambda\mathbf{a}_r^T\mathbf{a}_r - 2\boldsymbol{\theta}_r^T\mathbf{a}_r + \boldsymbol{\theta}_r^T\bar{\mathbf{a}}_r - \bar{\mathbf{a}}_r^T\mathbf{R}_A^{-1}\bar{\mathbf{a}}_r] \\ &= \sum_{r=1}^{R}(\mathbf{a}_r-\bar{\mathbf{a}}_r)^T(\lambda\mathbf{I}-\mathbf{R}_A^{-1})(\mathbf{a}_r-\bar{\mathbf{a}}_r) \end{aligned}$$

which is nonnegative from the definition of λ and, together with its gradient, vanish at $\bar{\mathbf{A}}$. ■

D.0.19 Proof of Lemma 5.2

Function $g(\mathbf{A},\bar{\mathbf{A}})$ in (5.27) is formed from $f(\mathbf{A})$ after substituting $g_1(\mathbf{A},\bar{\mathbf{A}})$ for $f_1(\mathbf{A})$, and $g_2(\mathbf{A},\bar{\mathbf{A}})$ for $f_2(\mathbf{A})$, respectively, as defined by

$$f_1(\mathbf{A}) := \mathrm{Tr}\left(\mathbf{A}^T\mathbf{R}_A^{-1}\mathbf{A}\right) \tag{D.20}$$

$$g_1(\mathbf{A},\bar{\mathbf{A}}) := \lambda\mathrm{Tr}\left(\mathbf{A}^T\mathbf{A}\right) - 2\mathrm{Tr}(\boldsymbol{\Theta}^T\mathbf{A}) + \mathrm{Tr}(\boldsymbol{\Theta}^T\bar{\mathbf{A}}) \tag{D.21}$$

where $\lambda := \lambda_{\max}(\mathbf{R}_A^{-1})$ and $\boldsymbol{\Theta} := (\lambda\mathbf{I}-\mathbf{R}_A^{-1})\bar{\mathbf{A}}$, and

$$f_2(\mathbf{A}) := -\mathbf{1}_M\boldsymbol{\Delta}\circledast\mathbf{Z}\log(\mathbf{A}\boldsymbol{\Pi}^T)\mathbf{1}_{NP} \tag{D.22}$$

$$g_2(\mathbf{A},\bar{\mathbf{A}}) := -\sum_{r=1}^{R}\sum_{m=1}^{M}\sum_{k=1}^{NP}\delta_{mk}z_{mk}\alpha_{mkr}\log\left(\frac{a_{mr}\pi_{kr}}{\alpha_{mkr}}\right) \tag{D.23}$$

with $\alpha_{mkr} := \bar{a}_{mr}\pi_{kr}/\sum_{r'=1}^{R}\bar{a}_{mr'}\pi_{kr'}$. Hence, properties i)-iii) will be satisfied by the functions $g(\mathbf{A},\bar{\mathbf{A}})$ and $f(\mathbf{A})$ in Lemma 5.2, as long as they are satisfied both by the pairs (D.20)-(D.21) and (D.22)-(D.23).

Focusing on the first pair, the arguments in the proof of Lemma 5.1 (D.0.18) imply that properties i)-iii) are satisfied by $g_1(\mathbf{A},\bar{\mathbf{A}})$ and $f_1(\mathbf{A})$. Considering the second pair, and expanding $f_2(\mathbf{A})$ yields

$$f_2(\mathbf{A}) = -\sum_{m=1}^{M}\sum_{k=1}^{NP}\delta_{mk}z_{mk}\log\left(\sum_{r=1}^{R}a_{mr}\pi_{kr}\right) \tag{D.24}$$

where the logarithm can be rewritten as (see also [48])

$$\log\left(\sum_{r=1}^{R}a_{mr}\pi_{kr}\right) = \log\left(\sum_{r=1}^{R}\alpha_{mkr}\frac{a_{mr}\pi_{kr}}{\alpha_{mkr}}\right) \tag{D.25}$$

$$\geq \sum_{r=1}^{R}\alpha_{mkr}\log\left(\frac{a_{mr}\pi_{kr}}{\alpha_{mkr}}\right) \tag{D.26}$$

and the inequality holds because of the concavity of the logarithm and the coefficients $\{\alpha_{mkr}\}_{r=1}^{R}$ summing up to one. Since substituting (D.26) for (D.25) in (D.24) results in (D.23), it follows that $g_2(\mathbf{A}, \bar{\mathbf{A}})$ and $f_2(\mathbf{A})$ satisfy property iii). The proof is complete after evaluating the pair of functions and their derivatives at $\bar{\mathbf{A}}$ to confirm that properties i) and ii) hold too.

The minimum $a^{\star}_{g,mr} := t_{mr} + \sqrt{t^2_{mr} + s_{mr}}$ is obtained readily after equating to zero the derivative of the corresponding summand in (5.27), and selecting the nonnegative root. ■