9-2-2014

Optimization algorithms for inference and classification of genetic profiles from undersampled measurements

Belhassen Bayar

OPTIMIZATION ALGORITHMS FOR INFERENCE AND CLASSIFICATION OF GENETIC PROFILES FROM UNDERSAMPLED MEASUREMENTS

by

Belhassen Bayar

A Thesis

Submitted to the
Department of Electrical & Computer Engineering
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science
at
Rowan University
June 2014

Thesis Chair: Nidhal Bouaynaya

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to Dr. Nidhal Bouaynaya, my supervisor who has always bothered to offer me the best working conditions possible. I thank her for her wide availability, her high scientific qualifications and her guidance, illuminating discussions related to this work and beyond, encouragement, moral and financial support in this research.

I express my appreciation and gratitude to Dr. Roman Shterenberg , Associate Professor at the University of Alabama at Birmingham USA, for the time he spent with me, his availability even when he was abroad and the valuable advice he has given me throughout my research.

I also would like to express my deep and sincere gratitude to Dr. Robi Polikar, Professor & Chair at the ECE Department, for the high quality courses he teaches, his availability and eagerness to provide the best learning experience for students at the department.

Many thanks to all the students who accompanied me during these years and have continued to create a good working atmosphere within the laboratory. Deepest thanks to my dear parents and grandmother to whom I owe so much. I would have neither the means nor the strength to accomplish this work without them. I also want to express my gratitude to my friends who have continued to give me the moral and intellectual support throughout my work during all the good and bad moments. They always say the best is for the end, that's why I dedicate this project to my dear sister, my little light that gave me energy and courage.

Abstract

Belhassen Bayar

OPTIMIZATION ALGORITHMS FOR INFERENCE AND CLASSIFICATION OF GENETIC PROFILES FROM UNDERSAMPLED MEASUREMENTS

2014/06

Nidhal Bouaynaya, Ph.D.

Master of Science in Electrical & Computer Engineering

In this thesis, we tackle three different problems, all related to optimization techniques for inference and classification of genetic profiles. First, we extend the deterministic Non-negative Matrix Factorization (NMF) framework to the probabilistic case (PNMF). We apply the PNMF algorithm to cluster and classify DNA microarrays data. The proposed PNMF is shown to outperform the deterministic NMF and the sparse NMF algorithms in clustering stability and classification accuracy. Second, we propose SMURC: Small-sample MUltivariate Regression with Covariance estimation. Specifically, we consider a high dimension low sample-size multivariate regression problem that accounts for correlation of the response variables. We show that, in this case, the maximum likelihood approach is senseless because the likelihood diverges. We propose a normalization of the likelihood function that guarantees convergence. Simulation results show that SMURC outperforms the regularized likelihood estimator with known covariance matrix and the state-of-the-art sparse Conditional Graphical Gaussian Model (sCGGM). In the third Chapter, we derive a new greedy algorithm that provides an exact sparse solution of the combinatorial ℓ_0-optimization problem in an exponentially less computation time. Unlike other greedy approaches, which are only approximations of the exact sparse solution, the proposed greedy approach, called Kernel reconstruction, leads to the exact optimal solution.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Research Objectives

We outline the goal of this research through the following objectives:

1. Study and analyse the Non-negative Matrix Factorization (NMF) and propose a probabilistic extension to NMF (PNMF) for data corrupted by noise.

2. Build a PNMF-based classifier and apply it for tumor classification from gene expression data.

3. Derive a convex optimization algorithm for the solution of an under-determined multivariate regression problem. Apply the proposed algorithm to infer genetic regulatory networks from gene expression data.

4. Derive a greedy algorithm for exact reconstruction of sparse signals from a limited number of observations.

1.2 Research Contribution

This work contributes to the field of computational bioinformatics and biology through the application of the signal processing algorithms aiming to study and analyze the microarray data. Our work shifts the focus of the genomic signal processing community from analyzing the genes expression patterns and samples clusters to considering

the mathematical aspect of the algorithm and deriving its application in the stochastic work. We also focus on solving under-determined multivariate regression systems in order to infer gene regulatory networks. These networks are known to be sparse, therefore, we have a great interest in studying the compressive sensing approach which recovers sparse signal from linear model. Specific contributions of this work include:

- The improvement of the mathematical proof for the NMF algorithm by providing a general evidence (see Appendix preposition 2).
- The development of a new NMF algorithm for the noisy Microarray data in order to improve the basic NMF approach and to predict some hidden data features.
- Solving under-determined multivariate regression systems to infer gene regulatory networks using our new SMURC algorithm.
- Recover k-sparse signal using our new approach, called Kernel Reconstruction, that guarantees an exact reconstruction and less computational time comparing to the ℓ_0-based compressive sensing approach [18].

1.3 Organization

This thesis is organized as follows.

In Chapter 2, we study and analyze the Non-negative Matrix Factorization and derive its probabilistic approach that we call PNMF algorithm and then we derive its corresponding update rules. The proof of the developed approaches is provided in

the Appendix chapter. We compare the performance of our PNMF approach with its homologues in clustering as well as classification.

In Chapter 3, we develop a new approach, called Small-sample MUltivariate Regression with Covariance Estimation (SMURC), to solve under-determined multivariate regression systems. We use this approach to infer gene regulatory networks. We compare our algorithm to other techniques cited in related works and using a synthetic data. Subsequently, we apply our approach to infer the know interactions in the Drosophila's 11-gene wing muscle network.

Finally, in Chapter 4 we provide a complete review of the compressive sensing technique. We also come up with a new approach that performs an exact reconstruction of a sparse signal. We call this approach, Kernel Reconstruction, and we compare it with what has been suggested in the related work.

Chapter 2

Probabilistic Non-negative Matrix Factorization: Theory and Application to Microarray Data Analysis

2.1 Introduction

Extracting knowledge from experimental raw data and measurements is an important objective and challenge in signal processing. Often data collected is high dimensional and incorporates several inter-related variables, which are combinations of underlying latent components or factors. Approximate low-rank matrix factorizations play a fundamental role in extracting these latent components [14]. In many applications, signals to be analyzed are non-negative, e.g., pixel values in image processing, price variables in economics and gene expression levels in computational biology. For such data, it is imperative to take the non-negativity constraint into account in order to obtain a meaningful physical interpretation. Classical decomposition tools, such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Blind Source Separation (BSS) and related methods do not guarantee to maintain the non-negativity constraint. Non-negative matrix factorization (NMF) represents non-negative data in terms of lower-rank non-negative factors. NMF proved to be a powerful tool in many applications in biomedical data processing and analysis,

such as muscle identification in the nervous system [54], classification of images [29], gene expression classification [10], biological process identification [32] and transcriptional regulatory network inference [38]. The appeal of NMF, compared to other clustering and classification methods, stems from the fact that it does not impose any prior structure or knowledge on the data. Brunet *et al.* successfully applied NMF to the classification of gene expression datasets [10] and showed that it leads to more accurate and more robust clustering than the Self-Organizing Maps (SOMs) and Hierarchical Clustering (HC). Analytically, the NMF method factors the original non-negative matrix V into two lower rank non-negative matrices, W and H such that $V = WH + E$, where E is the residual error. Lee and Seung [33] derived algorithms for estimating the optimal non-negative factors that minimize the Euclidean distance and the Kullback-Leibler divergence cost functions. Their algorithms, guaranteed to converge, are based on multiplicative update rules, and are a good compromise between speed and ease of implementation. In particular, the Euclidean distance NMF algorithm can be shown to reduce to the gradient descent algorithm for a specific choice of the step size [33]. Lee and Seung's NMF factorization algorithms have been widely adopted by the community [6, 10, 19, 59].

The NMF method is, however, deterministic. That is, the algorithm does not take into account the measurement or observation noise in the data. On the other hand, data collected using electronic or biomedical devices, such as gene expression profiles, are known to be inherently noisy and therefore, must be processed and analyzed by systems that take into account the stochastic nature of the data. Furthermore, the effect of the data noise on the NMF method in terms of convergence and robustness has

not been previously investigated. Thus, questions about the efficiency and robustness of the method in dealing with imperfect or noisy data are still unanswered.

In this chapter, we extend the NMF framework and algorithms to the stochastic case, where the data is assumed to be drawn from a multinomial probability density function. We call the new framework Probabilistic NMF or PNMF. We show that the PNMF formulation reduces to a weighted regularized matrix factorization problem. We generalize and extend Lee and Seung's algorithm to the stochastic case; thus providing PNMF updates rules, which are guaranteed to converge to the optimal solution. The proposed PNMF algorithm is applied to cluster and classify gene expression datasets, and is compared to other NMF and non-NMF approaches including sparse NMF (SNMF) and SVM.

The chapter is organized as follows: In Section 2.1.1, we discuss related work and clarify the similarities and differences between the proposed PNMF algorithm and other approaches to NMF present in the literature. In Section 2.2, we review the (deterministic) NMF formulation and extend Lee and Seung's NMF algorithm to include a general class of convergent update rules. In Section 2.3, we introduce the probabilistic NMF (PNMF) framework and derive its corresponding update rules. In Section 2.4, we present a data classification method based on the PNMF algorithm. Section 2.5 applies the proposed PNMF algorithm to cluster and classify gene expression profiles. The results are compared with the deterministic NMF, sparse NMF and SVM. Finally, a summary of the main contributions and concluding remarks are outlined in Section 2.6.

In this chapter, scalars are denoted by lower case letters, e.g., n, m; vectors are

denoted by bold lower case letters, e.g., $\boldsymbol{x}, \boldsymbol{y}$; and matrices are referred to by upper case letters, e.g., A, V. $\boldsymbol{x}_i$ denotes the i^{th} element of vector $\boldsymbol{x}$ and A_{ij} is the $(i,j)^{th}$ entry of matrix A. Throughout the chapter, we provide references to known results and limit the presentation of proofs to new contributions. All proofs are presented in the Appendix section.

2.1.1 Related work. Several variants of the NMF algorithm have been proposed in the literature. An early form of NMF, called Probabilistic Latent Semantic Analysis (PLSA) [27], [28], [37], was used to cluster textual documents. The key idea is to map high-dimensional count vectors, such as the ones arising in text documents, to a lower dimensional representation in a so-called *latent semantic space*. PLSA has been shown to be equivalent to NMF factorization with Kullback-Leibler (KL) divergence, in the sense that they have the same objective function and any solution of PLSA is a solution of NMF with KL minimization [17].

Many variants of the NMF framework introduce additional constraints on the non-negative factor matrices W and H, such as sparsity and smoothness. Combining sparsity with non-negative matrix factorization is partly motivated by modeling neural information processing, where the goal is to find a decomposition in which the hidden components are sparse. Hoyer [30] combined sparse coding and non-negative matrix factorization into *non-negative sparse coding* (NNSC) to control the trade-off between sparseness and accuracy of the factorization. The sparsity constraint is imposed by constraining the l_1-norm. The NNSC algorithm resorts to setting the

negative values of one of the factor matrices to zero. This procedure is not always guaranteed to converge to a stationary point. Kim and Park [31] solved the sparse NMF optimization problem via alternating non-negativity-constrained least squares. They applied sparse NMF to cancer class discovery and gene expression data analysis.

NMF has also been extended to consider a class of smoothness constraints on the optimization problem [41]. Enforcing smoothness on the factor matrices is desirable in applications such as unmixing spectral reflectance data for space object identification and classification purposes [41]. However, the algorithm in [41] forces positive entries by setting negative values to zero and hence may suffer from convergence issues. Similarly, different penalty terms may be used depending upon the desired effects on the factorization. A unified model of constrained NMF, called versatile sparse matrix factorization (VSMF), has been proposed in [34]. The VSMF framework includes both l_1 and l_2-norms. The l_1-norm is used to induce sparsity and the l_2-norm is used to obtain smooth results. In particular, the standard NMF, sparse NMF [30], [31] and semi-NMF [16], where the non-negativity constraint is imposed on only one of the factors, can be seen as special cases of VSMF.

Another variant of the NMF framework is obtained by considering different distances or measures between the original data matrix and its non-negative factors [49], [56]. Sandler and Lindenbaum [49] proposed to factorize the data using the earth movers distance (EMD). The EMD NMF algorithm finds the local minimum by solving a sequence of linear programming problems. Though the algorithm has shown significant improvement in some applications, such as texture classification and face recognition, it is computationally very costly. To address this concern, the authors

have proposed the wavelet-based approximation to the EMD distance, WEMD, and used it in place of EMD. They argued that the local minima of EMD and WEMD are generally collocated when using a gradient-based method. A similarity measure based on the correntropy, termed NMF MCC, has been proposed in [56]. The correntropy measure employs the Gaussian kernel to map the linear data space to a non-linear space. The optimization problem is solved using an expectation maximization based approach.

A collection of non-negative matrix factorization algorithms implemented for Matlab is available at `http://cogsys.imm.dtu.dk/toolbox/nmf/`. Except for PLSA, which was originally proposed as a statistical technique for text clustering, the presented NMF approaches do not explicitly assume a stochastic framework for the data. In other words, the data is assumed to be deterministic. In this work, we assume that the original data is a sample drawn from a multinomial distribution and derive the maximum a posteriori (MAP) estimates of the non-negative factors. The proposed NMF framework, termed Probabilistic NMF or PNMF, does not impose any additional constraints on the non-negative factors like SNMF or VSMF. Interestingly, however, the formulation of the MAP estimates reduces to a weighted regularized matrix factorization problem that resembles the formulations in constrained NMF approaches. The weighting parameters, however, have a different interpretation: they refer to signal to noise ratios rather than specific constraints.

2.2 Non-negative Matrix Factorization

The non-negative matrix factorization (NMF) is a constrained matrix factorization problem, where a non-negative matrix V is factorized into two non-negative matrices W and H. Here, non-negativity refers to elementwise non-negativity, i.e., all elements of the factors W and H must be equal to or greater than zero. The non-negativity constraint makes NMF more difficult algorithmically than classical matrix factorization techniques, such as principal component analysis and singular value decomposition. Mathematically, the problem is formulated as follows: Given a non-negative matrix $V \in \mathbb{R}^{n \times m}$, find non-negative matrices $W \in \mathbb{R}^{n \times k}$ and $H \in \mathbb{R}^{k \times m}$ such that $V \approx WH$. The optimal factors minimize the squared error and are solutions to the following constrained optimization problem,

$$(W^*, H^*) = \underset{W,H \geq 0}{\arg\min} f(W, H) = \|V - WH\|_F^2, \tag{2.1}$$

where $\|.\|_F$ denotes the Frobenius norm and f is the squared Euclidean distance function between V and WH. The cost function f is convex with respect to either the elements of W or H, but not both. Alternating minimization of such a cost leads to the ALS (Alternating Least squares) algorithm [25], [55], [1], which can be described as follows:

1. Initialize W randomly or by using any a priori knowledge.

2. Estimate H as $H = (W^T W)^- W^T V$ with fixed W.

3. Set all negative elements of H to zero or some small positive value.

4. estimate W as $W = VH^T(HH^T)^-$ with fixed H.

5. Set all negative elements of W to zero or some small positive value.

In this algorithm, A^- denotes the Moore-Penrose inverse of A. The ALS algorithm has been used extensively in the literature [25], [55], [1]. However, it is not guaranteed to converge to a global minimum nor even a stationary point. Moreover, it is often not sufficiently accurate, and it can be slow when the factor matrices are ill-conditioned or when the columns of these matrices are co-linear. Furthermore, the complexity of the ALS algorithm can be high for large-scale problems as it involves inverting a large matrix. Lee and Seung [33] proposed a multiplicative update rule, which is proven to converge to a stationary point, and does not suffer from the ALS drawbacks. In what follows, we present Lee and Seung's multiplicative update rule as a special case of a class of update rules, which converge towards a stationary point of the NMF problem.

Proposition 1. *The function* $f(W,H) = \|V - WH\|_F^2$ *is non-increasing under the update rules*

$$\begin{cases} \boldsymbol{h}^{k+1} = \boldsymbol{h}^k - K_h^{-1}(W^TW\boldsymbol{h}^k - W^T\boldsymbol{v}) \\ \\ \tilde{\boldsymbol{w}}^{k+1} = \tilde{\boldsymbol{w}}^k - K_w^{-1}(HH^T\tilde{\boldsymbol{w}}^k - H\tilde{\boldsymbol{v}}) \end{cases} \tag{2.2}$$

where $\tilde{\boldsymbol{w}}$ *and* $\tilde{\boldsymbol{v}}$ *are the columns of* W^T *and* V^T*, respectively, and* K_h *and* K_w *satisfy the following conditions*

a. K_h *and* K_w *are diagonal matrices with (strictly) positive elements for all vectors* $\boldsymbol{h}$ *and* $\tilde{\boldsymbol{w}}$*.*

b. *$K_h \boldsymbol{h}^k \geq W^T W \boldsymbol{h}^k$ and $K_w \tilde{\boldsymbol{w}}^k \geq H H^T \tilde{\boldsymbol{w}}^k$ where the inequality is elementwise.*

c. *The matrices $K_h - W^T W$ and $K_w - H H^T$ are positive semi-definite (p.s.d) for all $\boldsymbol{h}$ and $\tilde{\boldsymbol{w}}$.*

The function f is invariant under these update rules if and only if W and H are at a stationary point.

The following corollary presents a special choice of the matrices K_h and K_w, which leads to Lee and Seung's multiplicative rule for the NMF problem.

Corollary 2.2.1. *In Proposition 1, chose K_h and K_w as follows:*

$$(K_h)_{ij} = \delta_{ij} (W^T W \boldsymbol{h}^k)_i / \boldsymbol{h}_i^k, \tag{2.3}$$

$$(K_w)_{ij} = \delta_{ij} (H H^T \tilde{\boldsymbol{w}}^k)_i / \tilde{\boldsymbol{w}}_i^k, \tag{2.4}$$

Where $\boldsymbol{h}_i^k, \tilde{\boldsymbol{w}}_i^k$ are the i^{th} entries of the vectors $\boldsymbol{h}^k$ and $\tilde{\boldsymbol{w}}^k$, respectively, and δ_{ij} is the kronecker function, i.e., $\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases}$ This choice leads to the following update rule:

$$\begin{cases} H_{ij} \longleftarrow H_{ij} \frac{(W^T V)_{ij}}{(W^T W H)_{ij}} \\ \\ W_{ij} \longleftarrow W_{ij} \frac{(V H^T)_{ij}}{(W H H^T)_{ij}} \end{cases} \tag{2.5}$$

The function f is invariant under these updates if and only if W and H are at a stationary point.

Corollary 2.2.1 corresponds to the update rules proposed by Lee and Seung [33]. Proposition 1 presents a general class of update rules, which converge to a station-

ary point of the NMF problem. From the proof of the Proposition (detailed in the Appendix), it will be clear that conditions [a], [b] and [c] in Proposition 1 are only sufficient conditions for the update rules to converge towards a stationary point. That is, there may exist K_h and K_w that do not satisfy these conditions but that lead to update rules that converge towards a stationary point. The particular choice of K_h and K_w in Corollary 1 corresponds to the fastest convergent update rule among all matrices satisfying conditions [a]-[c] in Proposition 1. Observe also that since the data matrix V is non-negative, the update rule in (2.5) leads to non-negative factors W and H as long as the initial values of the algorithm are chosen to be non-negative.

2.3 Probabilistic Non-negative Matrix Factorization

2.3.1 The PNMF framework. In this section, we assume that the data, represented by the non-negative matrix V, is corrupted by additive white Gaussian noise. Then, the data follows the following conditional distribution,

$$p(V \mid W, H, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [\mathcal{N}(V_{ij} \mid \mathbf{u}_i^T \mathbf{h}_j, \sigma^2)], \tag{2.6}$$

where $\mathcal{N}(.|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and standard deviation σ, $\mathbf{u}_i$ and $\mathbf{h}_j$ denote, respectively, the i^{th} column of the matrix $U = W^T$ (or the i^{th} row of W) and the j^{th} column of the matrix H. Zero mean Gaussian priors are imposed on $\mathbf{u}_i$ and $\mathbf{h}_j$ to control the model parameters.

Specifically, we have

$$p(U \mid \sigma_W^2) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{u}_i \mid 0, \sigma_W^2 I) = p(W \mid \sigma_W^2). \tag{2.7}$$

$$p(H \mid \sigma_H^2) = \prod_{i=1}^{M} \mathcal{N}(\mathbf{h}_j \mid 0, \sigma_H^2 I). \tag{2.8}$$

We estimate the factor matrices W and H using the maximum a posteriori (MAP) criterion. The logarithm of the posterior distribution is given by

$$\ln(p(W, H \mid V, \sigma^2, \sigma_H^2, \sigma_W^2)) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} (V_{ij} - \mathbf{u}_i^T \mathbf{h}_j)^2$$
$$-\frac{1}{2\sigma_W^2} \sum_{i=1}^{N} \|\mathbf{u}_i\|^2 - \frac{1}{2\sigma_H^2} \sum_{j=1}^{M} \|\mathbf{h}_j\|^2 + C, \tag{2.9}$$

where C is a constant term depending only on the standard deviations σ, σ_W and σ_H. Maximizing (2.9) is equivalent to minimizing the following function

$$(W^*, H^*) = \arg\min_{W,H \geq 0} \|V - WH\|_F^2 + \lambda_W \|W\|_F^2$$
$$+ \lambda_H \|H\|_F^2, \tag{2.10}$$

where $\lambda_W = \frac{\sigma^2}{\sigma_W^2}$ and $\lambda_H = \frac{\sigma^2}{\sigma_H^2}$. Observe that the PNMF formulation in (2.10) corresponds to a weighted regularized matrix factorization problem. Moreover, the PNMF reduces to the NMF for $\sigma = 0$. The following proposition provides the update rules for the PNMF constrained optimization problem.

Proposition 2. *The function*

$$f(W,H) = \|V - WH\|_F^2 + \alpha\|W\|_F^2 + \beta\|H\|_F^2 \tag{2.11}$$

is non-increasing under the update rules

$$\begin{cases} H_{ij} & \longleftarrow & H_{ij}\frac{(W^TV)_{ij}}{(W^TWH+\beta H)_{ij}} \\ \\ W_{ij} & \longleftarrow & W_{ij}\frac{(VH^T)_{ij}}{(WHH^T+\alpha W)_{ij}} \end{cases} \tag{2.12}$$

The function f is invariant under these updates if and only if W and H are at a stationary point.

Observe that, since the data matrix V is non-negative, the update rules in (2.12) lead to non-negative factors W and H as long as the initial values of the algorithm are chosen to be non-negative.

2.4 PNMF-based Data Classification

In this section, we show how the PNMF output can be used to extract relevant features from the data for classification purposes. The main idea relies on the fact that metasamples extracted from the PNMF factorization contain the inherent structural information of the original data in the training set. Thus, each sample in a test set can be written as a sparse linear combination of the metasamples extracted from the training set. The classification task then reduces to computing the representation coefficients for each test sample based on a chosen discriminating function. The

sparse representation approach has been shown to lead to more accurate and robust results [64]. The sparsity constraint is imposed through an l_1-regularization term [64]. Thus, a test sample may be represented in terms of few metasamples.

2.4.1 Sparse Representation Approach. We divide the data, represented by the $n \times m$ matrix V, into training and testing sets, where the number of classes k is assumed to be known. In Section 2.5, we describe a method to estimate the number of classes based on the PNMF clustering technique. The training data is ordered into a matrix A with n rows of genes and r columns of training samples with $r < m$. Thus, A is a sub-matrix of V used to recognize any new presented sample from the testing set. We arrange the matrix A in such a way to group samples which belong to the same class in the same sub-matrix A_i where $(1 \leq i \leq k)$. Then A can be written as $A = [A_1, A_2, ..., A_k]$ and each matrix A_i is a concatenation of r_i columns of the i^{th} class $A_i = [\boldsymbol{c}_{i,1}, \boldsymbol{c}_{i,2}, ..., \boldsymbol{c}_{i,r_i}]$

A test sample $\boldsymbol{y} \in \mathbb{R}^n$ that belongs to the i^{th} class can be written as the following linear combination of the A_i columns,

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{c}_{i,1} + \alpha_{i,2}\boldsymbol{c}_{i,2} + ... + \alpha_{i,r_i}\boldsymbol{c}_{i,r_i}, \tag{2.13}$$

for some scalars $\alpha_{i,q} \in \mathbb{R}$, $1 \leq q \leq r_i$.

Equation (2.13) can be re-written as

$$\boldsymbol{y} = A\boldsymbol{x}, \tag{2.14}$$

where

$$\boldsymbol{x} = [0, ...0, \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,r_i}, 0..., 0]^T \in \mathbb{R}^r, \tag{2.15}$$

is the coefficient vector of the testing sample $\boldsymbol{y}$. $\boldsymbol{x}$ is a r_i-sparse vector whose nonzero entries are associated with the columns of the sub-matrix A_i, hence the name *sparse representation*. Therefore, predicting the class of test sample $\boldsymbol{y}$ reduces to estimating the vector $\boldsymbol{x}$ in Eq. (2.14).

We propose to find the sparsest least-squares estimate of the coefficient $\boldsymbol{x}$ as the solution to the following regularized least-squares problem [57]

$$\hat{\boldsymbol{x}} = \min_{\boldsymbol{x}} \{\|A\boldsymbol{x} - \boldsymbol{y}\|_2 + \lambda \|\boldsymbol{x}\|_1\}, \tag{2.16}$$

where $\|\boldsymbol{x}\|_1$ denotes the l_1-norm of vector $\boldsymbol{x}$, i.e., $\|\boldsymbol{x}\|_1 = \sum_i |\boldsymbol{x}_i|$, and λ is a positive scalar used to control the tradeoff between the sparsity of $\boldsymbol{x}$ and the accuracy of the reconstruction error. Donoho *et al.* showed that the l_1-norm approximates the l_0-norm, which counts the number of non-zero entries in a vector [18]. The l_0-norm problem, however, is NP hard, whereas the l_1-norm is convex. The optimization problem in (2.16) is therefore convex; thus, it admits a global solution, which can be efficiently computed using convex optimization solvers [23]. Actually, one can show that (2.16) is a Second-Order Cone Programming (SOCP) problem [9].

2.4.2 PNMF-based classification . The classifier's features are given by the metasamples computed by the PNMF algorithm. We first compute the PNMF factorization of each sub-matrix A_i as

$$A_i \sim W_i \times H_i, \tag{2.17}$$

where W_i and H_i are respectively $n \times k_i$ and $k_i \times r_i$ non-negative matrices. k_i refers to the number of metasamples needed to describe and summarize the i^{th} class. The value of k_i is experimentally determined and depends on the number of training samples r_i in each class and the total number of classes k. We subsequently concatenate all the W_i matrices to form the matrix $W = [W_1, W_2, .., W_k]$. Observe that the matrix W contains the metasamples of the entire training set. Therefore, a test sample $\boldsymbol{y}$ that belongs to the i^{th} class should approximately lie in the space spanned by the W_i columns.

The classification problem in (2.16) can therefore be re-written as

$$\hat{\boldsymbol{x}} = \min_{\boldsymbol{x}} \{\|W\boldsymbol{x} - \boldsymbol{y}\|_2 + \lambda\|\boldsymbol{x}\|_1\}, \tag{2.18}$$

Which can be easily solved using a SOCP solver [9].

PNMF-based classification algorithm The PNMF-based classification algorithm is summarized below.

Input: Gene expression data $V \in \mathbb{R}^{n\times m}$. It is assumed that V contains at least r labeled samples, which can be used in the learning or training process.

Step 1 Select the training samples $A \in \mathbb{R}^{n\times r}$ and the testing sample $\boldsymbol{y} \in \mathbb{R}^n$ from the original data V such that $\boldsymbol{y}$ is not a column of A.

Step 2 Reorder the training matrix $A = [A_1, A_2, ..., A_k]$ for k classes.

Step 3 Compute the matrix of features $W_i \in \mathbb{R}^{n\times k_i}$ from each sub matrix $A_i \in \mathbb{R}^{n\times r_i}$ Using the PNMF algorithm, $\quad i = 1 : k$

Step 4 Solve the optimization problem in (2.18) for $W = [W_1, W_2, ..., W_k]$ using, for instance, the *cvx* environment in MATLAB. Let the solution $\boldsymbol{x} = [\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_k^T]^T$, where $\boldsymbol{x}_i \in \mathbb{R}^{k_i\times 1}$.

Step 5 Compute the residuals $e_i(\boldsymbol{y}) = \|\boldsymbol{y} - W\delta_i(\boldsymbol{x})\|_2$, $i = 1 : k$, where $\delta_i(\boldsymbol{x}) = [0, \cdots, 0, \boldsymbol{x}_i^T, 0, \cdots 0]^T$.

Step 6 Associate class($\boldsymbol{y}$)=arg $\min_i e_i(\boldsymbol{y})$

2.5 Application to Gene Microarrays

We apply and compare the proposed PNMF-based clustering and classification algorithms with its homologue NMF-based clustering [10] and classification as well as

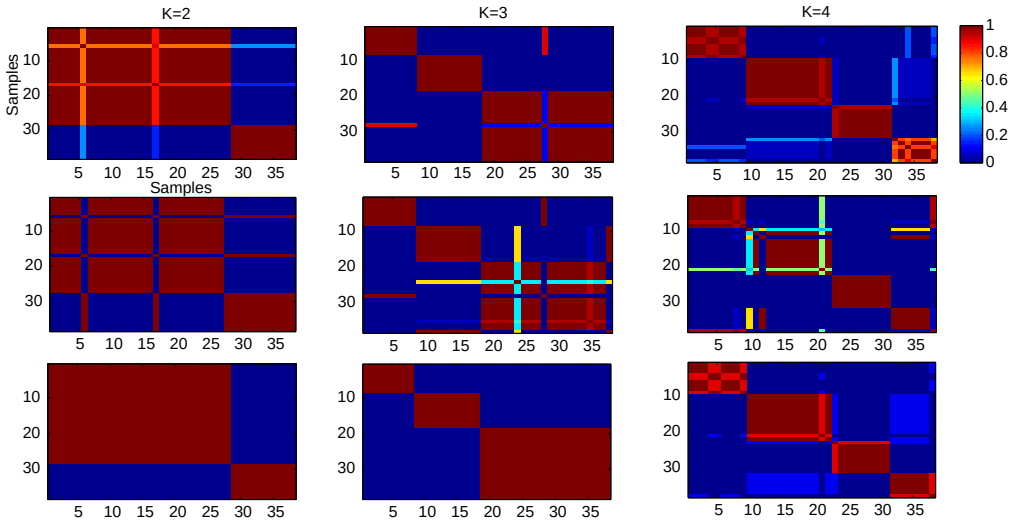


Figure 2.1: Clustering results for the Leukemia dataset: (a) Consensus matrices: Top row NMF-Euc, Second row NMF-Div, bottom row: PNMF; (b) Cophenetic coefficient versus the rank k (NMF-Euc in green, NMF-Div in red and PNMF in blue).

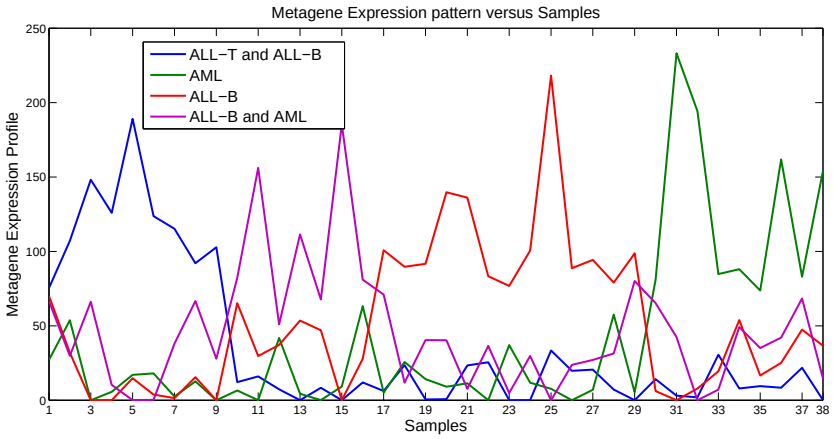
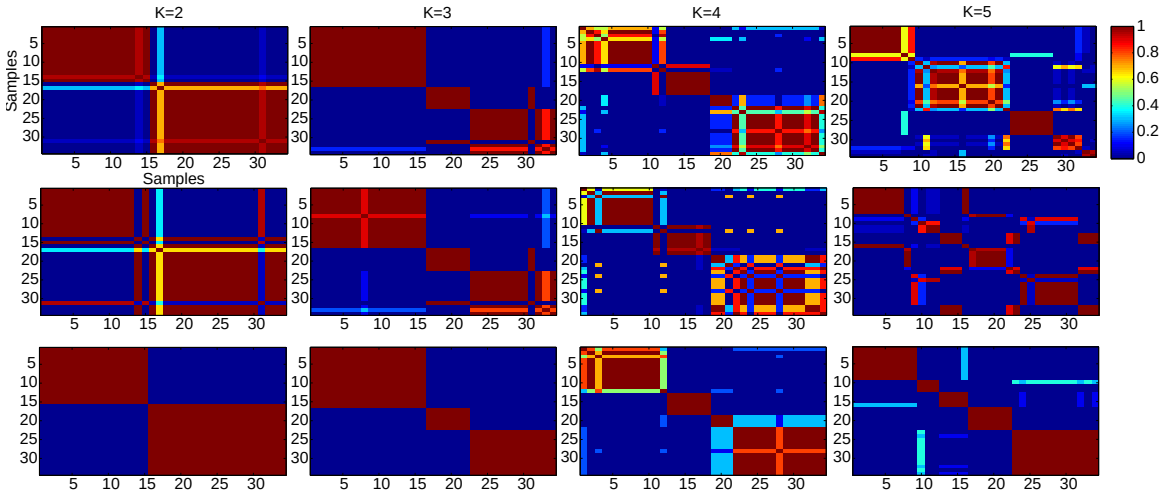


Figure 2.2: Metagenes expression patterns versus the samples for $k = 4$ in the Leukemia dataset.

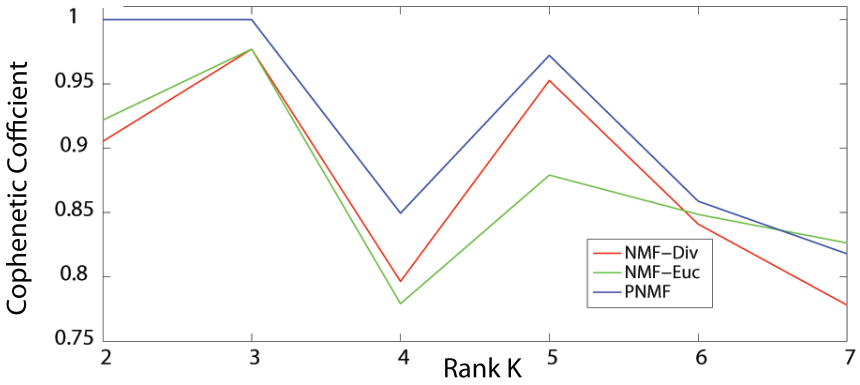
the sparse-NMF classification method presented in [64]. We first describe the gene expression dataset used and present the clustering procedure.

2.5.1 Data sets description. One of the important challenges in DNA microarrays analysis is to group genes and experiments/samples according to their similarity in gene expression patterns. Microarrays simultaneously measure the expression levels of thousands of genes in a genome. The microarray data can be represented by a gene-expression matrix $V \in \mathbb{R}^{n \times m}$, where n is the number of genes and m is the number of samples that may represent distinct tissues, experiments, or time points. The m^{th} column of V represents the expression levels of all the genes in the m^{th} sample.

We consider seven different microarray data sets: leukemia [10], medulloblastoma [10], prostate [51], colon [2], breast-colon [13], lung [7] and brain [44]. The leukemia data set is considered a benchmark in cancer clustering and classifica-



(a)



(b)

Figure 2.3: Clustering results for the Medulloblastoma dataset: (a) Consensus matrices: Top row NMF-Euc, Second row NMF-Div, bottom row: PNMF; (b) Cophenetic coefficient versus the rank k (NMF-Euc in green, NMF-Div in red and PNMF in blue).

tion [10]. The distinction between acute myelogenous leukemia (AML) and acute lymphoblastic leukemia (ALL), as well as the division of ALL into T and B cell subtypes, is well known [10]. We consider an ALL-AML dataset, which contains 5000 genes and 38 bone marrow samples (tissues from different patients for the considered genes) [10]. The considered leukemia dataset contains 19 ALL-B, 8 ALL-T and 11 AML samples.

The medulloblastoma data set is a collection of 34 childhood brain tumors samples from different patients. Each patient is represented by 5893 genes. The pathogenesis of these brain tumors is not well understood. However, two known histological subclasses can be easily differentiated under the microscope, namely, classic (C) and desmoplastic (D) medulloblastoma tumors [10]. The medulloblastoma dataset contains 25 C and 9 D childhood brain tumors.

The prostate data [51] contains the gene expression patterns from 52 prostate tumors (PR) and 50 normal prostate specimens (N), which could be used to predict common clinical and pathological phenotypes relevant to the treatment of men diagnosed with this disease. The prostate dataset contains 102 samples across 339 genes.

The colondataset [2] is obtained from 40 tumors and 22 normal colon tissue samples across 2000 genes. The breast and colon data [13] contains tissues from 62 lymph node-negative breast tumors (B) and 42 Dukes' B colon tumors (C). The lung tumor data [7] contains 17 normal lung tissues (NL), 139 adenocarcinoma (AD), 6 small-cell lung cancer (SCLC), 20 pulmonary carcinoids (COID) and 21 squamous cell lung carcinomas (SQ) samples across 12600 genes. The brain data [44] is the collection of

embryonal tumors of the central nervous system. This data includes 10 medulloblastomas (MD), 10 malignant gliomas (Mglio), 10 atypical teratoid/rhabdoid tumors (Rhab), 4 normal tissues (Ncer) and 8 primitive neuroectodermal tumors (PNET). The brain samples are measured across 1379 genes.

2.5.2 Gene expression data clustering. Applying the NMF framework to data obtained from gene expression profiles allows the grouping of genes as *metagenes* that capture latent structures in the observed data and provide significant insight into underlying biological processes and the mechanisms of disease. Typically, there are a few metagenes in the observed data that may monitor several thousands of genes. Thus, the redundancy in this application is very high, which is very profitable for NMF [14]. Assuming gene profiles can be grouped into j metagenes, V can be factored with NMF into the product of two non-negative matrices $W \in \mathbb{R}^{n\times j}$ and $H \in \mathbb{R}^{j\times m}$. Each column vector of W represents a metagene. In particular, w_{ij} denotes the contribution of the i^{th} genes into the j^{th} metagene, and h_{jm} is the expression level of the j^{th} metagene in the m^{th} sample.

Clustering performance evaluation

The position of the maximum value in each column vector of H indicates the index of the cluster to which the sample is assigned. Thus, there are j clusters of the samples. The stability of the clustering is tested by the so-called *connectivity matrix* $C \in \mathbb{R}^{m\times m}$ [10], which is a binary matrix defined as $c_{ij} = 1$ if samples i and j belong

to the same cluster, and $c_{ij} = 0$ otherwise. The connectivity matrix from each run of NMF is reordered to form a block diagonal matrix. After performing several runs, a *consensus matrix* is calculated by averaging all the connectivity matrices. The entries of the consensus matrix range between 0 and 1, and they can be interpreted as the probability that samples i and j belong to the same cluster. Moreover, if the entries of the consensus matrix were arranged so that samples belonging to the same cluster are adjacent to each other, perfect consensus matrix would translate into a block-diagonal matrix with non-overlapping blocks of 1's along the diagonal, each block corresponding to a different cluster [10]. Thus, using the consensus matrix, we could cluster the samples and also assess the performance of the number of clusters k. A quantitative measure to evaluate the stability of the clustering associated with a cluster number k was proposed in [31]. The measure is based on the correlation coefficient of the consensus matrix, ρ_k, also called the cophenetic correlation coefficient. This coefficient measures how faithfully the consensus matrix represents the similarities and dissimilarities among observations. Analytically, we have $\rho_k = \frac{1}{m^2} \sum_{ij} 4(c_{ij} - \frac{1}{2})^2$ [31]. Observe that $0 \leq \rho_k \leq 1$, and a perfect consensus matrix (all entries equal to 0 or 1) would have $\rho_k = 1$. The optimal value of k is obtained when the magnitude of the cophenetic correlation coefficient starts declining.

Clustering results

Brunet *et al.* [10] showed that the (deterministic) NMF based on the divergence cost function performs better than the NMF based on the Euclidean cost function. The

divergence cost function is defined as

$$\begin{aligned}(W^*, H^*) &= \arg\min_{W,H\geq 0} g(W,H) = \sum_{i,j}(V_{ij}\log(\frac{V_{ij}}{(WH)_{ij}}) \\ &- V_{ij} + (WH)_{ij}) \end{aligned} \tag{2.19}$$

The update rules for the divergence function are given by [33]

$$\begin{cases} H_{ij} \longleftarrow H_{ij}\frac{\sum_k (W_{ki}V_{kj})/(WH)_{kj}}{\sum_r W_{ri}} \\ \\ W_{ij} \longleftarrow W_{ij}\frac{\sum_k (H_{jk}V_{ik})/(WH)_{ik}}{\sum_r H_{jr}} \end{cases} \tag{2.20}$$

In this section, we compare the PNMF algorithm in (2.12) with both the Euclidean-based NMF in (2.5) and the divergence-based NMF in (2.19). We propose to cluster the leukemia and the medulloblastoma sample sets because the biological subclasses of these two datasets are known, and hence we can compare the performance of the algorithms with the ground truth. Figure 2.1(a) shows the consensus matrices corresponding to $k = 2, 3, 4$ clusters for the leukemia dataset. In this figure, the matrices are mapped using the gradient color so that dark blue corresponds to 0 and red to 1. We can observe the consensus matrix property that the samples' classes are laid in block-diagonal along the matrix. It is clear from this figure that the PNMF performs better than the NMF algorithm, in terms of samples' clustering. Specifically, the clusters, as identified by the PNMF algorithm, are better defined and the consensus matrices' entries are not overlapping and hence well clustered. In particular, PNMF with rank $k = 2$ correctly recovered the ALL-AML biological distinction with higher

accuracy than the deterministic NMFs (based on the Euclidean and divergence costs). Consistent clusters are also observed for rank $k = 3$, which reveal further portioning of the samples when the ALL samples are classified as the B or T subclasses. In particular, the nested structure of the blocks for $k = 3$ corresponds to the known subdivision of the ALL samples into the T and B classes. Nested and partially overlapped clusters can be interpreted with the NMF approaches. Nested clusters reflect local properties of expression patterns, and overlapping is due to global properties of multiple biological processes (selected genes can participate in many processes) [14]. An increase in the number of clusters beyond 3 ($k = 4$) results in stronger dispersion in the consensus matrix. However, Fig. 2.1(b) shows that the value of the PNMF cophenetic correlation for rank 4 is equal to 1, whereas it drops sharply for both the Euclidean and divergence-based NMF algorithms. The Hierarchal Clustering (HC) method is also able to identify four clusters [10]. These clusters can be interpreted as subdividing the samples into sub-clusters that form separate patterns within the whole set of samples as follows: {(11 ALL-B), (7 ALL-B and 1 AML), (8 ALL-T and 1 ALL-B), (10 AML)}.

Figure 2.2 depicts the metagenes expression profiles (rows of H) versus the samples for the PNMF algorithm. We can visually recognize the different four patterns that PNMF and HC are able to identify.

Figure 2.3 shows the consensus matrices and the cophenetic coefficients of the medulloblastoma dataset for $k = 2, 3, 4, 5$. The NMF and PNMF algorithms are able to identify the two known histological subclasses: classic and desmoplastic. They also predict the existence of classes for $k = 3, 5$. This clustering also stands out because

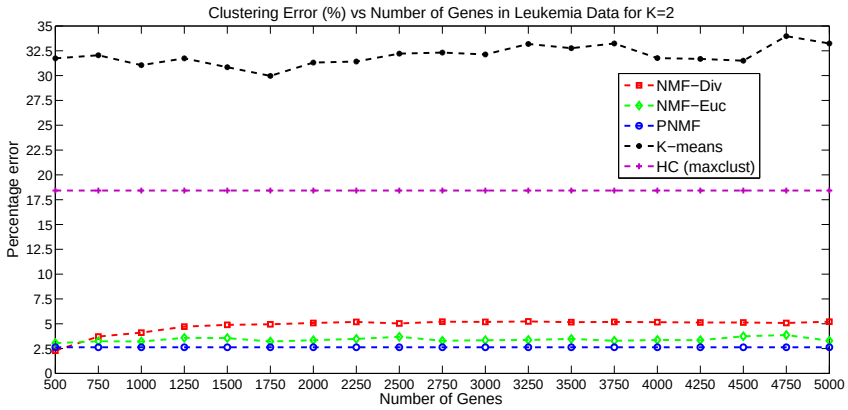


Figure 2.4: (NMF-Euc in green, NMF-Div in red and PNMF in blue, K-means in black and Hierarchical Clustering in purple) in Leukemia dataset for $k = 2$.

of the high values of the cophenetic coefficient for $k = 3, 5$ and the steep drop off for $k = 4, 6$. The sample assignments for $k = 2, 3$ and 5 display a nesting of putative medulloblastoma classes, similar to that seen in the leukemia dataset. From Fig. 2.3, we can see that the PNMF clustering is more robust, with respect to the consensus matrix and the cophenetic coefficient, than the NMF clustering. Furthermore, Brunet *et al.* [10] stated that the divergence-based NMF is able to recognize subtypes that the Euclidian version cannot identify. We also reach a similar conclusion as shown in Fig. 2.3 for $k = 3, 5$, where the Euclidian-based NMF factorization shows scattering from these structures. However, the PNMF clustering performs even better than the divergence-based NMF as shown in Figs. 2.3(a) and 2.3(b).

To confirm our results we compare our proposed PNMF algorithm with the standard NMF algorithms, distance criterion-based Hierarchical Clustering (HC) and K-means. We plot in figure 2.4 the curve Error vs. Number of genes in the labeled Leukemia data set. We select genes with small profile variance using the Bioinformatics toolbox in MATLAB from 500 to 5000 genes and the experimental points

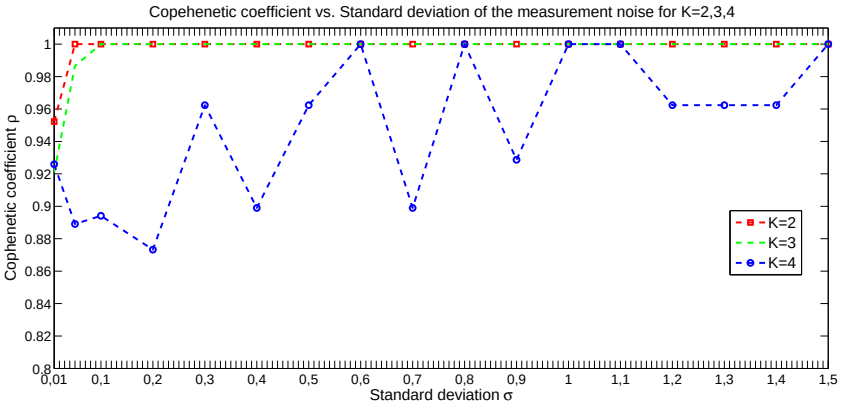


Figure 2.5: The cophenetic coefficient versus the standard deviation of the measurement noise for $k = 2$ (red), 3 (green) and 4 (blue) in the Leukemia dataset.

are equally spaced. We run 100 Monte Carlo simulation then we take the average of the error. Our simulation results show that PNMF outperforms other clustering approaches.

Robustness evaluation

In this subsection, we assess the performance of the PNMF algorithm with respect to the model parameters, especially the choice of the noise power. Recall that, in the probabilistic model, σ measures the uncertainty in the data or the noise power in the gene expression measurements. We set the prior standard deviations $\sigma_W = \sigma_H = 0.01$, and compute the cophenetic coefficient for varying values of σ between 0.01 and 1.5. Figure 2.5 shows the cophenetic coefficient versus the standard deviation σ in the leukemia data set for ranks $k = 2, 3, 4$. We observe that the PNMF is stable to a choice of σ between 0.05 and 1.5 for the ranks $k = 2$ and 3, which correspond to biologically relevant classes. In particular, when σ tends to zero, the PNMF algorithm reduces to the classic NMF, which explains the drop in the cophenetic coefficient for

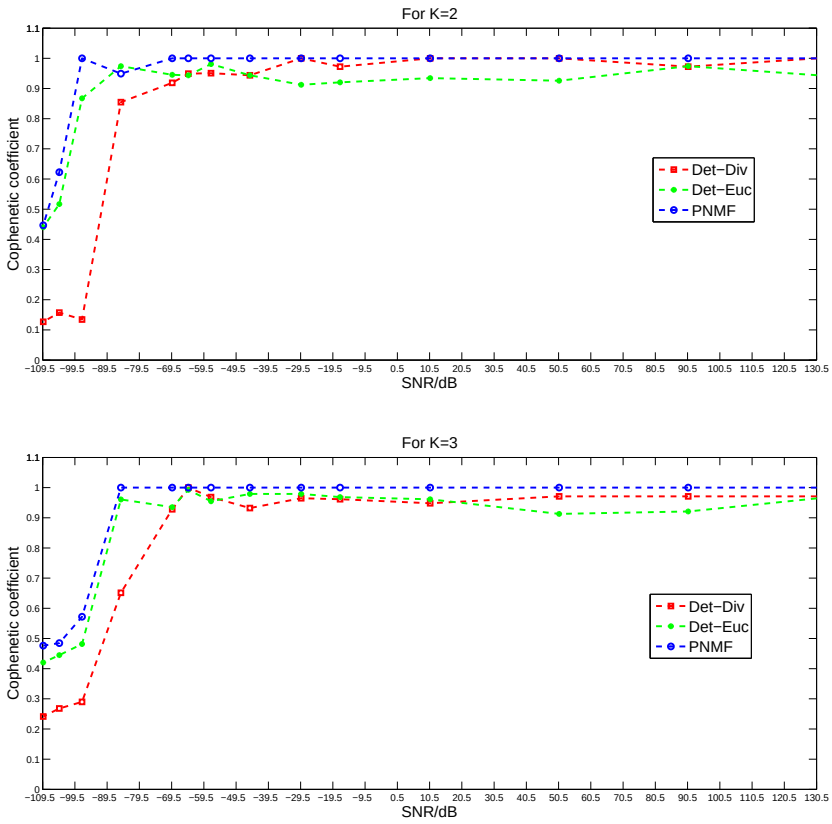


Figure 2.6: Cophenetic versus SNR in dB (NMF-Euc in green, NMF-Div in red and PNMF in blue) in Leukemia dataset for $k = 2$ and $k = 3$.

values of σ near zero.

We next study the robustness of the NMF and the proposed PNMF algorithms to the presence of noise in the data. To this end, we add white Gaussian noise, with varying power, to the leukemia dataset according to the following formula,

$$V_{noisy} = V + \sigma_n R, \tag{2.21}$$

where σ_n is the standard deviation of the noise, and R is a random matrix of the same size as the data matrix V, and whose entries are normally distributed with zero mean and unity variance. The signal to noise ratio (SNR) is, therefore, given

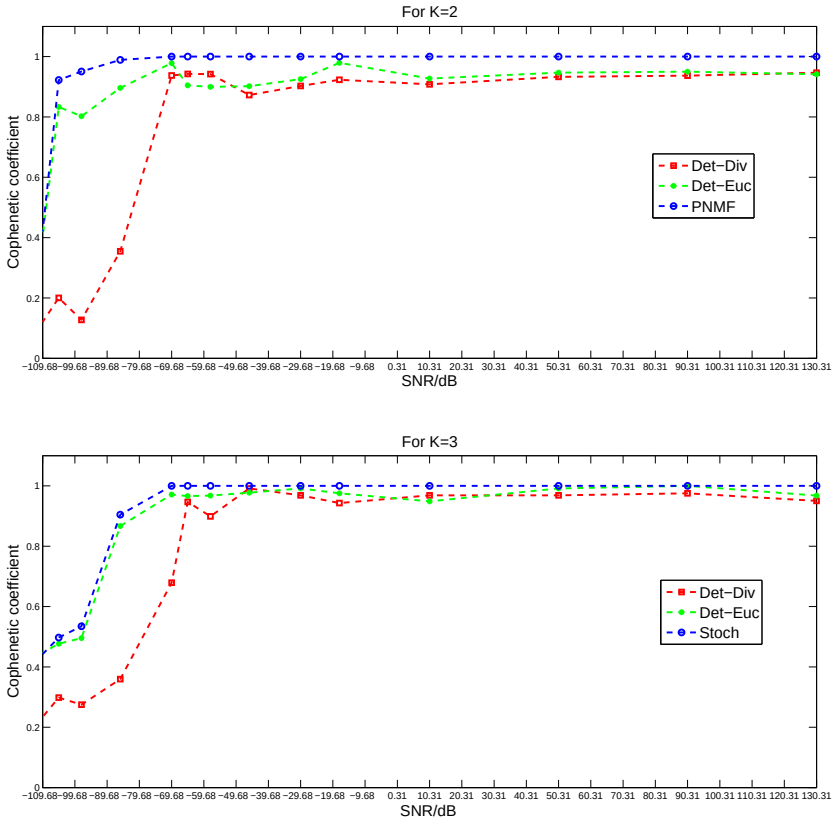


Figure 2.7: Cophenetic versus SNR in dB (NMF-Euc in green, NMF-Div in red and PNMF in blue) in Medulloblastoma dataset for $k = 2$ and $k = 3$.

Table 2.1: Smallest SNR value for which the cophenetic coefficient is higher or equal than 0.9.

Datasets	$k = 2$			$k = 3$		
	NMF-Euc	NMF-Div	PNMF	NMF-Euc	NMF-Div	PNMF
Leukemia	-93.50	-73.50	**-99.50**	-87	-71	**-88.50**
Medulloblastoma	-84.68	-70.68	**-104.68**	-86	-65.50	**-86**

by SNR $= \frac{P_V}{\sigma_n^2}$, where the signal power $P_V = \frac{1}{nm}\sum_i \sum_j v_{ij}^2 = \frac{1}{nm}\|V\|_F^2$. Since the cophenetic coefficient measures the stability of the clustering, we plot in Figures 2.6 and 2.7 the cophenetic coefficient versus the SNR, measured in dB, for both the Euclidean-based and divergence-based NMFs and PNMF algorithms using the leukemia and medulloblastoma data sets. We observe that the PNMF algorithm leads to more robust clustering than the deterministic NMF algorithms for all SNR values. Table 2.1 shows the minimum SNR values for which the cophenetic coefficient takes values higher or equal than 0.9. We say that the algorithm is "stable" for SNR values higher or equal than the minimum SNR. For the leukemia data, the Euclidean-based NMF and the divergence-based NMF algorithms stabilize respectively at $SNR = -93.5$ and $SNR = -73.5$ dB for $k = 2$, whereas the PNMF algorithm is stable at lower SNR values, $SNR = -99.5$ dB for $k = 2$. Similar results are obtained for the medulloblastoma dataset, where the NMF algorithms stabilize respectively as above at $SNR = -84.68$ and $SNR = -70.68$ dB, whereas the PNMF is stable at $SNR = -104.68$ dB. Thus, the PNMF algorithm is more stable than its deterministic homologue. Also, observe that the Euclidian-based NMF performs better than its divergence homologue for noisy data.

2.5.3 NMF-based tumor classification. Given that the proposed PNMF algorithm results in more stable clustering than its deterministic homologue, we expect that it will also lead to better feature extraction and classification. We classify the tumors in the seven gene expression datasets described in Section 2.5.1.

We assess the performance of the classification algorithm using the 10-fold cross-validation technique [64]. The number of metagenes k_i can be determined using the nested stratified 10-fold cross-validation. However, we follow the work in [64] and choose $k_i = 8$ if the number of samples in the i^{th} class $r_i > 8$. Otherwise we choose $k_i = r_i$. We selected the parameters α and β of PNMF in order to minimize the classification error in the training dataset based on a 10-fold cross-validation technique. The parameters of SNMF were selected using the same criterion and method, i.e. minimize the classification error in the training dataset. The classification results for the NMF, PNMF, SVM and SNMF [64] algorithms are summarized in Table 2.2. In particular, we compared the PNMF-based MSRC algorithm to the SVM algorithm which has been shown to outperform K-NN and neural network in tumor classification [52], [43]. In our experiment we use one-versus-rest SVM (OVRSVM) with Polynomial kernels approach which has been shown to be the best one [52]. The results can be obtained using the Gene Expression Model Selector (GEMS) publicly available online `http://www.gems-system.org/`. Observe that the PNMF-based classifier performs better than the other approaches for the considered data sets except for the prostate data where SVM achieves the highest classification accuracy. Moreover, the PNMF performs better than the SNMF for the prostate, lung and brain data sets. This

Table 2.2: Classification accuracy

Data sets	*Nbr.* of classes	NMF-Euc	NMF-Div	SNMF	SVM	PNMF
Prostate	2	85.29%	86.27%	88.24%	99%	**92.16%**
Medulloblastoma	2	85.29%	91.18%	94.12%	79.16%	**94.12%**
Colon	2	85.48%	88.71%	90.32%	89.04%	**90.32%**
Breast-Colon	2	98.08%	95.19%	98.08%	84.63%	**98.08%**
Leukemia	3	97.37%	97.37%	97.37%	95.50%	**97.37%**
Lung	5	92.61%	90.64%	93.60%	85.54%	**94.09%**
Brain	5	76.19%	78.57%	83.33%	77%	**85.71%**

is due to the high accuracy of the PNMF in feature extraction as compared to the SNMF algorithm, which is not guaranteed to converge to the optimal non-negative factorization [64].

2.6 Conclusion and Discussion

Studying and analyzing tumor profiles is a very relevant area in computational biology. Clinical applications include clustering and classification of gene expression profiles. In this work, we developed a new mathematical framework for clustering and classification based on the Probabilistic Non-negative Matrix Factorization (PNMF) method. We presented an extension of the deterministic NMF algorithm to the probabilistic case. The proposed PNMF algorithm takes into account the stochastic nature of the data due to the inherent presence of noise in the measurements as well as the internal biological variability. We subsequently casted the optimal non-negative probabilistic factorization as a weighted regularized matrix factorization problem. We derived updates rules and showed convergence towards the optimal non-negative factors. The derived update rules generalize Lee and Seung's multiplicative update

rules for the NMF algorithm. We have also generalized Lee and Seung's algorithm to include a general class of update rules, which converge towards a stationary point of the (deterministic) NMF problem. We next derived a PNMF-based classifier, which relies on the PNMF factorization to extract features and classify the samples in the data. The PNMF-based clustering and classification algorithms were applied to seven microarray gene expression datasets. In particular, the PNMF-based clustering was able to identify biologically significant classes and subclasses of tumor samples in the leukemia and medulloblastoma datasets. Moreover, the PNMF clustering results were more stable and robust to data corrupted by noise than the classic (deterministic) NMF.

Thanks to its high stability, robustness to noise and convergence properties, the PNMF algorithm yielded better tumor classification results than the NMF and the Sparse NMF (SNMF) algorithms. The proposed PNMF framework and algorithm can be further applied to many other relevant applications in biomedical data processing and analysis, including muscle identification in the nervous system, image classification, and protein fold recognition.

Chapter 3
SMURC: High-Dimension Small-Sample Multivariate Regression with Covariance Estimation

3.1 Introduction

Many engineering problems are formulated as an inverse problem. Examples in signal processing include source estimation of electroencephalographic (EEG) and magnetoencephalographic (MEG) data and inference or reverse-engineering of genetic regulatory networks from high-throughput gene expression data. These problems are sometimes referred to as *ill-posed* or *ill-defined* because the inverse problem has no unique solution, and there are infinitely many solutions that are equally compatible with the data. For instance, in EEG and MEG source estimation problems, if the source distribution contains more independent parameters than there is independent information in the recorded data, then the sources spatial distribution cannot be estimated. In genomics, the inference of genetic regulatory networks also suffers from the limited number of measurements available to unambiguously estimate the network connectivity. This problem, known as the "large p small n" problem, poses a challenge in estimation due to the identifiability problem, where a large class of solutions is consistent with the measurements and no unique solution exists.

The approaches proposed in the literature to tackle inverse problems can be classified into three groups: (1) the statistical approach, which finds the most likely solution that fits the data and any additional constraints that may be imposed; (2) the minimum norm approach, which finds a solution that is compatible with the data and satisfies additional constraints, e.g., on the amplitudes or covariances of the parameters; (3) the resolution optimization methods, which estimate the parameters as independently as possible from each other. It has been shown in [26] that all these approaches result in the same solution given the same a priori information. Moreover, if no a priori information is available, all three methods are equivalent to the classical minimum norm solution [26].

Let us consider the (under-determined) multivariate regression problem, which generalizes the classical regression problem of one response on p predictors to regressing q responses on p predictors. This model has various applications including genomics [42], neurology [35], imaging [35] and econometrics. Let $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ip})$ denote the predictors, $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{iq})$ denote the responses, and $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \cdots, \epsilon_{iq})$ the errors for the i^{th} sample. The multivariate regression model is given by

$$\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \cdots, n, \tag{3.1}$$

where $\boldsymbol{A}$ is a $q \times p$ regression matrix and n is the sample size. We make the standard assumption that $\boldsymbol{\epsilon}_1, \cdots, \boldsymbol{\epsilon}_n$ are i.i.d Gaussian with zero mean and covariance matrix

$\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. The model in (3.1) can be expressed in matrix notation as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}, \tag{3.2}$$

where $\boldsymbol{Y}$ is the $q \times n$ response matrix with its i^{th} column $\boldsymbol{y}_i$, $\boldsymbol{X}$ is the $p \times n$ predictor matrix with its i^{th} column $\boldsymbol{x}_i$ and $\boldsymbol{E}$ is the random error matrix. $\boldsymbol{X}$ is assumed to be full-rank. The system is under-determined when there are more parameters than samples, i,e, $q > p > n$.

The negative log-likelihood function of $(\boldsymbol{A}, \boldsymbol{\Omega})$, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, can be expressed up to a constant as,

$$g(\boldsymbol{A}, \boldsymbol{\Omega}) = \operatorname{tr}\left[\frac{1}{n}(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})^t \boldsymbol{\Omega}(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})\right] - \log|\boldsymbol{\Omega}|, \tag{3.3}$$

where tr denotes the trace operator. If $p \leq n$ (complete or over-determined system), the maximum likelihood estimator for $\boldsymbol{A}$ is simply given by $\hat{\boldsymbol{A}}^{\text{OLS}} = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$, which is independent of $\boldsymbol{\Omega}$ and amounts to performing q separate ordinary least-squares.

The multivariate regression problem becomes particularly challenging when the system is under-determined as it requires the estimation of pq parameters from $nq < qp$ predictors or $n < p$. Different approaches were proposed to reduce the number of parameters by minimizing (3.3) under various constraints on the regression matrix $\boldsymbol{A}$. Reduced-rank approaches restrict the rank of the estimated matrix of regression coefficients, $\operatorname{rank}(\boldsymbol{A}) \leq r \leq \min(p, q)$ [46]. The rank can also be reduced by imposing

a sparsity constraint on the singular values of $\boldsymbol{A}$ [61]. Sparsity can also be imposed to identify the main predictors [42], where a combined constraint function that includes l_1 and l_2 regularization, is used [40]. The l_1 constraint introduces sparsity in the entries of $\boldsymbol{A}$ and the l_2 regularization identifies irrelevant predictors (for all q responses) by introducing zeros for all entries in some rows of $\boldsymbol{A}$. However, all of these approaches do not account for correlated responses.

Exploiting the correlation in the response variables improves the prediction performance. For under-determined problems, however, the maximum likelihood (ML) approach with covariance estimation is senseless because there exist solutions satisfying $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ and $\boldsymbol{\Sigma}$ infinitely small. For these solutions, the negative log-likelihood in (3.3) tends to $-\infty$. Hence, the likelihood, as a function of the two variables $(\boldsymbol{A}, \boldsymbol{\Omega})$, diverges. Observe that the likelihood converges if the covariance matrix $\boldsymbol{\Sigma}$ is known (e.g., proportional to the Identity for uncorrelated measurements) or if the system is over-determined (in this case, there exists no solution that satisfies $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$).

Rothman *et al.* [48] proposed a regularized algorithm that simultaneously infers the regression coefficient matrix $\boldsymbol{A}$ and the inverse error covariance, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, by imposing sparsity constraints on $\boldsymbol{\Omega}$. The l_1-norm penalty on $\boldsymbol{\Omega}$ ensures the convergence of the regularized likelihood because it excludes exact solutions, for which the covariance is infinitely small or equivalently the inverse covariance is infinitely large. However, in many applications, the assumption of a sparse inverse covariance matrix may not be reasonable or have any physical justification. In particular, in the genetic regulatory network problem, there is no evidence for such an assumption. Moreover, the solution to the regularized problem in [48] relies on an iterative procedure that

finds the maximum over $\boldsymbol{A}$ then over $\boldsymbol{\Omega}$. That is because the problem is convex in each variable, $\boldsymbol{A}$ and $\boldsymbol{\Omega}$, but not convex in the pair $(\boldsymbol{A}, \boldsymbol{\Omega})$. This iterative procedure is not guaranteed to converge and if it does converge, then it may not reach the optimal solution. Additionally, the authors observed that this algorithm may take many iterations to converge for high-dimensional data. Subsequently, they proposed an approximate MRCE approach that prematurely terminates the iterative optimization procedure after two iterations.

Recently, Zhang *et al.* [62] proposed the sparse Conditional Gaussian Graphical Model (sCGGM). CGGM formulates the inference problem as a joint probabilistic graphical model. sCGGM minimizes the negative log-likelihood of the data with l_1 penalties on the autocorrelation and cross-correlation precision matrices [62]. The main advantage of CGGM over MRCE is that CGGM leads to a convex problem, whereas the MRCE estimation problem is only bi-convex, not jointly convex. However, as acknowledged by the authors, CGGM and MRCE are so similar that "MRCE was mistakenly called a sparse CGGM" [62]. In essence, both algorithms solve an under-determined linear regression problem by maximizing the Gaussian likelihood subject to sparse constraints on the correlation structure. Hence, the open question remains: "How can we perform maximum likelihood with covariance estimation for under-determined systems?"

The work that we present in this chapter addresses this question, namely the problem of ML estimation with unknown covariance in under-determined systems. We present a normalization of the likelihood function that guarantees convergence while still keeping the exponential form of the distribution.

In this chapter, scalars are denoted by lower case letters, e.g., n, m; vectors are denoted by bold lower case letters, e.g., $\boldsymbol{x}, \boldsymbol{y}$; and matrices are referred to by bold upper case letters, e.g., $\boldsymbol{A}, \boldsymbol{X}$. $\boldsymbol{I}$ denotes the identity matrix. $\boldsymbol{x}_i$ denotes the i^{th} element of vector $\boldsymbol{x}$ and a_{ij} is the $(i,j)^{th}$ entry of matrix $\boldsymbol{A}$. Throughout the chapter, we provide references to known results and limit the presentation of proofs to new contributions.

3.2 The Normalized-Likelihood

We propose to weight the likelihood function by the "energy" of the error, in order to guarantee the convergence of the energy-weighted likelihood function, while still keeping the exponential form of the density. Specifically, we define the normalized-likelihood of the under-determined ($p > n$) multiple regression model in (3.2), under the Gaussian assumption, as

Definition 3.2.1.

$$L_N(\boldsymbol{A}, \boldsymbol{\Omega}) = \frac{|(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})|^{\frac{n}{2}}}{(2\pi)^{\frac{np}{2}}} \exp[-\frac{1}{2} \operatorname{Tr}[(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})]], \tag{3.4}$$

where $|\cdot|$ is the matrix determinant operator.

Obviously, one can propose many possible normalizations of the Gaussian likelihood as a function of the pair $(\boldsymbol{A}, \boldsymbol{\Omega})$. Our particular "choice" in Definition 3.2.1 is motivated by finding a function that ensures a finite maximum of the likelihood

while keeping the form of the Gaussian density. This normalization of the Gaussian likelihood avoids exact solutions and subsequent divergence issues. The pair $(\boldsymbol{A}, \boldsymbol{\Omega})$ can then be computed to maximize the normalized-likelihood, L_N, i.e.,

$$(\boldsymbol{A}^*, \boldsymbol{\Omega}^*) = \arg\max_{\boldsymbol{A},\boldsymbol{\Omega}} L_N(\boldsymbol{A}, \boldsymbol{\Omega}), \tag{3.5}$$

Proposition 3. *The solution to (3.5) is given by*

$$(\boldsymbol{Y} - \boldsymbol{A}^*\boldsymbol{X})^T \boldsymbol{\Omega}^* (\boldsymbol{Y} - \boldsymbol{A}^*\boldsymbol{X}) = n\boldsymbol{I}, \tag{3.6}$$

where $\boldsymbol{I}$ denotes the $n \times n$ Identity matrix.

Proof of Proposition 3. Let $\boldsymbol{Z} = (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})$. Then, the normalized-likelihood can be written as the following function of the variable $\boldsymbol{Z}$,

$$L_N(\boldsymbol{Z}) = \frac{|\boldsymbol{Z}|^{\frac{n}{2}}}{(2\pi)^{\frac{nq}{2}}} \exp -\frac{1}{2} \operatorname{Tr}[\boldsymbol{Z}]. \tag{3.7}$$

To find the stationary point $\boldsymbol{Z}^*$, we set $\dfrac{\partial L_N(\boldsymbol{Z})}{\partial \boldsymbol{Z}} = \boldsymbol{0}$.

$$\begin{aligned}
\frac{\partial L_N(\boldsymbol{Z})}{\partial \boldsymbol{Z}} &= \frac{n}{2}|\boldsymbol{Z}|^{\frac{n}{2}-1}|\boldsymbol{Z}|\boldsymbol{Z}^{-1} \exp -\frac{1}{2}\operatorname{Tr}[\boldsymbol{Z}] \\
&- \frac{1}{2}|\boldsymbol{Z}|^{\frac{n}{2}} \exp -\frac{1}{2}\operatorname{Tr}[\boldsymbol{Z}] \\
&= \frac{1}{2}|\boldsymbol{Z}|^{\frac{n}{2}}[n\boldsymbol{Z}^{-1} - \boldsymbol{I}] \exp -\frac{1}{2}\operatorname{Tr}[\boldsymbol{Z}] \\
&= \boldsymbol{0} \\
\Rightarrow \quad & \boldsymbol{Z}^* = n\boldsymbol{I}.
\end{aligned} \tag{3.8}$$

Moreover, it can be easily derived that the Hessian at the stationary point $\boldsymbol{Z}^*$ is given by

$$\frac{\partial^2 L_N(\boldsymbol{Z})}{\partial \boldsymbol{Z}^2} \mid_{\boldsymbol{Z}=\boldsymbol{Z}^*} = -\frac{1}{2n} n^{\frac{n^2}{2}} e^{-\frac{n^2}{2}} < 0 \tag{3.9}$$

□

There are many pairs $(\boldsymbol{A}^*, \boldsymbol{\Omega}^*)$, which satisfy equality (3.6) and hence maximize the normalized-likelihood. The non-uniqueness of the solution is not surprising given that the problem is under-determined. Among all possible solutions of (3.6), we propose to find those that minimize the regularized error $\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{\Omega}\|_F^2$, where λ is a tuning parameter and $\|\cdot\|_F$ denotes the Frobenius norm. Observe that it is meaningful to consider the error as the objective function here, because the set of pairs $(\boldsymbol{A}, \boldsymbol{\Omega})$ satisfying (3.6) are not exact solutions, i.e., they do not satisfy the equality $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, and hence the minimum error is not trivially zero. Thus, an advantage of the normalized-likelihood is that it avoids considering exact solutions. In addition, we consider constraints on the regression matrix $\boldsymbol{A}$, which reflect prior knowledge about the nature of the regression model. For instance, $\boldsymbol{A}$ may be constrained to be sparse. Many applications assume a sparse regression matrix, e.g., robust face recognition, where the target can be represented as a sparse linear combination of the dataset [58] and structural equation models (SEM) to infer gene or phenotype networks [11]. For now, let us consider a general constraint set, $\boldsymbol{A} \in \mathcal{A} \subset \mathbb{R}^{q \times p}$. The

constrained optimization problem, thus, becomes

$$\begin{cases} \min\limits_{(\boldsymbol{A},\boldsymbol{\Omega})} \|\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X}\|_F^2+\lambda\|\boldsymbol{\Omega}\|_F^2 \\ \text{s.t.} \quad (\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})^T\boldsymbol{\Omega}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})=n\boldsymbol{I}, \\ \boldsymbol{A}\in\mathcal{A}. \end{cases} \tag{3.10}$$

Problem (3.10) is formulated in terms of the two coupled variables $\boldsymbol{A}$ and $\boldsymbol{\Omega}$, which satisfy (3.6) to maximize the normalized-likelihood function. The following lemma derives an analytical expression of $\boldsymbol{\Omega}$ as a function of $\boldsymbol{A}$, and hence reduces the problem to depend on only one variable $\boldsymbol{A}$. Before stating the lemma's result, we need the following definition of the polar decomposition of matrices.

Definition 3.2.2. *The polar decomposition of a matrix $\boldsymbol{B}\in\mathbb{C}^{p\times n}$ is given by*

$$\boldsymbol{B}=\boldsymbol{U}|\boldsymbol{B}|, \tag{3.11}$$

where $|\boldsymbol{B}|=(\boldsymbol{B}^T\boldsymbol{B})^{1/2}$, $(\cdot)^{1/2}$ is the principal square root operator and $\boldsymbol{U}:\mathbb{C}^n\longrightarrow$ Range$(\boldsymbol{B})$ is a $\mathbb{C}^{p\times n}$ isometry such that $\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}$.

Lemma 3.2.3. *Given $\boldsymbol{A}$, there exist many $\boldsymbol{\Omega}$ satisfying equality (3.6). The minimum Frobenius norm $\boldsymbol{\Omega}$, for a fixed $\boldsymbol{A}$, is given by*

$$\boldsymbol{\Omega}_A=n\,\boldsymbol{U}\left[(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})^T(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})\right]^{-1}\boldsymbol{U}^T, \tag{3.12}$$

where $\boldsymbol{U}$ is the isometry of the matrix $(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})$.

Proof of Lemma 3.2.3. Let $\boldsymbol{B} = (\boldsymbol{Y} - \boldsymbol{AX})$. Consider the polar decomposition of $\boldsymbol{B}$ given by

$$\boldsymbol{B} = \boldsymbol{U}|\boldsymbol{B}|, \quad \text{and} \quad |\boldsymbol{B}| = (\boldsymbol{B}^T\boldsymbol{B})^{1/2}. \tag{3.13}$$

Then, the equality in (3.6) becomes

$$\begin{aligned} & (\boldsymbol{Y} - \boldsymbol{AX})^T\boldsymbol{\Omega}(\boldsymbol{Y} - \boldsymbol{AX}) = n\boldsymbol{I} \\ \iff \quad & \boldsymbol{B}^T\boldsymbol{\Omega}\boldsymbol{B} = n\boldsymbol{I} \\ \iff \quad & |\boldsymbol{B}|\boldsymbol{U}^T\boldsymbol{\Omega}\boldsymbol{U}|\boldsymbol{B}| = n\boldsymbol{I} \\ \iff \quad & \boldsymbol{U}^T\boldsymbol{\Omega}\boldsymbol{U} = n|\boldsymbol{B}|^{-2} \end{aligned} \tag{3.14}$$

Since $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{U}^T$ restricted to the range of $\boldsymbol{B}$ is invertible, i.e., $\boldsymbol{U}^T \restriction_{\text{Range}(\boldsymbol{B})}$ is invertible. Let us write

$$\mathbb{C}^q = \text{Range}(\boldsymbol{B}) \oplus \text{Ker}(\boldsymbol{B}^T), \tag{3.15}$$

where $\oplus$ denotes the direct sum of the two subspaces Range($\boldsymbol{B}$) and Ker($\boldsymbol{B}^T$). Let $\boldsymbol{P}_B$ be the orthogonal projection onto Range($\boldsymbol{B}$). Then, we can decompose $\boldsymbol{\Omega}$ as

$$\boldsymbol{\Omega} = \boldsymbol{P}_B\Omega\boldsymbol{P}_B \oplus \boldsymbol{P}_B\Omega\boldsymbol{P}_{B^\perp} \oplus \boldsymbol{P}_{B^\perp}\Omega\boldsymbol{P}_B \oplus \boldsymbol{P}_{B^\perp}\Omega\boldsymbol{P}_{B^\perp}, \tag{3.16}$$

where $\boldsymbol{P}_B^\perp$ is the orthogonal projection onto the orthogonal space of Range($\boldsymbol{B}$), i.e., Ker($\boldsymbol{B}^T$). Recall that, by definition of the isometry $\boldsymbol{U}$, it satisfies the following

properties:

$$\boldsymbol{P}_{B^\perp}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{P}_{B^\perp} = \mathbf{0}. \tag{3.17}$$

Thus, from the decomposition of the matrix Ω in Eq. (3.16), we obtain

$$\boldsymbol{U}^T\boldsymbol{\Omega}\boldsymbol{U} \;\; = \;\; \boldsymbol{U}^T\boldsymbol{P}_B\;\boldsymbol{\Omega}\;\boldsymbol{P}_B\boldsymbol{U}. \tag{3.18}$$

From Eq. (3.14) and since $\boldsymbol{U}^T\restriction_{\mathrm{Range}(\boldsymbol{B})}$ is invertible, we have

$$\boldsymbol{U}^T\boldsymbol{\Omega}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{P}_B\;\boldsymbol{\Omega}\;\boldsymbol{P}_B\boldsymbol{U} = n|\boldsymbol{B}|^{-2}$$

$$\Longleftrightarrow$$

$$\boldsymbol{P}_B\;\boldsymbol{\Omega}\;\boldsymbol{P}_B = n\;\boldsymbol{U}|\boldsymbol{B}|^{-2}\boldsymbol{U}^T. \tag{3.19}$$

From the matrix decomposition in (3.16), for a fixed $\boldsymbol{A}$, $\boldsymbol{P}_B\;\boldsymbol{\Omega}\;\boldsymbol{P}_B$ is fixed. Thus, the minimum Frobenius norm matrix $\boldsymbol{\Omega}$ results by setting the three other terms in the matrix decomposition to zero, i.e., the minimum Frobenius norm matrix is of the form

$$\boldsymbol{\Omega} = \boldsymbol{P}_B\;\boldsymbol{\Omega}\;\boldsymbol{P}_B. \tag{3.20}$$

The result of Lemma 3.2.3 then follows from Eqs. (3.19) and (3.20). □

Using Lemma 3.2.3, the following proposition states the equivalent form of problem (3.10), where the optimization problem does not depend on the variable $\boldsymbol{\Omega}$.

Proposition 4. *The optimization problem in (3.10) is equivalent to*

$$\begin{cases} \min\limits_{\boldsymbol{S}} \operatorname{Tr}(\boldsymbol{S}^2) + \lambda\ n^2 \operatorname{Tr}(\boldsymbol{S}^{-4}) \\ \\ s.t. \quad \boldsymbol{S} = |\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}|,\ \boldsymbol{A} \in \mathcal{A} \end{cases} \tag{3.21}$$

Proof of Proposition 4. Replacing $\boldsymbol{\Omega_A}$ in the objective function of the optimization problem (3.10) by its expression obtained in Lemma 3.2.3, and letting $\boldsymbol{B} = \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}$, we obtain

$$\begin{aligned} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{\Omega}\|_F^2 &= \|\boldsymbol{B}\|_F^2 + \lambda\|n^2\boldsymbol{U}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{U}^T\|_F^2 \\ &= \operatorname{Tr}(\boldsymbol{B}^T\boldsymbol{B}) + \lambda n^2 \\ &\quad \operatorname{Tr}(\boldsymbol{U}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{U}^T\boldsymbol{U}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{U}^T) \\ &= \operatorname{Tr}(\boldsymbol{B}^T\boldsymbol{B}) + \lambda n^2 \operatorname{Tr}((\boldsymbol{B}^T\boldsymbol{B})^{-2}) \\ &= \operatorname{Tr}(\boldsymbol{S}^2) + \lambda n^2 \operatorname{Tr}(\boldsymbol{S}^{-4}), \end{aligned} \tag{3.22}$$

where $\boldsymbol{S}^2 = \boldsymbol{B}^T\boldsymbol{B} = (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})^T(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})$. □

Though the objective function in (3.21) is convex (as a function of the variable $\boldsymbol{S}$), the equality in the constraint (assuming that $\mathcal{A}$ is convex) is not affine and thus the optimization problem (3.21) is not convex [9]. We will, therefore, relax the minimization of (3.21) to a minimization over a convex set that is included in the original set. In what follows, we show that if the matrix regression $\boldsymbol{A}$ is sparse with a bounded norm, i.e., $\mathcal{A} = \{\boldsymbol{A} : \|A\|_1 \leq \epsilon\}$, then (3.21) can be approximated by

a convex optimization problem. Moreover, this approximation formulates a much simpler optimization problem than the initial setting in (3.21) because it depends only on $\boldsymbol{S}$ and is independent of $\boldsymbol{A}$.

Proposition 5. *If $\mathcal{A} = \{\boldsymbol{A} : \|A\|_1 \leq \epsilon\}$, then the optimization problem in (3.21) can be approximated by the following convex optimization problem*

$$\begin{cases} \min\limits_{\boldsymbol{S}} \operatorname{Tr}(\boldsymbol{S}^2) + \lambda\ n^2 \operatorname{Tr}(\boldsymbol{S}^{-4}) \\ \\ s.t. \quad \boldsymbol{S} \in \Lambda = \{\boldsymbol{S} \in \mathbb{S}_{n,n}\ :\ \|\boldsymbol{S} - |\boldsymbol{Y}|\|_F \leq \epsilon c^*\} \end{cases} \tag{3.23}$$

where $\mathbb{S}_{n,n}$ is the set of $n \times n$ symmetric positive semi-definite matrices and c^* is a small term which depends on X, Y but independent of ϵ.

Proof of Proposition 5. Let

$$\mathcal{S}_1 = \{\boldsymbol{S} : \boldsymbol{S} = |\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}|, \|\boldsymbol{A}\|_1 \leq \epsilon\}. \tag{3.24}$$

and let

$$\mathcal{S}_2 = \{\boldsymbol{S} \in \mathbb{S}_{n,n} : \|\boldsymbol{S} - |\boldsymbol{Y}|\|_F \leq \epsilon c^*\}. \tag{3.25}$$

We will show that $\mathcal{S}_2 \subseteq \mathcal{S}_1$. An illustration of these two sets is provided in Fig. 3.1. To this aim, we consider $\boldsymbol{S} \in \mathcal{S}_2$ and show that $\boldsymbol{S} \in \mathcal{S}_1$. Specifically, given $\boldsymbol{S} \in \mathcal{S}_2$ we find $\boldsymbol{A}$, such that $\boldsymbol{S} = |\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}|$ and $\|\boldsymbol{A}\|_1 \leq \epsilon$.

Given $\boldsymbol{S} \in \mathbb{S}_{n,n}$, we want to find $\boldsymbol{A}$ such that $\boldsymbol{S} = |\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}|$, i.e., for some isometry $\boldsymbol{U}$ we have $\boldsymbol{U}\boldsymbol{S} = \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}$. For every isometry $\boldsymbol{U}$, one can find corresponding

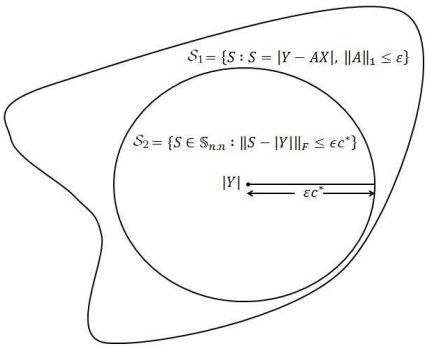


Figure 3.1: Approximation of the optimization problem in Proposition 4 by the convex optimization problem in Proposition 5. Illustration of the sets $\mathcal{S}_1$ and $\mathcal{S}_2$ in the proof of Proposition 5.

matrix $\boldsymbol{A}$ satisfying the previous identity. We will construct a specific matrix $\boldsymbol{A}$. Namely, we fix $\boldsymbol{U} = \boldsymbol{V}$, where $\boldsymbol{V}$ is the isometry from the polar decomposition $\boldsymbol{Y} = \boldsymbol{V}|\boldsymbol{Y}|$. Then, we need to find $\boldsymbol{A}$ such that

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{V}(|\boldsymbol{Y}| - \boldsymbol{S}). \tag{3.26}$$

$\boldsymbol{X}$ is full-rank; hence invertible from the right. Let us define

$$\tilde{\boldsymbol{X}} = \begin{cases} \boldsymbol{X}^{-1}\big|_{Range(X)}, \\ \boldsymbol{0}\big|_{[Range(X)]^\perp} \end{cases} \tag{3.27}$$

From the Definition of $\tilde{\boldsymbol{X}}$, we have $\boldsymbol{A}\boldsymbol{X}\tilde{\boldsymbol{X}}\Big|_{Range(X)} = \boldsymbol{A}\big|_{Range(X)}$ and $\boldsymbol{A}\boldsymbol{X}\tilde{\boldsymbol{X}}\Big|_{Range(X)^\perp} = \boldsymbol{0}$. Therefore, multiplying Eq. (3.26) to the right by $\tilde{\boldsymbol{X}}$, we see that $\boldsymbol{A}$ defined by

$$\boldsymbol{A} = \begin{cases} \boldsymbol{V}(|\boldsymbol{Y}| - \boldsymbol{S})\tilde{\boldsymbol{X}}\Big|_{Range(X)}, \\ \boldsymbol{0}\big|_{[Range(X)]^\perp} \end{cases} \tag{3.28}$$

solves Eq. (3.26). Now we estimate $\|A\|_1$,

$$\begin{aligned}
\|\boldsymbol{A}\|_1 &\leq \|\boldsymbol{V}\|(\||\boldsymbol{Y}| - \boldsymbol{S}\|_1)\|\tilde{\boldsymbol{X}}\| \\
&\leq n\|\boldsymbol{V}\|(\||\boldsymbol{Y}| - \boldsymbol{S}\|_F)\|\tilde{\boldsymbol{X}}\| \qquad (3.29) \\
&= C'\||\boldsymbol{Y}| - \boldsymbol{S}\|_F \qquad (3.30) \\
&\leq C'\epsilon c^* \qquad (3.31)
\end{aligned}$$

where (3.29) follows from the equivalence of norms and Cauchy-Schwartz. In (3.30), $C' = n\|\boldsymbol{V}\|\|\tilde{\boldsymbol{X}}\|$, which is a constant. The inequality in (3.31) follows from the fact that $\boldsymbol{S} \in \mathcal{S}_2$ and $\|\boldsymbol{S} - |\boldsymbol{Y}|\|_F \leq \epsilon c^*$. In (3.31), by choosing $c^* \leq \frac{1}{C'} = 1/(n\|\boldsymbol{V}\|\|\tilde{\boldsymbol{X}}\|)$, we obtain $\boldsymbol{A} \leq \epsilon$. This ends the proof that $\boldsymbol{S} \in \mathcal{S}_1$. □

The optimization problem (3.23) is convex, hence it admits a unique global solution $\boldsymbol{S}^*$. Given $\boldsymbol{S}^*$, the optimal regression matrix, $\hat{\boldsymbol{A}}$, is found by solving $\boldsymbol{S}^* = |\boldsymbol{Y} - \hat{\boldsymbol{A}}\boldsymbol{X}|$. There are many possible such solutions $\hat{\boldsymbol{A}}$. We propose to find the sparsest matrix, in the sense of minimization of the l_1-norm.

$$\begin{cases} \min\limits_{\boldsymbol{A},\boldsymbol{U}} \|\boldsymbol{A}\|_1 \\ \\ \text{s.t.} \quad \boldsymbol{A}\boldsymbol{X} = \boldsymbol{Y} - \boldsymbol{U}\boldsymbol{S}^*, \end{cases} \qquad (3.32)$$

where $\boldsymbol{U}$ is an isometry matrix. For every isometry $\boldsymbol{U}_0$, we can find the minimum l_1-norm $\boldsymbol{A}(\boldsymbol{U}_0)$. The optimal matrix $\boldsymbol{A}$ is, thus, found by minimizing over $\boldsymbol{U}$ and $\boldsymbol{A}$. Let $\boldsymbol{V}$ be the isometry of the matrix $\boldsymbol{Y}$. Assuming that $\boldsymbol{A}$ is sparse, we can chose

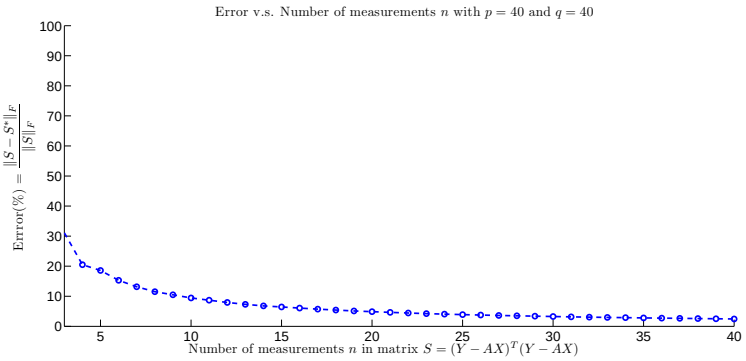


Figure 3.2: Approximation error $||\boldsymbol{S} - \boldsymbol{S}*||_F/||\boldsymbol{S}||_F$ versus $n = 1, \cdots, p$ for $p = 40$.

$\boldsymbol{U}$ to be the isometry of $\boldsymbol{Y}$. By replacing $\boldsymbol{U}$ by $\boldsymbol{V}$ in (3.32), we may increase the minimum but we reduce the problem to a convex setting in the unique variable $\boldsymbol{A}$. Finally, the estimated regression matrix is the unique global solution of the following convex optimization problem,

$$\begin{cases} \min\limits_{\boldsymbol{A}} \|\boldsymbol{A}\|_1 \\ \\ \text{s.t.} \quad \boldsymbol{AX} = \boldsymbol{Y} - \boldsymbol{VS}^*, \end{cases} \tag{3.33}$$

SMURC algorithm. The SMURC algorithm is summarized below.

Input: The matrices $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{q\times n}$ according to the multivariate regression model in Eq. (3.2) with $q > n$.

Step 1 Solve the convex optimization problem in (3.23). The solution of this

problem is a p.s.d. matrix $\boldsymbol{S}^* \in \mathbb{R}^{n \times n}$

Step 2 Given $\boldsymbol{S}^*$, the optimal regression matrix is obtained as the solution to the convex optimization problem in (3.33).

Steps 1 and 2 can be implemented efficiently using the Matlab Software for Disciplined Convex Programming, *cvx* [23], [22].

The following corollary provides an upper bound on the l_1-norm of the optimal connectivity matrix

Corollary 3.2.4. *The norm of the optimal connectivity matrix, given by the solution of the convex optimization problem in (3.33), is bounded above by*

$$\|\boldsymbol{A}^*\|_1 \leq \|\boldsymbol{V}(|\boldsymbol{Y}| - \boldsymbol{S}^*)\tilde{\boldsymbol{X}}\|_1 \leq \epsilon, \tag{3.34}$$

where $\boldsymbol{V}$ is the isometry in the polar decomposition of $\boldsymbol{Y}$, $\boldsymbol{S}^*$ is the global solution of the convex optimization problem in (3.23) and $\tilde{\boldsymbol{X}}$, defined in (3.27), is the right inverse of the matrix $\boldsymbol{X}$.

Proof. The proof follows from the proof of Proposition 5, and specifically from Eq. (3.29). □

The SMURC algorithm involved an approximation of the original optimization problem (3.10) by the convex optimization problem in (3.23). It is thus important to assess the effect of this convex approximation on the final solution. An analytical

derivation to bound this approximation is difficult and cumbersome. We, therefore, provide a numerical assessment of this approximation by computing the average error between the exact solution of (3.21) and the approximate solution of (3.23), $||\boldsymbol{S} - \boldsymbol{S} * ||_F / ||\boldsymbol{S}||_F$. In synthetic data, the exact solution $\boldsymbol{S}$ is known. The error graph, displayed in Fig. 3.2 shows that this approximation error decreases to a very small value when the number of measurements n approaches the number of unknowns p.

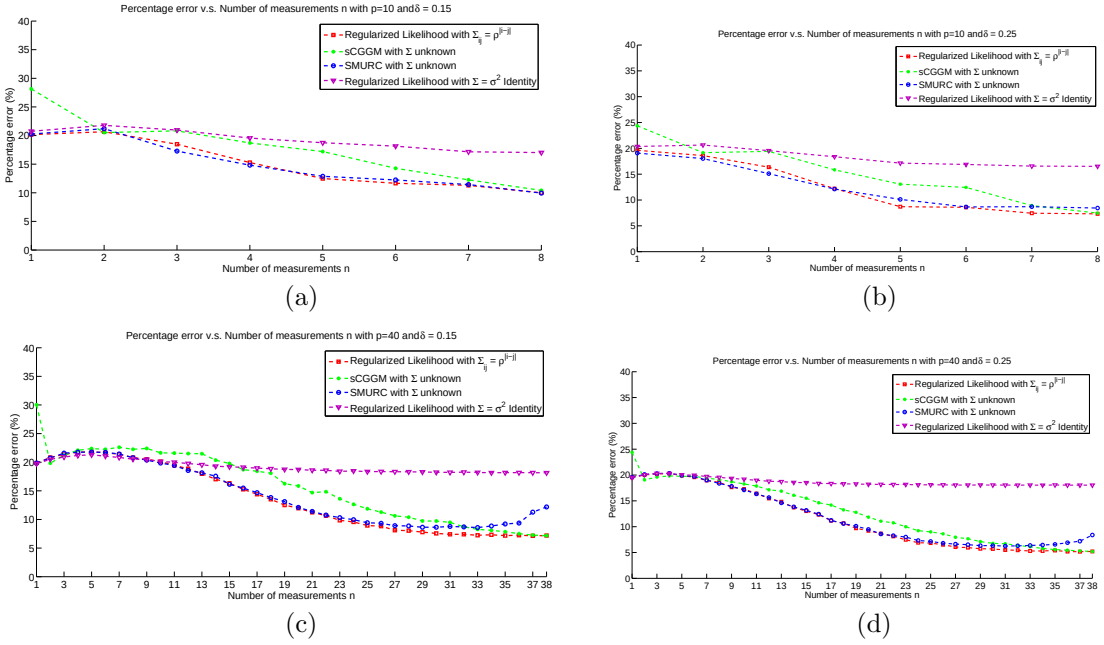


Figure 3.3: Performance comparison of SMURC with sCGGM, the l_1-regularized maximum likelihood (RMLE) with known true covariance Σ_{true} and the l_1-regularized maximum likelihood (RMLE) with covariance $\Sigma_{ij} = \sigma^2\ I$ for different network sizes with %80 sparsity. Blue: SMURC with unknown covariance; Green: sCGGM with unknown covariance; Red: RMLE with $\Sigma = \Sigma_{true} = \rho^{|i-j|}$; Purple: RMLE with $\Sigma = \sigma^2\ I$. (a) $(p, \delta) = (10, 0.15)$; (b) $(p, \delta) = (10, 0.25)$; (c) $(p, \delta) = (40, 0.15)$; (d) $(p, \delta) = (40, 0.25)$.

3.3 Application: Genetic Regulatory Networks

An application of interest, which suffers from the high-dimension, small sample-size problem is the reconstruction, also called *reverse engineering*, of genetic regulatory

networks (GRNs), where only few samples, denoting time points or tissue samples, are available. Inference of genetic regulatory networks is important for understanding the dynamics of genetic interactions and harnessing this understanding into an educated intervention of the cell. The behavior of the regulatory network can be modeled by a system of linear differential equations near a steady-state [4, 5, 15, 21, 45]:

$$\dot{x}_i(t_k) = \sum_{j=1}^{N} a_{ij} x_j(t_k) + b_i u(t_k) + \epsilon_i(t_k), \tag{3.35}$$

where $i = 1, \cdots, p, k = 1, \cdots, n$, p is the number of genes, n is the number of experiments or time points, $x_i(t)$ is the expression of gene i at time t, $\dot{x}_i(t)$ is the rate of change of expression of gene i at time t, a_{ij} represents the influence of gene j on gene i, b_i is the effect of the external perturbation on gene i and $u(t)$ denotes the external perturbation at time t. $\epsilon_i(t_k)$ models the measurement and model error at time step k. The goal is to infer the gene interactions $\{a_{ij}\}_{i,j=1}^{p}$, given a certain number of measurements n. Introducing the new variable $y_i(t) = \frac{dx_i}{dt} - b_i u(t)$, we can write the ODE model in vector form for the p genes as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{3.36}$$

where $\boldsymbol{y} = [y_1, y_2, \cdots, y_p]^T, \boldsymbol{x} = [x_1, x_2, \cdots, x_p]^T, \boldsymbol{\epsilon} = [\epsilon_1, \cdots, \epsilon_p]^T$ and $\boldsymbol{A} = \{a_{ij}\}_{i,j=1}^{p}$. Performing n different experiments , we obtain n measurements and can write the results as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}, \tag{3.37}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n]$, $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$ and $\boldsymbol{E} = [\boldsymbol{\epsilon}_1, \cdots, \boldsymbol{\epsilon}_n]$. That is, every column of $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{E}$ represents a single experiment and there are $n < p$ columns representing n experiments. The goal of reverse-engineering the network is to estimate the connectivity matrix $\boldsymbol{A}$ given a number of measurements and in the presence of correlated noise with unknown covariance matrix Σ.

3.3.1 Simulation results. Before considering a real dataset, we generate synthetic data and compare the proposed SMURC algorithm with the l_1-regularized maximum likelihood estimator in [45], where an l_1-norm penalty is imposed on the connectivity matrix $\boldsymbol{A}$. The regularized MLE finds the optimal connectivity matrix, which minimizes the following convex function

$$\begin{aligned} f(\boldsymbol{A}) =& \mathrm{Tr}\left[\frac{1}{n}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})^T\boldsymbol{\Sigma}^{-1}\right] + \ln|\boldsymbol{\Sigma}| \\ &+ \alpha\sum_{i=1}^{p}\sum_{j=1}^{p}|a_{i,j}|, \end{aligned} \tag{3.38}$$

where $\boldsymbol{\Sigma}$, the covariance matrix of the data, is assumed to be known and α is a tuning parameter that controls the sparsity level of the matrix $\boldsymbol{A}$.

We generate synthetic gene networks with varying size p, varying number of measurements $n < p$, and covariance structure $\boldsymbol{\Sigma}$. Gene regulatory networks are known to be sparse: every gene interacts only with few other genes. Thus, the connectivity matrix $\boldsymbol{A}$ is sparse. In the presented simulations, we assume 80% sparsity level, i.e., $\|\boldsymbol{A}\|_0 = 0.2p^2$, where $\|\cdot\|_0$ denotes the number of non-zero elements. The perfor-

mance of the algorithm is similar for other sparsity levels as long as the system is under-determined. The entries of the matrix $\boldsymbol{A}$ are drawn from a standard normal distribution with zero-mean and unit variance, i.e., $a_{i,j} \sim \mathcal{N}(0,1)$. The performance of the algorithm is assessed using the following measure suggested in [60]:

$$E = \sum_{i=1}^{p} \sum_{i=j}^{p} e_{i,j} \quad \text{with}$$

$$e_{i,j} = \begin{cases} 1, & \text{if} \quad |a_{ij} - \hat{a}_{ij}| > \delta |a_{ij}| \\ 0, & \text{otherwise,} \end{cases} \tag{3.39}$$

where a_{ij} is the $(i,j)^{th}$ element of the true genetic interaction matrix and $\hat{a}_{ij}$ is the estimate of a_{ij}. δ is a threshold parameter. The percentage error is computed as E/p^2.

Figure 3.3 shows the percentage error versus the number of measurements n for $p = 10$ and $p = 40$-gene networks, which are 80% sparse. We considered a threshold of error corresponding to $\delta = 0.15$ and $\delta = 0.25$. Observe that, though the system is sparse, it is still under-determined, i.e., the number of "effective" unknowns is larger than the number of independent observations. We compare the proposed SMURC algorithm (which assumes an unknown covariance matrix) with the regularized MLE (RMLE) with known covariance matrix [45], with covariance matrix $\Sigma = \sigma^2 \, I$ and the latest sCGGM algorithm [62]. It was shown in [62] that sCGGM outperforms Rothman *et al.* MRCE and approximate MRCE. We used the optimized code for sCGGM available at `http://www.cs.cmu.edu/~sssykim/softwares/softwares.html`. We

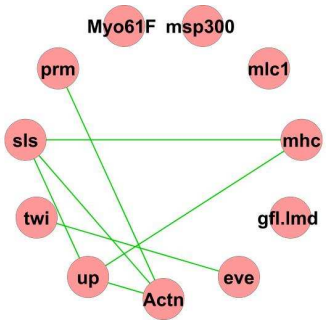


Figure 3.4: Flybase: The known undirected gene interactions in the Drosophila's 11-gene wing muscle network [36].

assess the algorithms with a covariance $\mathbf{\Sigma}_{true} = \rho^{|i-j|}$ with $\rho = 0.7$. Fifty Monte Carlo simulations were performed for each experiment. From Fig. 3.3, we observe that the proposed SMURC algorithm outperforms the sCGGM algorithm and has similar performance as the regularized MLE with known covariance matrix equal to the true covariance matrix.

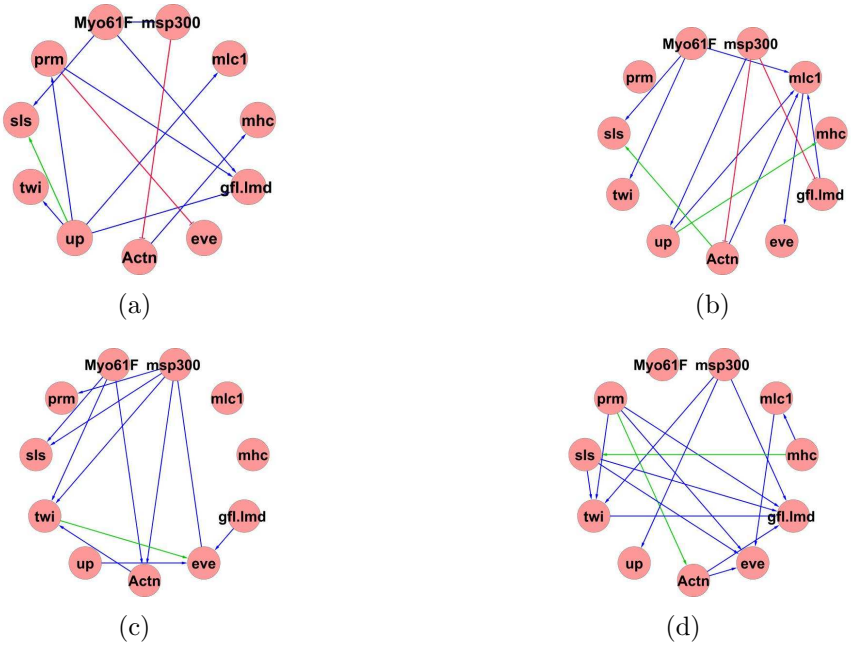


Figure 3.5: Estimated gene regulatory networks of the Drosophila during four developmental phases using the SMURC algorithm. Blue and red edges denote, respectively, positive and negative interactions. The green edges are interactions reported in Flybase. (a) Embryonic; (b) Larval; (c) Pupal; (d) Adulthood.

3.3.2 Drosophila Melanogaster gene expression data. To assess our algorithm on real data, we tested it on the Drosophila melanogaster gene expression levels [3]. The data contains 4028 genes in wild-type flies examined during 66 sequential time periods beginning at fertilization and spanning embryonic, larval, pupal and the first 30 days of adulthood. Since early embryos change rapidly, overlapping 1-hour periods were sampled; adults were sampled at multiday intervals. The time points span the embryonic (samples 1-30; time E01h till E2324h), larval (samples 31-40; time L24h till L105h), pupal (samples 41-58; M0h till M96h) and adulthood (samples 59-66; A024h till A30d) periods of the organism. A list of known undirected gene interactions is hosted in Flybase [36].

A set of 11 genes that regulate the wing muscle development has been considered in [24, 39, 47, 63]. The 11-gene network, with the interactions reported in Flybase, is depicted in Fig. 3.4. We reconstructed the genetic network between these 11 genes during the four developmental phases using the SMURC algorithm. In the embryonic and pupal phases, 9 time points, undersampled from the original time points (30 for embryonic and 18 for pupal), were used to reconstruct the 11-gene network during these two developmental periods. In the larval and adulthood phases, the entire 9 larval and 7 adulthood time points were used to reconstruct the network during the larval and adulthood development phases, respectively. In summary, the connectivity matrix of the 11-gene Drosophila development network was estimated using the SMURC algorithm with 9 time points in the embryonic phase, 9 time points in the larval phase, 9 time points in the pupal phase and 7 time points in the adulthood

phase. Observe that in all four developmental phases, the system is underdetermined.

The reconstructed networks using the SMURC algorithm are shown in Fig. 3.5. The SMURC algorithm was able to detect six out of the seven Flybase interactions during different developmental phases of the organism: (*up,sls*) appears during the embryonic period; (*Actn,sls*) and (*up,mhc*) appear during the larval phase; (*twl,eve*) appears during the pupal phase; (*prm,Actn*) and (*mhc,sls*) appear during the adulthood stage of the development.

We compare the SMURC findings with the results in [63], [24], [47], [39]. Though these references are not directly related to the problem of under-determined regression systems with unknown covariance structure, their work aims at reverse-engineering the connectivity of genetic regulatory networks. In particular, they all consider the Drosophila's 11-gene wing muscle network. Zhao *et al.* [63] infer a single directed network using the minimum description length principle. They used all 66 time points to identify a single network that characterizes the entire Drosophila's life cycle. In [24], a time-varying undirected network is learnt using an exponential random graph model model. A dynamic Bayesian network was used in [47], and [39] proposed a non-parametric Bayesian regression approach for gene regulatory network inference. Table 3.1 shows the detection of the known interactions in Flybase by the five approaches, E,L,P,A stand, respectively, for the embryonic, larval, pupal and adulthood phases. Though the proposed SMURC algorithm relies on fewer time points than the other approaches, it detected the most number of known interactions cited in Flybased and reported in FLIGHT website `http://flight.icr.ac.uk/search/search_interactions.jsp`. Additionally, the

Table 3.1: Detection of the known gene interactions in Flybase

	(*prm,Actn*)	(*sls,mhc*)	(*mhc,up*)	(*sls,Actn*)	(*sls,up*)	(*twi,eve*)	(*up,Actn*)
SMURC	✓ (A)	✓ (A)	✓ (L)	✓ (L)	✓ (E)	✓ (P)	×
Minimum description length [63]	✓	✓	×	×	×	✓	×
Random graph model [24]	×	×	✓ (E,L,P,A)	✓ (P,A)	✓ (E,L,P,A)	×	×
Dynamic Bayesian network [47]	×	✓ (E,L,P,A)	×	×	×	×	×
Nonparametric Bayesian regression [39]	×	×	×	×	×	✓ (E)	×

SMURC algorithm found two directed interactions (*msp 300* $\rightarrow$ *prm*) and (*msp300* $\rightarrow$ *up*) in common with the works in [63], [24], [47], and three directed interactions during the embryonic phase in common with [39] (the networks in the other phases were not reported in [39]) , (*up* $\rightarrow$ *twi*), (*up* $\rightarrow$ *mlc1*) and (*msp300* $\longrightarrow$ *Myo61F1*). It is striking that all detected interactions that are shared with previous work [24,39,47,63] have also the same direction.

3.4 Conclusion and Discussion

In this chapter, we showed that the Gaussian likelihood, as a function of the regression coefficients and the covariance matrix, diverges when the multivariate regression system is under-determined. We subsequently proposed a normalized likelihood function that guarantees convergence while still keeping the Gaussian form of the data. The maximum normalized likelihood, however, admits multiple solutions because the system is still under-determined. Using the polar decomposition of matrices, we provided an expression of the covariance matrix in terms of the regression coefficients. This provided an equivalent representation of the optimization problem in one variable only, namely the regression matrix. We then relaxed the optimization problem into a convex one by considering a convex set included in the original constraint set. The optimal sparse regression matrix is found as the global solution to a convex

optimization problem.

We applied the proposed Small-sample MUltivariate Regression with Covariance estimation (SMURC) algorithm to infer the wing muscle genetic regulatory networks of the Drosophila melanogaster during the four phases of its development: embryonic, larval, pupal and adulthood. Genetic regulatory networks are known to be sparse and often the number of measurements is smaller than the number of genes, which makes the network inference problem unidentifiable. SMURC was able to detect six out of the seven interactions reported in Flybase. Other algorithms aimed at reverse-engineering dynamic gene regulatory networks were able to detect a maximum of three out of the seven interactions.

Chapter 4
Kernel Reconstruction versus the Combinatorial ℓ_0-based Compressive Sensing Algorithm

4.1 Introduction

The Compressed Sensing method is known to reconstruct large dimensional signal from a small number of measurements, where the original signal admits a sparse representation in a certain basis. It has been widely used and implemented in many application such as computized tomography [12], wireless communication [53], image processing [8] and camera design [20] where the task is the reconstruction of a signal or an image from linear measurements while taking many of them. We will focus in this chapter on the mathematical aspect and proofs to see how far the CS based on convex programming is asymptotically optimal. Actually, the CS technique is an under-determined system where there is no guarantee that it yields a solution. Thus, in order to make sure that our system converges to the optimal solution, some constraints are imposed. In fact, this technique employs a few number of non adaptative-projections of the sparse signal to compute a set of M measurements, i.e. encoding matrix. Therefore, this framework is characterized by M equations for N unknowns. To solve this problem, this system needs to be well-conditioned in order to yield a solution. In

other words, we are looking to guarantee the stability of these projections.

More specifically, CS solves ℓ_0-based optimization problem where the ℓ_0 norm corresponds to the number of nonzero entries in a vector. The main goal is to minimize the number of non zero element in a vector given a linear model. In other words, we are looking to find the sparsest solution within all possible ones. This achieved by doing combinatorial search which is computationally very expensive. Therefore, Donoho in [18] suggested to replace ℓ_0-norm by the ℓ_1 and the ℓ_2 norms and stated that they yield the same solutions. However, minimizing either the ℓ_1 or the ℓ_2 norm does not guarantee an exact solution. Therefore, in this chapter will present a new approach that performs the exact reconstruction of sparse signals given a linear model for under-determined systems.

This chapter is organized as follows: In Section 4.2 we present an overview of the Compressed Sensing technique and the most important theoretical details. Section 4.3 we present our approach called the the Kernel Reconstruction and we compare it to the alternatives techniques suggested in [18]. Finally, we briefly discuss in Section 4.4 the most important results of this research and what are the points that should be investigated for the future work.

4.2 Compressed Sensing

4.2.1 Preliminaries and Notations. This section introduces the different mathematical modeling and analysis needed for this study. We mostly consider complex vectors in $\mathbb{C}^N$, however, sometimes the considerations will be restricted to $\mathbb{R}^N$.

We denote vectors by lower-case letters and matrices by upper-case letters. The ℓ_p-norm of a vector $x \in \mathbb{C}^N$ is given by

$$\begin{aligned} \|x\|_p &= (\sum_{j=1}^{N} |x|^p)^{1/p} \\ \|x\|_\infty &= \max_{j=1,..,N} |x_j|, \end{aligned} \tag{4.1}$$

where $1 \leq p \leq \infty$. For $0 \leq p \leq 1$, ℓ_p is called quasi-norm. However, in contrast to the definition, the ℓ_0-norm $\|x\|_0$ is the number of the nonzero entries in x and it's not considered as a quasi-norm. The support of a vector x is denoted $supp(x) = \{j,\ x_j \neq\}0$. Moreover, x is called k-sparse if $\|x\|_0 \leq k$ where $k \in \{1, .., N\}$. The set of k-sparse vectors is denoted by $\Sigma_k = \{x \in \mathbb{C}^N, \|x\|_0 \leq k\}$. In addition, the best k-term approximation error of a vector x in ℓ_p is defined as

$$\sigma_k(x)_p = \inf_{z \in \Sigma_k} \|x - z\|_p. \tag{4.2}$$

The ℓ_p^N notation refers to the ℓ_p-norm for the N-dimensional vector in $\mathbb{C}^N$ or $\mathbb{R}^N$. $B_p^N = \{x \in \mathbb{C}^N, \|x\|_p \leq 1\}$ denotes the unit ball in ℓ_p^N. The operator norm of a matrix $\phi \in \mathbb{C}^{M \times N}$ $\|.\| : \ell_p^N \to \ell_p^M$ is denoted $\|\phi\|_{p \to p} = \max_{\|x\|_p = 1} \|\phi x\|_p$. Specifically if $p = 2$, the operator norm is the maximal singular value $\sigma_{\max}(\phi)$ of ϕ. We consider the subset $T \subset \{1, .., N\}$ where x_T is the N-dimensional vector whose entries in T are equal to the corresponding ones in $x \in \mathbb{C}^N$ and zero outside T. The complement of T is denoted by T^c and $\sharp$ refers to the cardinality.

4.2.2 Mathematical Analysis. In the last decade, the CS was introduced in literature [18], [50] based on mathematical viewpoint to prove its feasibility. This method deals with the original signal in a certain k-sparse representation computed by the transform coding (TC) [18] knowing that many signals are sparse in some bases, e.g. Gabor, Fourier and Wavelet bases. A signal x with dimension N is called k-sparse when it can be written in linear combination of few K basis vectors with large coefficients. The remained $(N - K)$ lowest values are set to zero where $N >> K$. x is called compressible when it has few large coefficients and many small values. The main task of CS is how to directly sense the compressible signal. In general, as we mentioned above, we apply the TC to find the sparse vector representation. The k-sparse signal seems to be sufficient and suitable for any application since it contains a few coefficients. It so happens that TC has numerous inefficiencies that CS rigorously addresses. Specifically, in some cases, for example seismic and video signals, the dimension N of the signal may blow up to a big number regardless whether the desired K is small. In addition, the entire transform coefficients in the new space must be computed. Finally the locations of the sparse coefficients are unknown, therefore they must be encoded. Consequently this leads to an overhead in memory. Actually, to solve this bunch of problems CS employs a few numbers of non-adaptive linear projections of the k-sparse signal. From the latter we can get a set of measurements which will replace the acquired samples. Explicitly, the measurements set can be

represented by the following mathematical model:

$$y_0 = \phi \times x, \tag{4.3}$$

where y_0, x, ϕ are respectively the $M \times 1$ vector of noiseless measurements, the k-sparse $N \times 1$ signal and the $M \times N$ measurement matrix. The k-sparse signal x has $(N - K)$ zeroed entries and the matrix ϕ has M different rows to compute each corresponding measurement. The term non-adaptive is used here to mention that the linear projections do not depend on the signal x. Infering the sparse vector x is equivalent to solving the ℓ_0-minimization problem:

$$\min \|x\|_0 \text{ subject to } \quad y_0 = \phi \times x, \tag{4.4}$$

Ideal CS reconstruction of K-sparse signal is to find the sparsest solution from the infinitely many solutions $\hat{x}$. It has been shown in [18] that if $M \leq K$ then there is no solution and if $M \leq k + 1$ then there is a high probability of high reconstruction. Unfortunately, this combinational minimization problem is computationally NP-hard and intractable. Donoho *et al.* [18] proposed two practical alternatives, i.e. $\ell_1$minimization and ℓ_2-minimization. The $\ell_1$minimization technique considers the following problem:

$$\min \|x\|_1 \text{ subject to } \quad y_0 = \phi \times x, \tag{4.5}$$

which is a convex optimization problem and can be seen as a convex relaxation of Eq. (4.4). The $\ell_2$minimization technique considers the following problem:

$$\min \|x\|_2 \text{ subject to } \quad y_0 = \phi \times x, \tag{4.6}$$

using the $\ell_2$minimization, our problem still convex and can be solved.

As we mentioned above, our system in Eq. (4.3) is under-determined since we have M equations for N unknowns where $M << N$. Consequently, some constraints are imposed on ϕ in order to yield a solution. Therefore the properties of the measurement matrix has been extensively studied in [50] where they revealed to be useful for CS based on convex programming recovery. Thus, based on lemma and theorems; which have been proven in [50]; we show how the ℓ_2-recovery implies the ℓ_1-recovery. We need first to introduce the following lemma which states that ℓ_q-balls where $q \leq 1$ are good models for compressible vectors.

Lemma 4.2.1. *Let $0 < q < p \leq \infty$ and set $r = \frac{1}{q} - \frac{1}{p}$. Then*

$$\sigma_k(x)_p \leq k^{-r} \quad k=1,..,N \quad \forall x \in B_q^N.$$

Proof. Let T be the set of indices of the k-largest entries of x in absolute value. The nonincreasing rearrangement satisfies $|r_k(x)| \leq |x_j|$ for all $j \in T$, and therefore we have

$$k r_k(x)^q \leq \sum_{j \in T} |x_j|^q \leq \|x\|_q^q \leq 1.$$

Hence, $r_k(x) \leq k^{\frac{1}{q}}$. Thus

$$\sigma_k(x)_p^p = \sum_{j \notin T} r_k(x)^{p-q} |x_j|^q \leq k^{\frac{p-q}{q}} \|x\|_q^q \leq k^{\frac{p-q}{q}},$$

which implies $\sigma_k(x)_p \leq k^{-r}$. □

From lemma 4.2.1, it's very interesting to see that $\sigma_k(x)_p$ is bounded.

One fundamental property in ℓ_1-analysis that ϕ needs to satisfy is the Null Space Property (NSP).

Definition 4.2.2. *A matrix $\phi \in \mathbb{C}^{M \times N}$ is said to satisfy the null space property (NSP) of order k with constant $\gamma \in (0,1)$ if*

$$\|\eta_T\| \leq \gamma \|\eta_{T^c}\|,$$

for for all sets $T \subset 1, .., N$, $\sharp T \leq k$ and $\forall \eta \in \ker(\phi)$

The following theorem prove the ℓ_1-recovery of x.

Theorem 4.2.3. *Let $\phi \in \mathbb{C}^{M \times N}$ be a matrix that satisfies the NSP of order k with constant $\gamma \in (0,1)$. Let $x \in \mathbb{C}^N$, $y = \phi \times x$ and let $\hat{x}$ be a solution of the Eq. (4.5). Then*

$$\|x - \hat{x}\|_1 \leq \frac{2(1+\gamma)}{1-\gamma} \sigma_k(x)_1. \tag{4.7}$$

In particular, if x is k-sparse then $\hat{x} = x$.

Proof. Let $\eta = \hat{x} - x$. Then $\eta \in \ker \phi$ and

$$\|\hat{x}\|_1 \leq \|x\|_1$$

because $\hat{x}$ is a solution of the ℓ_1-minimization problem in Eq. (4.5). Let T be the set of the k-largest entries of x in absolute value. we have

$$\|\hat{x}_T\|_1 + \|\hat{x}_{T^c}\|_1 \leq \|x_T\|_1 + \|x_{T^c}\|_1$$

using the triangular inequality we have

$$\|x_T\|_1 - \|\eta_T\|_1 + \|\eta_{T^c}\|_1 - \|x_{T^c}\|_1 \leq \|x_T\|_1 + \|x_{T^c}\|_1.$$

Therefore,

$$\|\eta_{T^c}\|_1 \leq \|\eta_T\|_1 + 2\|\eta_{T^c}\|_1 \leq \gamma\|\eta_{T^c}\|_1 + 2\sigma_k(x)_1.$$

which is equivalent to have

$$\|\eta_{T^c}\|_1 \leq \frac{2}{1-\gamma}\sigma_k(x)_1. \tag{4.8}$$

Then,

$$\|x - \hat{x}\|_1 = \|\eta_T\|_1 + \|\eta_{T^c}\|_1 \leq \frac{2(1+\gamma)}{1-\gamma}\sigma_k(x)_1. \tag{4.9}$$

□

Then Using Lemma 4.2.1 and Eq. (4.7) in theorem 4.2.3 we can show that the

reconstruction error $\|x - \hat{x}\|_1$ is bounded. Hence if all possible k-sparse signals x can be ℓ_1-recovered then necessarily ϕ satisfies the Null Space Property (NSP). It's clear here that the NSP is equivalent to sparse ℓ_1-recovery.

For ℓ_2-analysis, we consider the Restricted Isometry Property (RIP). The matrix ϕ must satisfy this property in order to prove the ℓ_2-recovery.

Definition 4.2.4. *The restricted isometry constant constant δ_k of a matrix $\phi \in \mathbb{C}^{M\times N}$ is the smallest number such that*

$$(1 - \delta_k)\|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k)\|x\|_2^2 \quad \forall x \in \Sigma_k,$$

A matrix ϕ is said to satisfy the restricted isometry property of order k with constant δ_k if $\delta_k \in (0, 1)$. It's easily seen that δ_k can be equivalently defined as

$$\delta_k = \max_{T\subset\{1,..,N\}} \|\phi_T^*\phi_T - Id\|_{2\to 2},$$

where ϕ_T^* is the hermitian matrix of ϕ_T.
In other words, all columns submatrices of ϕ with at most k columns are required to be well conditioned.

From the previous definition, if ϕ satisfies the RIP then any matrix formed from the columns of indices $j \in T$ is well-conditioned. Given that x is sparse, then the vector of measurement y_0 can be written as the summation of k inner products between the same k-sparse vector x and the corresponding columns of the matrix ϕ that correspond to the nonzero entries in x. Thus, the RIP implies the stability of the system.

Furthermore, we can show the relationship between the NSP and the RIP constraints for ϕ. The following lemma proves this result.

Lemma 4.2.5. *If $\phi \in \mathbb{C}^{M\times N}$ satisfies the RIP of order $K = k + h$ with constant $\delta_K \in (0, 1)$, then ϕ has the NSP of order k with constant $\gamma = \sqrt{\frac{k}{h}\frac{1+\delta_K}{1-\delta_K}}$*

Proof. Let $\eta \in \mathcal{N} = \ker(\phi)$ and $T \subset \{1, \ldots, N\}$, $\#T \leq k$. Define $T_0 = T$ and $T_1, T_2, \ldots, T_s$ to be disjoint sets of indexes of size at most h, associated to a nonincreasing rearrangement of the entries of $\eta \in \mathcal{N}$, i.e.

$$|\eta_j| \leq |\eta_i| \quad \text{for all } j \in T_\ell, i \in T_{\ell'}, \ell \geq \ell' \geq 1. \tag{4.10}$$

We have $\phi\eta = 0$ which is equivalent to have $\phi\eta_{T_0\cup T_1} = -\sum_{j=2}^{s} \phi\eta_{T_j}$. Using the Cauchy-Schwartz inequality, the RIP, and the triangular inequality we have

$$\begin{aligned}
\|\eta_T\|_1 &\leq \sqrt{k}\|\eta_T\|_2 \leq \sqrt{k}\|\eta_{T_0\cup T_1}\|_2 \\
&\leq \sqrt{\frac{k}{1-\delta_K}}\|\phi\eta_{T_0\cup T_1}\|_2 = \sqrt{\frac{k}{1-\delta_K}}\|\phi\eta_{T_2\cup T_3\cup\cdots\cup T_s}\|_2 \\
&\leq \sqrt{\frac{k}{1-\delta_K}}\sum_{j=2}^{s}\|\phi\eta_{T_j}\|_2 \leq \sqrt{\frac{1+\delta_K}{1-\delta_K}}\sqrt{k}\sum_{j=2}^{s}\|\eta T_j\|_2
\end{aligned} \tag{4.11}$$

Now consider Eq. (4.10), then we have$|\eta_i| \leq |\eta_\ell|$ for all $i \in T_{j+1}$ and $\ell \in T_j$. Taking the sum over $\ell \in T_j$ and next the ℓ_2-norm over $i \in T_{j+1}$ gives

$$|\eta_i| \leq h^{-1}\|\eta_{T_j}\|_1; \text{and} \quad \|\eta_{T_{j+1}}\|_2 \leq h^{-1/2}\|\eta_{T_j}\|_1.$$

Using the inequality in Eq. (4.11) we have

$$\|\eta_T\|_1 \leq \sqrt{\frac{1+\delta_K}{1-\delta_K}\frac{k}{h}}\sum_{j=1}^{s-1}\|\eta_{T_j}\|_1 \leq \sqrt{\frac{1+\delta_K}{1-\delta_K}\frac{k}{h}}\|\eta_{T^c}\|_1. \tag{4.12}$$

□

Lemma 4.2.5 shows that the RIP implies the NSP. Then we can conclude that the ℓ_2-recovery implies the ℓ_1-recovery. The following theorem prove the ℓ_2-recovery of x.

Theorem 4.2.6. *Assume $\phi \in \mathbb{C}^{M\times N}$ satisfies the RIP of order $3k$ with $\delta_{3k} < 1/3$. For $x \in \mathbb{C}^N$, let $y_0 = \phi x$ and $\hat{x}$ be the solution of the Eq. (4.6). Then*

$$\|x - \hat{x}\|_2 \leq \frac{C}{\sqrt{k}}\sigma_k(x)_1, \tag{4.13}$$

with $C = \frac{2}{1-\gamma}(\frac{\gamma+1}{\sqrt{2}} + \gamma)$, $\gamma = \sqrt{\frac{1+\delta_{3k}}{2(1-\delta_{3k})}}$.

Proof. This proof is similar to Lemma 4.2.5, where $\eta = \hat{x} - x \in \mathcal{N} = \ker(\phi)$, $T_0 = T$ the set of the $2k$-largest entries of η in absolute value, and T_j's of size at most k corresponding to the nonincreasing rearrangement of η. Then using the inequality in Eq. (4.11)and (4.12) with $h = 2k$ of the previous proof we find

$$\|\eta_T\|_2 \leq \sqrt{\frac{1+\delta_{3k}}{2(1-\delta_{3k})}}k^{-1/2}\|\eta_{T^c}\|_1. \tag{4.14}$$

Given that $\delta_{3k} < 1/3$ we have $\gamma = \sqrt{\frac{1+\delta_{3k}}{2(1-\delta_{3k})}} < 1$. Using the results of Lemma 4.2.1

and 4.2.5 we can find

$$\begin{aligned} \|\eta_{T^c}\|_2 &= \sigma_{2k}(\eta)_2 \leq (2k)^{-1/2}\|\eta\|_1 = (2k)^{-1/2}(\|\eta_T\|_1 + \|\eta_{T^c}\|_1) \\ &\leq (2k)^{-1/2}(\|\gamma\eta_{T^c}\|_1 + \|\eta_{T^c}\|_1) \end{aligned} \tag{4.15}$$

We know that T is the set of $2k$-largest entries of η in absolute value, then we have

$$\|\eta_{T^c}\|_1 \leq \|\eta_{supp\ x^c_{[2k]}}\|_1 \leq \|\eta_{supp\ x^c_{[k]}}\|_1, \tag{4.16}$$

where $x[k]$ is the best k-term approximation to x. From Eqs. (4.16) and (4.8) we can finally find

$$\begin{aligned} \|x - \hat{x}\|_2 &\leq \|\eta_T\|_2 + \|\eta^c_T\|_2 \\ &\leq (\frac{\gamma+1}{\sqrt{2}} + \gamma)k^{-1/2}\|\eta_{T^c}\|_1 \\ &\leq \frac{2}{1-\gamma}(\frac{\gamma+1}{\sqrt{2}} + \gamma)k^{-1/2}\sigma_k(x)_1. \end{aligned} \tag{4.17}$$

□

Then similarly to the ℓ_1-analysis, using lemma 4.2.1 and Eq. (4.13)in theorem 4.2.6 we can show that the reconstruction error $\|x - \hat{x}\|_2$ is bounded. Hence if all possible k-sparse signals x can be ℓ_2-recovered then necessarily ϕ satisfies the RIP.

RIP in Linear Algebra Explicitly, the RIP condition is equivalent to have all the elements of $\ker(\phi)$ should have at least $K+1 = 3k+1$ non zero entries, i.e. $K \leq M$. This would guarantee that the measurement matrix is full rank and hence invertible.

Proof. Assume that $K > M$ and let G be a subspace having $N - K$ zeros where $\dim(G) = K$. If ϕ is invertible in this space this is equivalent that there is no *kernel* in G which is impossible since $\dim(\phi) = M$ □

Thus, we have $M \geq K + 1$ which is a theoretical bound of the number of measurements. In [50], it has been suggested a more precise theoretical bound which is in terms of the sparsity K. The following corollary from [50] shows a very important result in CS community.

Corollary 4.2.7. *Suppose that $\phi \in \mathbb{R}^{M \times N}$ such that*

$$\|x - x^*\|_2 \leq C_{\frac{\sigma_k(x)_1}{\sqrt{k}}},$$

for all $x \in B_1^N$ and some constant $C > 0$. Then necessarly

$$M \geq C^{'} K \log(2N/M). \tag{4.18}$$

Therefore, the most important property that ϕ needs to satisfy is the Restricted Isometry Property. Actually it has been shown that RIP implies robustness under noise on measurements. The following theorem gives an upper bound for the ℓ_2-norm error between the original vector x and the sensed vector x^* under noisy measurements.

Theorem 4.2.8. *Assume that the restricted isometry constant δ_{2k} of the matrix $\phi \in \mathbb{C}^{M\times N}$ satisfies*

$$\delta_{2k} < \frac{2}{3+\sqrt{7/4}} \simeq 0.4627.$$

Then the following holds for all $x \in \mathbb{C}^N$. Let noisy measurements $y = Ax + e$ be given with $\|e\|_2 \leq \epsilon$. Let x^* be the solution of

$$\min \|z\|_1 \quad \textit{subject to } \|\phi z - y\|_2 \leq \eta,$$

Then

$$\|x - x^*\|_2 \leq C_1\epsilon + C_2\frac{\sigma_k(x)_1}{\sqrt{k}}, \tag{4.19}$$

for some constants C_1,$C_2 > 0$ that depend only on δ_{2k}.

Comparing to the bound in Eq. (4.13), Eq. (4.19) has an extra term related to the noise e which is in terms of its upper bound ϵ.

The recovery abilities of a measurement matrix is assessed by the coherence coefficient μ computed for ℓ_2 normalized columns matrix $\|\phi_n\|_2 = 1$ for $n = 1, .., N$; $\mu = \max_{n\neq k} |\langle\phi_n|\phi_k\rangle|$. This allows deriving numerous inequalities and conditions on the number of measurements M. Furthermore it was proven in [18] that $M = \mathcal{O}(K \log \frac{N}{K}) << N$ for ℓ_1 and ℓ_2-minimization recovery approaches. We will discuss in the next section what can be said about M in practical application. The M linear projections of x are non-adaptive so we can choose random encoding entries for ϕ. Different useful types of random matrices have shown to satisfy the RIP, e.g. Gaussian, Bernoulli and Partial Fourier random matrices [18]. From the latter, it's indeed

easier to deal with such matrices.

In this section we have presented the ℓ_1-based and the ℓ_2-based CS alternative approaches for inferring k-sparse signals. We have also presented the two main conditions that are required to be satisfied by a matrix of measurements in order to find an upper bound for the ℓ_1 and the ℓ_2 errors. In the next section we present our new approach that guarantees an exact reconstruction of a k-sparse signal

4.3 Kernel Reconstruction

In the previous section we have shown from [18] that the ℓ_0-based CS approach could be replace by either ℓ_1-based or ℓ_2-based approach were the error $\|x - x^*\|$ is upper-bounded for both norms. In this section, we present an alternative approach to the ℓ_0-based approach where it requires to go through all possible $x \in \Sigma_k = \{x \in \mathbb{C}^N, \|x\|_0 \leq k\}$ to find the sparsest solution. This requires to make a search of $\sum_{k=1}^{K} \binom{N}{k}$ combinations to find the optimal solution of the ℓ_0 optimization problem.

We consider the linear operator $\Phi : \mathbb{C}^N \longrightarrow \text{Range}(\Phi)$ and we know that $\mathbb{C}^n = \text{Range}(\Phi^T) \oplus \text{Ker}(\Phi)$ where $\dim(\text{Ker}(\Phi)) = S$. Let $\boldsymbol{x}_0 \in \text{Range}(A^T)$ be a particular solution. We have

$$\begin{aligned} y &= P_{Range(\Phi)} \Phi P_{Range(\Phi^T)} x \\ &= \Phi\big(\Phi^T\Phi\big)^{-1}\Phi^T\Phi\Phi^T\big(\Phi\Phi^T\big)^{-1}x \\ &= Ax \\ \Longrightarrow \quad & x_0 = \big(P_{Range(\Phi)}\Phi P_{Range(\Phi^T)}\big)^{-1} y. \end{aligned} \tag{4.20}$$

Let $B = Null(\Phi)$ be the $N \times S$ matrix whose columns are the S vectors that span the subspace Ker(Φ). Therefore $\forall x \in \mathbb{C}^N$ we have

$$\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{j=1}^{S} a_j \boldsymbol{b}_j, \tag{4.21}$$

where $\boldsymbol{b}_j$'s are the kernel vectors and a_j's are the coefficients of the linear combination in Ker(Φ). The matrix form of Eq. (4.21) is as follow

$$\boldsymbol{x} = \boldsymbol{x}_0 + B\boldsymbol{a}, \tag{4.22}$$

Thus, to find x we need to compute the entries in the vector $\boldsymbol{a}$. To do this, we assume that we have AT LEAST $S = \dim(\text{Ker}(\Phi))$ zeros in the vector x. rank$(B) = S$, therefore $\exists$ S linearly independent rows of B that span a space that we call L where these entries are equal to zero.

Let P_s be the projection matrix that projects x onto the space L. We choose P_s to be the $n \times n$ matrix which has 1's on the diagonal entries that correspond to the s selected rows of B and 0's elsewhere.

Finally, $\boldsymbol{a}$ can be computed as follow

$$P_s\boldsymbol{x} = P_s\boldsymbol{x}_0 + P_sB\boldsymbol{a} = 0 \implies \boldsymbol{a} = -\big(P_sB\big)^{-1}P_s\boldsymbol{x}_0. \tag{4.23}$$

Thus, if we find the vector a, then we can recover $\boldsymbol{x}$ from its expression in 4.22. One should notice that we don't know the indices of the S linearly independent rows in B. Therefore, a combinatorial search should be performed in order to find the exact

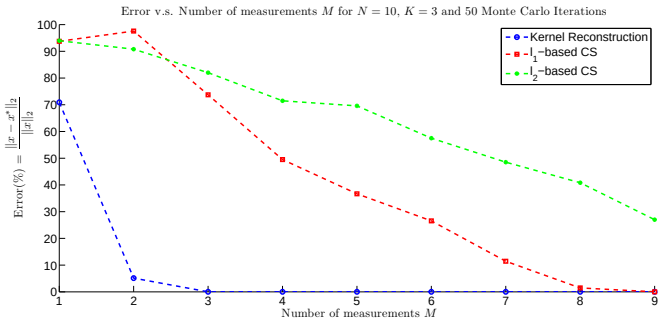


Figure 4.1: Performance comparison of Kernel Reconstruction with ℓ_1-based and ℓ_2-based CS for $N = 10$ and $K = 3$

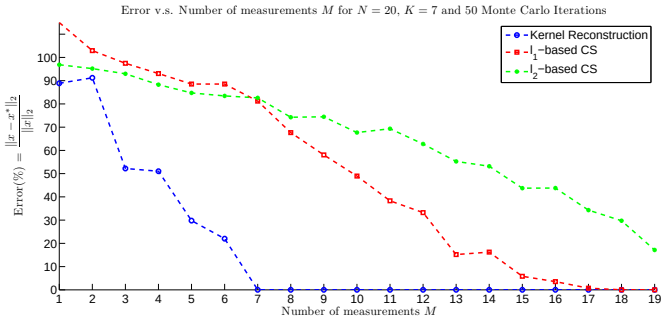


Figure 4.2: Performance comparison of Kernel Reconstruction with ℓ_1-based and ℓ_2-based CS for $N = 20$ and $K = 7$

$\boldsymbol{x}$. Comparing to the ℓ_0-based CS approach, our algorithm requires $\binom{N}{S} << \sum_{k=1}^{K} \binom{N}{k}$ combinatorial search to find the optimal solution.

We plot in Figs 4.1 and 4.2 the percentage error $\frac{\|x-x^*\|_2}{\|x\|_2}$ versus number of measurements M respectively for $(N, K) = (10, 3)$ and $(N, K) = (20, 7)$. We compared our Kernel Reconstruction algorithm to the ℓ_1 and the ℓ_2 approaches. Observe from Figs 4.1 and 4.2 that our approach can exactly recover the K-sparse signal respectively for $M \geq 3$ and $M \geq 7$. This is mainly because at these point, i.e., $M = 3$ and $M = 3$, we consider that there are at least respectively $S = 7$ and $S = 13$ zeros in x. However, respectively for $M < 3$ and $M < 7$, we don't have an exact reconstruction because we assume that we have at least $S = 8$ and $S = 14$ zeros in x.

4.4 Conclusion

In this chapter, we have presented a complete overview about the Compressive Sensing approach. Subsequently, we have presented a new algorithm that recovers exact k-sparse signals. The CS is based on the ℓ_0-norm optimization. Donoha in [18], showed that the ℓ_p-based approaches yield very close solutions with upper-bounded error. In this research, we have suggested a new algorithm that recovers exactly a k-sparse vector from a given matrix Φ and a vector of measurements y. We assume in our approach that x has at least S zero entries where S corresponds to the dimension of the kernel of the matrix of measurements Φ. Although, the technique that we suggest requires $\binom{N}{S}$ combinatorial search to find the optimal solution, it is still computationally not expensive comparing to the ℓ_0-based approach which requires $\sum_{k=1}^{K}\binom{N}{k}$ combinatorial search. Finally, we compared our algorithm to the two ℓ_p approaches and we showed that our technique does exactly recover a sparse signal, whereas, its homologues can infer the same signal with a certain error. Therefore, that would be very interesting to test our approach in solving multivariate regression model, that we have studied in Chapter 3, by considering multidimensional CS framework.

Bibliography

[1] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. *Matrix*, 2006.

[2] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S.Ybarra, D.Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Cell Biology*, 1999.

[3] M. N. Arbeitman, E. E. Furlong, F. Imam, E. Johnson, B. H. Null, B. S. Baker, M. A. Krasnow, M. P. Scott, R. W. Davis, and K. P. White. Gene expression during the life cycle of drosophila melanogaster. *Science*, 297(5590):2270–2275, 2002.

[4] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(78), February 2007.

[5] M. Bansal, G. D. Gatta, and D. di Bernardo. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22(7):815–822, April 2006.

[6] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmon. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 2007.

[7] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–13795, November 2001.

[8] J. Bobin, J.-L. Starck, and R. Ottensamer. Compressed sensing in astronomy. *Selected Topics in Signal Processing, IEEE Journal of*, 2(5):718–726, 2008.

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

[10] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P.Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, March 2004.

[11] X. Cai, J. A. Bazerque, and G. B. Giannakis. Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations. *PLoS Computational Biology*, 9(5), May 2013.

[12] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

[13] D. Chowdary, J. Lathrop, J. Skelton, K. Curtin, T. Briggs, Y. Zhang, J. Yu, Y. Wang, and A. Mazumder. Prognostic gene expression signatures can be measured in tissues collected in rnalater preservative. *The Journal of Molecular Diagnostics*, 8(1):31–39, February 2006.

[14] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari. *Matrix and Tensors Factorizations Application to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley & Sons, November 2009.

[15] M. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtilis using differential equations. In *Pacific Symposium on Biocomputing*, pages 17–28, 2003.

[16] C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:4555, 2010.

[17] C. Dinga, T. Lib, and W. Pengb. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis*, 52:39133927, 2008.

[18] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52:1289–1306, April 2006.

[19] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[20] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):83–91, 2008.

[21] H. M. Fathallah-Shaykh, J. L. Bona, and S. Kadener. Mathematical model of the Drosophila circadian clock: Loop regulation and transcriptional integration. *Biophysical Journal*, 97(9):2399–2408, November 2009.

[22] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[23] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, 2013.

[24] F. Guo, S. Hanneke, W. Fu, and E. P. Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the international conference on Machine learning*, pages 321–328, 2007.

[25] T. M. Hancewicz and J. H. Wang. Discriminant image resolution: a novel multivariate image analysis method utilizing a spatial classification constraint in addition to bilinear nonnegativity. *Chemometrics and Intelligent Laboratory Systems*, 2005.

[26] O. Hauk. Keep it simple: a case for using classical minimum norm estimation in the analysis of EEG and MEG data. *Neuroimage*, 21(4):1612–1621, April 2004.

[27] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann, 1999.

[28] T. Hofmann. *Advances in Neural Information Processing Systems 12*, chapter Learning the Similarity of Documents : an information-geometric approach to document retrieval and categorization, pages 914–920. MIT Press, 2000.

[29] J. hoon Ahn, S. ki Kim, J. hoon Oh, and S. Choi. Multiple nonnegative-matrix factorization of dynamic pet images. In *Asian Conference on Computer Vision*, pages 1009–1013, 2004.

[30] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 2004.

[31] H. Kim and H. Park. Sparse non-negative matrix factorization via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12 2007):1495–1502, March 2007.

[32] P. M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome research*, 13(7):1706–1718, February 2003.

[33] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[34] Y. Li and A. Ngom. Versatile sparse matrix factorization and its applications in high-dimensional biological data analysis. *Patten Recognition in Bioinformatics*, 7986:91–101, 2013.

[35] F. D. Martinoa, A. W. de Borsta, G. Valentea, R. Goebela, and E. Formisanoa. Predicting EEG single trial responses with simultaneous fmri and relevance vector machine regression. *NeuroImage*, 56(2):826–836, May 2011.

[36] S. J. Marygold, P. C. Leyland, R. L. Seal, J. L. Goodman, J. Thurmond, V. B. Strelets, R. J. Wilson, et al. Flybase: improvements to the bibliography. *Nucleic acids research*, 41(D1):751–757, 2013.

[37] M. Masseroli, D. Chicco, and P. Pinoli. Probabilistic latent semantic analysis for prediction of gene ontology annotations. In *International Joint Conference on Neural Networks*, pages 1 – 8, June 2012.

[38] O. Michael and F. Elana. Matrix factorization for transcriptional regulatory network inference. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on*, pages 387–396. IEEE, 2012.

[39] H. Miyashita, T. Nakamura, Y. Ida, T. Matsumoto, and T. Kaburagi. Nonparametric Bayes-based heterogeneous "drosophila melanogaster" gene regulatory network inference: T-process regression. In *Proceedings of the International Conference on Artificial Intelligence and Applications*, pages 51–58, 2013.

[40] N. B. Mohammed Mohammed-Rasheed and H. Fathallah-Shaykh. A combined constraint approach for inference of sparse large-scale biomolecular networks. In *International Conference on Control, Engineering & Information Technology*, volume 1, pages 38–42, 2013.

[41] V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra Appl.*, 416:29–54, 2006.

[42] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics*, 4(1):53–77, 2010.

[43] N. Pochet, F. D. Smet, J. A. K. Suykens, and B. L. R. D. Moor. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, 20(17):3185–3195, 2004.

[44] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio,

S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, January 2002.

[45] G. Rasool, N. Bouaynaya, H. M. Fathallah-Shaykh, and D. Schonfeld. Inference of genetic regulatory networks using regularized likelihood with covariance estimation. In *IEEE Statistical Signal Processing Workshop*, August 2012.

[46] G. Reinsel and R. Velu. *Multivariate Reduced-rank Regression: Theory and Applications*. Springer, New York, 1998.

[47] J. W. Robinson and A. J. Hartemink. Non-stationary dynamic Bayesian networks. In *Advances in Neural Information Processing Systems*, pages 1369–1376, 2008.

[48] A. J. Rothman, E. Levina, and J. Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 2010.

[49] R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. 33(8):1590–1602.

[50] O. Scherzer. *handbook of Mathematical Methods in Imaging*. Springer, 2011.

[51] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, Marsh 2002.

[52] A. R. Statnikov, C. F. Aliferis, I. Tsamardinos, D. P. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. 21(5):631–643.

[53] G. Tauboeck, F. Hlawatsch, and H. Rauhut. Compressive estimation of doubly selective channels: exploiting channel sparsity to improve spectral efficiency in multicarrier transmissions. *CoRR*, abs/0903.2774, 2009.

[54] M. C. Tresch, V. C. K. Cheung, S. A. O. A. D'avella, G. T. Oviedo, L. H. Ting, N. Krouchev, J. F. Kalaska, T. Drew, J. M. Macpherson, Y. P. Ivanenko, and F. Lacquaniti. Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets. *Journal of Neurophysiology*, 95:2199–2212, January 2006.

[55] J. H. Wang, P. K. Hopke, T. M. Hancewicz, and S. L. Zhang. Application of modified alternating least squares regression to spectroscopic image analysis. *Analytica Chimica Acta*, 2003.

[56] J. J.-Y. Wang, X. Wang, and X. Gao. Non-negative matrix factorization by maximizing correntropy for cancer clustering. *BMC Bioinformatics*, 1(1):1590–1602, 2013.

[57] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31:210–227, February 2009.

[58] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on pattern Analysis and Machine Intelligence*, 31(2):210 – 227, February 2009.

[59] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference*, pages 267–273. ACM Press, 2003.

[60] M. K. S. Yeung, J. Tegner, and J. J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *PNAS*, 99(9):6163–6168, April 2002.

[61] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):1935, 2007.

[62] L. Zhang and S. Kim. Learning gene networks under SNP perturbations using eQTL datasets. *PLoS Computationaly Biology 10(4)*, 10, February 2014.

[63] W. Zhao, E. Serpedin, and E. R. Dougherty. Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics*, 22(17):2129–2135, 2006.

[64] C. H. Zheng, L. Zhang, T. Y. Ng, S. C. Shiu, and D. S. Huang. Metasample-based sparse representation for tumor classification. *IEEE Trans. on Computational Biology and Bioinformatics*, 8:1273–1282, September/October 2011.

Appendix A

To prove the results in Chapter 2, we need to define the notion of an auxiliary function.

Definition A.0.1. *$G(\boldsymbol{h}, \boldsymbol{h'})$ is an auxiliary function for $f(\boldsymbol{h})$ if $G(\boldsymbol{h}, \boldsymbol{h'}) \geq f(\boldsymbol{h})$ and $G(\boldsymbol{h}, \boldsymbol{h}) = f(\boldsymbol{h})$.*

The following lemma in [33] shows the usefulness of the auxiliary function.

Lemma A.0.2. *[33] if G is an auxiliary function, then f is nonincreasing under the update*

$$\boldsymbol{h}^{(k+1)} = \arg\min_{\boldsymbol{h}} G(\boldsymbol{h}, \boldsymbol{h}^{(k)}). \tag{A.1}$$

Proof of Proposition 1. We will prove the update rule for H. A similar reasoning would provide the update rule for W. Consider the two-variable matrix

$$\begin{aligned} G(\boldsymbol{h}, \boldsymbol{h}^{(k)}) = \quad & f(\boldsymbol{h}^{(k)}) + (\boldsymbol{h} - \boldsymbol{h}^{(k)})^T \nabla f(\boldsymbol{h}^{(k)}) + \\ & \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T K_h(\boldsymbol{h}^{(k)})(\boldsymbol{h} - \boldsymbol{h}^{(k)}), \end{aligned} \tag{A.2}$$

where K_h is any function satisfying conditions [a]-[c] and $f(\boldsymbol{h}) = 1/2 \sum_i (\boldsymbol{v}_i - \sum_j W_{ij} \boldsymbol{h}_j)^2$. We show that G is an auxiliary function for f. It is straightforward to verify that

$G(\boldsymbol{h}, \boldsymbol{h}) = f(\boldsymbol{h})$. We only need to show that $G(\boldsymbol{h}, \boldsymbol{h}^k) \geq f(\boldsymbol{h})$. To do this, we compare

$$\begin{aligned} f(\boldsymbol{h}) = \quad & f(\boldsymbol{h}^{(k)}) + (\boldsymbol{h} - \boldsymbol{h}^{(k)})^T \nabla f(\boldsymbol{h}^{(k)}) + \\ & \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T (W^T W)(\boldsymbol{h} - \boldsymbol{h}^{(k)}) \end{aligned} \tag{A.3}$$

With Eq. (A.2) to find that $G(\boldsymbol{h}, \boldsymbol{h}^k) \geq f(\boldsymbol{h})$ is equivalent to

$$(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T [K_h(\boldsymbol{h}^{(k)}) - W^T W](\boldsymbol{h} - \boldsymbol{h}^{(k)}) \geq 0, \tag{A.4}$$

From Condition [c], we have that $K_h - W^T W$ is positive semi-definite; thus, Eq. (A.4) is satisfied and $G(\boldsymbol{h}, \boldsymbol{h}^k) \geq f(\boldsymbol{h})$, proving that G is an auxiliary function of f. We next show that $\boldsymbol{h}$ is positive elementwise at every iteration k. From lemma A.0.2, and taking the derivative of G with respect to $\boldsymbol{h}$, we obtain that

$$\begin{aligned} \boldsymbol{h}^{(k+1)} &= \boldsymbol{h}^{(k)} - K_h^{-1} \nabla f(\boldsymbol{h}^{(k)}) \\ &= \boldsymbol{h}^{(k)} - K_h^{-1}(W^T W \boldsymbol{h}^{(k)} - W^T \boldsymbol{v}) \\ &= [I - K_h^{-1} W^T W] \boldsymbol{h}^{(k)} + K_h^{-1} W^T \boldsymbol{v}, \end{aligned} \tag{A.5}$$

Let us assume that $\boldsymbol{h}^k$ is positive and show that $\boldsymbol{h}^{k+1}$ is also positive. From condition [a], K_h is diagonal and positive (elementwise). Therefore, K_h^{-1} is also diagonal and positive. Given that W and V are also positive, we have that $K_h^{-1} W^T \boldsymbol{v}$ is positive. From condition [b], we have that $[I - K_h^{-1} W^T W] \boldsymbol{h}^{(k)}$ is positive. Thus, $\boldsymbol{h}^{k+1}$ is positive

(elementwise). In particular, by choosing the initial point $\boldsymbol{h}^0$ positive, all iterations $\boldsymbol{h}^k$ are guaranteed to be positive.

This ends the proof of Proposition 1. Next, we show that Lee and Seung's choice of $(K_h)_{ij} = \delta_{ij}(W^T W \boldsymbol{h}^{(k)})_i / \boldsymbol{h}_i^{(k)}$ corresponds to the fastest convergent update rule among the class of matrices K_h that satisfy conditions [a]-[c].

From Eq. (A.5), we have

$$\begin{aligned} \|\boldsymbol{h}^{(k+1)} - \boldsymbol{h}^{(k)}\| &= \|K_h^{-1}(W^T W \boldsymbol{h}^{(k)} + W^T \boldsymbol{v})\| \\ &\leq \|K_h^{-1}\| \|W^T W \boldsymbol{h}^{(k)} + W^T \boldsymbol{v}\|. \end{aligned} \tag{A.6}$$

Thus, the smaller the norm of K_h (or the larger the norm of K_h^{-1}), the faster the convergence rate. From condition (b), we have that $K_h \boldsymbol{h}^k \geq W^T W \boldsymbol{h}^k$. Hence, the smallest choice of K_h corresponds to $(K_h)_{ij} = \delta_{ij}(W^T W \boldsymbol{h}^{(k)})_i / \boldsymbol{h}_i^{(k)}$. □

Proof of Proposition 2. The following lemma provides an auxiliary function for the objective function f in (2.11).

Lemma A.0.3. *Consider the diagonal matrix*

$$\Phi_{ij}(\boldsymbol{h}^{(k)}) = \delta_{ij}(W^T W \boldsymbol{h}^{(k)})_i / \boldsymbol{h}_i^{(k)} + \beta. \tag{A.7}$$

We show that

$$\begin{aligned} G(\boldsymbol{h}, \boldsymbol{h}^{(k)}) = \quad & f(\boldsymbol{h}^{(k)}) + (\boldsymbol{h} - \boldsymbol{h}^{(k)})^T \nabla f(\boldsymbol{h}^{(k)}) + \\ & \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T \Phi(\boldsymbol{h}^{(k)})(\boldsymbol{h} - \boldsymbol{h}^{(k)}) \end{aligned} \tag{A.8}$$

is an auxiliary function for $f(\boldsymbol{h}) = \sum_i (\boldsymbol{v}_i - \sum_j W_{ij}\boldsymbol{h}_j)^2 + \alpha\|W\|_F^2 + \beta\sum_i \|\boldsymbol{h}_i\|^2$.

The fact that $G(\boldsymbol{h}, \boldsymbol{h}) = f(\boldsymbol{h})$ is obvious. Therefore, we need only to show that $G(\boldsymbol{h}, \boldsymbol{h}^{(k)}) \geq f(\boldsymbol{h})$. To do this, we compare

$$\begin{aligned} f(\boldsymbol{h}) = \quad & f(\boldsymbol{h}^{(k)}) + (\boldsymbol{h} - \boldsymbol{h}^{(k)})^T \nabla f(\boldsymbol{h}^{(k)}) + \\ & \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T (W^T W + \beta I)(\boldsymbol{h} - \boldsymbol{h}^{(k)}) \end{aligned} \tag{A.9}$$

with Eq. (A.8) to find that $G(\boldsymbol{h}, \boldsymbol{h}^{(k)}) \geq f(\boldsymbol{h})$ is equivalent to

$$(\boldsymbol{h} - \boldsymbol{h}^{(k)})^T [K(\boldsymbol{h}^{(k)}) - W^T W](\boldsymbol{h} - \boldsymbol{h}^{(k)}) \geq 0, \tag{A.10}$$

The proof of the semi-definiteness of the matrix in (A.10) is provided in [33]. Replacing G in Eq. (A.2) by its expression in Eq. (A.8) results in the update rule

$$\boldsymbol{h}^{(k+1)} = \boldsymbol{h}^{(k)} - \Phi(\boldsymbol{h}^{(k)})^{-1} \nabla f(\boldsymbol{h}^{(k)}). \tag{A.11}$$

Since G is an auxiliary function of f, f is non-increasing under this update rule. Writing the components of Eq. (A.11), we obtain

$$\boldsymbol{h}_i^{(k+1)} = \boldsymbol{h}_i^{(k)} \frac{(W^T \boldsymbol{v})_i}{(W^T W \boldsymbol{h}^{(k)} + \beta \boldsymbol{h}^{(k)})_i}. \tag{A.12}$$

Similarly, we can obtain the update rule for W. □

Proof of Corollary 2.2.1. Consider the diagonal matrices

$$(K_h)_{ij} = \delta_{ij}(W^T W H^k)_{ij}/H^k_{ij}. \tag{A.13}$$

$$(K_w)_{ij} = \delta_{ij}(W_k H H^T)_{ij}/W^k_{ij}. \tag{A.14}$$

It can be easily shown that K_h and K_w in Eqs. (A.13) and (A.14) satisfy conditions [a]-[c]. Corollary 2.2.1 follows directly from Proposition 1 by choosing K_h and K_w in proposition 1 as above. □