UNIVERSITY OF CALIFORNIA

Los Angeles

Numerical Optimization Methods for Image Processing and Machine Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Joseph Woodworth

2016

Abstract of the Dissertation

Numerical Optimization Methods for Image Processing and Machine Learning

by

Joseph Woodworth

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2016

Professor Andrea Bertozzi, Chair

In this dissertation, numerical optimization methods for three different classes of problems are presented : statistical modeling of crime, compressive sensing, and ordinal embedding. A common aspect of each of these problems is their need for computational efficiency. The input data sets can be large or high-dimensional, so each method sparsifies or reduces the dimension of the data, while preserving essential structure. We make use of several popular paradigms of modern optimization such as non-local operators, randomized sampling, sparse regulation, relaxations of intractable problems, and divide-and-conquer. The novel variations and analysis of these approaches suggest promising directions of further study in image processing and machine learning.

First, we are given a discrete sample of event locations, and we wish to produce a probability density that models the relative probability of events occurring in a spatial domain. Standard density estimation techniques do not incorporate priors informed by spatial data. Such methods can result in assigning significant positive probability to locations where events cannot realistically occur. In particular, when modeling residential burglaries, standard density estimation can predict residential burglaries occurring where there are no residences. Incorporating the spatial data can inform the valid region for the density. When modeling very few events, additional priors can help to correctly fill in the gaps. Learning and enforcing correlation between spatial data and event data can yield better estimates from fewer events. We propose a non-local version of Maximum

Penalized Likelihood Estimation based on the H^1 Sobolev seminorm regularizer that computes non-local weights from spatial data to obtain more spatially accurate density estimates. We evaluate this method in application to a residential burglary data set from San Fernando Valley with the non-local weights informed by housing data or a satellite image.

Second, we analyze a method for compressed sensing. The ℓ^0 minimization of compressed sensing is often relaxed to ℓ^1, which yields easy computation using the shrinkage mapping known as soft thresholding, and can be shown to recover the original solution under certain hypotheses. Recent work has derived a general class of shrinkages and associated nonconvex penalties that better approximate the original ℓ^0 penalty and empirically can recover the original solution from fewer measurements. We specifically examine *p-shrinkage* and *firm thresholding*. We prove that given data and a measurement matrix from a broad class of matrices, one can choose parameters for these classes of shrinkages to guarantee exact recovery of the sparsest solution. We further prove convergence of the algorithm *iterative p-shrinkage* (IPS) for solving one such relaxed problem.

Lastly, we consider the problem of embedding unweighted, directed k-nearest neighbor graphs in low-dimensional Euclidean space. The k-nearest neighbors of each vertex provide ordinal information on the distances between points, but not the distances themselves. Relying only on such ordinal information, along with the low-dimensionality, we recover the coordinates of the points up to arbitrary similarity transformations (rigid transformations and scaling). Furthermore, we also illustrate the possibility of robustly recovering the underlying density via the Total Variation Maximum Penalized Likelihood Estimation (TV-MPLE) method. We make existing approaches scalable by using an instance of a local-to-global algorithm based on group synchronization, recently proposed in the literature in the context of sensor network localization, and structural biology, which we augment with a scale synchronization step. We show our approach compares favorably to the recently proposed Local Ordinal Embedding (LOE) algorithm even in the case of smaller sized problems, and also demonstrate its scalability on large graphs. The above divide-and-conquer paradigm can be of independent interest to the machine learning community when tackling geometric embeddings problems.

The dissertation of Joseph Woodworth is approved.

Paul Jeffrey Brantingham

Luminita Aura Vese

Stanley J. Osher

Andrea Bertozzi, Committee Chair

University of California, Los Angeles

2016

To my family, friends, teachers, and collaborators . . .

who give me invaluable guidance and support.

Table of Contents

List of Figures

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Andrea Bertozzi for her continued guidance and support, which made possible all of the research presented in this dissertation. She has been a great mentor for me. Andrea, thank you for introducing me to interesting problems and methodologies, as well researchers I have learned a lot from, including Mihai Cucuringu, Rick Chartrand, Brendt Wohlberg, George Mohler, and several others.

I also wish to thank my other committee members, Jeff Brantingham, Stan Osher, and Luminita Vese, for their time, advice, and instruction. They demonstrate the ideals of scientific research that I strive to achieve. In addition to the professors I have had research discussions and collaborations with, I am very grateful to the instructors of all of my courses at UCLA who imparted upon me the foundational knowledge to pursue research at a professional level, and an appreciation for several disciplines outside my main field of study. I owe much to my collaborators Mihai Cucuringu, Rick Chartrand, Brendt Wohlberg, George Mohler, and Alan Mackey, for their ideas and contributions to our work. They have helped shape me as a mathematician.

Chapter 2 is a version of [138]. George Mohler advised in the statistical tests of the method, Andrea Bertozzi advised in the use of graph methods, and Jeff Brantingham advised on the sociological context of such an approach. Chapter 3 is a version of [137]. Rick Chartrand advised on the properties of generalized shrinkage penalties, prior methods in compressed sensing, and techniques for proving convergence. Chapter 4 is a version of [43]. Mihai Cucuringu advised on the ASAP framework and its prior applications.

The research presented here was made possible in part by a Girsky Fellowship from the Girsky Students Award Fund, NSF grant no. DGE-1144087, DMS-0968309, and DMS-1417674; the UC Lab Fees Research grant 12-LR-236660, the W. M. Keck Foundation, ONR grants no. N000141210040 and N000141210838, AFOSR MURI grant no. FA9550-10-1-0569, ARO grant no. W911NF1010472, and the U.S. Department of Energy through the LANL/LDRD Program.

Vita

2010	Daniel Sweet Memorial Fellow, Norbert Wiener Center for Harmonic Analysis and its Applications
2011	B.S. Mathematics, University of Maryland College Park, *Magna Cum Laude*
2011 - 2015	NSF Graduate Research Fellow
2012 Summer	Undergraduate Research Project Mentor, UCLA REU
2012-2013, 2015	Graduate Teaching Assistant, UCLA
2013 Summer 2014 Summer	Research Assistant, Los Alamos National Laboratory
2013-2015	Graduate Research Assistant, UCLA
2015 Summer	Software Engineer Intern, Google
2015 - 2016	Girsky Fellowship Award

Publications

Mihai Cucuringu, and Joseph Woodworth. Parallel Point Localization and Density Estimation from Ordinal kNN graphs using Eigenvector Synchronization. *Submitted*.

Mihai Cucuringu, and Joseph Woodworth. Ordinal embedding of unweighted kNN graphs via synchronization. *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on, 2015.*

Joseph Woodworth, and Rick Chartrand, Compressed Sensing Recovery via Nonconex Shrinkage Penalties. *Submitted*, preprint : `http://arxiv.org/abs/1504.02923.`

Joseph Woodworth, George Mohler, Andrea Bertozzi and P. Jeff Brantingham. Non-local Crime Density Estimation Incorporating Housing Information. *Philosophical Transactions of the Royal Society of London A : Mathematical, Physical, and Engineering Sciences* , November 2014 Vol. 372 Issue 2028, 20130403.

John Benedetto, Rob Benedetto, and Joseph Woodworth. Optimal ambiguity functions and Weil's exponential sum bound. *Journal of Fourier Analysis and Applications*, June 2012 Vol. 18 Issue 3 pp 471-487.

CHAPTER 1

Introduction

This dissertation is based on the work from the following published and submitted papers : Nonlocal Crime Density Estimation Incorporating Housing Information [138], Compressed Sensing Recovery via Nonconvex Shrinkage Penalties [137], and Ordinal Embedding Of Unweighted kNN Graphs Via Synchronization [43]. These are all collaborative works. The author of this dissertation was the primary or equal contributor to each of these works. See the Acknowledgements for specific lists of co-author contributions.

In real-world applications, satellite images, housing data, census data, and other types of geographical data become highly relevant for modeling the probability of a certain type of event. The methodology presented in Chapter 2 provides a general framework paired with fast algorithms for incorporating external information in density estimation computations.

In density estimation, one is given a discrete sample of event locations, drawn from some unknown density u on the spatial domain, and tries to approximately recover u [118]. Relating the events to the additional data allows one to search over a smaller space of densities, which can yield more accurate results with fewer events. We refer to the additional data source as the function $g(x)$ defined over the spatial domain Ω. We may typically assume two things about the relationship between g and u : 1) g informs the support of u via $g(x) = 0 \Rightarrow u(x) = 0$ and 2) u varies smoothly with g in a non-local way (explained below). This method allows the additional information in g to significantly improve the recovery of u.

Although there are other classes of methods in the density estimation literature which are quite popular (such as average shifted histogram and kernel density estimation [109]), in Chapter 2 we shall focus on Maximum Penalized Likelihood Estimation (MPLE). MPLE provides a general framework for finding an approximate density from sampled events. The likelihood of events occurring at the locations $\{x_i\}_{i=1}^n$ according to a proposed probability u is the product of the probability evaluated at each of those locations:

$$\mathcal{L}\left(u, \{x_i\}_{i=1}^n\right) = \prod_{i=1}^n u(x_i).$$

MPLE approximates u as the maximizer of a log-likelihood term combined with a penalty term, typically enforcing smoothness [51],

$$\hat{u} = \underset{u \geq 0, \int_\Omega u dx = 1}{\arg\max} \sum_{i=1}^n \log(u(x_i)) - P(u).$$

Without some kind of penalty term, the solution is just a weighted sum of Dirac deltas located at the training samples. Typical choices of $P(u)$ include the TV-norm, $P(u) = \lambda \int_\Omega |\nabla u| dx,$ and the H^1 Sobolev seminorm $P(u) = \frac{\lambda}{2} \int_\Omega |\nabla u|^2 dx.$ λ is the parameter which controls the amount of regularization. This is typically chosen via cross-validation, when it is computationally feasible.

The H^1 seminorm is a common, well-understood regularizer in image processing related to Poisson's equation, the heat equation, and the Weiner filter, producing visually smooth surfaces. For this reason, it is often a default choice when little is known about the data being modeled. H^1 MPLE has further justification in crime density estimation from the ”broken window” effect [136, 116, 115]. This observation states that after a burglary has occurred at a given house, burglaries are more likely to occur at the same house or nearby houses for some period of time afterwards. Initial burglaries give criminals information about what valuables remain and the schedule of inhabitants in the area. Additionally, a successful burglary leaves environmental clues, such as broken windows, that indicate an area is more crime-tolerant than others. This effect leads to repeat and near-repeat burglaries. More generally, criminals tend to move in a bounded region around a few key nodes and have limited awareness of potential for criminal activity outside of familiar areas [15, 9, 114]. Within neighborhoods, risk factors are typically homogeneous[127, 88, 80]. All of this explains why observed incidence rates of burglaries are locally smooth.

However, local smoothness is not always appropriate and in practice there is much room for improvement. In recent years several studies on the application of MPLE to crime data [97, 120, 81] emphasize the fact that crime density should have boundaries corresponding to the local geography. Mohler et al. and Kostic et al. model this by choosing penalty functions that are edge-preserving, TV and Ginzburg-Landau respectively [97, 81]. Smith et al. more closely follows the idea presented here. That work introduces a modified H^1 MPLE, which based the penalty term on an additional component of the data [120]. The method assumes that the the valid region of the probability density estimate is known a priori. In their application to residential burglary the valid region was the approximate support of the housing density in the region. If we denote the valid region by D, then the modified penalty term is just a standard H^1 MPLE with a factor z_ϵ^2 in the integral, where z_ϵ is a smooth Ambrosio-Tortorelli approximation of $(1 - \delta(\partial D))$:

$$\hat{u} = \underset{u \geq 0, \int_\Omega u = 1}{\arg\min} \frac{1}{2} \int_\Omega z_\epsilon^2 |\nabla u|^2 dx - \mu \sum_{i=1}^{n} \log(u(x_i)),$$

$$z_\epsilon(x) = \begin{cases} 1 & \text{if } d(x, \partial D) > \epsilon, \\ 0 & \text{if } x \in \partial D. \end{cases}$$

In spectral graph theory, data is represented as nodes of a weighted graph, where the weight on each edge indicates the similarity between the two nodes. Such data structures have been very successfully applied to data clustering problems and image segmentation [40, 69, 113]. The standard theory behind this is described in [35, 96] and a tutorial on spectral clustering is given in [129]. A theory of non-local calculus was developed first by Zhou and Schölkopf in 2004 [144] and put in a continuous setting by Gilboa and Osher in 2008 [62]. Such methods were originally used for image denoising [19, 62], but the general framework led to methods for inpainting, reconstruction, and deblurring [94, 61, 143, 101, 89]. Compared with local methods, non-local methods are generally better able to handle images with patterns and texture. Further, by choosing an appropriate affinity function, the methods can be made suitable for a wide variety of different of data sets : not just images.

Chapter 2 presents non-local H^1 MPLE (NL H^1 MPLE), which modifies the standard H^1

MPLE energy to account for spatial inhomogeneities, but unlike Smith et al. [120], we do so in a non-local way, which has the benefit of leveraging recent fast algorithms and the potential to generalize to other applications.

The organization of the chapter is as follows: In Sec. 2.1, we introduce the NL H^1 MPLE method and review the non-local calculus and numerical methods on which it is based. In Sec. 2.2 we demonstrate the advantages of NL H^1 MPLE by comparing it with standard H^1 MPLE when applied to modeling residential burglary. In Sec. 2.3 we summarize our conclusions and discuss directions for future research.

Chapter 3 analyzes a method for compressed sensing. Compressed sensing has been successfully applied in a multitude of scientific fields, ranging from image processing tasks to radar to coding theory, making the potential impact of advancements in theory and practice rather large. Compressed sensing methods rely on the notion of sparsity, which is primarily approximated via the ℓ^1 norm [22, 47]. The nature and limitations of this relaxation have been well-studied [18, 20, 45, 48, 52, 53, 54, 72, 73], as well as some alternative relaxations, such as the ℓ^p quasinorm [32, 139, 3, 34, 45, 55, 72, 73, 85, 107, 121, 122]. The nonconvex ℓ^p quasinorm approaches present a tradeoff: closer approximation of sparsity for harder analysis and computation. Recent work has introduced generalized nonconvex penalties [27, 29, 31, 33, 30, 28, 4] that have thus far demonstrated strong empirical performance [27, 132, 31, 30]. In Chapter 3, we prove conditions that guarantee good performance of these generalized penalties.

Compressed sensing seeks to represent a signal from a small number of linear measurements. We let the vector $x \in \mathbb{R}^n$ represent the original signal. The linear measurements are the result of an application of the short and fat measurement matrix $A \in \mathbb{R}^{m \times n}$, with $m \ll n$. One is given the measurements $b := Ax$ and wants to recover x. Of course $m \ll n$ implies that $Ax = b$ is an underdetermined linear system in x, so additional assumptions must be made about x. Thus one assumes that x is *sparse*, meaning that it has few nonzero entries. By considering the standard definition of p norms for vectors,

$$\|w\|_p^p := \sum_i |w_i|^p, \tag{1.1}$$

and taking the limit as p approaches 0 from above, we get the ℓ^0 penalty, $\|w\|_0$, which counts the number of nonzero entries of w. One would like to find the sparsest vector $w \in \mathbb{R}^n$ whose measurements are b, which suggests the following optimization problem:

$$\min_w \|w\|_0 \text{ subject to } Aw = b. \tag{1.2}$$

Unfortunately, this problem is known to be NP-hard (Non-deterministic Polynomial-time hard) in general [99, Sec. 9.2.2]. In other words, without making further assumptions on A and x, an algorithm solving this problem would be computationally intractable. For this reason, one relaxes the problem, replacing the ℓ^0 penalty with other penalties.

The ℓ^1 relaxed version of the compressed sensing problem is as follows:

$$\min_w \|w\|_1 \text{ subject to } Aw = b. \tag{1.3}$$

In contrast to the combinatorial ℓ^0 problem, this problem minimizes a convex energy subject to linear constraints, and can be recast as a linear program. Extensive theory has been developed to study the properties of solutions to convex problems [17]. Further, a subproblem related to the ℓ^1 relaxation of compressed sensing has a closed-form solution, given by an application of a shrinkage operator:

Definition 1.0.1. Soft thresholding is given by the following formula:

$$S_{\lambda,1}(x)_i = s_{\lambda,1}(|x_i|)\operatorname{sign}(x_i) = \max\{|x_i| - \lambda, 0\}\operatorname{sign}(x_i). \tag{1.4}$$

The role soft thresholding plays is as the *proximal mapping* of the ℓ^1 norm:

$$S_{\lambda,1}(x) = \operatorname{prox}_\lambda \|\cdot\|_1(x) := \arg\min_w \lambda\|w\|_1 + \tfrac{1}{2}\|w-x\|_2^2. \tag{1.5}$$

Several algorithms for compressed sensing make use of this proximal mapping, such as iterative soft thresholding [44], alternating direction method of multipliers (ADMM) [63, 58, 65, 16], and the Chambolle-Pock algorithm [23]. The explicit formula for (1.5) makes the use of ℓ^1 regularization particularly convenient.

All of this suggests why the ℓ^1 relaxation of compressed sensing is nice to solve, but does not motivate it as the right problem to solve. In particular, one is interested in conditions under which the solution to the ℓ^1 relaxation (1.3) of compressed sensing equals or approximately equals the solution of the original ℓ^0 compressed sensing problem (1.2). The papers [22, 47] developed theory for the recovery of the ℓ^0 solution by the ℓ^1 problem. In the years the followed, getting looser conditions for exact ℓ^1 recovery received continuing interest [18, 20, 45, 48, 52, 53, 54, 55, 72, 73]. One type of condition for recovery of the ℓ^0 solution from the ℓ^1 problem relies on the *restricted isometry constants* associated with the measurement matrix A. The restricted isometry constant of order k associated with the matrix $A \in \mathbb{R}^{m\times n}$ is the smallest $\delta_k \geq 0$ such that the following holds for all $x \in \mathbb{R}^n$ with $\|x\|_0 \leq k$ [21]:

$$(1-\delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_k)\|x\|_2^2. \tag{1.6}$$

Note that when $\delta_k > 1$ the lower bound becomes trivial and the upper bound can be improved by rescaling A. Thus any measurement matrix, with appropriate rescaling, can achieve $\delta_k = 1$, so one typically only regards $\delta_k \in [0, 1)$. One of the best current ℓ^1 recovery results states that for sufficiently large n, a sparse vector $x \in \mathbb{R}^n$ with $\|x\|_0 = k$ can be recovered by ℓ^1 minimization as long as $k < m/2$ and the restricted isometry constant of order $2k$ associated with A satisfies $\delta_{2k} \leq 1/2$ [20].

A similar relaxation of the ℓ^0 problem that achieves recovery results in broader cases is ℓ^p minimization for $0 < p < 1$. In contrast to the ℓ^1 norm, the ℓ^p quasinorms for $0 < p < 1$

are not convex. Hence much of the theory of convex analysis no longer applies, making solution uniqueness and convergence results more complicated. However, the loss of convexity comes with the benefit that ℓ^p is better able to approximate the original ℓ^0 than ℓ^1 can. As a result, one can show that for any given measurement matrix with restricted isometry constant $\delta_{2k} < 1$, there exists some $p \in (0, 1)$ that will guarantee exact recovery of signals with support smaller than $k < m/2$ by the ℓ^p minimization problem [139]. It has also been demonstrated empirically that ℓ^p minimization gives better sparse recovery results than ℓ^1 minimization [24, 26, 25], with improved robustness [107, 3, 121].

Consider the proximal mapping of the ℓ^p quasinorm (to the p^{th} power, for simplicity), that is,

$$\operatorname{prox}_\lambda \| \cdot \|_p^p(x) := \arg\min_w \lambda \|w\|_p^p + \tfrac{1}{2}\|w - x\|_2^2. \tag{1.7}$$

Unfortunately, (1.7) is a discontinuous mapping [142], and there is no closed-form expression for (1.7) for general p. (The expression given in [90] is incorrect. For the special cases of $p = 1/2$ or $2/3$, the proximal mapping can be expressed in terms of the solution of a cubic or quartic equation, explicitly but cumbersomely.) This prevents several efficient algorithms from being generalized from ℓ^1 to ℓ^p minimization.

The need for an explicit proximal mapping motivates the approach of specifying a shrinkage mapping, and minimizing an implicitly-defined penalty function whose proximal mapping is the specified shrinkage [4, 27, 29, 31]. In Chapter 3, we extend theoretical results for recovery of sparse signals to the case of penalty functions induced by two families of shrinkages, p-shrinkage and firm thresholding (see Defs. 3.1.1, 3.1.2 below). While some recovery results corresponding to a firm thresholding algorithm are presented in [49], those results are probabilistic in nature, characterizing probabilities in the limiting behavior as the dimension approaches infinity, in contrast to the deterministic results presented here. In Section 3.1 we describe these shrinkage mappings, and how they are the proximal mappings of nonconvex penalty functions. In Section 3.2 we prove conditions for the exact recovery of sparse signals via minimizing such nonconvex penalty functions. In Section 3.3 we demonstrate the stability of signal recovery to noisy measurements and approximately sparse signals, and in Section 3.4 we show the algorithmic convergence of *iterative p-shrinkage* (IPS).

In Chapter 4 we present a method for embedding unweighted k-nearest neighbor (kNN) graphs. This is a special case of ordinal or non-metric embedding, where one seeks a spatial embedding of n points $\{\vec{x}_i\}_{i=1}^n$ in $\mathbb{R}^d$ such that

$$\forall (i_1, j_1, i_2, j_2) \in \mathcal{C}, \quad \|\vec{x}_{i_1} - \vec{x}_{j_1}\|_2 < \|\vec{x}_{i_2} - \vec{x}_{j_2}\|_2, \tag{1.8}$$

where $\mathcal{C}$ denotes the set of ordinal constraints. Ordinal constraints are sometimes also specified as triplets [1]. In the case of unweighted kNN graph embedding,

$$\mathcal{C} = \mathcal{C}(G) = \left\{ (a, b, a, c) \middle| ab \in E(G), ac \notin E(G) \right\},$$

where $E(G)$ is the set of directed edges in the kNN graph G.

Graph-based methods are of utmost importance in several modern machine learning methods with applications such as clustering, dimensionality reduction, and ranking. Many such methods rely on weighted graphs, with weights often based on similarity functions, i.e., $w_{ij} = S(x_i, x_j)$. From a practical standpoint, storing unweighted kNN graphs instead would allow for a very sparse representation of the data. If one could recover the source data $\{x_i\}_{i=1}^n$ from unweighted kNN graphs, such a computationally efficient sparser representation would incur no information loss. Because of the extreme sparsity of the representation, this is generally a hard problem. Just recently, a method for recovering data distributions from unweighted kNN graphs was introduced in [130]. Another motivation for this problem comes from an instance of the popular sensor network localization problem, where each sensor is able to transmit only limited connectivity information to a central location (ID names of its k nearest neighbors), but transmits neither the distance measurements nor a complete list of all its neighbors within a given fixed radius. While most of kNN-related work in the literature focuses on actually efficiently finding the nearest neighbors in the given space[140], we consider the inverse problem of recovering the embedding from the kNN queries.

The original work on this problem dates back to Shepard [112] and Kruskal [82, 83], and lately has been studied intensively in the machine learning literature [1, 111, 92, 102, 123, 78, 105, 2, 100, 93, 77, 86, 134]. In Chapter 4, we compare against and extend the recent Local Ordinal Embedding method [124], which enjoys several favorable comparisons with other modern methods. Another motivation for this problem comes from an instance of the popular sensor network localization problem, where each sensor is able to transmit only limited connectivity information to a central location, in the form of ID names of its k nearest neighbor sensors, but transmits neither the estimated distance measurements nor a complete list of all its neighbors within a given fixed radius. Note that either of these last two scenarios renders the localization problem (of estimating the sensor coordinates) easier to solve. Similar to the sensor network application, one could potentially apply this framework to cooperative control and sensing involving swarms of robot micro-vehicles with limited payloads communicating via radio with limited bandwidth [91, 66]. Our key ingredient is a modified version of the As-Synchronized-As-Possible (ASAP) algorithm introduced in [41], which makes existing embedding methods scalable via a divide-and-conquer, non-iterative local to global approach, reduces computational complexity, allows for massive parallelization of large problems, and increases robustness to noise. The ASAP algorithm introduced in [41], on which we rely in Chapter 4, renders our approach to reconstruct kNN graphs scalable to graphs with thousands or even tens of thousands of nodes, and is an example of a local-to-global approach that integrates local ordinal information into a global embedding calculation.

We detail in Section 4.2.1 the exact approach used to decompose the initial kNN graph into many overlapping subgraphs, that we shall refer to as patches from now on. Each resulting patch is then separately embedded in a coordinate system of its own using an ordinal embedding algorithm, such as the recent Local Ordinal Embedding (LOE) algorithm [124]. In the hypothetical scenario when LOE recovers the actual ground truth coordinates of each patch, such local coordinates agree with the global coordinates up to scaling and some unknown rigid motion (such as rotation, reflection and translation), in other words, up to a similarity transformation. However, in most practical instances, it is unreasonable to expect that the LOE algorithm will recover the exact coordinates only from ordinal data. On a related note, we point out the recent work of Kleindessner and von

Luxburg [131], who settled a long-known conjecture claiming that, given knowledge of all ordinal constraints of the form $||x_i - x_j|| < ||x_k - x_l||$ between an unknown set of points $x_1, \ldots, x_n \in \mathbb{R}$ (for finite n), it is possible to approximately recover the ground truth coordinates of the points up to similarity transformations. Furthermore, the same authors show that the above statement holds even when we only have **local** information such as the distance comparisons between points in small neighborhoods of the graphs, thus giving hope for a local-to-global approach, in the spirit of the one we propose in Chapter 4.

Our contributions are: **1.** We present a local-to-global approach for the problem of embedding clouds of points from ordinal information, which is scalable to very large graphs, and can be computed efficiently and robustly in a distributed manner. Specifically, we extend the ASAP framework to the setting of ordinal embeddings, by augmenting it with a scale synchronization step. We believe that local-to-global strategies could benefit many problems in the machine learning community. The scale of data involved in many interesting problems poses a challenge to direct, holistic approaches. **2.** We extend the ordinal embedding pipeline to perform density estimation via Total Variation Maximum Penalized Likelihood Estimation. This demonstrates the similarity between the point localization and density estimation problems. Sufficiently simple point distributions can be well estimated by applying a short postprocessing step to an approximate embedding. **3.** We present preliminary results for a very simple, straightforward ordinal embedding method.

The chapter is organized as follows. Section 4.1 is a summary of existing methods for related embedding problems. Section 4.2 details the pipeline of the ASAP framework, including the scale synchronization step in Section 4.2.2. In Section 4.3 we remark on the connection to the density estimation problem, and describe the post-processing step performed via Total-Variation Maximum Penalized Likelihood Estimation. Section 4.4 shows the results of several experiments recovering point embeddings from a variety of data sets, and compares to the existing LOE algorithm, as well as presenting results for the density estimation problem. In Section 4.5 we discuss an entirely different approach to ordinal embedding, and present some preliminary results which suggest more modifications are needed. We conclude our primary discussion in Section 4.6 and summarize in Appendix B some related rigidity theory.

Finally, we would like to make the disclaimer that some symbols and notation have distinct use in each of the following chapters. For example n denotes either the number of events, the dimension of the sparse vector, or the number of nodes in a graph. We use standard notation in these instances rather than use distinct but non-standard notation. We intend to make the specific meaning of each multiply-defined symbol clear from context. Moreover, each chapter's notation is self-contained, e.g. in Chapter 3 k always refers to the number of non-zero elements of the vector, while in Chapter 4 k always refers to the number of nearest neighbors in the kNN graph.

CHAPTER 2

Non-local Crime Density Estimation Incorporating Houses

2.1 Non-Local Crime Density Estimation

We propose replacing the H^1 seminorm regularizer of H^1 MPLE with a linear combination of an H^1 regularizer and a non-local smoothing term $\iint_{\Omega\times\Omega} (\nabla_{w,s} u(x,y))^2 \, dxdy$ where $\nabla_{w,s}$ denotes the non-local symmetric-normalized gradient depending on an affinity function w derived from the spatial data, g. More details are found in Sec. 2.1.2. The energy we optimize is thus

$$\hat{u} = \underset{u\geq 0, \int_\Omega u=1}{\arg\max} \sum_{i=1}^{n} \log\left(u(x_i)\right) - \alpha \iint_{\Omega\times\Omega} (\nabla_{w,s} u(x,y))^2 \, dxdy - \frac{\beta}{2}\int_\Omega |\nabla u(x)|^2 dx. \tag{2.1}$$

The non-local term in equation (2.1) is tolerant of sharp changes in the probability density estimate, as long as they coincide with sharp non-local changes in the spatial data. The mathematical formulation of this statement follows from the definitions presented in the following sections and is presented in the Appendix A. Before reviewing the non-local calculus behind this energy, we motivate why a non-local regularizer is good for crime density estimation. Many cities grow in a dispersal colony-like fashion, i.e. colony patches start growing at dispersed location at the same time with the same architectural or cultural model as a starting point, generating non-local similarities [75]. Dissimilar colony patches grow and meet to form diffuse interface-like boundaries [6]. Thus housing data typically contains similar features spread across the domain, along with interfaces between different types of areas. Whereas opposite sides of these interfaces are spatially close, they are non-locally well-separated.

The clearest advantage of non-local regularization is that it allows for sharp changes in crime density across interfaces of distinct housing regions. In particular, since the residential areas are non-locally well-separated from the non-residential areas, the non-local regularized estimate correctly captures the support of the residential burglary density. This feature has been studied for its own sake in prior work and non-local regularization addresses it in an automatic, hands-off way.

Another, more subtle advantage of non-local regularization is that it encourages distant, but non-locally similar regions (e.g. colony patches based on the same model) to have similar crime density values. The assumption is that the layout of a neighborhood and its crime density are both tied to underlying socio-economic factors. When one has these relevant factors, one can perform Risk Terrain Modeling [80], combining the factors in the way that is most consistent with the observed data. Non-Local regularization implicitly measures correlation between housing features and levels of crime, presumably explained by these unknown factors. The regularization encourages those relationships to remain consistent across the entire domain and all data. In this work, we base the non-local similarity of two locations on the similarity of surrounding housing density patches. For simplicity, one could consider basing it on only the housing density in the immediate vicinity. This would encourage the crime density to be a smooth function of the immediate housing density. Likely, one would estimate residential burglaries as roughly proportional to the housing density. This would be a simple, but reasonable null model, assuming that burglary depends heavily on opportunity. One would balance the spatial smoothness and smoothness as a function of housing density with cross-validation, allowing for varying results depend on what the data shows. Our non-local weights are based on housing density patches, which makes them more noise-robust and representative of more complex housing features. This approach is general, relates to previous work in image processing, and produces favorable results.

2.1.1 Non-Local means

Non-Local means was originally developed for the application of image denoising, but can also be interpreted as an affinity function. The formula for the non-local means affinity, $w_{\mathbf{Im}}$, is given by

[19]

$$w_{\mathbf{Im}}(x,y) = \exp\left(-\frac{\left(K_r * |\mathbf{Im}(x+\cdot) - \mathbf{Im}(y+\cdot)|^2\right)(0)}{\sigma^2}\right). \tag{2.2}$$

Here **Im** is the image the non-local means weights are based on, K_r is a nonnegative weight kernel of size $(2r+1)\times(2r+1)$, and σ is a scaling parameter. This function measures similarity between two pixels based on a weighted ℓ_2 difference between patches surrounding them in the image. In our experiments, the image **Im** is either a housing image or a satellite image. In practical settings, computing and storing all function values of w is a very computationally intensive task, so we use the fast approximation : Nyström's extension (see Sec. 2.12.1.4).

2.1.2 Non-Local calculus and graphs

Non-Local calculus was introduced in its discrete form by Zhou and Schölkopf [144] and put in a continuous framework by Gilboa and Osher [62]. In these definitions, $w(x,y)$ is a general nonnegative symmetric affinity function which generally measures similarity between the points x and y.

Let $\Omega \subset \mathbb{R}^n$, and $u(x)$ be a function $u : \Omega \to \mathbb{R}$. Then the non-local gradient of u at the point $x \in \Omega$ in the direction of $y \in \Omega$ is given by

$$\left(\nabla_w u\right)(x,y) = \left(u(y) - u(x)\right)\sqrt{w(x,y)}.$$

This suggests an analogous generalization of divergence, which in turn leads to the following definition of the non-local Laplacian:

$$\Delta_w u(x) = \int_\Omega \left(u(y) - u(x)\right) w(x,y)dy. \tag{2.3}$$

Now let $\{p_i\}_{i=1}^n$ be a discrete subset of Ω and let $w_{ij} = w(p_i,p_j)$ if $i \neq j$ and $w_{ii} = 0$. We then let $\{p_i\}_{i=1}^n$ be vertices and w_{ij} the edge weights on a weighted graph. Let $d_i = \sum_{j=1}^n w_{ij}$ be the weighted degree of the ith node. Then the graph Laplacian applied to the function on the graph, u, is given by Lu where

$$L_{ij} = \begin{cases} d_i & \text{if } i = j \\ -w_{ij} & \text{otherwise} \end{cases}, \quad \text{and so } (Lu)_i = \sum_{j=1}^{n} (u_i - u_j)\, w_{ij}.$$

To keep the spectrum of the graph Laplacian in a fixed range as the the number of samples in increased and thus to guarantee consistency, we must normalize the graph Laplacian. See Bertozzi and Flenner 2012 [10] for a more in depth discussion of this. We use the symmetric normalization.

$$L_{sym} := D^{-1/2} L D^{-1/2}, \quad D_{ij} = \begin{cases} d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Because we express our energy as applied to functions over continuous domains, we also introduce the following notation for the symmetric-normalized non-local gradient.

$$\nabla_{w,s} u(x,y) := \frac{\nabla_w u(x,y)}{\left(\int_\Omega w(x,z)dz \int_\Omega w(y,z)dz\right)^{1/4}}.$$

2.1.3 Numerical optimization

We must numerically find an approximate solution. The unconstrained energy has gradient flow

$$u_t = \alpha \Delta_{w,s} u + \beta \Delta u + \frac{1}{u} \sum_{i=1}^{n} \delta(x - x_i).$$

We evolve this equation, projecting onto the space of probability densities after each step. We discretize the equation as

$$\frac{u^{k+1} - u^k}{\delta t} = -\alpha L_{sym} u^{k+1} + \beta \Delta_h u^{k+1} + \frac{1}{u^k} \sum_{i=1}^{n} \delta(x - x_i).$$

Here Δ_h denotes the discrete Laplacian from the 5-point finite difference stencil with mesh size $h = 1$. Solving for u^{k+1} yields

$$u^{k+1} = (I + \alpha \delta t L_{sym} - \beta \delta t \Delta_h)^{-1} \left(\frac{\delta t}{u^k} \sum_{i=1}^{n} \delta(x - x_i) + u^k \right).$$

To approximate this, we use a split-time method

$$\begin{aligned} u^{k+1/2} &= \left(I + \alpha \frac{\delta t}{2} L_{sym} \right)^{-1} \left(\frac{\delta t}{u^k} \sum_{i=1}^{n} \delta(x - x_i) + u^k \right), \\ u^{k+1} &= \left(I - \beta \frac{\delta t}{2} \Delta_h \right)^{-1} \left(\frac{\delta t}{u^{k+1/2}} \sum_{i=1}^{n} \delta(x - x_i) + u^{k+1/2} \right). \end{aligned}$$

To apply these operators, we use a spectral method. This has two advantages over forming and multiplying the matrices. First, we can approximate the projection onto the constraint by using the spectral decomposition of the discrete Laplacian (shown in Table 2.1). Second, the computation required to form and apply the entire symmetric graph Laplacian is too intensive. Fortunately, we can apply Nyström's extension (discussed in Sec. 2.1.4), which is a popular method for approximating a portion of the eigenvectors and eigenvalues which approximate the operator well. To project onto the eigenvectors of Δ_h we apply the 2D Fast Fourier Transform.

In both the case of applying $(I + \alpha \frac{\delta t}{2} L_{sym})^{-1}$ and $(I - \beta \frac{\delta t}{2} \Delta_h)^{-1}$ we are applying operators of the form $(I + \delta t P)^{-1}$ where P is symmetric and positive semidefinite. In general, if P has spectral decomposition $P = \Phi \Lambda \Phi^T$ then we apply $(I + \delta t P)^{-1}$ to $\vec{w}$ by first projecting onto the eigenvectors : $\vec{a} = \Phi^T \vec{w}$, updating the coefficients $\tilde{a}_m = a_m / (1 + \delta t \lambda_m)$, and finally transforming back to the standard basis : $(I + \delta t P)^{-1} \vec{w} = \Phi \vec{\tilde{a}}$. We summarize the steps of our algorithm in Table 2.1.

2.1.4 Nyström's extension

To apply the spectral method described in the previous section we need to approximate the eigenvectors and eigenvalues of the symmetric graph Laplacian. Here we present the Nyström's exten-

```
Nyström $(\mathbf{Im}_g) \rightarrow \Phi, \Lambda : L_{sym} \approx \Phi\Lambda\Phi^T$.
Initialize $u^0 \equiv 1/|\Omega|, succDiff = \infty, k = 0$.
while $succDiff > 10^{-7}$ and $k < maxSteps = 800$
    • $k = k + 1$
    • $\vec{b} = \Phi^T \left[u^{k-1} + \frac{\delta t}{u^{k-1}} \sum_{i=1}^{n} \delta(x - x_i)\right]$
    • $a_i = \frac{b_i}{1+\alpha \frac{\delta t}{2} \lambda_i}$
    • $\vec{u}^{k-1/2} = \Phi \vec{a}$
    • $\vec{b} = \text{fft2} \left[u^{k-1/2} + \frac{\delta t}{u^{k-1/2}} \sum_{i=1}^{n} \delta(x - x_i)\right]$
    • $a_i = \frac{b_i}{1+2\beta\delta t\pi^2(m^2+n^2)}, i \sim (m,n)^{th}$ Fourier mode,
      $a_1 = 1$ (guarantees integral 1 constraint)
    • $\vec{u}^k = \text{ifft2}\left(\vec{a}\right)$
    • $\vec{u}^k = \max\left(\vec{u}^k, 0\right)$
    • $succDiff = \|u^k - u^{k-1}\|_2^2 / \|u^k\|_2^2$
```

Table 2.1: Non-Local H^1 MPLE Algorithm

sion method and refer the reader to [56, 10, 94] for further discussion and analysis. Nyström's extension is a technique for performing matrix completion, well-known within the spectral graph theory community. In this setting, Nyström's extension is applied to the normalized affinity matrix $W_{sym} = D^{-1/2}WD^{-1/2}$ where the (i, j)th entry of W is the affinity between node i and j. Note that the matrices W_{sym} and L_{sym} have the same eigenvectors, and λ is an eigenvalue of W_{sym} if and only if $1 - \lambda$ is an eigenvalue of L_{sym}.

We let N denote the set of nodes in our complete weighted graph, then take X to be a small random sample from N, and Y its complement. Up to a permutation of the nodes we can write the affinity matrix as

$$W = \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{pmatrix},$$

where the matrix $W_{XY} = W_{YX}^T$ consists of weights between nodes in X and nodes in Y, W_{XX} consists of weights between pairs of nodes in X, and W_{YY} consists of weights between pairs of nodes in Y. Nyström's extension approximates the eigenvalues and eigenvectors of the affinity matrix by manipulating the approximation:

$$W \approx \hat{W} = \begin{pmatrix} W_{XX} \\ W_{YX} \end{pmatrix} W_{XX}^{-1} \begin{pmatrix} W_{XX} & W_{XY} \end{pmatrix}.$$

This approximates $W_{YY} \approx W_{YX}W_{XX}^{-1}W_{XY}$. The error due to this approximation is determined by how well the rows of W_{XY} span the rows of W_{YY}. If the affinity matrix W is positive semidefinite then we can write it as a matrix transpose times itself : $W = V^TV$. In [8] the authors show that the Nyström extension thus approximates the unknown part of V (corresponding to W_{YY}) by orthogonally projecting it onto the range of the known part (corresponding to W_{XY}). In this setting it is clear that as the size of X grows, the approximation improves. Further, a random choice of X is likely to yield W_{XY} full-rank if the rank of the rank of W is sufficiently large.

Next we must incorporate the normalization factors into the above approximation. The degrees are approximated by applying their definition to the approximation. Note that $d_i = \sum_{j=1}^n w_{ij}$ can also be written as $d = W\mathbf{1}_n$ where $\mathbf{1}_n$ is the n length vector of ones. This yields

$$\begin{aligned}\hat{d}_X =& W_{XX}\mathbf{1}_{|X|} + W_{XY}\mathbf{1}_{|Y|},\\ \hat{d}_Y =& W_{YX}\mathbf{1}_{|X|} + W_{YX}W_{XX}^{-1}W_{XY}\mathbf{1}_{|Y|}.\end{aligned}$$

In this way we approximate the degrees without forming any matrices of size larger than $|X| \times |Y|$. Define also the vectors $s_X = d_X^{-1/2}$, $s_Y = d_Y^{-1/2}$. Normalizing our approximation of W gives

$$W_{sym} \approx \hat{W}_{sym} = \begin{pmatrix} W_{XX} \odot \left(s_X s_X^T\right) & W_{XY} \odot \left(s_X s_Y^T\right) \\ W_{YX} \odot \left(s_Y s_X^T\right) & \left(W_{YX}W_{XX}^{-1}W_{XY}\right) \odot \left(s_Y s_Y^T\right) \end{pmatrix},$$

where $\odot$ denotes component-wise product. For notational convenience going forward, let us define $W_{XX}^{sym} = W_{XX} \odot \left(s_X s_X^T\right)$ and $W_{XY}^{sym} = W_{XY} \odot \left(s_X s_Y^T\right)$.

In practice, one uses a diagonal decomposition of such a formula to avoid forming and applying the full matrix. It follows from analysis discussed in [56] that if W_{XX}^{sym} is positive definite, the diagonal decomposition of the approximation is given by $\hat{W}_{sym} = V\Lambda_S V^T$, where

$$S = W_{XX}^{sym} + \left(W_{XX}^{sym}\right)^{-1/2} W_{XY}^{sym} W_{YX}^{sym} \left(W_{XX}^{sym}\right)^{-1/2},$$

S has diagonal decomposition $S = U_S \Lambda_S U_S^T$, and

$$V = \begin{bmatrix} W_{XX}^{sym} \\ W_{YX}^{sym} \end{bmatrix} \left(W_{XX}^{sym}\right)^{-1/2} U_S \Lambda_S^{-1/2}.$$

Note that S is size $|X| \times |X|$ and V is size $|N| \times |X|$. Their computation never requires computing or storing matrices larger than size $|N| \times |X|$. Thus V is a matrix of $|X|$ approximate eigenvectors of W_{sym} with corresponding eigenvalues Λ_S. For more detailed discussion on Nyström's extension, see [56, 10, 94].

2.1.5 Cross-validation

Cross-validation is a methodology for choosing the smoothing parameter λ which yields probability densities that are predictive of the missing data [108]. Because our method consists primarily of simple coefficient updates after mapping to different eigenspaces, it is fast relative to methods with similar goals ([120] for instance). This speed increase allows us to perform 10-fold cross-validation, which requires many evaluations of the density estimation method. In V-fold cross validation we randomly partition the data points into V disjoint subsets $X = \sqcup_{v=1}^{V} X_v$ with complements $X_{-v} = X \backslash X_v$. We let $u_{\lambda,-v}$ denote the density estimate using parameter λ trained on the data X_{-v}. The objective we minimize is an application of the Kullback-Leibler divergence, an asymmetric distance measure for probabilities given by

$$D_{KL}\left(p, q\right) = \int_{\Omega} \log\left(\frac{p(x)}{q(x)}\right) p(x) dx.$$

We select the parameter λ that minimizes the average KL divergence between the density estimates, $u_{\lambda,-v}$, and the discrete distributions on the withheld data points :

$$p_v(x) = \frac{1}{|X_v|} \sum_{x_i \in X_v} \delta(x - x_i).$$

This yields the following optimization:

$$\begin{aligned}\hat{\lambda} &= \arg\min_{\lambda} \frac{1}{V} \sum_{v=1}^{V} D_{KL}(p_v, u_{\lambda,-v}) \\ &= \arg\max_{\lambda} \frac{1}{V} \sum_{v=1}^{V} \sum_{x_i \in X_v} \log(u_{\lambda,-v}(x_i)).\end{aligned}$$

The result can also be interpreted as maximizing the average log-likelihood that the missing events are drawn from the corresponding estimated densities. We approximate this optimization via a grid search (note that $\lambda = (\alpha, \beta)$ is 2 dimensional). The search requires the computation of all the density estimates $u_{\lambda,-v}$. In particular, for 10-fold cross-validation, we must compute $10 \times |\alpha \text{ values}| \times |\beta \text{ values}|$ densities.

When evaluating the energy, it is important to ensure that nonnegativity and sum-to-one constraints hold strictly for the input densities. If a density is slightly negative somewhere, it could add complex terms to the objective, and if a density has sum slightly larger than 1, it could unfairly achieve a slightly higher objective. Further, in the strictest interpretation, if a density has a value 0 at the location of a missing event, the objective will take value $-\infty$. We relax this penalty by replacing $u_{\lambda,-v}(x_i)$ with $\max\{u_{\lambda,-v}(x_i), 10^{-16}\}$.

2.2 Numerical experiments

In this section, we demonstrate the advantage NL H^1 MPLE method over standard H^1 MPLE by evaluating its performance on residential burglary data from San Fernando Valley in Los Angeles, California, using of corresponding housing data and a satellite image to inform the non-local weights.

2.2.1 Residential burglary

We perform experiments on residential burglary data from San Fernando Valley in 2005-2013, getting substantially different results than those shown in [97, 120, 81]. In Fig. 2.1 we show the data used (locations of residential burglaries in Fig. 2.1(a), housing in Fig. 2.1(b), satellite image in Fig. 2.1(c)), H^1 MPLE (Fig. 2.1(d)), housing-based NL H^1 MPLE (Fig. 2.1(c)), and satellite-based NL H^1 MPLE (Fig. 2.1(d)) density estimates on increasing subsets of data from 2005-2008. To evaluate performance, we compute the log-likelihood of each density on the residential burglaries from 2009-2013 (shown in Table 2.2).

As one would predict, the locations of residential burglaries in Fig. 2.1(a) are primarily restricted to the support of the housing density image Fig. 2.1(b). There are some locations in the burglary data set that correspond to locations with no residences (4,173 events out of 23,725 total), which we attribute to imprecision in the burglary data. Most such misplaced events occur on streets, suggesting that the actual event took place at a residence facing that street. Because of this inconsistency between the data sets, for the experiments which use the housing data, we adjust the residential burglary data for training and testing (for both H^1 and NL H^1), moving each event to the nearest house if it is within 2 pixels, and dropping the event otherwise. This results in 603 dropped events. For the experiments which do not use housing data, we work with the raw burglary data for training and testing.

We implement H^1 MPLE by applying our algorithm, described in Table 2.1 with $\alpha = 0$ and $\Phi = Id$. We choose the value of the regularization parameter β for each training data set by performing 10-fold log-likelihood cross-validation, searching over $\beta =$`[0,10.^(-2:8)]`. We apply H^1 MPLE to both the raw and corrected burglary data.

For housing-based NL H^1 MPLE , we perform Nyström's extension with non-local means applied to g, the housing density image shown in Fig. 2.1(c). We use 400 random samples for Nyström's extension. We use the first 300 eigenvectors and eigenvalues in our computations. The non-local means weights are based on differences between patches of size 11×11 and $\sigma = 1 \cdot std(g)$, the standard deviation of the housing image. The weight kernel K_r, $r = 5$, is given as follows.

$$K_r(1+r+i, 1+r+j) = \frac{1}{r} \sum_{d=\max(|i|,|j|,1)}^{r} \frac{1}{(2d+1)^2}, \quad i, j = -r, \dots, r.$$

To choose the regularization parameters α, β, we perform 10-fold log-likelihood cross-validation, searching over $\alpha =$`[0,10.^(-2:12)]`, $\beta =$`[0,10.^(-2:8)]`. We apply housing NL H^1 MPLE to the corrected burglary data.

For satellite-based NL H^1 MPLE, we perform Nyström's extension with non-local means applied to g, the Google Maps image shown in Fig. 2.1(c). In applying non-local means to a color image, we interpret the image as a vector valued function with 3 components (one for each color channel) and so in equation (2.2) the expression $|\mathbf{Im}(x+\cdot)-\mathbf{Im}(y+\cdot)|^2$ is size $(2r+1)\times(2r+1)\times 3$. We use 800 random samples for Nyström's extension. We use the first 600 eigenvectors and eigenvalues in our computations. The non-local means weights are based on differences between patches of size 11×11 and $\sigma = 1 \cdot std(g)$, the standard deviation of the Google Maps image. The weight kernel is as in the previous case, but repeated on each color channel. To choose the regular-

Training Data Set (corrected)	scaled Histogram	H^1	Housing NL H^1
50 random from 2008	-3.6039×10^5	**-1.3386** $\times 10^5$	-1.3396×10^5
100 random from 2008	-3.5991×10^5	**-1.3369** $\times 10^5$	-1.3369×10^5
500 random from 2008	-3.5197×10^5	-1.3282×10^5	**-1.3004** $\times 10^5$
1000 random from 2008	-3.4350×10^5	-1.3246×10^5	**-1.2953** $\times 10^5$
2008	-3.1905×10^5	-1.3189×10^5	**-1.2888** $\times 10^5$
2007-2008	-2.9846×10^5	-1.3174×10^5	**-1.2850** $\times 10^5$
2006-2008	-2.8152×10^5	-1.3136×10^5	**-1.2815** $\times 10^5$
2005-2008	-2.6847×10^5	-1.3121×10^5	**-1.2774** $\times 10^5$

Traing Data Set (raw)	scaled Histogram	H^1	Satellite NL H^1
50 random from 2008	-3.6959×10^5	-1.3733×10^5	**-1.3553** $\times 10^5$
100 random from 2008	-3.6822×10^5	-1.3732×10^5	**-1.3553** $\times 10^5$
500 random from 2008	-3.6342×10^5	-1.3583×10^5	**-1.3524** $\times 10^5$
1000 random from 2008	-3.5733×10^5	-1.3598×10^5	**-1.3525** $\times 10^5$
2008	-3.3313×10^5	-1.3535×10^5	**-1.3494** $\times 10^5$
2007-2008	-3.1326×10^5	-1.3525×10^5	**-1.3482** $\times 10^5$
2006-2008	-2.9630×10^5	-1.3496×10^5	**-1.3449** $\times 10^5$
2005-2008	-2.8334×10^5	-1.3488×10^5	**-1.3431** $\times 10^5$

Table 2.2: Log-likelihood of densities on residential burglaries from 2009-2013 (corrected & raw)

ization parameters α, β for each training set, we perform 10-fold log-likelihood cross-validation, searching over $\alpha =$`[0,10.^(-2:12)]`, $\beta =$`[0,10.^(-2:8)]`. We apply satellite NL H^1 MPLE to the raw burglary data.

The H^1 MPLE results transition from a completely smooth uniform density to a probability density with more apparent structure as the amount of training data increases. The NL H^1 MPLE housing and satellite results exhibit a similar trend, but are able to better approximate the correct support of the density with many fewer data points. The measurable benefit of non-local smoothing is shown by the log-likelihood values in Table 2.2. NL H^1 generally gets higher log-likelihood than H^1. This means the densities estimated by housing NL H^1 on corrected 2005-2008 data are more congruous with the corrected 2009-2013 data than the H^1 densities, and the densities estimated by satellite NL H^1 on raw 2005-2008 data are more congruous with the raw 2009-2013 data than the H^1 densities.

The added complexity of our algorithm results in an increase in run time from the standard H^1 MPLE, but the difference is not too substantial. We compare run times on a laptop with one Intel

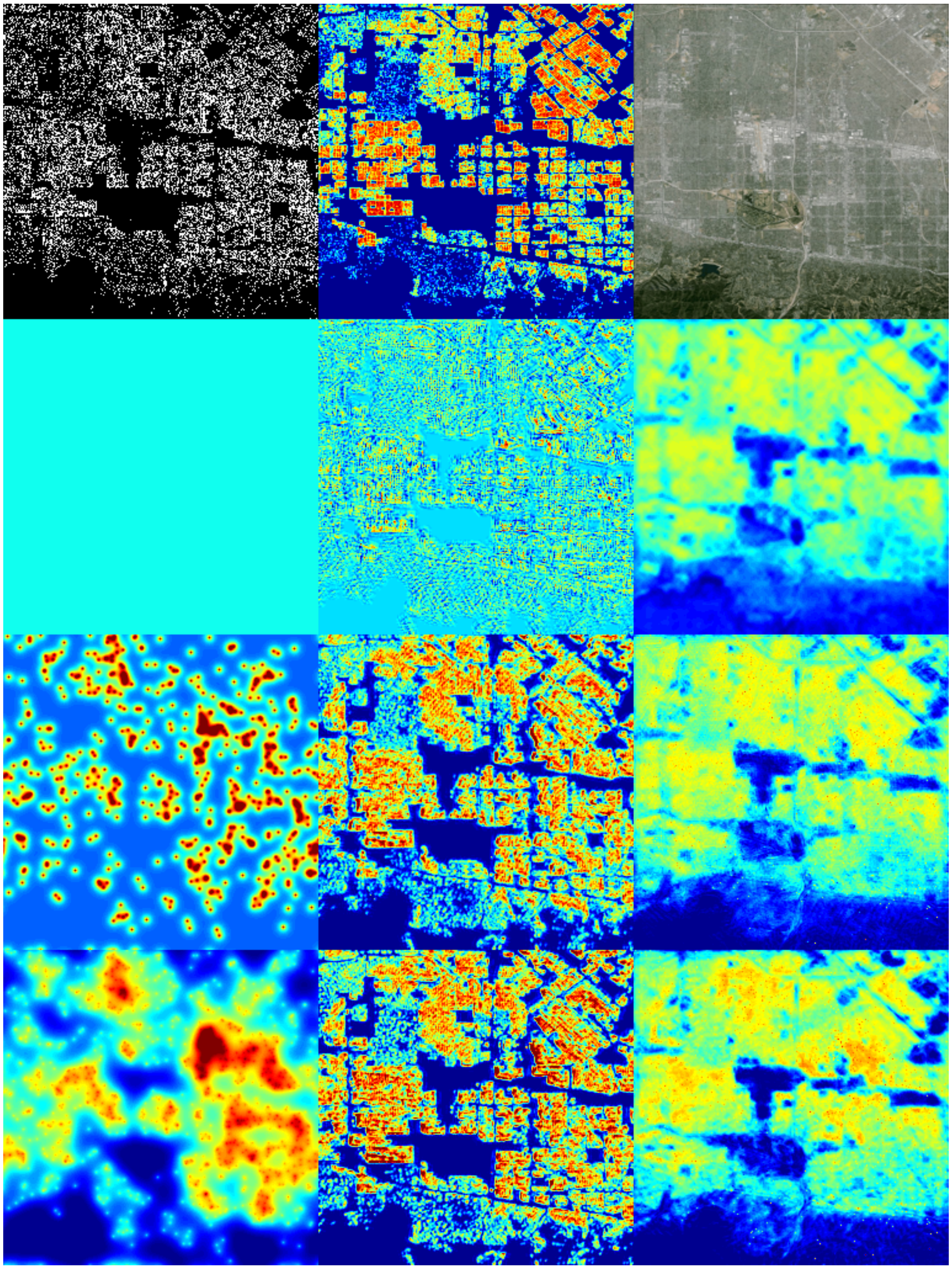


Figure 2.1:
Top row: data
(a) 2005-2013 Residential burglaries in San Fernando Valley (from LAPD)
(b) San Fernando Valley $\log(\min(\text{\# housing units}, 7) + 1)$ (from LA County Tax Assessor)
(c) Satellite image of San Fernando Valley (from Google Maps)
Bottom three rows : MPLE of 50, 500, and 1000 random samples from '08 residential burglaries
(d) Column 1 : H^1 MPLE
(e) Column 2 : Housing NL H^1 MPLE
(f) Column 3 : Satellite NL H^1 MPLE

Core i7 processor that has two cores with processor speed 2.67GHz and 4GB of memory. The run time for Nyström applied to the housing image is typically about 17 seconds. The run time for Nyström applied to the satellite image is typically about 36 seconds. For cross-validation purposes, Nyström can be run once outside of the loop and the results used for all combinations of data sets and parameters. The run time for H^1 MPLE with parameters as chosen by cross-validation on the residential burglaries from 2005-2008 is typically about half a second. The run time for housing NL H^1 MPLE with parameters as chosen by cross-validation on the the residential burglaries from 2005-2008 is typically about 2.3 seconds. The run time for satellite NL H^1 MPLE with parameters as chosen by cross-validation on the the residential burglaries from 2005-2008 is typically about 1.5 seconds. The cross-validation run times depend on what range of parameters are being tested, but can easily be run in parallel across several computing nodes.

2.2.2 Synthetic Density

To further verify that NL H^1 MPLE is correctly performing density estimation, we test the method's ability to recover a given density. We start with a known density, draw events from it, and attempt to recover it. Because the method assumes a relationship between the spatial data g and the density u, we generate a synthetic density which is closely related to the housing data, shown in the bottom left of Fig. 2.2. This density is given by taking a random linear combination of the first 5 approximated eigenvectors of the graph Laplacian (with weights based on the housing image) and then shifting and normalizing the result to yield a probability density. The coefficients are chosen uniformly at random in $[0, 1]$ and the non-local weights are based on the housing data as they were in the previous section. This randomly generated density was chosen over others because it looks like a potential probability density for residential burglary. It should be noted that this choice of synthetic density is quite ideal for the proposed method. The hope is that very good density recovery of ideal probability densities extends to good density recovery of less ideal probability densities.

We sample events according to this density by generating numbers uniformly at random in $[0, 1]$ and inverting the cumulative distribution function associated with the density. In the top row of Fig. 2.2 we show the H^1 MPLE result on the 400 events ($\beta = 5 \times 10^4$), the housing NL H^1 MPLE result on the 400 events ($\alpha = 100, \beta = 0$), and the NL H^1 MPLE result on 400 events restricted to the first 5 eigenvectors. In the bottom row of Fig. 2.2 we show the synthetic density, the H^1 MPLE result on the 4,000 events ($\beta = 10^5$), the housing NL H^1 MPLE result on the 4,000 events ($\alpha = 10^8, \beta = 0$), and the NL H^1 MPLE result on 4,000 events restricted to the first 5 eigenvectors. In all cases, smoothing parameters were chosen to minimize mean absolute error of the probability density. The NL H^1 results and the restricted NL H^1 results do a substantially better job at recovering the probability density than H^1 MPLE. This is expected of course, from the construction of the probability. The comparison merely suggests that if the correct density is well-approximated by a combination of eigenvectors of the graph Laplacian, enforcing non-local smoothness can substantially improve recovery of the density. It is, in general, difficult to determine when a density is well-approximated by a graph Laplacian's eigenbasis. The assumption is that the primary and non-local data have some meaningful, consistent connection. We refer the reader to Sec. 2.1 for heuristics on this connection and Appendix A for some more precise formulations. It is also worth noting that if unrelated non-local data is used, cross-validation will likely yield $\alpha = 0$, reverting the model back to standard H^1 MPLE.

2.3 Conclusions and Future work

In this chapter we have looked at the problem of obtaining spatially accurate probability density estimates. The need for new approaches is demonstrated by the inadequate performance of standard techniques such as H^1 MPLE.

Our proposed solution accomplishes this by incorporating a non-local regularity term based on the H^1 regularizer and non-local means which fuses geographical information into the density estimate. Our experiments with the San Fernando Valley residential burglary data set demonstrate that this method does yield a probability density estimate with the correct support which also gives favorable log-likelihood results. Further, our results based on the Google Maps image suggest we can apply NL H^1 MPLE to a wide variety of geographic regions without obtaining specialized

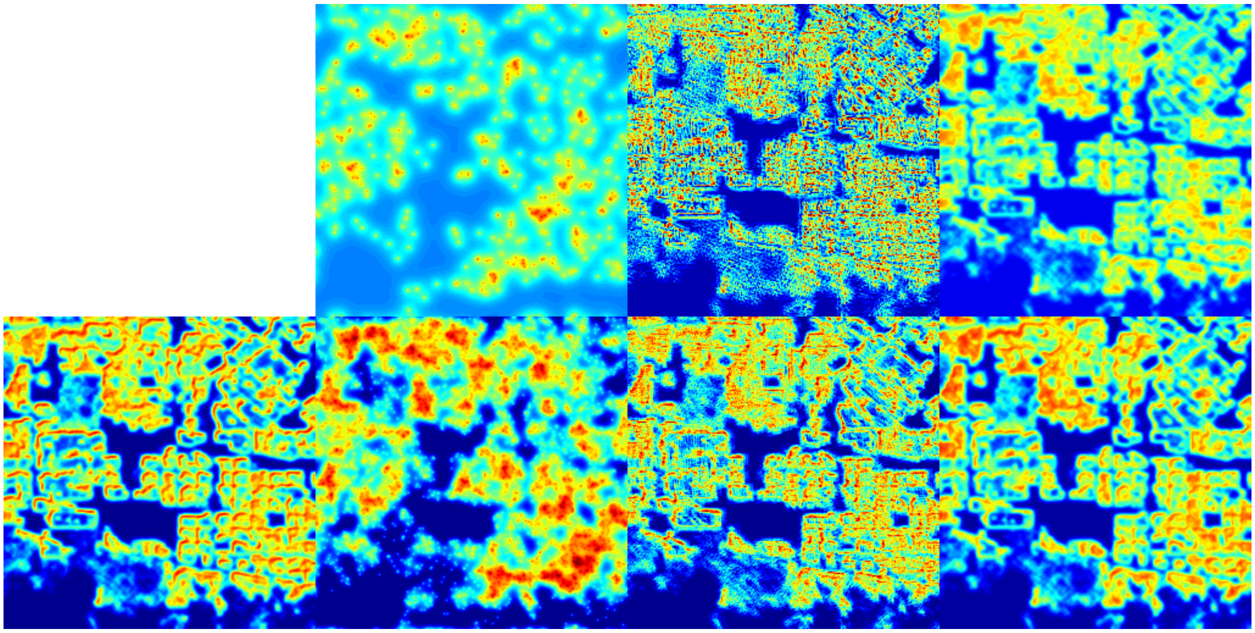


Figure 2.2: Synthetic density recovery (see Sec. 2.2 2.2.2)
Top row : density estimates based on 400 samples from synthetic density
$\overline{|\text{error}|}$: H^1 7.12473×10^{-6}, NL H^1 5.26617×10^{-6} , NL H^1 restricted 2.55042×10^{-6}
Bottom row : synthetic density and density estimates on 4,000 samples
$\overline{|\text{error}|}$: H^1 5.05662×10^{-6}, NL H^1 2.52831×10^{-6} , NL H^1 restricted 1.36416×10^{-6}

geographic data.

There are several others aspects of this and related problems to explore. In general, testing the method on other datasets would be interesting. This may present the added challenge of dealing with other types of geographical information since high-resolution housing density data may not be readily available. In modeling the density of other types of events, the geographical data may not be related to housing at all. As the problem dictates, the non-local weights can be replaced with whatever weights seem appropriate for the data at hand. We have yet to incorporate time, leading indicators of crime, or census data into model. Any of these could further improve results and allow one to use density estimation in place of risk terrain modeling.

Finally, our method need not stand alone. Several sophisticated spatio-temporal models for probabilistic events make use of density estimation, typically using the standard methods [98, 87, 133]. By replacing the standard density estimation techniques with a non-locally regularized MPLE such as ours, the density estimates in these models could improve, thus improving the overall result of the resulting simulation.

CHAPTER 3

Compressed Sensing Recovery via Nonconvex Shrinkage Penalties

3.1 Generalized shrinkage penalties

As described above, nonconvex penalty functions have been shown both theoretically and empirically to give better results for compressed sensing than the ℓ^1 norm. In order to make use of any of several efficient algorithms, we wish to consider penalty functions with explicit proximal mappings. In this section, we consider two such families of functions.

3.1.1 p-shrinkage and firm thresholding

First we consider a shrinkage mapping, a version of which first appeared in [27], that has some qualitative resemblance to the ℓ^p proximal mapping, while being continuous and explicit:

Definition 3.1.1. For $\lambda > 0$, the p-shrinkage mapping $S_p = S_{\lambda,p}$ for $p \in \mathbb{R}$ is defined by $S_p(x)_i = s_p(|x_i|)\operatorname{sign}(x_i)$, where the shrinkage function $s_p = s_{\lambda,p}$ is defined by

$$s_p(t) = \max\{t - \lambda^{2-p} t^{p-1}, 0\}. \tag{3.1}$$

See Figure 3.1 for example plots. When $p = 1$, p-shrinkage and soft thresholding coincide. The smaller the value of p, the less p-shrinkage shrinks large inputs. In the limit as $p \to -\infty$, p-shrinkage tends pointwise to *hard thresholding*:

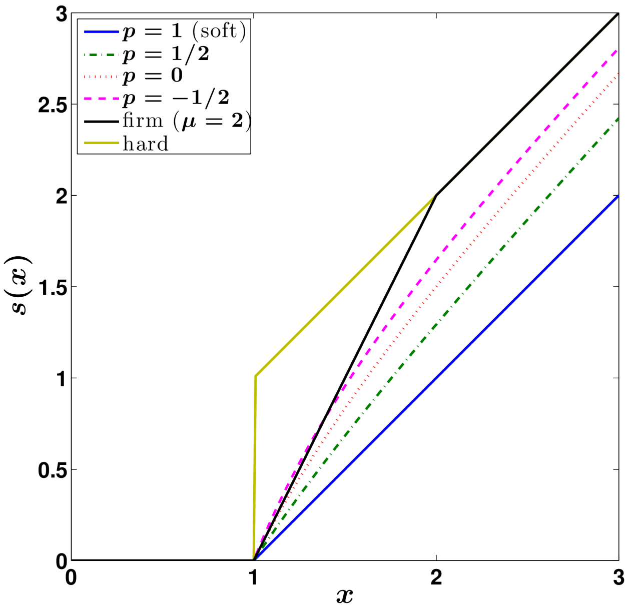


Figure 3.1: Plot of several shrinkage functions, all with $\lambda = 1$. The smaller the value of p, the smaller the bias applied to large inputs. Firm thresholding removes the bias completely for large enough inputs, without the discontinuity of hard thresholding.

Definition 3.1.2. For $\lambda > 0$, the hard thresholding mapping H_λ is defined by

$$H_\lambda(x)_i = \begin{cases} 0 & \text{if } |x_i| \le \lambda, \\ x_i & \text{if } |x_i| > \lambda. \end{cases} \tag{3.2}$$

Hard thresholding is related to the proximal mapping of the ℓ^0 penalty function:

$$H_{\sqrt{2\lambda}} \in \text{prox}_\lambda \| \cdot \|_0, \tag{3.3}$$

the right side of (3.3) being two-valued in components satisfying $x_i^2 = 2\lambda$. Hard thresholding imposes no bias on large inputs, but its discontinuity makes it very unstable when used with ADMM [46].

Another shrinkage mapping we consider is *firm thresholding*, a continuous, piecewise-linear approximation of hard thresholding. Firm thresholding was first introduced in [59] in connection with the WaveShrink procedure for denoising and non-parametric regression. It was not known at the time to be the proximal operator of a given penalty function.

Definition 3.1.3. For $\lambda > 0$ and $\mu > \lambda$, the firm thresholding mapping $S_{\text{firm}} = S_{\lambda,\mu,\text{firm}}$ is defined by $S_{\text{firm}}(x)_i = s_{\text{firm}}(|x_i|)$, where $s_{\text{firm}} = s_{\lambda,\mu,\text{firm}}$ is defined by

$$s_{\text{firm}}(t) = \begin{cases} 0 & \text{if } t \leq \lambda, \\ \frac{\mu}{\mu-\lambda}(t-\lambda) & \text{if } \lambda \leq t \leq \mu, \\ t & \text{if } t \geq \mu. \end{cases} \tag{3.4}$$

Note that $S_{\lambda,\lambda,\text{firm}} = H_\lambda$, and $\lim_{\mu\to\infty} S_{\lambda,\mu,\text{firm}}(x) = S_{\lambda,1}(x)$ pointwise. Thus both p-shrinkage and firm thresholding can be seen as generalizing both soft and hard thresholding.

3.1.2 Shrinkage-induced penalty functions

Our motivation for considering alternative shrinkage mappings is to have them as closed-form proximal mappings. This requires that the shrinkages actually be the proximal mappings of penalty functions. The following theorem guarantees this. It is proved in [31, Thm. 1], and strengthens the earlier result of Antoniadis [4, Prop. 3.2].

Theorem 3.1.4. *Suppose $s : [0,\infty) \to \mathbb{R}$ is continuous, satisfies $x \leq \lambda \Rightarrow s(x) = 0$ for some $\lambda \geq 0$, is strictly increasing on $[\lambda,\infty)$, and $s(x) \leq x$. Define $S(x)_i = s(|x_i|)\,\operatorname{sign}(x_i)$, for each i. Then S is the proximal mapping of a penalty function $G(w) = \sum_i g(w_i)$ where g is even, strictly increasing and continuous on $[0,\infty)$, differentiable on $(0,\infty)$, and nondifferentiable at 0 iff $\lambda > 0$ (in which case $\partial g(0) = [-1,1]$). If also $x - s(x)$ is nonincreasing on $[\lambda,\infty)$, then g is concave on $[0,\infty)$ and G satisfies the triangle inequality.*

Both p-shrinkage and firm thresholding satisfy all hypotheses of the theorem for all parameter values. The proof of the theorem constructs g using the Legendre-Fenchel transform [103] of an antiderivative of s. Because of the nature of the Legendre-Fenchel transform, this often does not produce a closed-form expression for g. We consider this as an acceptable price to pay for having an explicit proximal mapping, which is much more useful for most of today's state-of-the-

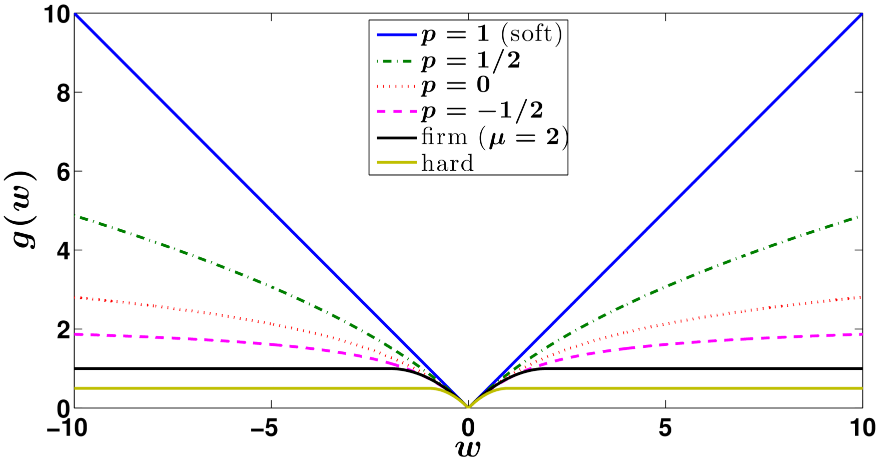


Figure 3.2: Plot of penalty component function g induced by several shrinkage mappings, all with $\lambda = 1$. The smaller the value of p, the slower the growth of the p-shrinkage penalty function, being bounded above when $p < 0$. Both firm and hard thresholding have penalty functions that are quadratic near the origin, then constant.

art algorithms for compressed sensing than having an explicit penalty function. In the case of the penalty function G_p induced by p-shrinkage, we can compute $g_p(w)$ numerically, and example plots are in Figure 3.2. In addition to the properties guaranteed by Thm. 3.1.4, it can be shown that $\lim_{w\to\infty} g_p(w) - w^p/p - C_p = 0$ for $p \neq 0$ and constant C_p depending only on p. This includes $p < 0$, in which case it follows that $g_p(w)$ is bounded above. For $p = 0$, we have $\lim_{w\to\infty} g_0(w) - \log w - C = 0$ instead.

In the case of the penalty function G_{firm} induced by firm thresholding, g_{firm} does have a closed form:

$$g_{\text{firm}}(w) = \begin{cases} |w| - w^2/(2\mu) & \text{if } |w| \leq \mu, \\ \mu/2 & \text{if } |w| \geq \mu. \end{cases} \tag{3.5}$$

Note that $g_{\text{firm}}(w)$ is independent of λ, except that $\mu \geq \lambda$ is required by the definition of g_{firm}.

Although the statement of Thm. 3.1.4 excludes hard thresholding (being discontinuous), the construction in the proof does produce a penalty function G_{hard}. It coincides with G_{firm} for $\mu = \lambda$. The part of the conclusion of the theorem that doesn't hold is that $\operatorname{prox}_\lambda G_{\text{hard}}(\lambda)$ is the entire interval $[0, \lambda]$, while $H_\lambda(\lambda)$ is generally defined to take on a single value from this interval (namely 0 in our definition (3.2)).

3.1.3 Example

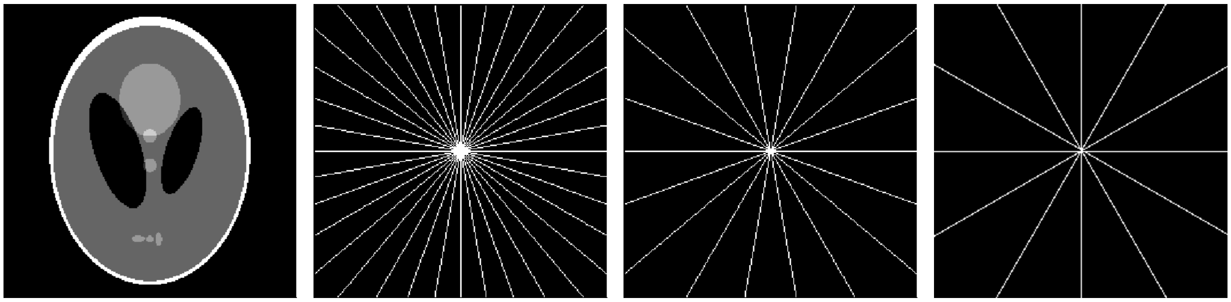
To motivate the consideration of p-shrinkage and firm thresholding, we consider a generalization of an example appearing in the first compressed sensing paper [22]. We seek to reconstruct the 256×256 Shepp-Logan phantom image from samples of its 2-D discrete Fourier transform (DFT), taken along radial lines, thereby simulating both MRI and X-ray CT data (the latter by way of the Fourier slice theorem). See Figure 3.3. Since the phantom has a sparse gradient, we seek to solve the following optimization problem:

$$\min_x G(\nabla x) \text{ subject to } \mathcal{F}x = b, \tag{3.6}$$

where G is one of the penalty functions being compared, ∇ is a discrete gradient using forward differences and periodic boundary conditions, $\mathcal{F}$ is the 2-D DFT, and b contains the sample data. We solve (3.6) with ADMM, where the shrinkage mapping is p-shrinkage with $p \leq 1$ or firm thresholding. See [30] for details, being also a straightforward generalization of the algorithm of [65].

With $G = G_1 = \|\cdot\|_1$, 18 lines are required for exact reconstruction, while using $G = G_{-1/2}$, 9 lines suffice, as shown in [27], the latter being the fewest that had been demonstrated at that time. In [29] (see also [31]), 6 lines were shown to suffice using the G induced by a shrinkage mapping that is a C^∞ approximation of hard thresholding. This is the fewest possible, since with 5 lines, there are fewer measurements than nonzero gradient pixels, so that the phantom will not even be a local minimizer of the problem with $G = \|\cdot\|_0$. However, here we report that using $G = G_{\text{firm}}$ (with $\lambda = 0.1$ and $\mu = 2.5$), 6 lines also suffice, and many fewer ADMM iterations are needed (337 versus 2213).

While this example is an ideal case, using a very sparse image and noise-free measurements, this does demonstrate that p-shrinkage and firm thresholding induce penalty functions that can be useful for recovering sparse signals. Now we turn to a theoretical analysis of the sparse recovery performance of minimizing these penalty functions.



(a) Shepp-Logan phantom (b) $p = 1$, 18 lines (c) $p = -1/2$, 9 lines (d) firm, 6 lines

Figure 3.3: The Shepp-Logan phantom, and the number of radial lines of Fourier samples needed to reconstruct the phantom perfectly using different penalty functions.

3.2 Exact recovery

In this section, we establish sufficient conditions for exact recovery of sparse signals from noise-free measurements by solving a minimization problem with penalty function G:

$$\min_{w} G(w) \text{ subject to } Aw = b. \tag{3.7}$$

Our objective is to determine sufficient conditions in the case where G is a penalty function induced by a shrinkage mapping; however, we will establish conditions for a somewhat more general class of penalty functions G. We shall assume that the measurement matrix $A \in \mathbb{R}^{m \times n}$ has the Unique Representation Property (URP) [67], i.e., any m columns of A are linearly independent. This implies that any vector in $\ker(A)$ has at least $m + 1$ nonzero entries. The URP can be regarded as a *generic* property of matrices; for example, a matrix whose entries are independently and identically distributed samples drawn from any absolutely continuous probability distribution will have URP with probability 1.

Remark 3.2.1. The URP implies that the m rows of A are linearly independent. Thus an orthonormal basis for the span of the rows can be formulated as linear combinations of the rows of A. So if we multiply A by a product of elementary matrices, E, corresponding to the necessary elementary row operations, the resulting product will have orthonormal rows. Since elementary matrices are invertible, $Aw = b$ is equivalent to $EAw = Eb$. Also, since each elementary matrix is invertible,

A_T being full rank for $|T| = m$ implies EA_T is full rank as well, and so A satisfying the URP implies EA satisfies the URP. Thus we can always transform the problem so that the rows of A are orthonormal, i.e., $AA^T = I$, and so without loss of generality, we assume that the A given satisfies $AA^T = I$.

We shall also assume that $G(w) = \sum_i g(w_i)$ with

I) $g(0) = 0$, and g even on $\mathbb{R}$; and

II) g is continuous on $\mathbb{R}$, and either strictly increasing and strictly concave on $\mathbb{R}$, or strictly increasing and strictly concave on $(0, \gamma]$ and constant on $[\gamma, \infty)$ for some $\gamma > 0$.

These conditions imply that g is nondecreasing and concave on $[0, \infty)$, is everywhere nonnegative, and satisfies the triangle inequality.

Lemma 3.2.2. *The penalty functions G_{firm} and G_p (for $-\infty < p < 1$) satisfy the above conditions.*

Proof. It is clear from the expression (3.5) for g_{firm} that G_{firm} satisfies the conditions with $\gamma = \mu$.

For G_p, by Thm. 3.1.4 we get condition I, and that g_p is differentiable on $(0, \infty)$ with $g_p' > 0$. It suffices to prove that g_p is twice differentiable on $(0, \infty)$ with $g_p'' < 0$; it will be no more difficult to show that $g_p \in C^\infty(0, \infty)$. We need some details from the construction of g_p, from [31]. We have

$$g_p(w) = (f_p^*(w) - w^2/2)/\lambda, \tag{3.8}$$

where $f_p' = s_p$ and f_p^* is the Legendre-Fenchel transform of f_p. Since s_p is continuous and nondecreasing, f_p is C^1 and convex. Then by [103, Prop. 11.3], we have that

$$x \in \partial f_p^*(w) \Leftrightarrow w = f_p'(x) = s_p(x). \tag{3.9}$$

Fix $w > 0$, and let x be such that $w = s_p(x)$. From (3.1), we must have $x > \lambda$, so $w = x - \lambda^{2-p}x^{p-1}$. If we define $F(x, w) = x - \lambda^{2-p}x^{p-1} - w$, we have that $F(\cdot, w)$ is C^∞ on $(0, \infty)$, and $\frac{\partial^k F}{\partial x^k}(x, w) \neq 0$ for $x \in (\lambda, \infty)$. Thus by the implicit function theorem, f_p^* is C^∞ on $(0, \infty)$, hence g_p is as well by (3.8).

Returning to $w = x - \lambda^{2-p}x^{p-1}$, by (3.8), (3.9), and the differentiability of f_p^*, we have

$$g_p'(w) = ((f_p^*)'(w) - w)/\lambda = (\lambda/x)^{1-p}. \tag{3.10}$$

Thus $g_p'(w)$ is decreasing in x on (λ, ∞), and since x is a strictly increasing function of w on $(0, \infty)$, $g_p''(w) < 0$ on $(0, \infty)$. □

Lemma 3.2.3. *Assume $A \in \mathbb{R}^{m \times n}$ satisfies the URP and G satisfies (I,II) above. Then the global minimizer of* (3.7) *has m or fewer nonzero entries.*

Proof. Consider w such that $Aw = b$ and $\|w\|_0 > m$. Define the matrix M to have the columns $-w_i e_i$. The set of vectors Mv with $\mathrm{supp}(v) \subset \mathrm{supp}(w)$ span a subspace of dimension greater than m. Since $\dim(\ker(A)) = n - m$, we can choose a v with $Mv \in \ker(A)$ and $\|v\|_\infty = 1$.

For all $t \in \mathbb{R}$, $w + tMv$ is feasible. Define $T = \{i : v_i \neq 0 \text{ and } |w_i| < \gamma\}$ (taking $\gamma = +\infty$ if the first case of assumption II holds). First suppose $T \neq \emptyset$. Then by assumption II, the function $t \mapsto G(w + tMv)$ is strictly concave on an interval $[-\delta, \delta]$, with $\delta > 0$ chosen small enough that every $(w+tMv)_i$ has the same sign as w_i for all $|t| \leq \delta$. Then $G(w) > \min\{G(w-\delta Mv), G(w+\delta Mv)\}$, and w is not a global minimizer.

Otherwise, we have $v_i \neq 0 \Rightarrow |w_i| \geq \gamma$. Let $t_0 = \sup\{t : \forall i \min\{|(w - tMv)_i|, |(w + tMv)_i|\} \geq \gamma\}$. Then taking $t_1 = t_0 + \delta$ with $\delta > 0$ again small enough that every $(w \pm t_1 Mv)_i$ has the same sign as w_i, then one of $|(w \pm t_1 Mv)_i|$ is less than γ for at least one i, giving a smaller value of g. Since all other components keep g constant, we have one of $G(w \pm t_1 Mv)$ being smaller than $G(w)$. □

Lemma 3.2.4. *Assume $A \in \mathbb{R}^{m \times n}$ satisfies the URP. Then the magnitudes of nonzero entries of vectors y satisfying $Aw = b$ with m or fewer nonzero entries are uniformly bounded below by some positive constant α and uniformly above by some positive constant β.*

Proof. By the URP, every m columns of A can admit no more than one solution. Thus there are no more than $\binom{n}{m}$ vectors w satisfying $Aw = b$ with m or fewer nonzero entries. Thus the set of nonzero entries of these vectors is finite and bounded below and above by α, β respectively. Neither constant depends on G in any way. □

Note that Lemma 3.2.3 and Lemma 3.2.4 imply that the global minimizer of the equality-constrained G minimization problem has nonzero entries with magnitude bounded below by α and above by β.

Next we introduce the G Nullspace Property, a generalization of the ℓ^1 Nullspace Property introduced in [36] for norms and implicitly in [73] for penalty functions belonging to a particular class. We denote $\{1, 2, \ldots, n\} = [n]$, and T^c denotes the complement of T in $[n]$.

Definition 3.2.5. The *G Nullspace Property* (or *G NSP*) of order k for the matrix A is satisfied when for all $h \in \ker(A)\backslash\{0\}$ and $T \subset [n]$ with $|T| \leq k$, one has $G(h_T) < G(h_{T^c})$.

Proposition 3.2.6. *For a penalty function G satisfying the triangle inequality, the G NSP implies exact recovery.*

Proof. We simply observe that the proof of [73] works assuming only that the penalty function satisfies the triangle inequality. □

Definition 3.2.7. Let the matrix $A \in \mathbb{R}^{m \times n}$ and the vector $b \in \mathbb{R}^m$ be given. Let x be the sparsest solution to $Aw = b$, $k = \|x\|_0$ with $2k \leq m$, and $T = \operatorname{supp}(x)$. We say the *G Restricted Nullspace Property* (or *G RNSP*) of order k is satisfied if whenever w satisfies $Aw = b$ and $\|w\|_0 \leq m$, then for $h = x - w$, we have either $h = 0$ or $G(h_T) < G(h_{T^c})$.

Note that the G NSP of order k for A implies the G RNSP of order k for A. However, examining the proof of Proposition 3.2.6 from [73] and applying Lemma 3.2.3 shows that in fact G RNSP suffices for exact recovery. We assume $2k \leq m$ to guarantee that the sparsest solution of $Aw = b$ is unique, as URP ensures that a second solution must have more than $m - k$ nonzero components.

Proposition 3.2.8. *For penalty function G satisfying the triangle inequality, G RNSP implies exact recovery.*

Theorem 3.2.9 (G exact recovery)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP and G satisfies (I,II) above. For given b, let x^* be the global minimizer of* (3.7) *and x the sparsest feasible vector. Let $k = \|x\|_0$, and define α, β to be the lower and upper bound of magnitudes of nonzero entries of feasible vectors with m or fewer nonzero components as in Lemma 3.2.4. If $2k \leq m$ and $kg(2\beta) < (m+1-k)g(\alpha)$ then $x^* = x$.*

Proof. Let $h = x^* - x$. Since x is supported on T, $h_{T^c} = x^*_{T^c}$, and so for all $t \in T^c$, $|h(t)|$ is either zero or at least α. Also, since $h \in \ker(A)$, if $h \neq 0$, then $\|h_{T^c}\|_0 \geq m+1-k$ (otherwise we would have $\|h\|_0 \leq m$, violating URP), so that $G(h_{T^c}) \geq (m+1-k)g(\alpha)$. Also,

$$G(h_T) \leq \sum_{i\in T} g(|x_i^*| + |x_i|) \leq kg(2\beta) < (m+1-k)g(\alpha), \tag{3.11}$$

by assumption. Thus either $h = 0$ or $G(h_T) < G(h_{T^c})$, so G RNSP is satisfied. □

Corollary 3.2.1 (G_{firm} exact recovery)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies URP and $G = G_{\text{firm}}$, the penalty corresponding to firm thresholding. For given b, let x^* be the global minimizer of* (3.7) *and x the sparsest feasible vector. Let $k = \|x\|_0$. If $2k \leq m$ and*

$$\mu < \min\left\{\alpha\frac{m+1-k}{k}\left(1+\sqrt{1-\frac{k}{m+1-k}}\right), 2\beta\right\}, \tag{3.12}$$

then $x^* = x$.

Proof. Since A satisfies URP and G satisfies (I,II), we may apply Theorem 3.2.9. The inequality conditions from Theorem 3.2.9 are $2k \leq m$ and $kg(2\beta) < (m+1-k)g(\alpha)$. We know $\alpha < 2\beta$. If we have $\mu \leq \alpha$, then the inequality becomes $k\mu/2 < (m+1-k)\mu/2$ which follows automatically from $2k \leq m$. And so we satisfy the hypotheses of Theorem 3.2.9, and thus have exact recovery.

If instead we have $\alpha < \mu < 2\beta$, we can evaluate the desired inequality as follows:

$$\frac{k\mu}{2} \leq (m+1-k)(\alpha - \alpha^2/2\mu), \tag{3.13}$$

$$\mu^2 - 2\frac{m+1-k}{k}\alpha\mu + \frac{m+1-k}{k}\alpha^2 < 0, \tag{3.14}$$

$$\left|\mu - \alpha\frac{m+1-k}{k}\right| < \alpha\frac{m+1-k}{k}\sqrt{1 - \frac{k}{m+1-k}}, \tag{3.15}$$

$$\alpha\frac{m+1-k}{k}\left(1 - \sqrt{1 - \frac{k}{m+1-k}}\right) < \mu < \alpha\frac{m+1-k}{k}\left(1 + \sqrt{1 - \frac{k}{m+1-k}}\right). \tag{3.16}$$

The left bound is always looser than the assumed $\alpha < \mu$ (for $2k < m+1$), so the condition $\mu < \alpha\frac{m+1-k}{k}\left(1 + \sqrt{1 - \frac{k}{m+1-k}}\right)$ gives the desired inequality and guarantees exact recovery. □

Corollary 3.2.2 (G_p exact recovery)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP and $G = G_p$, the $p-$shrinkage penalty. For given b, let x^* be the global minimizer of* (3.7) *and x the sparsest feasible vector. Let $k = \|x\|_0$. If $2k \leq m$ then there exist $\lambda > 0$ and $0 < p < 1$ sufficiently small that $x^* = x$. For any $p < 0$ there also exists $\lambda > 0$ sufficiently small that $x^* = x$.*

Proof. Since A satisfies the URP and G_p satisfies (I,II), we may apply Theorem 3.2.9. The inequality conditions from Theorem 3.2.9 are $2k \leq m$ and $kg(2\beta) < (m+1-k)g(\alpha)$.

Fix $w > 0$. As in the proof of Lemma 3.2.2, we have

$$g_p(w) = (f_p^*(w) - w^2/2)/\lambda, \tag{3.17}$$

where $f_p' = s_p$ and f_p^* is the Legendre-Fenchel transform of f_p and is smooth at w. Let $x = (f_p^*)'(w)$, noting that while w is fixed, x depends on λ and p. By (3.9), we have $s_p(x) = w$, so that

$$x - w = \lambda^{2-p}x^{p-1}. \tag{3.18}$$

Furthermore, by [103, Prop. 11.3], we have

$$x = \arg\min_x \big(xw - f_p(x)\big), \tag{3.19}$$

so that by definition of the Legendre-Fenchel transform,

$$f_p^*(w) = xw - f_p(x). \tag{3.20}$$

Combining (3.17), (3.18), and (3.20), we obtain

$$\begin{aligned} g_p(w) &= (xw - f_p(x) - w^2/2)/\lambda \\ &= \big(xw - x^2/2 + \lambda^{2-p}x^p/p - \lambda^2(1/p - 1/2) - w^2/2\big)/\lambda \qquad (3.21) \\ &= \lambda^{1-p}x^p/p - (x-w)^2/(2\lambda) - \lambda(1/p - 1/2) \\ &= \frac{\lambda}{p}\,(x/\lambda)^p - \frac{\lambda}{2}(x/\lambda)^{2p-2} - \lambda(1/p - 1/2). \qquad (3.22) \end{aligned}$$

(In (3.21), the expression for $f_p(x)$ is obtained by antidifferentiating s_p with $f_p(0) = 0$.)

Case $0 < p < 1$ We want to show that for sufficiently small $0 < \lambda$ and $0 < p < 1$, $g(2\beta)/g(\alpha) < (m+1-k)/k$. By hypothesis, $(m+1-k)/k > 1$. So it suffices to show for any fixed α, β with $0 < \alpha < 2\beta$, that $g(2\beta)/g(\alpha) \to 1$ as $(p, \lambda) \to (0^+, 0^+)$.

By (3.18), $x > w$ for any λ and p, so $\lim_{\lambda\to 0^+}(x/\lambda) = \infty$. Then for $p < 1$,

$$\lim_{\lambda\to 0^+} g_p(w) - \frac{\lambda}{p}\big[(x/\lambda)^p - 1\big] = 0. \tag{3.23}$$

Now

$$\frac{\lambda}{p}\big[(x/\lambda)^p - 1\big] = \frac{\lambda}{p}\big[\exp\!\big(p\log(x/\lambda)\big) - 1\big] = \frac{\lambda}{p}\big[p\log(x/\lambda) + o\big(p\log(x/\lambda)\big)\big], \tag{3.24}$$

where the little-o is as $p\log(x/\lambda) \to 0^+$, which we wish to establish as $p, \lambda \to 0^+$. Since $x > w$, we have that

$$p\log(x/\lambda) = p\log(w/\lambda + (x/\lambda)^{p-1}) < p\log(w/\lambda + (w/\lambda)^{p-1}) \to 0^+, \tag{3.25}$$

provided $p \to 0^+$ fast enough, such as if $p \sim \lambda^q$ for any $q > 0$. This yields

$$\begin{aligned}\liminf_{(\lambda,p)\to(0^+,0^+)} \frac{g_p(2\beta)}{g_p(\alpha)} &= \liminf_{(\lambda,p)\to(0^+,0^+)} \frac{\lambda\log(x(2\beta)/\lambda)}{\lambda\log(x(\alpha)/\lambda)} \\ &\leq \liminf_{(\lambda,p)\to(0^+,0^+)} \frac{\log(2\beta/\lambda + (2\beta/\lambda)^{p-1})}{\log(\alpha/\lambda)} = \liminf_{\lambda\to 0^+} \frac{\log(2\beta)-\log(\lambda)}{\log(\alpha)-\log(\lambda)} = 1. \quad (3.26)\end{aligned}$$

Therefore, there exist $\lambda > 0, p > 0$ sufficiently small that $kg(2\beta) < (m+1-k)g(\alpha)$.

Case $p < 0$ Since g_p is strictly increasing on $[0,\infty)$, we take $w \to \infty$ to determine an upper bound. Note that $x(w) > w$ implies that $x(w) \to \infty$ as $w \to \infty$. Then from (3.22), since now $p < 0$, we obtain

$$\lim_{w\to\infty} g_p(w) = \lambda(1/2 - 1/p). \quad (3.27)$$

Thus for $p < 0$ and all w, λ, we have $g_p(w) \leq \lambda(1/2 - 1/p)$. Applying this with $w = 2\beta$ and using (3.22),

$$\liminf_{\lambda\to 0^+} \frac{g_p(2\beta)}{g_p(\alpha)} \leq \liminf_{\lambda\to 0^+} \frac{\lambda(1/2-1/p)}{\lambda\left[\frac{1}{p}(x(\alpha)/\lambda)^p - \frac{1}{2}(x(\alpha)/\lambda)^{2p-2} - (1/p - 1/2)\right]}. \quad (3.28)$$

As before, $(x/\lambda) \to \infty$ as $\lambda \to 0^+$. Then

$$\liminf_{\lambda\to 0^+} \frac{g_p(2\beta)}{g_p(\alpha)} \leq \lim_{\lambda\to 0^+} \frac{\lambda(1/2-1/p)}{\lambda(1/2-1/p)} = 1. \quad (3.29)$$

Thus for every $p < 0$ there exists $\lambda > 0$ sufficiently small that $kg(2\beta) < (m+1-k)g(\alpha)$. □

3.3 Stability

Next we consider the case of noisy measurements of an approximately sparse signal. Let x be the original signal with $\|Ax - b\|_2 \leq \epsilon$ whose k-sparse approximation is supported on T, i.e.

$x_T = \arg\min_w G(x-w)$ subject to $\|w\|_0 = k$. We wish to bound $G(x^* - x)$ where

$$x^* = \arg\min_w G(w) \text{ subject to } \|Aw-b\|_2 \le \epsilon. \tag{3.30}$$

We shall bound the recovery error by the sum of a term dependent on the noise level and a term dependent on the sparse approximation error.

We shall first need two results: bounds on the magnitudes of nonzero entries of local minima of (3.30) and an extension of those bounds to the error vector projected onto the null space of A. Recall that $\|w\|_{-\infty} := \min_i |w_i|$.

Lemma 3.3.1. *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP and G satisfies (I,II) above. Let $b \in \mathbb{R}^m$ be given. For $S \subset [n]$ with $|S| = m$ define $\alpha_S = \|A_S^{-1}b\|_{-\infty}$ and $\beta_S = \|A_S^{-1}b\|_\infty$. If $\epsilon < \min_S(\alpha_S/\|A_S^{-1}\|)$, then the magnitudes of components of feasible vectors of* (3.30) *are bounded below by $\alpha := \min_S(\alpha_S - \|A_S^{-1}\|\epsilon) > 0$ and bounded above by $\beta := \max_S(\beta_S + \|A_S^{-1}\|\epsilon)$.*

The assumption that $\alpha_S > 0$ for all S has a similar character to the URP, in that it is true with probability 1 for random data drawn from an absolutely continuous distribution.

Proof. First, note that the error-bounded problem (3.30) is equivalent to taking the G minimizer from a set of equality-constrained G minimizers (with different equality constraints): For all feasible w, we must have $Aw = b+\eta$ for some $\|\eta\|_2 \le \epsilon$. Thus by Lemma 3.2.3 the minimizer of (3.30) has m or fewer nonzero entries. By the URP, any m columns S of A give exactly one solution to $A_S w = b + \eta$. So we have

$$\begin{aligned}\|w\|_{-\infty} &= \|A_S^{-1}(b+\eta)\|_{-\infty} \ge \min_i(|A_S^{-1}b| - |A_S^{-1}\eta|)_i \\ &\ge \|A_S^{-1}b\|_{-\infty} - \|A_S^{-1}\eta\|_\infty \ge \alpha_S - \|A_S^{-1}\eta\|_2 \ge \alpha_S - \|A_S^{-1}\|\epsilon \\ &\ge \alpha,\end{aligned} \tag{3.31}$$

and

$$\begin{aligned}\|w\|_\infty &= \|A_S^{-1}(b+\eta)\|_\infty \leq \|A_S^{-1}b\|_\infty + \|A_S^{-1}\eta\|_\infty \\ &\leq \beta_S + \|A_S^{-1}\eta\|_2 \leq \beta_S + \|A_S^{-1}\|\epsilon \\ &\leq \beta. \end{aligned} \tag{3.32}$$

□

Lemma 3.3.2. *Assume G satisfies (I,II). Let x^* be the global minimizer of* (3.30)*, x the original signal with $\|Ax - b\| \leq \epsilon$, and let T be the support of the k-sparse approximation of x. Let α_S, α, and β be as in Lemma 3.3.1. Define $\alpha' := \alpha - \|x_{T^c}\|_\infty - 2\epsilon$ and $\beta' := \beta + \epsilon$. If A satisfies the URP, $AA^T = I$, $\min_S \alpha_S > \|x_{T^c}\|_\infty$ (requiring that x be nearly k sparse), and $\epsilon < \min_S\{(\alpha_S - \|x_{T^c}\|_\infty)/(2 + \|A_S^{-1}\|)\}$, then the orthogonal projection w of $h = x^* - x$ onto the nullspace of A satisfies*

$$\alpha' \leq \|w_{T^c}\|_{-\infty} \textit{ and } \|w_{T^c}\|_\infty \leq 2\beta'. \tag{3.33}$$

Proof. First, consider the bound $\epsilon < \min_S\{(\alpha_S - \|x_{T^c}\|_\infty)/(2+\|A_S^{-1}\|)\}$. Note that this is stronger than the bound on ϵ from Lemma 3.3.1, and it implies $2\epsilon + \|x_{T^c}\|_\infty < \alpha$. We see this from the following inequalities:

$$\begin{aligned}\alpha &= \min_S\{\alpha_S - \epsilon\|A_S^{-1}\|\} \\ &> \min_S\{\alpha_S - (\alpha_S - \|x_{T^c}\|_\infty)\|A_S^{-1}\|/(2 + \|A_S^{-1}\|)\} \\ &= \min_S\left\{\frac{2\alpha_S}{2 + \|A_S^{-1}\|} + \frac{\|A_S^{-1}\|\|x_{T^c}\|_\infty}{2 + \|A_S^{-1}\|}\right\} \\ &= \min_S\left\{\frac{2\alpha_S - 2\|x_{T^c}\|_\infty}{2 + \|A_S^{-1}\|} + \frac{(2 + \|A_S^{-1}\|)\|x_{T^c}\|_\infty}{2 + \|A_S^{-1}\|}\right\} \\ &> 2\epsilon + \|x_{T^c}\|_\infty. \end{aligned} \tag{3.34}$$

We shall use this below to guarantee $\alpha' > 0$.

Note that the hypotheses of Lemma 3.3.1 are satisfied, giving $\|x^*\|_{-\infty} \geq \alpha$ and $\|x^*\|_\infty \leq \beta$, $\|x\|_\infty \leq \beta$. Since $AA^T = I$, the orthogonal projection of h onto the nullspace of A is $(I - A^T A)h$. The desired lower bound comes from the following sequence of inequalities, using the given lower bound on nonzero elements of x^*, the feasibility of x^* and x, the fact $\|A^T A\| = 1$, and the assumed bound on ϵ:

$$\begin{aligned}
\|[(I - A^T A)h]_{T^c}\|_{-\infty} &\geq \|h_{T^c}\|_{-\infty} - \|A^T Ah\|_\infty \\
&\geq \|x^*_{T^c} - x_{T^c}\|_{-\infty} - \|A^T Ah\|_2 \\
&\geq \|x^*_{T^c}\|_{-\infty} - \|x_{T^c}\|_\infty - \|h\|_2 \\
&\geq \alpha - \|x_{T^c}\|_\infty - 2\epsilon = \alpha' > 0.
\end{aligned}$$

The upper bound comes from a completely analogous argument:

$$\begin{aligned}
\|(I - A^T A)h\|_\infty &\leq \|h\|_\infty + \|A^T Ah\|_\infty \\
&\leq \|x^* - x\|_\infty + \|A^T Ah\|_2 \\
&\leq 2\beta + 2\epsilon = 2\beta'.
\end{aligned}$$

□

Definition 3.3.3. The *G Noisy Nullspace Property* (or *G NNSP*) of order k for the matrix A is satisfied when for all $h \in \mathbb{R}^n$ and $S \subset [n]$ with $|S| \leq k$, there are constants $0 \leq \tau < 1$ and $D \geq 0$ such that

$$G(h_S) \leq \tau G(h_{S^c}) + D\|Ah\|_2. \tag{3.35}$$

Proposition 3.3.4. *Assume G satisfies the triangle inequality. For given A, b, let x^* be the global minimizer of* (3.30) *and let x be the original signal with $\|Ax - b\|_2 \leq \epsilon$ whose k-sparse approximation is supported on T. Then the G NNSP of order k for A implies the following stability* *bound:*

$$G(x^* - x) \leq C_1\epsilon + C_2 G(x_{T^c}), \tag{3.36}$$

with $C_1 = 4D/(1-\tau)$ *and* $C_2 = 2(1+\tau)/(1-\tau)$*, where* τ *and* D *satisfy* (3.35).

Proof. Define the error vector $h = x^* - x$. Since x^* and x are both feasible and $\|A\| = 1$, $\|Ah\|_2 \leq 2\epsilon$. Then by the triangle inequality of G,

$$G(x_T) - G(-h_T) \leq G(x_T + h_T). \tag{3.37}$$

Since G decouples across components,

$$G(x_T + h_T) + G(h_{T^c}) = G(x_T + h_T + h_{T^c}) = G(x^* - x_{T^c}). \tag{3.38}$$

Then

$$\begin{aligned} G(h_{T^c}) &\leq G(x^* - x_{T^c}) + G(h_T) - G(x_T) \\ &\leq G(x^*) + G(x_{T^c}) + G(h_T) - G(x_T) \\ &\leq G(x) + G(x_{T^c}) + G(h_T) - G(x_T) \\ &= 2G(x_{T^c}) + G(h_T). \end{aligned} \tag{3.39}$$

Now apply G NNSP to h on T:

$$G(h_T) \leq \tau G(h_{T^c}) + D\|Ah\|_2, \tag{3.40}$$

so that

$$G(h_T) \leq \frac{2}{1-\tau}\big(D\epsilon + \tau G(x_{T^c})\big). \tag{3.41}$$

Using (3.39), we obtain

$$G(h_{T^c}) \le 2G(x_{T^c}) + G(h_T) \le \frac{2D\epsilon}{1-\tau} + \frac{2}{1-\tau}G(x_{T^c}). \tag{3.42}$$

Now we add (3.41) and (3.42) to get the desired inequality:

$$\begin{aligned} G(h) &= G(h_T) + G(h_{T^c}) \\ &\le \frac{2}{1-\tau}\big(D\epsilon + \tau G(x_{T^c})\big) + \frac{2D}{1-\tau}\epsilon + \frac{2}{1-\tau}G(x_{T^c}) \\ &= \frac{4D}{1-\tau}\epsilon + \frac{2(1+\tau)}{1-\tau}G(x_{T^c}). \end{aligned} \tag{3.43}$$

□

Theorem 3.3.5 (G stability)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP, $AA^T = I$, G satisfies (I,II) above, and $G(v) \le C\sqrt{n}\|v\|_2$ for some constant $C > 0$. For given b, let x be the original signal with $\|Ax - b\|_2 \le \epsilon$, let T be the support of its k-sparse approximation, and suppose* $\min_S\{\alpha_S\} > \|x_{T^c}\|_\infty$*. Let x^* be the global minimizer of* (3.30)*, where $\epsilon < \min_S\{(\alpha_S - \|x_{T^c}\|_\infty)/(2 + \|A_S^{-1}\|)\}$ (with α_S defined as in Lemma 3.3.1). Define α', β' as in Lemma 3.3.2. Assume that $2k < n$ and $kg(2\beta') < (n-k)g(\alpha')$. Then*

$$G(x^* - x) \le 2\left(1 - \frac{kg(2\beta')}{(n-k)g(\alpha')}\right)^{-1}\left[2C\sqrt{n}\epsilon + \left(1 + \frac{kg(2\beta')}{(n-k)g(\alpha')}\right)G(x_{T^c})\right]. \tag{3.44}$$

Proof. We shall show that the given hypotheses allow for the same application of the G NNSP as in Proposition 3.3.4, and in a similar way, arrive at stability. Define $h = x^* - x$. Since G satisfies the triangle inequality, we have $G(h_{T^C}) \le G(h_T) + 2G(x_{T^C})$, as in the proof of Proposition 3.3.4.

Next we write h as the sum of its orthogonal projections onto $\ker(A)$ and $\ker(A)^\perp$, which we denote by w and v respectively. First, suppose that there exists some $0 \le \tau < 1$ such that

$G(w_T) \leq \tau G(w_{T^c})$ (which we will prove below). Then we have:

$$\begin{aligned}
G(h_T) &\leq G(w_T) + G(v_T) \\
&\leq \tau G(w_{T^c}) + G(v_T) = \tau G(w_{T^c} + v_{T^c} - v_{T^c}) + G(v_T) \\
&\leq \tau G(h_{T^c}) + G(v_{T^c}) + G(v_T) \\
&= \tau G(h_{T^c}) + G(v) \\
&\leq \tau G(h_{T^C}) + C\sqrt{n}\|v\|_2. \qquad (3.45)
\end{aligned}$$

Since $AA^T = I$ and $v \in \ker(A)^\perp$, it follows that $v = A^T Av$. Hence $\|v\|_2^2 = \|Av\|_2^2$. Then from (3.45) we obtain

$$G(h_T) \leq \tau G(h_{T^c}) + C\sqrt{n}\|Av\|_2. \qquad (3.46)$$

And so we have the application of the G NNSP to h on T with constants τ and $D = C\sqrt{n}$. From here the stability inequality (3.44) follows as in Proposition 3.3.4.

Now we go back to prove $G(w_T) \leq \tau G(w_{T^c})$. We shall use the lower bound $\|w_{T^c}\|_{-\infty} \geq \alpha'$ and the upper bound $\|w_T\|_\infty \leq \beta'$ from Lemma 3.3.2. We overestimate $G(w_T)$ and underestimate $G(w_{T^c})$ as follows:

$$G(w_T) \leq kg(2\beta'), \quad G(w_{T^c}) \geq (n-k)g(\alpha'). \qquad (3.47)$$

So to get $G(w_T) \leq \tau G(w_{T^c})$, it suffices to have $kg(2\beta') \leq \tau(n-k)g(\alpha')$, and thus $kg(2\beta') < (n-k)g(\alpha')$ guarantees some $0 \leq \tau < 1$. The condition $k < n-k$ gives $(n-k)/k > 1$ and thus makes the inequality possible for $\alpha' < 2\beta'$.

Plugging in $\tau = \frac{kg(2\beta')}{(n-k)g(\alpha')}$ to the stability inequality we get from the previous argument gives

$$G(h) \leq 2\left(1 - \frac{kg(2\beta')}{(n-k)g(\alpha')}\right)^{-1}\left[2C\sqrt{n}\epsilon + \left(1 + \frac{kg(2\beta')}{(n-k)g(\alpha')}\right)G(x_{T^c})\right]. \qquad (3.48)$$

□

Corollary 3.3.1 (G_{firm} stability)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP, $AA^T = I$, and $G = G_{\text{firm}}$, the penalty corresponding to firm thresholding. For given b, let x be the original signal with $\|Ax - b\|_2 \leq \epsilon$ whose k-sparse approximation is supported on T, with $\min_S\{\alpha_S\} > \|x_{T^c}\|_\infty$, and x^* be the global minimizer of* (3.30)*, where $\epsilon < \min_S\{(\alpha_S - \|x_{T^c}\|_\infty)/(2 + \|A_S^{-1}\|)\}$ (with α_S defined as in Lemma 3.3.1). Define α', β' as in Lemma 3.3.2. If $2k < n$ and $\mu < \min\{\alpha' \frac{n-k}{k}\left(1 + \sqrt{1 - \frac{k}{n-k}}\right), 2\beta'\}$ then x^* is stable, satisfying the following inequality:*

$$G_{\text{firm}}(x^* - x) \leq 2\left(1 - \frac{kg_{\text{firm}}(2\beta')}{(n-k)g_{\text{firm}}(\alpha')}\right)^{-1}\left[2C\sqrt{n}\epsilon + \left(1 + \frac{kg_{\text{firm}}(2\beta')}{(n-k)g_{\text{firm}}(\alpha')}\right)G_{\text{firm}}(x_{T^c})\right]. \tag{3.49}$$

The proof of Corollary 3.3.1 is an application of Theorem 3.3.5 combined with the corresponding computations from the proof of Corollary 3.2.1.

Corollary 3.3.2 (G_p stability)**.** *Assume $A \in \mathbb{R}^{m\times n}$ satisfies the URP, $AA^T = I$, and $G = G_p$, the penalty corresponding p-shrinkage. For given b, let x be the original signal with $\|Ax - b\|_2 \leq \epsilon$ whose k-sparse approximation is supported on T, with $\min_S\{\alpha_S\} > \|x_{T^c}\|_\infty$, and x^* be the global minimizer of* (3.30)*, where $\epsilon < \min_S\{(\alpha_S - \|x_{T^c}\|_\infty)/(2 + \|A_S^{-1}\|)\}$ (with α_S defined as in Lemma 3.3.1). If $2k < n$ then there exist $0 < p < 1, 0 < \lambda$ sufficiently small so that x^* is stable, satisfying the following inequality.*

$$G_p(x^* - x) \leq 2\left(1 - \frac{kg_p(2\beta')}{(n-k)g_p(\alpha')}\right)^{-1}\left[2C\sqrt{n}\epsilon + \left(1 + \frac{kg_p(2\beta')}{(n-k)g_p(\alpha')}\right)G(x_{T^c})\right]. \tag{3.50}$$

Also, for any $p < 0$ there exists $\lambda > 0$ sufficiently small such that x^* is stable, and the above inequality holds.

The proof of Corollary 3.3.2 is an application of Theorem 3.3.5 combined with the corresponding computations from the proof of Corollary 3.2.2.

3.4 Convergence of iterative p-shrinkage

Now we consider an algorithm that employs generalized shrinkage. Consider the following optimization problem:

$$\min_x F_p(x) := \lambda G_p(x) + \tfrac{1}{2}\|Ax - b\|_2^2, \tag{3.51}$$

where $\|A\| < 1$. Applying forward-backward splitting to this problem gives *iterative p-shrinkage* (IPS):

$$x^{n+1} = S_p(x^n - A^T(Ax^n - b)). \tag{3.52}$$

This generalizes the *iterative soft thresholding* algorithm (ISTA) [44], which is the case $p = 1$. ISTA was shown in [44] to be globally convergent to a global minimizer (necessarily, since F_1 is convex). In this section, we prove global convergence of IPS for general $p < 1$, though only to a stationary point of F_p. Portions of the proof appeared in [132], though statements there concerning convergence to a local minimizer are incorrect.

Recall from Lemma 3.2.2 that g_p is C^∞ on $(0, \infty)$. A closer examination of the proof shows that g_p on $[0, \infty)$ is the restriction of a function that is C^∞ on $\mathbb{R}$, so g_p is one-sided differentiable to all orders at $w = 0$.

The following follows exactly as in the known case of $p = 1$ [44]:

Lemma 3.4.1 ([132])**.** *Let $\lambda > 0$ and $p \in \mathbb{R}$, and define $\{x^n\}$ by* (3.52)*, with x^0 arbitrary.*

1. *$F(x^{n+1}) \leq F(x^n)$ for all n, and $F(x^{n+1}) < F(x^n)$ unless x^n is a fixed point of the algorithm.*

2. *$\|x^{n+1} - x^n\|_2 \to 0$.*

Lemma 3.4.2. *Let $\lambda > 0$ and $p \in \mathbb{R}$. The fixed points of* (3.52) *are precisely the stationary points of F_p.*

Proof. The iteration (3.52) can be seen as minimizing the surrogate functional

$$\lambda G_p(x) + \tfrac{1}{2}\|Ax - b\|_2^2 + \tfrac{1}{2}\|x - w\|_2^2 - \tfrac{1}{2}\|Ax - Aw\|_2^2, \tag{3.53}$$

with fixed $w = x^n$, by expanding the quadratic terms and rearranging to express the minimizer in terms of the proximal mapping of G_p. Therefore the first-order optimality condition of this functional is satisfied at $x = x^{n+1}$. Also, the first-order optimality condition of this functional at $x = x^n$ is the same as the first-order optimality condition of F_p at $x = x^n$. Hence $x^{n+1} = x^n$ if and only if the first-order optimality condition of F_p at $x = x^n$ is satisfied. □

The lemma shows why it is not possible to show that IPS converges to a local minimizer: if the algorithm happens to be initialized with a stationary point that is not a local minimizer (*i.e.*, a saddle point or local maximizer), then the initializer is a fixed point of the algorithm, so the algorithm cannot converge to a local minimizer in such a case.

Lemma 3.4.3. *Fix $\lambda > 0$, $p \in (-\infty, 1)$. We have $g_p''' > 0$ on $(0, \infty)$, $g_p''' < 0$ on $(-\infty, 0)$, $g_p'''(0+) > 0$, and $g_p'''(0-) < 0$.*

Proof. Since g_p is even, it suffices to consider $w > 0$. Above we had that $x = x(w) = (f_p^*)'(w)$ satisfies $x - \lambda^{2-p}x^{p-1} = w$. Differentiating with respect to w, we have that

$$x' - \lambda^{2-p}(p-1)x^{p-2}x' = 1, \tag{3.54}$$

so

$$x' = \big(1 - \lambda^{2-p}(p-1)x^{p-2}\big)^{-1}. \tag{3.55}$$

Since $p < 1$, $(f_p^*)''(w) = x'(w) > 0$ for all $w > 0$.

Differentiating (3.54), we get

$$x'' - \lambda^{2-p}(p-1)\big[(p-2)x^{p-3}(x')^2 + x^{p-2}x''\big] = 0, \tag{3.56}$$

or

$$x''\big(1 - \lambda^{2-p}(p-1)x^{p-2}\big) = \lambda^{2-p}(p-1)(p-2)x^{p-3}(x')^2, \tag{3.57}$$

implying that x'' has the same sign as x. Since $x(w)$ has the same sign as w, we have that $(f_p^*)'''(w)$ has the same sign as w for $w \neq 0$.

Differentiating the relation (3.8) defining g_p, we obtain $w + \lambda g_p'(w) = (f_p^*)'(w)$, $1 + \lambda g_p''(w) = (f_p^*)''(w)$, and $\lambda g_p'''(w) = (f_p^*)'''(w)$. Thus $g_p'''(w)$ has the same sign as w for $w \neq 0$ as well. Also, $\lambda g_p'''(0+) = (f_p^*)'''(0+) = \lim_{w \to 0^+} x''(w)$. Since $\lim_{w \to 0^+} x(w) = \lambda$, we obtain from (3.55) and (3.57) that $(f_p^*)'''(0+) = \frac{1-p}{(2-p)^2}\lambda^{-1} > 0$. Thus $g'''(0+) > 0$. □

Lemma 3.4.4. *Let* $p \geq 0$*. Then* $\{x^n\}$ *is bounded.*

Proof. Since $\{F_p(x^n)\}$ decreases monotonically, it suffices to show that F_p is coercive, which we establish be showing coercivity of g_p. By (3.18), if $w \to \infty$, then $x \to \infty$. For $p > 0$, that $g_p(w) \to \infty$ follows from (3.22). The $p = 0$ case is similar, but f_0 has a different form:

$$\begin{aligned} g_0(w) &= (xw - f_0(x) - w^2/2)/\lambda \\ &= \big(xw - x^2/2 + \lambda^2 \log x - \lambda^2(\log\lambda - 1/2) - w^2/2\big)/\lambda \\ &= \lambda \log x - (x-w)^2/(2\lambda) - \lambda(\log\lambda - 1/2) \\ &= \lambda \log x - \tfrac{\lambda}{2}(x/\lambda)^{-2} - \lambda(\log\lambda - 1/2). \end{aligned} \tag{3.58}$$

From this the coercivity of g_0 follows. □

Lemma 3.4.5. *Let* $p < 0$*, and assume* $\lambda^2 > p\|b\|_2^2/(p-2)$*. Let* $x^0 = 0$*. Then* $\{x^n\}$ *is bounded.*

Proof. From Lemma 3.4.1, we know that $F_p(x^n)$ decreases (strictly except at a fixed point, in which case we are done). Then for $n \geq 1$,

$$F_p(x^n) < F_p(x^0) = \|b\|_2^2/2, \tag{3.59}$$

so

$$G_p(x^n) \leq F_p(x^n)/\lambda < \|b\|_2^2/(2\lambda). \tag{3.60}$$

By (3.27), $g_p(w) < (1/2 - 1/p)\lambda$. Combining this bound with (3.60), we obtain for each j,

$$g_p(x_j^n) \leq G_p(x^n) < \|b\|_2^2/(2\lambda) < (1/2 - 1/p)\lambda. \tag{3.61}$$

Letting t be the unique positive number satisfying $g(t) = \|b\|_2^2/(2\lambda)$, we obtain $\|x^n\|_\infty < t$ independently of n. □

Now we can establish convergence of our algorithm.

Theorem 3.4.6. *Let $\lambda > 0$, $p \in (-\infty, 1)$. Let the sequence $\{x^n\}$ be defined by* (3.52)*, with x^0 arbitrary for $p \geq 0$, and $x^0 = 0$ for $p < 0$ in which case we further assume $\lambda^2 > p\|b\|_2^2/(p-2)$. Then $\{x^n\}$ converges to a stationary point of F.*

Proof. We have that $F_p(x^{n+1}) < F_p(x^n)$ unless x^n is a fixed point, F is continuous, and the sequence $\{x^n\}$ is bounded. Then by [95, Thm. 3.1], we have that either $\{x^n\}$ converges or its limit points form a continuum. (A continuum is a compact, connected set; here we also exclude the degenerate case of a singleton.) Since we already know that any limit point of $\{x^n\}$ will be a stationary point of F_p, we complete the proof by showing that the stationary points of F_p cannot form a continuum.

Let E be the set of stationary points of F_p, and suppose E is a continuum. Fix $\bar{x} \in E$. For any $\epsilon > 0$, it cannot be that $\mathcal{N}(\bar{x}; \epsilon) \cap E = \{\bar{x}\}$, otherwise $\{\bar{x}\}$ would be both open and closed in E, contrary to E being connected. Thus there is a sequence of stationary points $\bar{x} + v^n$ with $v^n \neq 0$, $v^n \to 0$.

Since $\{v^n/\|v^n\|\}$ is a sequence of unit vectors, it cannot converge to zero. Then we can fix j such that $\{v_j^n/\|v^n\|\}$ does not tend to zero, though of course $v_j^n \to 0$. First suppose that $\bar{x}_j \neq 0$. By considering a tail of v_j^n, we can assume that $\bar{x}_j + v_j^n \neq 0$ for all n. Then g_p is differentiable at $\bar{x}_j$ and $\bar{x}_j + v_j^n$, and since $\bar{x}$ and $\bar{x} + v^n$ are fixed points,

$$\lambda^{2-p} g_p'(\bar{x}_j + v_j^n) + \left[A^T(A(\bar{x} + v^n) - b)\right]_j = 0, \tag{3.62}$$

and

$$\lambda^{2-p} g_p'(\bar{x}_j) + \left[A^T(A\bar{x} - b)\right]_j = 0. \tag{3.63}$$

Define $\varphi(x) = \lambda g_p'(x_j) + \left[A^T(Ax - b)\right]_j$. All derivatives of φ exist at every $x \neq 0$. Letting

(a_i) denote the columns of A, if $i \neq j$, we have $\partial\varphi/\partial x_i(\bar{x}) = \langle a_i, a_j\rangle$, while $\partial\varphi/\partial x_j(\bar{x}) = \lambda g''(\bar{x}_j) + \|a_j\|^2$. Also, $\varphi(\bar{x}) = 0$ and each $\varphi(\bar{x} + v^n) = 0$. By differentiability of φ, we have

$$\frac{\varphi(\bar{x} + v^n) - \varphi(\bar{x}) - \nabla\varphi(\bar{x}) \cdot v^n}{\|v^n\|} \to 0. \tag{3.64}$$

Since the first two terms of (3.64) are zero, $\nabla\varphi(\bar{x}) \cdot v^n = o(\|v^n\|)$ as well. By continuity of $\nabla\varphi$ at $\bar{x}$, it is straightforward to show that $\nabla\varphi(\bar{x} + v^n) \cdot v^n = o(\|v^n\|)$ also.

Now we consider second derivatives. $\partial^2\varphi/\partial x_i \partial x_k(\bar{x}) = 0$, unless $i = k = j$, while $\partial^2\varphi/\partial x_j^2(\bar{x}) = \lambda g_p'''(\bar{x}_j)$. Now by the differentiability of $\nabla\varphi$,

$$\|\nabla\varphi(\bar{x} + v^n) - \nabla\varphi(\bar{x}) - \nabla^2\varphi(\bar{x})\, v^n\| = o(\|v^n\|), \tag{3.65}$$

so

$$\nabla\varphi(\bar{x} + v^n) \cdot v^n - \nabla\varphi(\bar{x}) \cdot v^n - v^n \cdot \nabla^2\varphi(\bar{x})\, v^n = o(\|v^n\|^2). \tag{3.66}$$

But from the above we have that the first two terms are $o(\|v^n\|^2)$, so $v^n \cdot \nabla^2\varphi(\bar{x})\, v^n = o(\|v^n\|^2)$ as well. But this is $\lambda g_p'''(\bar{x}_j)(v_j^n)^2$; since $(v_j^n)^2/\|v^n\|^2$ does not tend to zero by choice of j, it must be that $g_p'''(\bar{x}_j) = 0$, a contradiction.

Thus we must have $\bar{x}_j = 0$. By choice of j, infinitely many $v_j^n \neq 0$, so by passing to a subsequence we may assume that either all $v_j^n > 0$ or $v_j^n < 0$. By the one-sided differentiability of g_p, we can then repeat the above argument using a smooth extension of g_p to $\mathbb{R}$. Since neither $g_p'''(0+)$ nor $g_p'''(0-)$ are zero, we will obtain the same contradiction. Therefore E cannot be a continuum, and the sequence $\{x^n\}$ defined by (3.52) is convergent to a stationary point of F_p. □

3.5 Conclusion

We have shown that for given signals with reasonable sparsity assumptions and a broad class of measurement matrices, the families of penalties corresponding to p-shrinkage and firm thresholding, like the ℓ^p quasinorms, provide a candidate penalty that is able to exactly recover the given data with the given measurement matrix. Further we have shown that these penalties behave well with respect to the addition of noise in the measurements, or only approximately sparse signals (as is often the case in practical settings). Finally, we have shown that iterative p-shrinkage converges to stationary points of the unconstrained energy. These results, together with empirical results (see [31], and Figure 3.3), further support the idea that generalized shrinkage penalties can be an advantageous alternative to standard ℓ^1 compressed sensing, or ℓ^p compressed sensing.

Further work could benefit from exploring in what generality these type of results hold. The theory of generalized shrinkage allows for an endless possibility of other shrinkages and penalties to study. Additionally, the methods of proof may apply to compressed sensing relaxations that arise in other ways. Generally speaking, determining conditions under which convex optimization results can be extended to handle nonconvex functionals may continue to be a fruitful area of research. Lastly, we make no claims that the approximations made in these proofs give the tightest results possible, so further refinement of these results may be possible and interesting.

CHAPTER 4

Point Localization and Density Estimation from Ordinal kNN graphs using Synchronization

4.1 Related Work

4.1.1 Multidimensional Scaling

Broadly speaking, multidimensional scaling (MDS) refers to a number of related problems and methods. In Classical Multidimensional Scaling (CMDS) [126], one is given all Euclidean Squared-Distance measurements $\Delta_{ij} = \|\vec{x}_i - \vec{x}_j\|_2^2$ on a set of points $X = \{\vec{x}_i\}_{i=1}^n$ and wishes to approximate the points, assuming that they approximately lie in a low-dimensional space $d \ll n$. Note that the solution for the coordinates is unique only up to rigid transformations, and that solutions do not exist for all possible inputs Δ.

One can generalize CMDS to incorporate additional nonnegative weights W_{ij} on each distance, useful when some distances are missing, or most distances are noisy, but some are known. The optimization involves minimizing an energy known in the literature as *stress* [84]. One approach to minimize stress is to iteratively minimize a majorizing function of two variables. A further generalization of MDS is non-metric MDS, or Ordinal Embedding, in which the input D is assumed to be an increasing function applied to distance measurements [112]. This may be the case if D represents dissimilarity between points, as opposed to measured distances. The problem can again be expressed with stress functionals and is usually solved with isotonic regression [83].

4.1.2 Semidefinite Programming methods

Semidefinite Programming methods (SDP) have been applied frequently to MDS and related problems. Classical MDS can be stated as an SDP, with a closed form solution. Any formulation of the problem that optimizes over the Gram matrix requires the semidefinite constraint $K \in \mathbb{S}^n_+$. Indeed, for metric MDS, if one penalizes the squared error on the squared distance measurements, the problem can be written as

$$\begin{aligned} &\min_{X \in \mathbb{R}^{d \times n}} \sum_{ij} W_{ij}(\Delta_{ij} - \Delta_{ij}(X))^2 \\ = &\min_{K \in \mathbb{S}^n_+, X \in \mathbb{R}^{d \times n}} \sum_{ij} W_{ij}(\Delta_{ij} - (K_{ii} - 2K_{ij} + K_{jj}))^2 \\ &\text{s.t. } K = X^T X. \end{aligned}$$

Constraints of the form $K = X^T X$ are usually not allowed however, and are typically relaxed to [125, 110]

$$\begin{bmatrix} I & X \\ X^T & K \end{bmatrix} \succeq 0.$$

via Schur's Lemma. Furthermore, one encourages K to be approximately low-rank by introducing a nuclear norm or trace penalty $\|K\|_* = \|\sigma(K)\|_1 = tr(K)$, as a convex relaxation of a rank constraint. Intuitively, since the ℓ_1 norm promotes sparsity, the nuclear norm should promote few nonzero singular values. Elsewhere [135], it is argued that one should maximize $tr(K)$, in the spirit of the popular Maximum Variance Unfolding approach [135]. Neither minimizing nor maximizing the trace actually imposes an exact rank constraint, which is non-convex and NP-hard. One approach that could achieve exact rank constraints would be to use the Majorized Penalty Approach of Gao and Sun [60] with an alternating minimization method.

A group of methods have studied the graph realization problem, where one is asked to recover the configuration of a cloud of points given a sparse and noisy set of pairwise distances between the points [14, 12, 13, 11, 145]. One of the proposed approaches involves minimizing the following energy

$$\min_{p_1,\ldots,p_n \in \mathbb{R}^2} \sum_{(i,j)\in E} \left(\|p_i - p_j\|^2 - d_{ij}^2 \right)^2 . \tag{4.1}$$

which unfortunately is nonconvex, but admits a convex relaxation into a SDP program. We refer the reader to Section 2 of [41] for several variations of this approach, some of which have been shown to be more robust to noise in the measured distances.

4.1.3 Local Ordinal Embedding

Terada and von Luxburg [124] have recently proposed an algorithm for ordinal embedding and kNN embedding specifically, called Local Ordinal Embedding (LOE), which minimizes a soft objective function that penalizes violated ordinal constraints.

$$\min_{X \in \mathbb{R}^{d\times n}} \sum_{i<j,k<l,(i,j,k,l)\in\mathcal{C}} \max\left[0, D_{ij}(X) + \delta - D_{kl}(X)\right]^2. \tag{4.2}$$

The energy takes into account not only the number of constraints violated, but the distance by which the constraints are violated, penalizing large violations more heavily.

An advantage of this energy in contrast to ones that normalize by the variance of X (to guarantee nondegeneracy) is its relatively simple dependence on X, making the above energy easier to minimize. Instead, the scale parameter δ guarantees nondegeneracy, and fixes the scale of the embedding (which is indeterminable from ordinal data alone).

The authors introduce algorithms to minimizing the above energy, based on majorization minimization and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of Newton's method, and prove that ordinal embedding is possible when only local information is given (e.g. a k nearest neighbor graph). The algorithm recovers not only the ordinal constraints, but the density structure of the data as well. The algorithm applies to ordinal constraints associated with kNN graphs as well more general sets of ordinal constraints. We will use this crucial property when solving subproblems in the method presented here, as the corresponding subgraphs are generally not kNN graphs.

4.2 ASAP & Scale Synchronization for Ordinal Embeddings

In this section we detail the steps of the ASAP algorithm, central to the divide-and-conquer algorithm we propose for the ordinal embedding problem. Note that the difference between the original ASAP algorithm introduced in [41] and our approach lies in the decomposition method from Section 4.2.1 and the scale synchronization step from Section 4.2.2. The ASAP approach starts by decomposing the given graph G into overlapping subgraphs (referred to as *patches*), which are then embedded via the method of choice (in our case LOE). To every local patch embedding, there corresponds a scaling and an element of the Euclidean group Euc(d) of d-dimensional rigid transformations, and our goal is to estimate the scalings and group elements that will properly align all the patches in a globally consistent framework. The local optimal alignments between pairs of overlapping patches yield noisy measurements for the ratios of the above unknown group elements. Finding group elements from noisy measurements of their ratios is also known as the group synchronization problem. for which Singer [119] introduced spectral and semidefinite programming (SDP) relaxations over the group SO(2) of planar rotations, which is a building block for the ASAP algorithm [41].

Table 1 gives an overview of the steps of our approach. The inputs are an ordinal graph (we consider kNN graphs) $G = (V, E)$, where edge $ij \in E$ and non-edge $il \notin E$ indicates that $d_{ij} \leq d_{il}$, the max patch size parameter MPS, the target dimension d, and a base-case ordinal embedding method $OrdEmbed : G \mapsto X \in \mathbb{R}^{d \times n}$ for embedding each patch, such as LOE.

4.2.1 Break up the kNN graph into patches and embed

The first step we use in breaking the kNN graph into patches is to apply normalized spectral clustering [129] to a symmetrized version of the graph. Normalized spectral clustering partitions the nodes of a graph into $N \ll n$ clusters by performing k-means on the embedding given by

Algorithm 1 Modified ASAP algorithm that incorporates the scale synchronization step.

INPUT	$G = (V, E),\ \lvert V\rvert = n,\ \lvert E\rvert = m,\ MPS,\ d,\ OrdEmbed(\cdot)$
Choose Patches Embed Patches	1. Break G into N overlapping globally rigid patches $P_1, \ldots, P_N$ following the steps in Sec. 4.2.1. 2. Embed each patch P_i separately based on the ordinal constraints of the corresponding subgraph of G using $OrdEmbed(\cdot)$.
Step 1 Scale	1. If a pair of patches (P_i, P_j) have enough nodes in common, let Λ_{ij} be the median of the ratios of distances realized in the embedding of P_i and their corresponding distances in P_j as in (4.3); otherwise set $\Lambda_{ij} = 0$. 2. Compute the eigenvector v_1^{Λ} corresponding to the largest eigenvalue of the sparse matrix Λ. 3. Scale each patch P_i by $v_1^{\Lambda}(i)$, for $i = 1, \ldots, n$
Step 2 Rotate & Reflect	1. Align all pairs of patches (P_i, P_j) that have enough nodes in common. 2. Estimate their relative rotation and possibly reflection $H_{ij} \in O(d) \subset \mathbb{R}^{d \times d}$. 3. Build a sparse $dN \times dN$ symmetric matrix $H = (H_{ij})$ where entry ij is itself a matrix in $O(d)$. 4. Define $\mathcal{H} = D^{-1}H$, where D is a diagonal matrix with $D_{1+d(i-1),1+d(i-1)} = \ldots = D_{di,di} = deg(i)$, $i = 1, \ldots, N$, where $deg(i)$ is the node degree of patch P_i. 5. Compute the top d eigenvectors $v_i^{\mathcal{H}}$ of $\mathcal{H}$ satisfying $\mathcal{H}v_i^{\mathcal{H}} = \lambda_i^{\mathcal{H}} v_i^{\mathcal{H}}, i = 1, \ldots, d$. 6. Estimate the global reflection and rotation of patch P_i by the orthogonal matrix $\hat{h}_i$ that is closest to $\widetilde{H}_i$ in Frobenius norm, where $\widetilde{H}_i$ is the submatrix corresponding to the i$^{\text{th}}$ patch in the $dN \times d$ matrix formed by the top d eigenvectors $[v_1^{\mathcal{H}} \ldots v_d^{\mathcal{H}}]$. 7. Update the embedding of patch P_i by applying the above orthogonal transformation $\hat{h}_i$
Step 3 Translate	Solve $m \times n$ overdetermined system of linear equations (4.5) for optimal translation in each dimension.
OUTPUT	Estimated coordinates $\hat{x}_1, \ldots, \hat{x}_n$

the top N eigenvectors of the random-walk normalized graph Laplacian. It is shown [129] that normalized spectral clustering minimizes a relaxation of the normalized graph cut problem. Next, we enlarge the clusters by adding the graph-neighbors of each node, so that the resulting patches have significant overlap, a prerequisite for the ASAP synchronization algorithm. The higher the overlap between the patches, the more robust the pairwise group ratio estimates would be, thus leading overall to a more accurate final global solution. Finally, we use an iterative procedure to remove nodes from the graph relying on tools from rigidity theory. [1] If a patch is not globally rigid, we drop a constant fraction of the added nodes. At each round we choose to drop a quarter of the nodes with the lowest degree while retaining all nodes that were in the original cluster generated by k-means in the corresponding patch. This uses the heuristic that low-degree nodes tend to render a graph not globally rigid. After dropping nodes, we check the remaining patch for globally rigidity again. We stop the pruning process when the patch contains fewer than $4/3$ the number of nodes in the original cluster, or the patch is globally rigid.

We refer the readers to Appendix B for for a brief description of global rigidity, and relevant results in the literature, and use the remainder of this section as a brief discussion of the main definitions. In the *graph realization problem* (GRP), one is given a graph $G = (V, E)$ together with a non-negative distance measurement d_{ij} associated with each edge, and is asked to compute a realization of G in $\mathbb{R}^d$. In other words, for any pair of adjacent nodes i and j, the distance $d_{ij} = d_{ji}$ is available, and the goal is to find a d-dimensional embedding $p_1, p_2, \ldots, p_n \in \mathbb{R}^d$ such that $\|p_i - p_j\| = d_{ij},$ for all $(i, j) \in E$. The main difference between the GRP and the problem we aim to address in this chapter is the input information available to the user. Unlike the GRP problem where distances are available to the user, here we only have information of the adjacency matrix of the graph and have the knowledge that it represents a kNN graph. Both problems aim to recover an embedding of the initial configuration of points.

A graph is globally rigid in $\mathbb{R}^d$ if there is a unique (up to the trivial Euclidean isometries) embedding of the graph $\mathbb{R}^d$ such that all distance constraints are preserved. It is well known that a necessary condition for global rigidity is 3-connectivity of the graph. Since the problem at hand

[1] A graph is globally rigid if all realizations of it are congruent up to a rigid transformation.

that we are trying to solve is harder (as we do not have distance information available) we require that the patches we generate are globally rigid graphs. Even in the favorable scenario when we do have available distance measurements (which we do not in the present problem, but only ordinal information), any algorithm seeking an embedding of the graph would fail if the graph were to have multiple non-congruent realizations.

4.2.2 Scale Synchronization

Before applying the original ASAP algorithm to the embedded patches, we introduce an additional step that further improves our approach and is independent of the dimension d. In the *graph realization problem* which motivated ASAP, one is given a graph $G = (V, E)$ and non-negative distance measurement d_{ij} associated with each edge $ij \in E(G)$, and is asked to compute a realization of G in $\mathbb{R}^d$. The distances are readily available to the user and thus the local embedding of each patch is on the same scale as the ground truth. However, in the kNN embedding problem, distances are unknown and the scale of one patch relative to another must be approximated. Any ordinal embedding approach has no way of relating the scaling of the local patch to the global scale. To this end, we augment the ASAP algorithm with a step where we synchronize local scaling information to recover an estimate for the global scaling of each patch, thus overall synchronizing over the group of similarity transformations.

We accomplish this as follows. Given a set of patches, $\{P_i\}_{i=1}^N$, we create a patch graph in which two patches are connected if and only if they have at least $d+1$ nodes in common. We then construct a matrix $\Lambda \in \mathbb{R}^{N\times N}$ as

$$\Lambda_{ij} = \begin{cases} \text{median}\left\{\frac{D_{a,b}^{P_i}}{D_{a,b}^{P_j}}\right\}_{a\neq b\in P_i\cap P_j} & \text{if } P_i \sim P_j,\, i \leq j, \\ 1/\Lambda_{ji} & \text{if } P_i \sim P_j,\, i > j, \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

where $D_{a,b}^{P_i}$ is the distance between nodes a and b as realized in the embedded patch P_i. The matrix Λ approximates the relative scales between patches. If all distances in all patches were recovered correctly up to scale, and all patches had sufficient overlap with each other, then each row of Λ would be a scalar multiple of the others and each column of Λ would be scalar multiple of the others, thus rendering Λ a rank-1 matrix. For the noisy case, in order to get a consistent estimate of global scaling, we compute the best rank-1 approximation of Λ, given by its leading eigenvector $v_1^{(\Lambda)}$. We use this approximation of global scaling to rescale the embedded patches before running ASAP. Note that the connectivity of the patch graph and the non-negativity of Λ guarantee, via the Perron-Frobenius Theorem, that all entries of $v_1^{(\Lambda)}$ are positive. We refer the reader to Figure 4.2, which illustrates on an actual example the importance of this scaling synchronization step.

4.2.3 Optimal Rotation, Reflection and Translation

After applying the optimal scaling to each patch embedding, we use the original ASAP algorithm to integrate all patches in a global framework, as illustrated in the pipeline in Figure 4.1. We estimate, for each patch P_i, an element of the Euclidean group Euc(d) $=$ O(d) $\times \mathbb{R}^d$ which, when applied to that patch embedding P_i, aligns all patches as best as possible in a single coordinate system. In doing so, we start by estimating, for each pair of overlapping patches P_i and P_j, their optimal relative rotation and reflection, i.e., an element H_{ij} of the orthogonal group O(d) that best aligns P_j with P_i. Whenever the patch embeddings perfectly match the ground truth, $H_{ij} = O_i O_j^{-1}$. We refer the reader to [41] for several methods on aligning pairs of patches and computing their relative reflections and rotations $H_{i,j}$. Finding group elements $\{O_i\}_{i=1}^N$ from noisy measurements H_{ij} of their ratios is also known as the group synchronization problem. Since this problem is NP-hard, we rely on the spectral relaxation [119] of

$$\min_{O_1,\dots,O_N \in O(d)} \sum_{P_i \sim P_j} \|O_i O_j^{-1} - H_{ij}\|_F^2. \tag{4.4}$$

for synchronization over O(2), and estimate a consistent global rotation of each patch from the top d eigenvectors of the graph Connection Laplacian, as in Step 2.4 in Table 1. We estimate the optimal translation of each patch by solving, in a least squares sense, d overdetermined linear

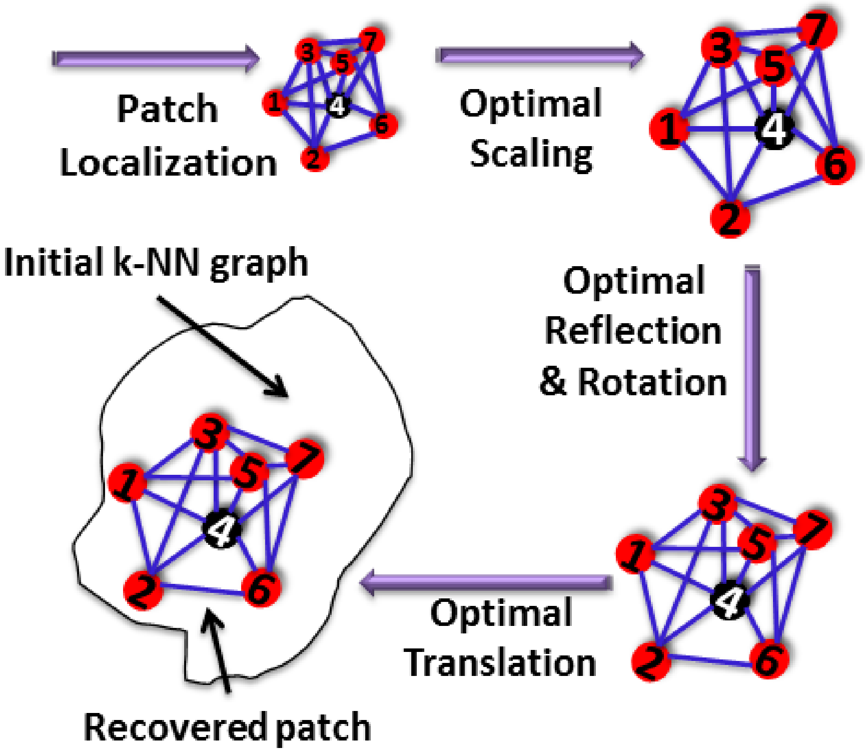


Figure 4.1: ASAP and scale synchronization pipeline.

systems

$$x_i - x_j = x_i^{(k)} - x_j^{(k)}, \quad (i,j) \in E_k, \quad k = 1, \ldots, N, \tag{4.5}$$

where x_i, respectively $x_i^{(k)}$, denote the unknown location of node i we are solving for, respectively, the known location of node i in the embedding of patch P_k. We refer the reader to [41] for a description of computing the optimal translations.

4.2.4 Extension to higher dimensions

Although we present experiments here on 2D and 3D data, the ASAP approach extends naturally to higher dimensions. In the 3D case, ASAP has been recently used as a scalable robust approach to the molecule problem in structural biology [42]. For the d-dimensional general case, one can extend ASAP by first using the same approach for scaling synchronization from Section 4.2.2, then synchronizing over O(d), and finally estimating the optimal translations over R^d by solving d overdetermined systems of linear equations via least-squares. The LOE approach that can be used to obtain the local patch embeddings required by ASAP, has a natural extension to the d-dimensional case, thus rendering the entire pipeline amenable to dealing with higher-dimensional data.

4.2.5 Complexity Analysis

Here we briefly summarize the time complexity of each of the steps of the proposed algorithm. Several of these estimates come directly from the analogous step presented in the original ASAP work [41].

Choosing patches (detailed in Section 4.2.1) consists of spectral clustering, enlarging clusters, and pruning. Spectral clustering in general has complexity $\mathcal{O}(n^3)$ where n is the number of nodes, though faster variants exist : [141] achieves $\mathcal{O}(N^3) + \mathcal{O}(Nnt_{\text{k-means}})$, where N is the number of clusters (patches in this case), $t_{\text{k-means}}$ is the number of iterations of k-means used. Expanding the patches in the kNN graph takes $\mathcal{O}(kn)$ time. Iteratively pruning patch P_i takes $\mathcal{O}(|P_i|)$ time. Pruning all patches collectively takes certainly no more than $\mathcal{O}(Nn)$ time. Thus the patch selection process takes $\mathcal{O}((n^2 + k + N)n)$ if standard spectral clustering methods are used, or potentially as little as $\mathcal{O}((k + Nt_{\text{k-means}})n) + \mathcal{O}(N^3)$ if more efficient approximate spectral clustering methods are used.

Embedding the patches depends upon the size and number of patches, and the complexity of the embedding algorithm used. In the case of a constant number of iterations of LOE, embedding a patch P_i takes $\mathcal{O}(k|P_i|^2)$. If one assumes the patches evenly distribute the nodes, this is $\mathcal{O}(k(n/N)^2)$ per patch and $\mathcal{O}(kn^2/N)$ in total. A very crude upper bound for the general behavior would be $\mathcal{O}(kn^2N)$.

Computing all patch intersections takes $\mathcal{O}(N^2 \max |P_i|)$ (where $\max |P_i|$ is the maximum patch size achieved). This could be crudely bounded by $\mathcal{O}(nN^2)$.

Computing the scale matrix Λ is similar to the above with an additional median computation, so we get a crude upper bound of $\mathcal{O}(nN^2 \log(n))$.

Computing the leading eigenvector of Λ take $\mathcal{O}(Mt_{\text{power}})$ where M patches have substantial intersection ($M < N$ choose 2), and t_{power} is the number of iterations of the power method.

The remaining steps follow quite similarly to the standard ASAP case presented in [41]. Using those results, we find that the overall time complexity of the ASAP LOE method (assuming efficient spectral clustering) is $\mathcal{O}(n^2 \text{poly}(k, N, t_{\text{k-means}}, t_{\text{power}}) + m\sqrt{\kappa})$ where κ is a condition number related to the linear system solved in the translation step. The highest order complexity step (with respect to n) is embedding each patch via LOE.

4.3 Density Estimation

In this section, we remark on the explicit connection between the graph embedding problem considered in this chapter and the density estimation problem. In particular, one may approach the problem of recovering the unknown coordinates underlying the kNN graph by first aiming to estimate the density function that generates the coordinates. Suppose for example that one is able to estimate the pointwise density $u : \Omega \subseteq \mathbb{R}^d \to [0, 1]$, up to some constant multiple, evaluated at each vertex of the graph, x_i. Next, as outlined in [130], one can assign weights to the originally unweighted kNN graph, defined by $w(x_i, x_j) = (u^{-1/d}(x_i) + u^{-1/d}(x_j))/2$. Furthermore, it can be shown that the shortest path distance in the resulting weighted kNN graphs converges to the Euclidean distance of the original points as the number of points increases. In other words, applying multidimensional scaling to the shortest path distances on the weighted kNN graph will yield increasingly accurate embeddings of the original points $\{x_i\}_{i=1}^n$ as $n \to +\infty$.

In contrast to finding an approximate embedding from a density estimate, under certain conditions, the reverse process is also straightforward. With sufficiently many points and sufficiently strong priors on the distribution, the methodology of Maximum Penalized Likelihood Estimation (MPLE) applies [51]. One first assumes that the locations correspond to points drawn independently identically distributed according to some unknown underlying spatial distribution. MPLE approximates the most likely spatial distribution given the points observed and some assumed prior distribution on the space of distributions. The data fidelity term comes in the form of a log-likelihood term, a function of the distribution estimate and the point locations, and is given by

$$L(u, \{x_i\}_{i=1}^n) = \sum_{i=1}^n \log(u(x_i)),$$

and the penalty term, $P(u)$ enforces the prior distribution on the space of distributions. Typical choices for $P(u)$ include the H^1-seminorm regularizer, $P(u) = \frac{\lambda}{2}\int_\Omega |\nabla u|^2 dx$, enforcing smoothness, and Total Variation (TV) norm regularization, $P(u) = \lambda \int_\Omega |\nabla u| dx$, which enforces smoothness, but also allows for edges. Therefore, general MPLE seeks to optimize the following energy over all probability distributions on the spatial domain $\Omega \subseteq \mathbb{R}^d$

$$\hat{u} = \underset{u \geq 0, \int_\Omega u dx = 1}{\arg\max} \; L(u, \{x_i\}_{i=1}^n) - P(u).$$

The form and scale of P encodes different types and amounts of regularity in the resulting density estimate u. In practical settings, cross-validation should be performed to determine the appropriate amount of regularity to impose on a given data set.

For the purpose of using kNN graphs to recover densities, we will include a post-processing step for a subset of the embedding experiments, to which we apply a standard implementation of TV MPLE [97] to the embedded points. TV is a good choice of penalty because we will be applying it to points that are drawn from a piecewise constant density. The good density estimates based on good embeddings shown in Section 4.4 illustrate that there is in fact a strong connection between the embedding and density estimation problems.

The actual implementation of the TV MPLE relies on the Split Bregman (equivalently Alternating Direction Method of Multipliers) minimization technique in which one introduces a splitting and equality constraints that are enforced by performing saddle-point optimization of the augmented Lagrangian. This results in an iterative update procedure given by Algorithm 2. The first minimization step is actually replaced by minimizing over u, and d individually, making use of the shrinkage proximal operator associated with the ℓ^1 norm.

Algorithm 2 TV MPLE

```
INPUT : $\{x_i\}, \rho, \gamma$
$y = 0, z = 0$
For $numberIterations$ {
    $\left(\hat{u}, \hat{d}\right) =$
    $\underset{u \geq 0, d}{\arg\min} \left\{ \|d\|_1 - \sum_{i=1}^{n} \log(u(x_i)) + \frac{\rho}{2} \|\nabla u - d + y\|_2^2 + \frac{\gamma}{2} \left( \|u\|_1 - 1 + z \right)^2 \right\}$
    $y = y + \tilde{\nabla} \hat{u} - \hat{d}$
    $z = z + \|\hat{u}\|_1 - 1$
}
```

4.4 Experiments

Our experiments compare embeddings of points drawn from three different 2D synthetic densities: piecewise constant half-planes (**PC**), piecewise constant squares (**PCS**), and Gaussian (**Gauss**), and a 3D synthetic density : piecewise constant half-cubes (**halfcube**), each with $n = \{500, 1000, 5000\}$ points, as well as points drawn uniformly from a 3D donut shape (**Donut**) with $n = 500$, and the actual 2D coordinates of $n = 1101$ cities in the US (**US cities**). For a given set of data points, we use its kNN adjacency matrix as input to each ordinal embedding method. Separate from these datasets with a clear correct geometric embeddings, we find embeddings of points in a co-authorship network of network scientists (**NetSci2010**) with $n = 552$ (see Section 4.4.6). We test Laplacian Eigenmaps [7], the LOE BFGS and LOE MM methods [124], and ASAP with LOE BFGS used for the patch embeddings. As LOE was already compared with several methods in [124], attaining better performance than LOE may suggest better performance than a number of relevant methods including Kamada and Kawai [79], and Fruchterman and Reingold [57]. We remark that our approach deals with a different input than that of the **t-SNE** method in [128], which is generally used for embeddings of high dimensional data where some of the constraints are deliberately violated, which is not necessarily the case in our setting. We evaluate the methods based on (wall-clock) runtime and two different error metrics, Procrustes alignment error[117], and *A-error* ($\mathcal{E}_A$) defined as the percentage of edge disagreements between the kNN adjacency matrix of the proposed embedding $\tilde{X}$ and the ground truth matrix

$$\text{error}(\tilde{X}, X): \quad \mathcal{E}_A \overset{\text{def}}{=} \frac{1}{n^2} \sum_{i,j=1}^{n} \left| \left(A^k_{\tilde{X}}\right)_{ij} - \left(A^k_X\right)_{ij} \right|, \tag{4.6}$$

where $A^k_X \in \{0,1\}^{n \times n}$ denotes the adjacency matrix of the corresponding kNN graph. We set varying limits on the number of LOE iterations $\{5, 10, 50, 100, 300, 500\}$, and we use varying maximum patch sizes (MPS) for ASAP . The LOE and ASAP methods give, for each distribution and values n and k, an error-runtime Pareto curve (with low values in both coordinates being best). In Table 4.1, we establish some shorthand notation for the methods and parameters used in this section. For fair comparisons, we pass the same randomly sampled data to each of the methods. Ideally, one would run these experiments many times over and average the results (to get an estimate of average performance), but this is effect already partially accomplished by running the LOE and ASAP methods with multiple parameters to get a more holistic measurement of performance. It is worth mentioning that while LOE BFGS and LOE MM are iterative methods which should converge to the best estimate of the solution as the number of iterations increases, ASAP is not iterative and the results of ASAP LOE with a given MPS, do not inform the results of ASAP LOE with another MPS. This aspect, combined with the randomized k-means spectral clustering used to choose patches means that we do not generally expect the recovery errors of ASAP LOE to be monotonically decreasing with MPS or time (as higher MPS generally leads to longer computational time). A principled way of choosing the best MPS for a given application of ASAP LOE could be of further interest.

4.4.1 The need for scale synchronization

First, to illustrate the importance of the scale synchronization introduced in Section 4.2.2, we compare in Figure 4.2 ASAP synchronized embeddings with and without this step. Clearly, this step significantly improves the recovered solutions.

Recovery Method	
LE	Laplacian Eigenmaps embedding
LOE MM	Local Ordinal Embedding using majorization minimization
LOE BFGS	Local Ordinal Embedding using BFGS
ASAP LOE	ASAP & LOE BFGS patch embeddings
Parameters	
sparse k	$k = \lceil 2\log(n) \rceil$
dense k	$k = \lceil \sqrt{n \log(n)} \rceil$
MPS	maximum patch size (for ASAP)
Iter.	number of iterations (of LOE)
Data sets	
PC	2D piecewise constant half-planes
PCS	2D piecewise constant squares
Gauss	2D Gaussian
halfcube	3D piecewise constant half-cubes
Donut	3D Donut
US cities	2D coordinates of US cities
NetSci2010	co-authorship network of scientists

Table 4.1: Notation for plotting experimental results.

4.4.2 Simulations with $n = 500, 1000, 5000$ with sparse and dense k

We show $\mathcal{E}_A$ versus runtime for recovering $n = \{500, 1000, 5000\}$ points sampled from the PC (Figure 4.3), PCS (Figure 4.4), and Gaussian (Figure 4.5), with each figure showing results in the sparse and dense k regime (see Table 4.1). We also show $\mathcal{E}_A$ versus runtime for $n = \{500, 1000, 5000\}$ points drawn from the halfcube (Figure 4.6) distribution for $k = 50, 150, 250, 450$. Even for lower values of n, we find that ASAP LOE is often either faster than or better-performing than LOE BFGS, or both. This seems to be especially true in the sparse k domain. This is partly due to the massively parallel embedding step in ASAP, which can take advantage of multiple cores as the problem scales. One would expect that as n continues to grow, if more processors are made available and memory increases sufficiently, the advantage of embedding parallelization would continue to increase.

To further illustrate how the methods perform, we plot the embeddings of $n = 1000$ point sampled from the 2D densities in Figure 4.7. In each case, the ASAP LOE with MPS=400 takes less time to run and yields smaller $\mathcal{E}_A$ errors than the LOE BFGS with 100 maximum iterations. We only run LOE MM for $n = 500$ because of difficulties we had when trying to get the provided

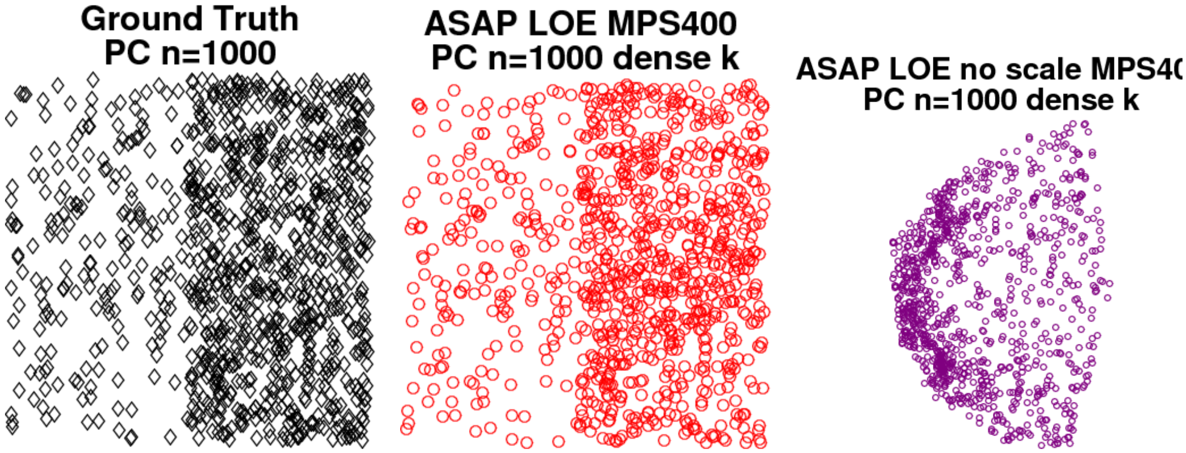


Figure 4.2: Left: Ground truth, $n = 1000, k = 14$. Middle: ASAP LOE with scale synchronization: $\mathcal{E}_A = 0.007$. Right: ASAP LOE without scale synchronization: $\mathcal{E}_A = 0.038$.

R implementation to run on our Linux-based remote computing resource. We ran into no problems with the LOE BFGS implementation. The computers used have 12 CPU cores which are Intel(R) Xeon(R) X5650 @ 2.67GHz, and have 48GB ram. The R implementation of LOE does not (as far as its authors are aware) take advantage of multiple cores, and runs a single process on a single core. In contrast, our ASAP Matlab implementation uses the Multicore package to divide up the local embedding problems among the available cores.

To demonstrate that this approach is not limited to the 2D case, nor does it only perform well on synthetic data, we plot in Figure 4.9 the embeddings Procrustes aligned with points sampled from a 3D donut shape, and actual coordinates of $n = 1101$ US cities. In both cases, ASAP LOE with MPS=300 runs faster and yields smaller $\mathcal{E}_A$ than LOE BFGS with 500 maximum iterations, the latter of which produces twisted or folded results.

4.4.3 Large n : 50,000

In Table 4.2 we show $\mathcal{E}_A$ vs runtime for ASAP LOE on a data set of $n = 50,000$ points and $k = 22$. While this size is completely prohibitive for LOE BFGS, ASAP LOE produces good results in less than 4 hours. The worst possible result would be all edges of original graph misplaced, meaning $\mathcal{E}_A = 2 \cdot 50k \cdot 22/(50k)^2 = 8.8 \times 10^{-4}$. $\mathcal{E}_A = 2 \times 10^{-4}$ means we get approximately $3/4$ of the

MPS	100	300	500
PCS $\mathcal{E}_A$	5.1×10^{-4}	5.6×10^{-4}	1.9×10^{-4}
PC $\mathcal{E}_A$	5.8×10^{-4}	4.7×10^{-4}	3.0×10^{-4}

Table 4.2: Recovery results for $n = 50,000$ for ASAP LOE.

edges correct.

4.4.4 Increasing k

We show in Figure 4.10, scaled $(n/k) \times \text{NumberNonzero}(A - A_0)$ (this is a rescaling of $\mathcal{E}_A$ proportional to number of misplaced edges, more comparable for different values of k) and procrustes error versus increasing values of k for $n = \{5000\}$ points drawn from the piecewise constant half-planes distribution using the method ASAP LOE with MPS=300. We see that for large n, adjacency matrix error and Procrustes error remain relatively small and stable over a range of small increasing k. Additionally, we show in Figure 4.15 some of the embeddings corresponding to these results. Like the Procrustes error plot, these embeddings suggest that for a range of k small relative to n and not too large relative to MPS, ASAP LOE BFGS returns sensible, although not perfect results. As k gets too large however the results are quite poor. We suspect this is a result of k being too large relative to MPS, leading to patches which are overly dense. When an ordinal graph contains nearly all possible edges, it essentially provides no information. When such data is of specific interest, one could either increase the mps as computational resources and time allow, or potentially use an alternate method for breaking the graph into overlapping patches which are not too dense.

4.4.5 Density Estimation Experiments

In Figure 4.11 we show the results of applying TV MPLE to some of the embeddings shown in Figure 4.7. The regularization parameter used is .0001 . This is not obtained by cross-validation, but it simply seems to perform well on the originally sampled points. The densities of the approximate embeddings are as expected, with ASAP LOE BFGS recovering the density best, with LOE

BFGS behind, and LE doing the worst. This altogether suggests that better embedding results do lead to better density estimation, if that is the end goal.

4.4.6 Network of network scientists embedding

To further illustrate potential of ordinal embedding to broad categories of data, we present here an experiment embedding data that does not have an apparent ground-truth geometry. We use data from a co-authorship network of network scientists [50] from 2010, which was studied in [104] to evaluate methods of computing core-periphery structure. The network contains nonegatively weighted undirected edges where the weights are based on the number of papers they have co-authored. The network has 552 nodes and 1318 edges, with the number of edges attached to each node ranging from 1 to 38. The mean number of edges attached to each node is 4.7754 and the median is 4.

To embed the data, we treat the co-authorship links as nearest neighbor relationships. In other words, if X and Y have authored papers together, but X has not authored any papers with Z, we impose that the distance between X and Y should be smaller than the distance between X and Z. We used LOE BFGS and ASAP LOE BFGS to perform these embeddings in 2D and 3D. In this case, the LOE results were ultimately best with the 2D LOE BFGS 500 iteration embedding misplacing 754 of the 1318 nearest neighbor edges and the 3D LOE BFGS 500 Iteration embedding misplacing 272 of the nearest neighbor edges, while the 2D ASAP LOE BFGS mps 500 misplaced 910 edges, and the 3D ASAP LOE BFGS mps 500 misplaced 467 edges. That being said, several of the runtimes for the ASAP LOE results beat the LOE results. We speculate that the reasons LOE outperforms ASAP LOE in accuracy in this case are twofold : 1) the number of nodes, $n = 552$, is too small to make the LOE method applied to the full data sufficiently intensive, and 2) the wide distribution of degrees of the nodes in the network perhaps does not go well with our approach of breaking up the network via spectral clustering. Perhaps other methods for braking up the network should be considered when the degree distribution is highly varied.

Independent of the comparison of the two methods, we look at the best 2D and 3D embeddings from LOE (shown in Figure 4.13), to see if the embeddings preserve any interesting structure in the network. Since the network was previously studied for core-periphery detection, we color the nodes based on the corescore computed by the method proposed in [104] (mapping low values to blue and high values to red), and label the names of the authors with the top 10 corescores. These red, core authors appear primarily central to the embeddings, suggesting that these embeddings preserve important structural properties in the original network.

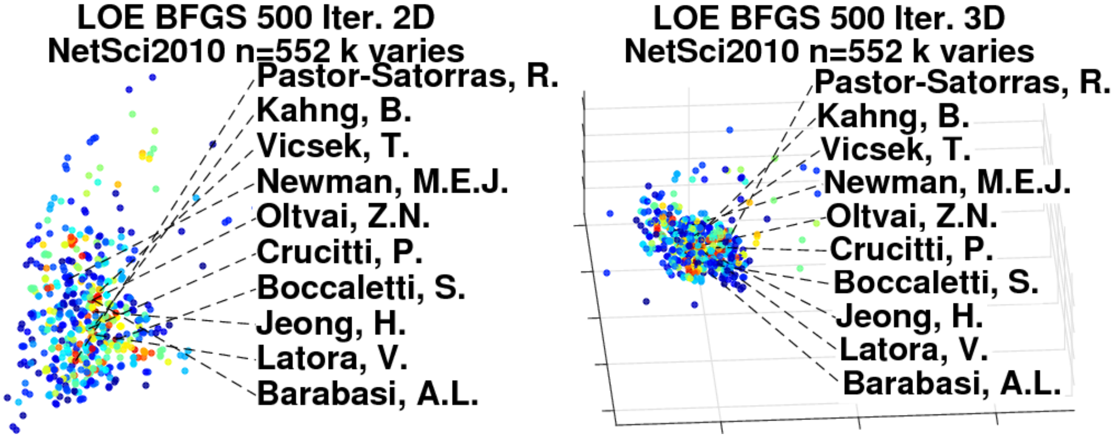


Figure 4.13: LOE BFGS 2d and 3d embeddings of data from NetSci2010 data set, $n = 552$, where co-authorship imposes that authors should be close

4.5 A Linear Program Alternative to SDP embedding

In this section we present the algorithm and a few results for a Linear Program Embedding approach using metric MDS (LPEm) for ordinal embedding. Though the results are ultimately not competitive with Local Ordinal Embedding, the approach is different enough so that the ideas may be of independent interest. In contrast to the SDP methods which cast embedding problems in terms of the Gram matrix K our LPEm approach for kNN embedding optimizes over the variables D (the distance matrix), R (the radius at each node), and the slack variables. The radius at each node i, denoted by R_i is defined to be the distance between node i and its k-th closest neighbor. Thus R_i is the radius of the neighborhood at node i. In kNN embedding, the objective and constraints can be written as linear constraints in D, R and the slack variables, altogether leading to a linear program which is computationally cheaper to solve than an SDP. Although SDP-based methods can encompass a larger class of problems, they currently do not approach the scalability or numerical maturity of LP and SOCP solvers.

After the LP returns a candidate distance matrix D and radii R, we pass D into a standard mdscale, here using metric multidimensional scaling (see Algorithm 3), where by $\mathcal{T}$ we mean the set of triangle inequalities we considered (ordered set (i, j, k)). If $(i, j, k) \in \mathcal{T}$, the same holds true for the two other permutations. The full set of triangle inequalities are necessary, though not sufficient, for the matrix D to correspond to an Euclidean distance matrix. If one omits slack variables, there are $n(n-1)/2$ distance values to solve for along with n radii, and thus $n(n+1)/2$ unknowns in total. Considering the ordinal constraints, for the upper bounds on the entries D_{ij}, there are n ways to choose i, and for each i there are k ways to choose j, thus $nk/2$ constraints (accounting for symmetric distances). For the lower bounds on the entries D_{ij} there are n ways to choose i and for each i there are $n-k-1$ ways to choose j, giving $n(n-k-1)/2$ constraints. So there are $n(n-1)/2$ ordinal constraints on relating the $n(n-1)/2$ distances and n radii. In other words, the intuition behind the added triangle inequalities is that they help to better constrain the system. There are on the order of n^3 triangle inequalities (choose any three points), so for large n, there are many more constraints than unknowns.

Algorithm 3 LP approach

$$
\begin{array}{rl}
(D^*, R^*) = \underset{\alpha,\beta,R,D}{\arg\min} & \displaystyle\sum_{ij \in E(G)} \alpha_{ij} + \sum_{ij \notin E(G)} \beta_{ij} \\
\text{subject to} & \alpha, \beta \in \mathbb{R}_{+}^{n\times n}, R \in \mathbb{R}_{+}^{n}, D \in \mathbb{R}_{+,sym}^{n\times n} \\
 & D_{ij} \leq R_i + \alpha_{ij}, \text{ if } ij \in E(G) \\
 & D_{ij} > R_i - \beta_{ij}, \text{ if } ij \notin E(G) \\
 & \displaystyle\sum_{i=1}^{n} R_i = V \\
 & D_{ij} + D_{ik} \leq D_{kl}, (i, j, k) \in \mathcal{T} \\
X = \text{mds}\,(D^*, d) &
\end{array}
$$

To avoid the added complexity from imposing all triangle inequalities, one could consider models that impose only a fraction of such constraints via either imposing them locally, for k-hop neighboring triples of points, or globally, such as picking edges via an Erdős-Rényi model, or mixing the two approaches.

We remark that dropping triangle inequalities altogether could certainly speed up the embedding process. The resulting non-metric D may correspond to an increasing function of distance (e.g., distance squared), which suggests that non-metric MDS would be appropriate.

In general, even if the recovered distance metric corresponds to a metric distance, this is not a guarantee that the distance is realizable in a low-dimensional space. That requires a rank constraint on D, which is non-convex and is computationally intractable for an LP or SDP. The ultimate embedding into a low-dimensional space thus potentially gives up some structure in both the LP and SDP formulation, and it can be argued that this effect is lessened via the local to global approach.

In Figure 4.14 we show an example with points drawn from the densities discussed in the previous section along with points embedded using the LPEm approach. In these experiments we use a very dense value of k, $k = n/2 = 50$, which is where the approach seemed to work the best. The recovery of the piecewise constant half-planes is the best, but the preliminary results led us to decide not to experiment with this method further for the time being. The method was implemented using the CVX library, a package for specifying and solving convex programs ([71, 70]). Overall, we find the LP formulation appealing due to its simplicity. It would be interesting if a similarly simple approach could obtain competitive results on the problem of ordinal embedding, especially since until the work of von Luxburg and Alamgir [130], it was unknown to the community whether the problem was practically solvable at all.

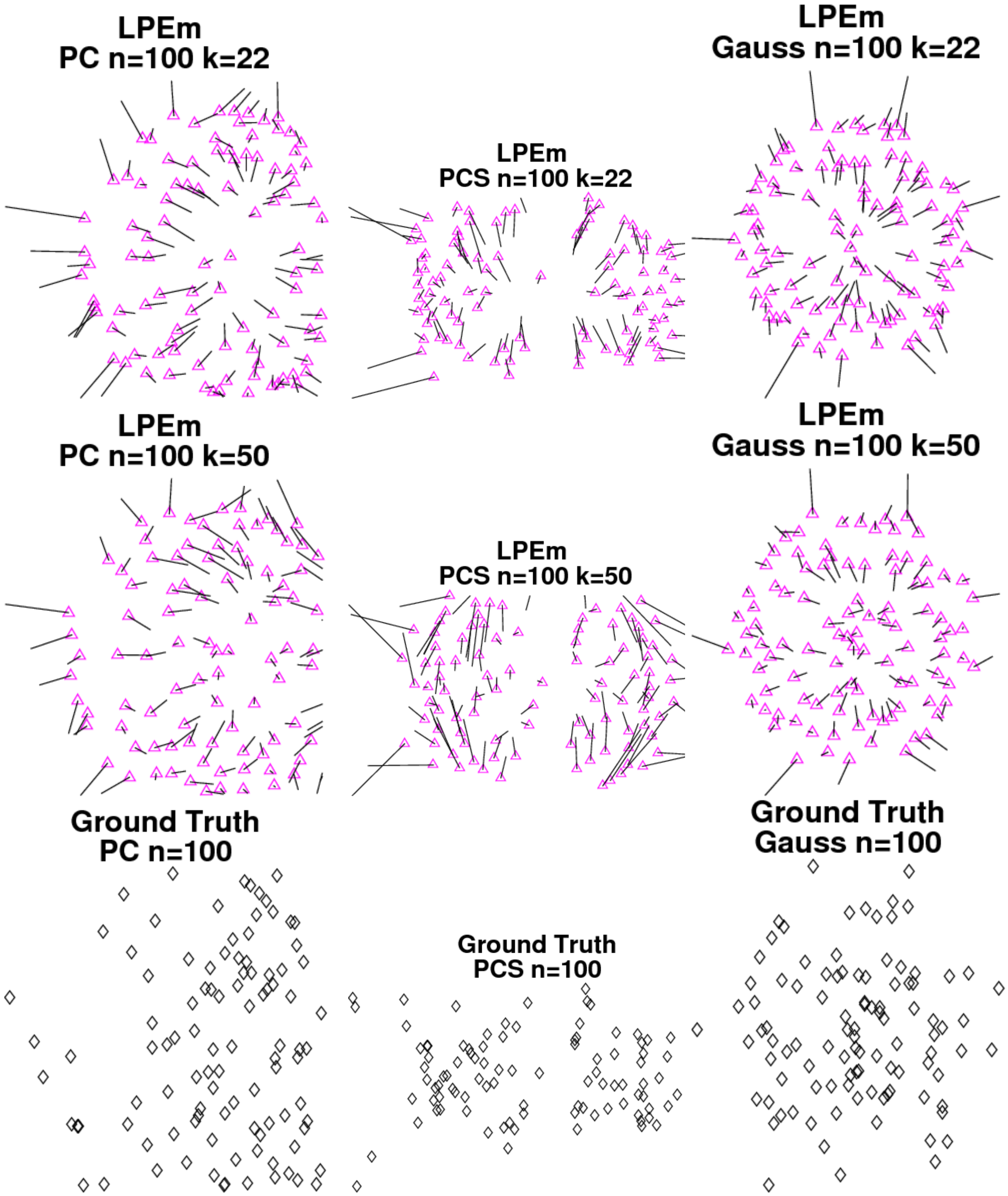


Figure 4.14: Linear Program Embeddings for the PC (left), PCS (middle), and Gauss (right) data sets with $n = 100$, Row 1 : $k = 22$ Row 2: $k = 50$ Row 3: ground truth. Line segments highlight the displacement of each point.

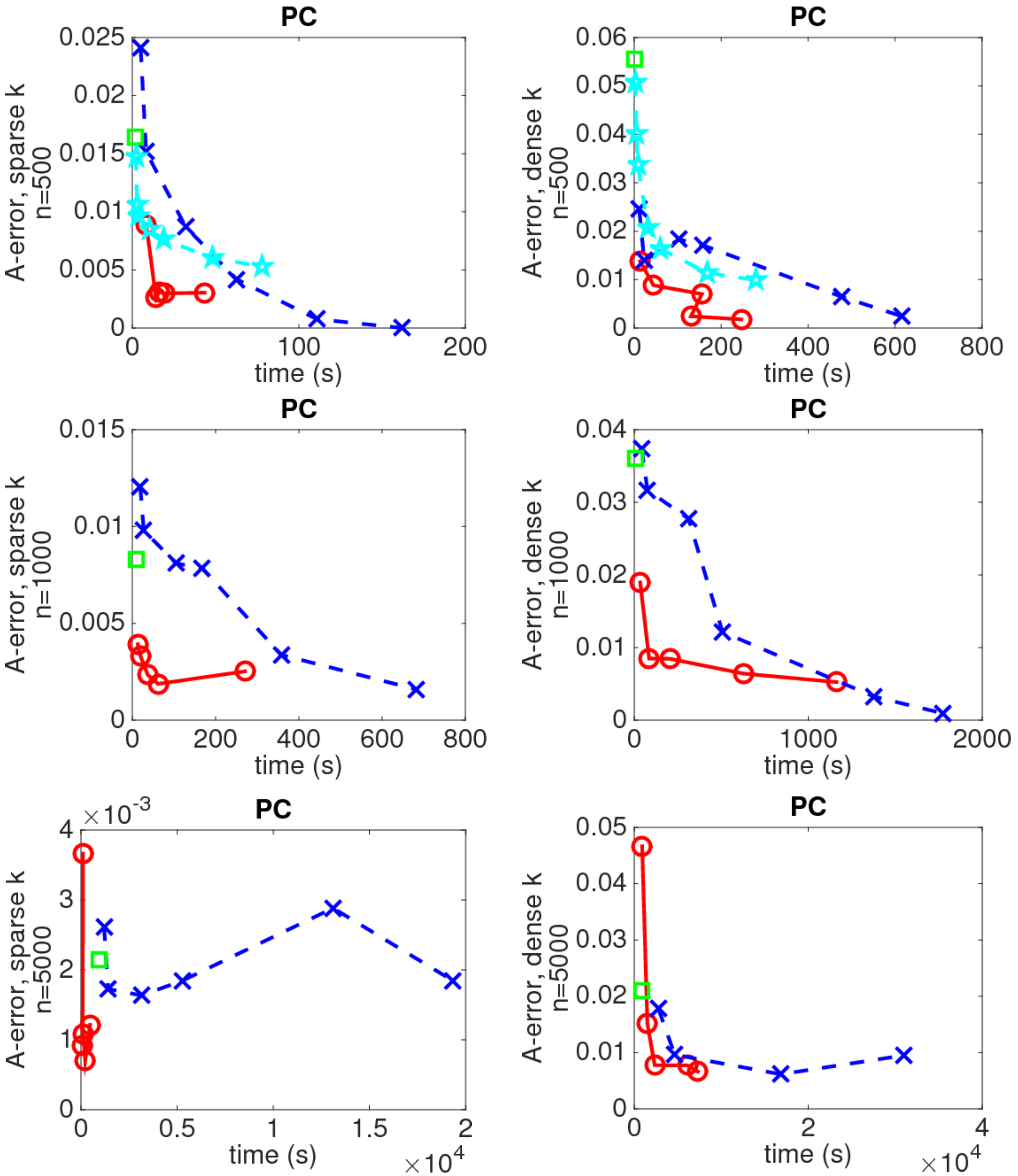


Figure 4.3: $\mathcal{E}_A$ vs. time, $n = \{500, 1000, 5000\}$, Left : k sparse, Right : k dense, piecewise constant half-planes, ○ ASAP LOE, × LOE BFGS, □ LE , ★ LOE MM

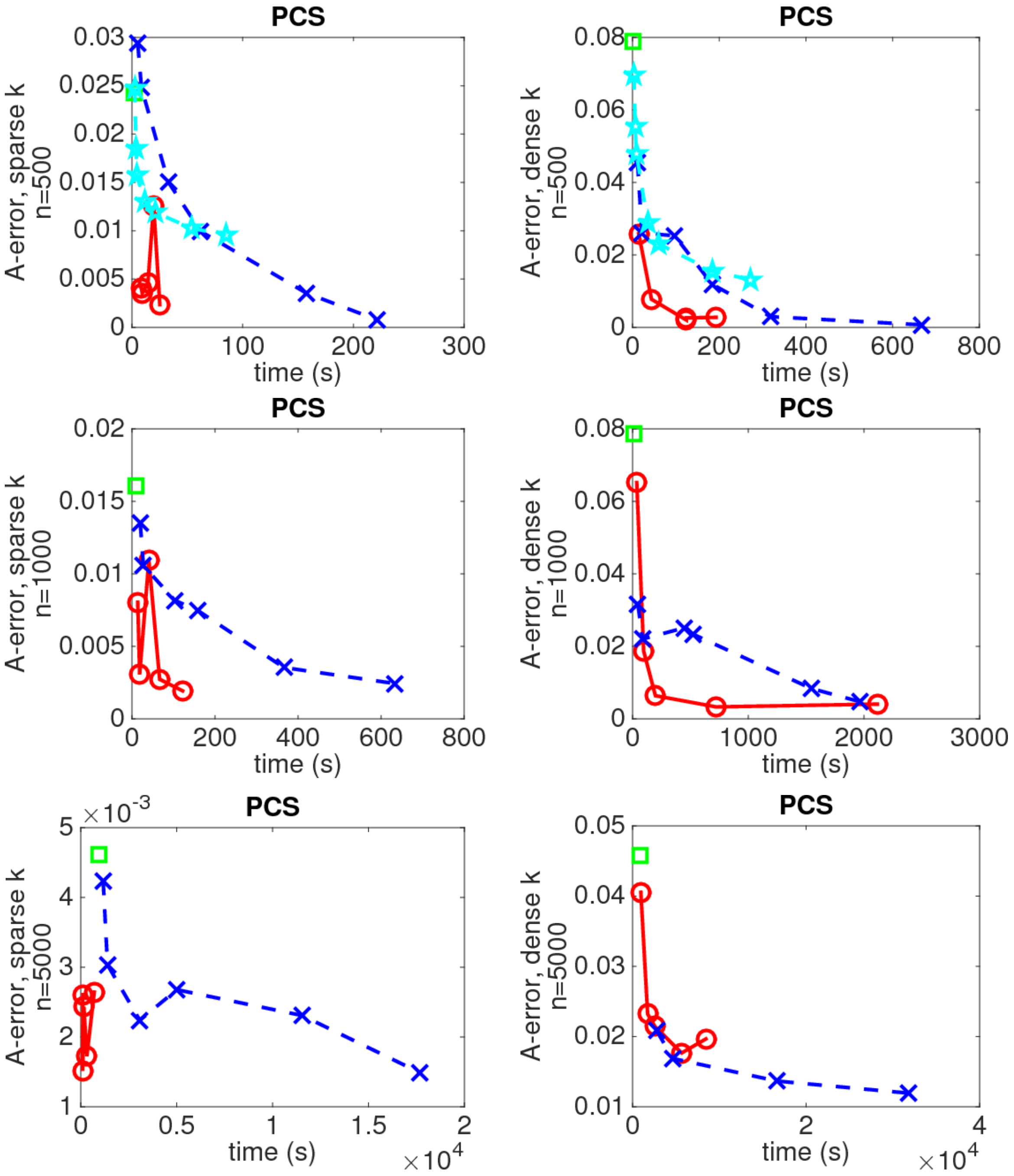


Figure 4.4: $\mathcal{E}_A$ vs. time, $n = \{500, 1000, 5000\}$, Left : k sparse, Right : k dense, piecewise constant half-planes, ○ ASAP LOE, × LOE BFGS, □ LE , ⋆ LOE MM

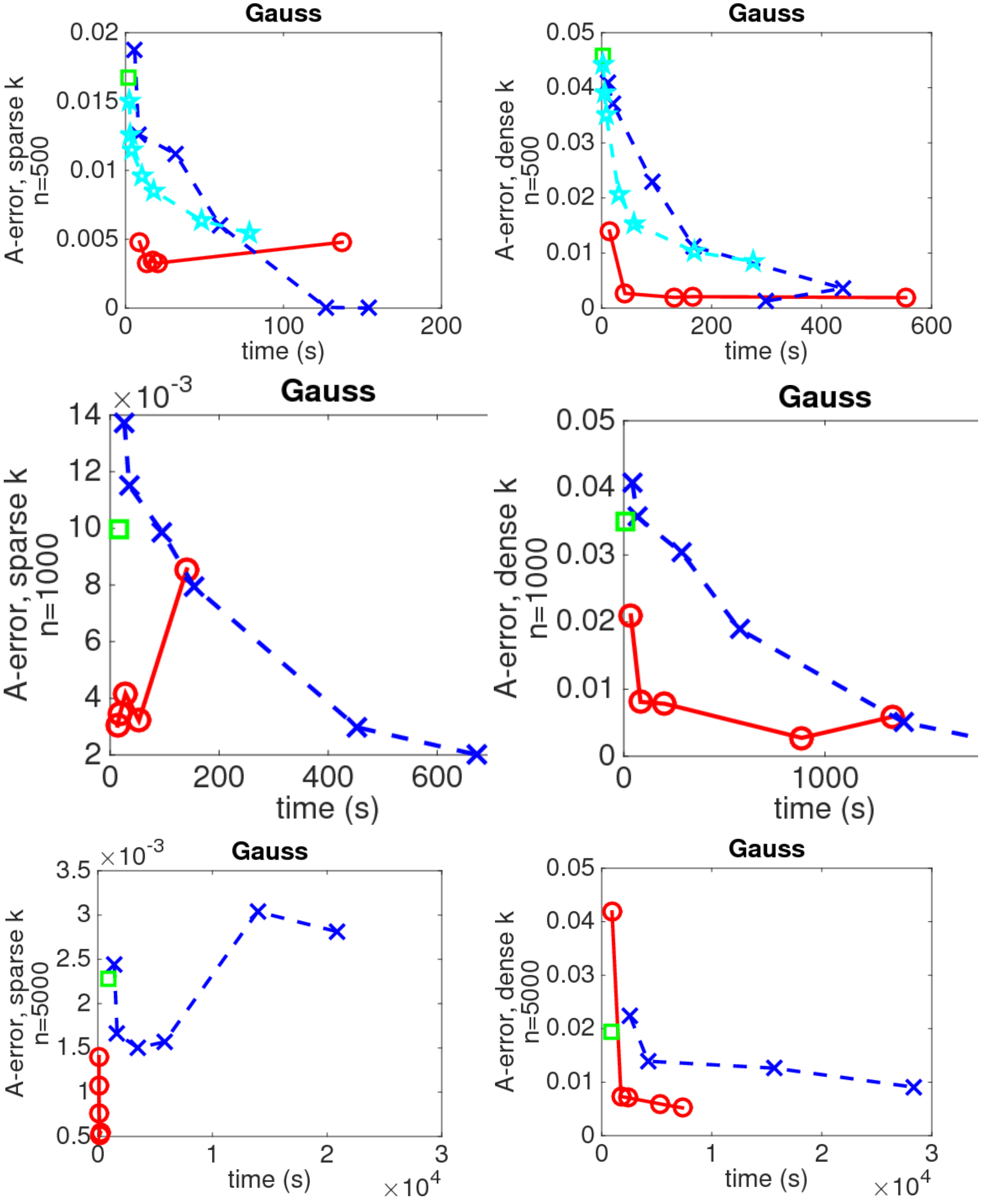


Figure 4.5: $\mathcal{E}_A$ vs. time, $n = \{500, 1000, 5000\}$, Left : k sparse, Right : k dense, Gaussian density ○ ASAP LOE, × LOE BFGS, □ LE , ⋆ LOE MM

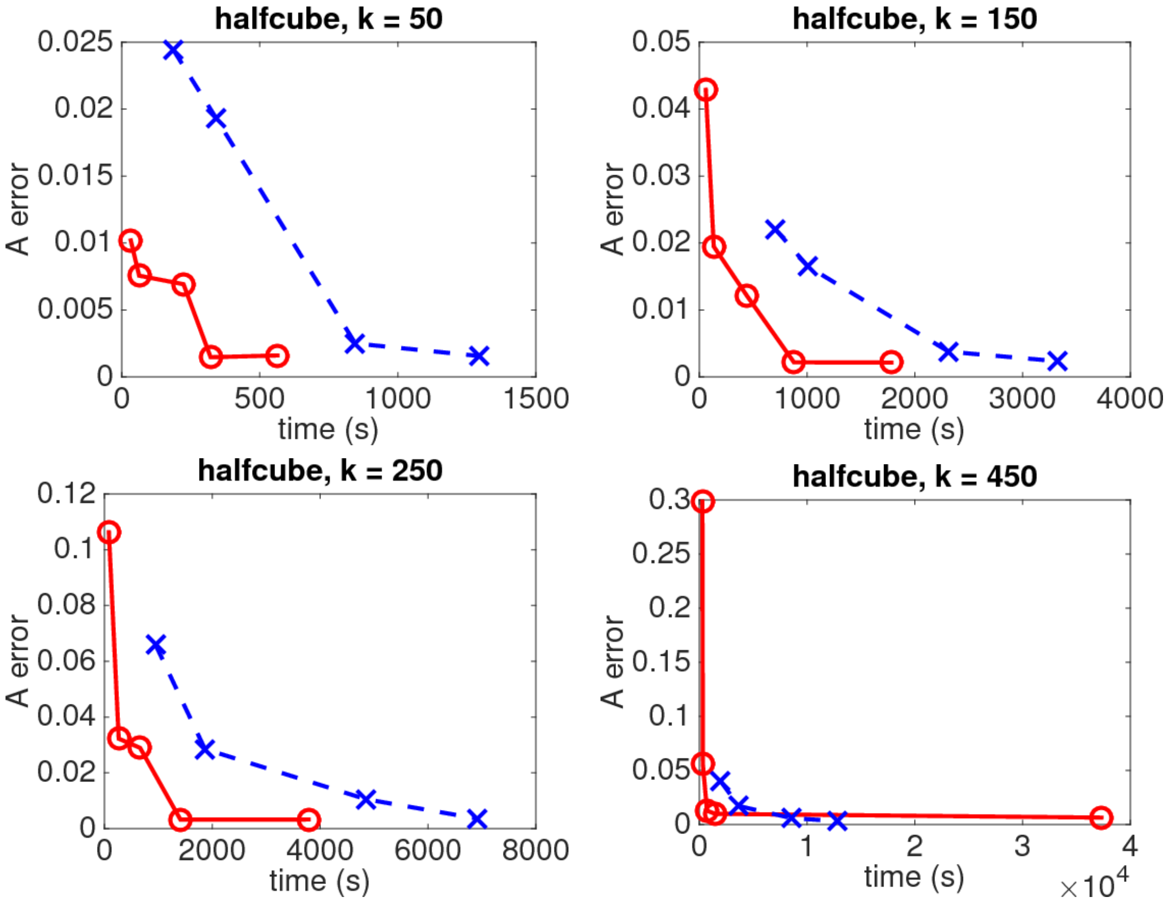


Figure 4.6: $\mathcal{E}_A$ vs. time, $n = \{500, 1000, 5000\}$, $k = 50, 150, 250, 450$, 3D half-cube density ○ ASAP LOE, × LOE BFGS, □ LE , ★ LOE MM

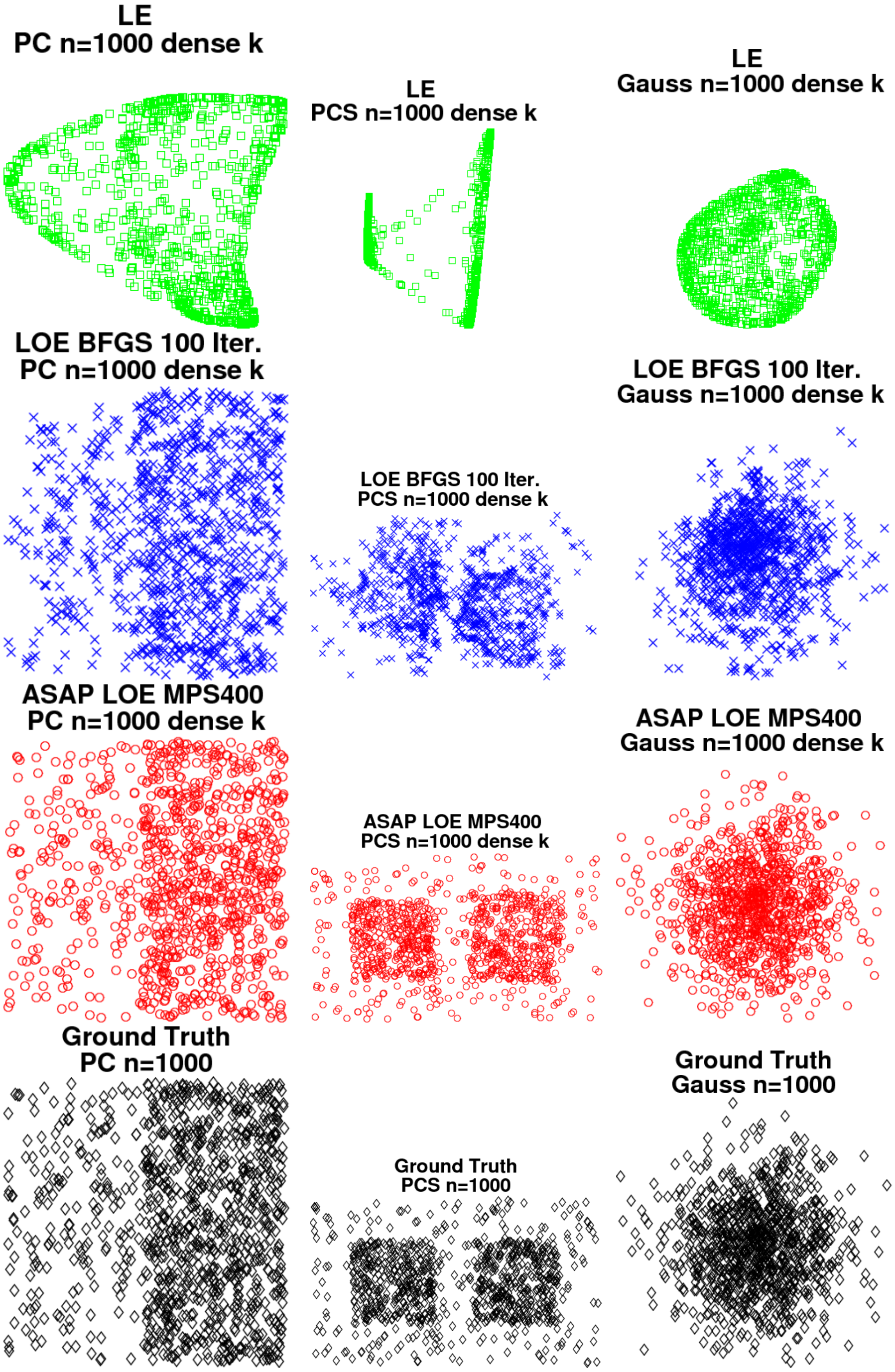


Figure 4.7: Embeddings for the PC (left), PCS (middle), and Gauss (right) data sets with $n = 1000$, and k dense. Row 1 : LE. Row 2: LOE BFGS Iter.=100. Row 3: ASAP LOE with MPS = 400 (with each ASAP result obtained is less time than the corresponding LOE result). Row 4: ground truth.

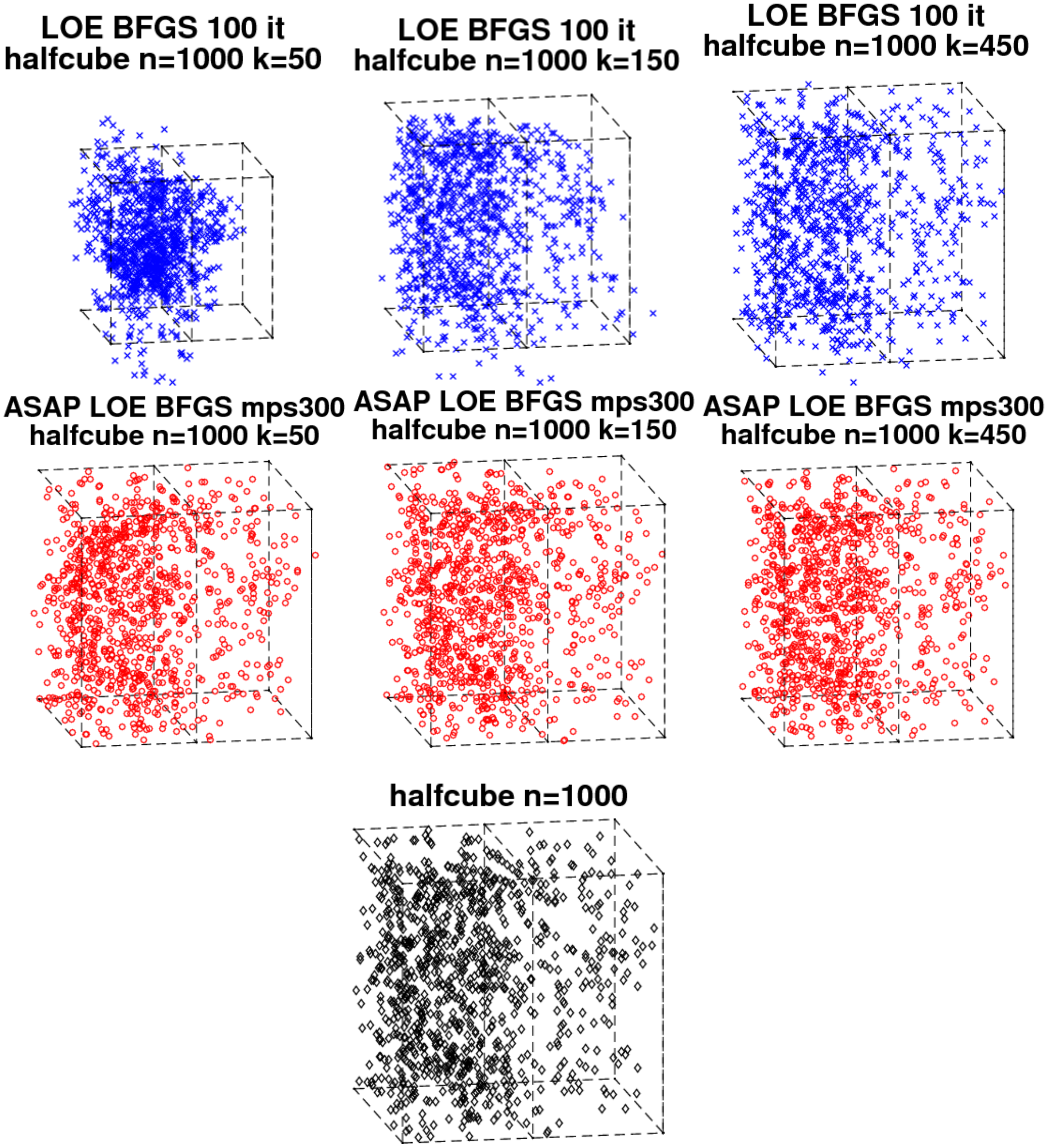


Figure 4.8: Embeddings for halfcube data sets with $n = 1000$, and $k = 50$ (left), 150 (middle), 450 (right) Row 2: LOE BFGS Iter.=100. Row 3: ASAP LOE with MPS = 300 (with each ASAP result obtained is less time than the corresponding LOE result). Row 4: ground truth.

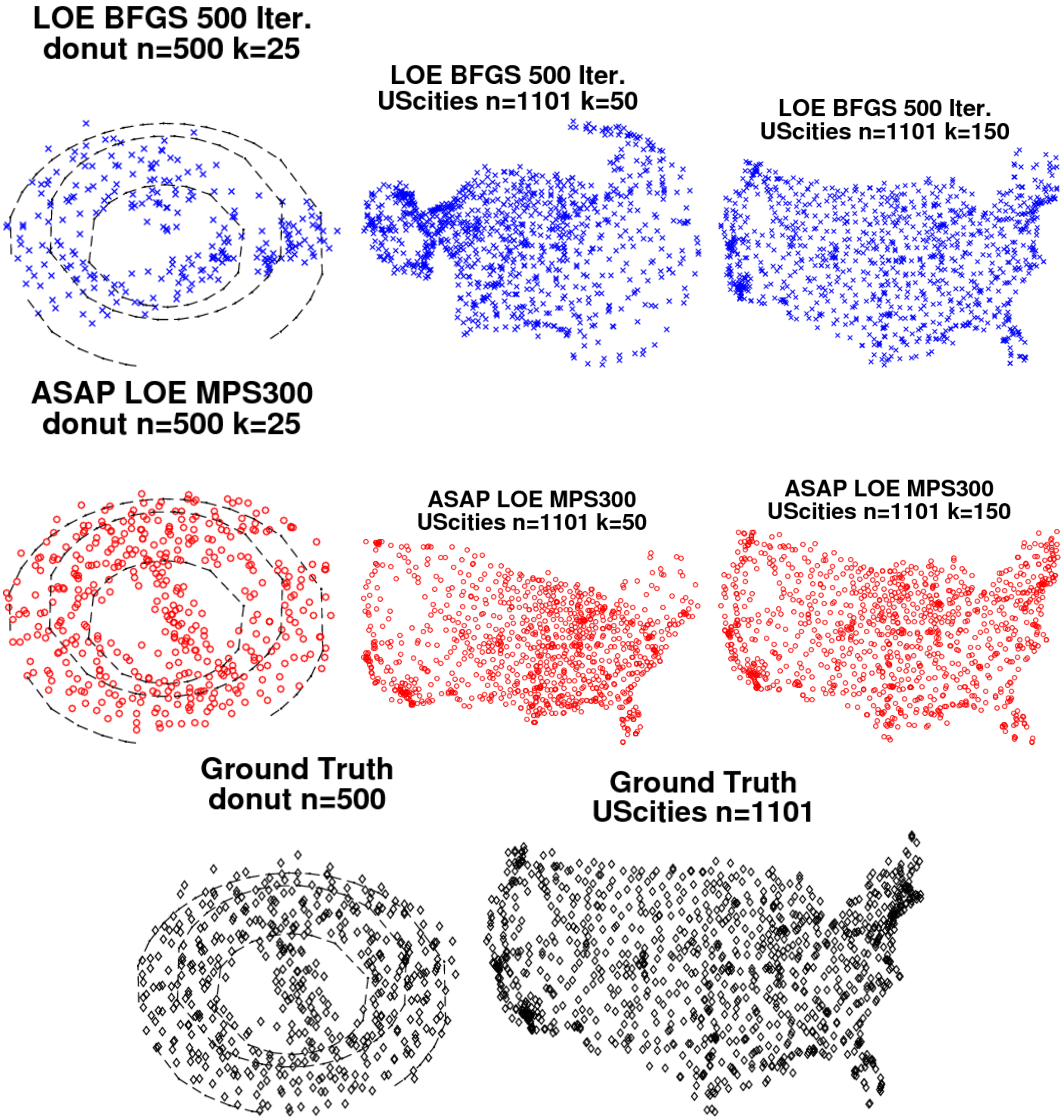


Figure 4.9: Embeddings of Donut (3D) and US Cities (2D) data sets. Row 1: LOE BFGS Iter.=500. Row 2: ASAP LOE MPS=300 (with each ASAP result obtained in less time than the corresponding LOE result). Row 3: Ground truth.

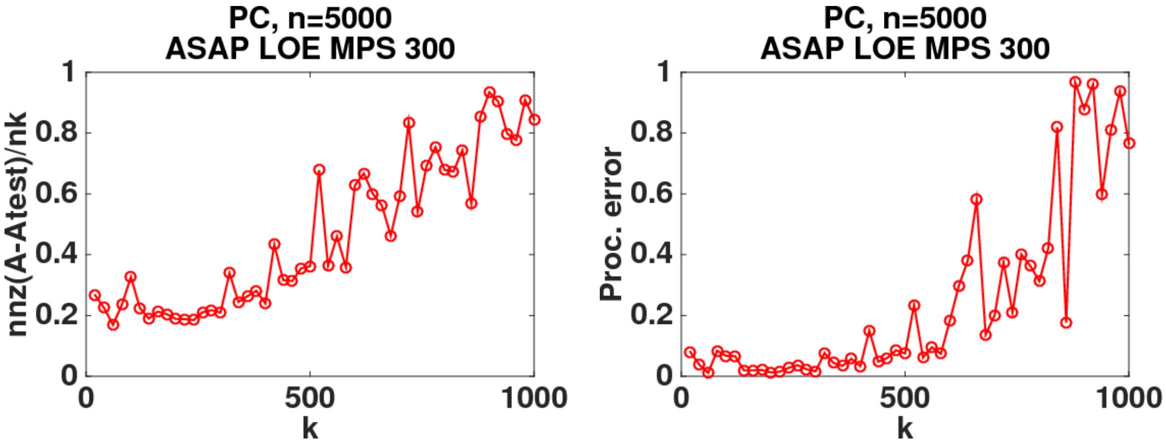


Figure 4.10: ASAP LOE MPS=300, $n = 5000$, k increasing by 20, Left: number of differences in adjacency matrix divided by number of edges, nk, Right: Procrustes error.

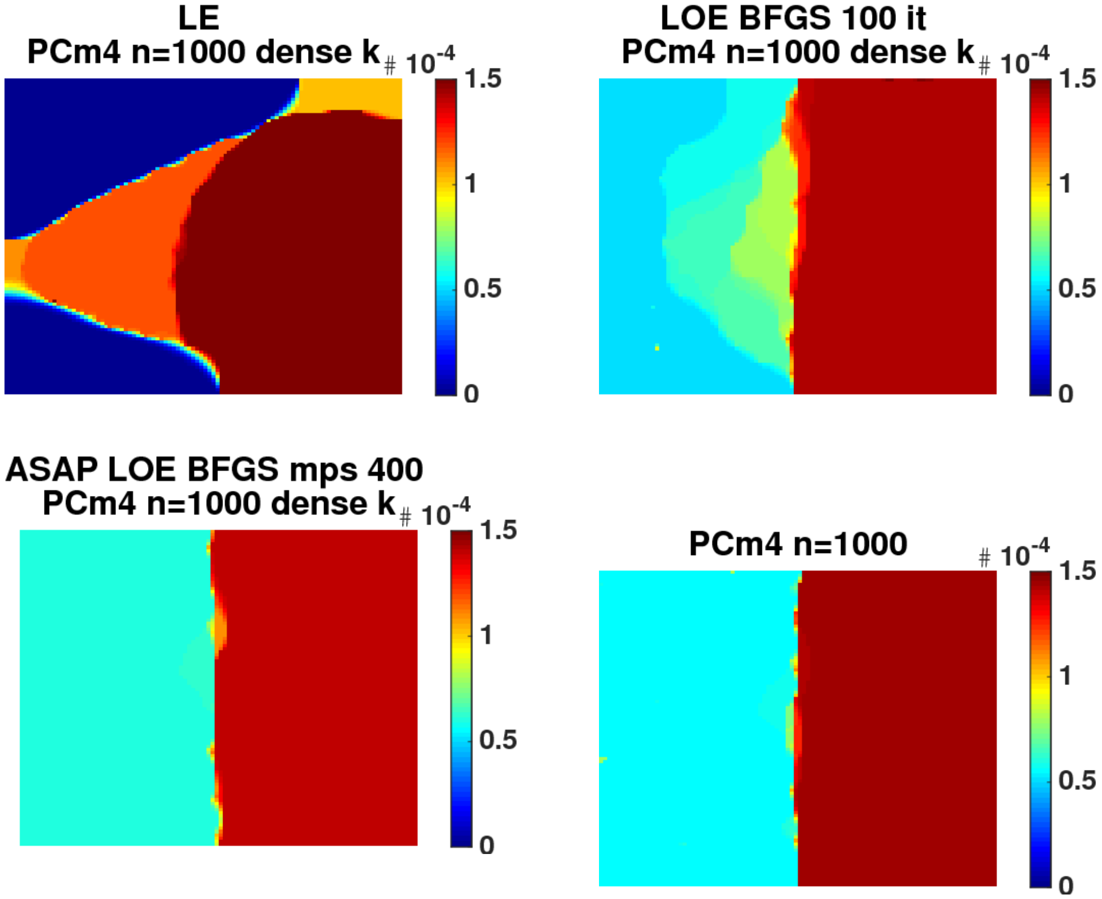


Figure 4.11: TV MPLE applied to example embeddings of PC $n = 1000$, k dense, and top left : LE, top right : LOE BFGS maxIt=100, bottom left : ASAP LOE BFGS max patch size 400, bottom right : estimated density from ground truth points, see column 1 of Figure 4.7

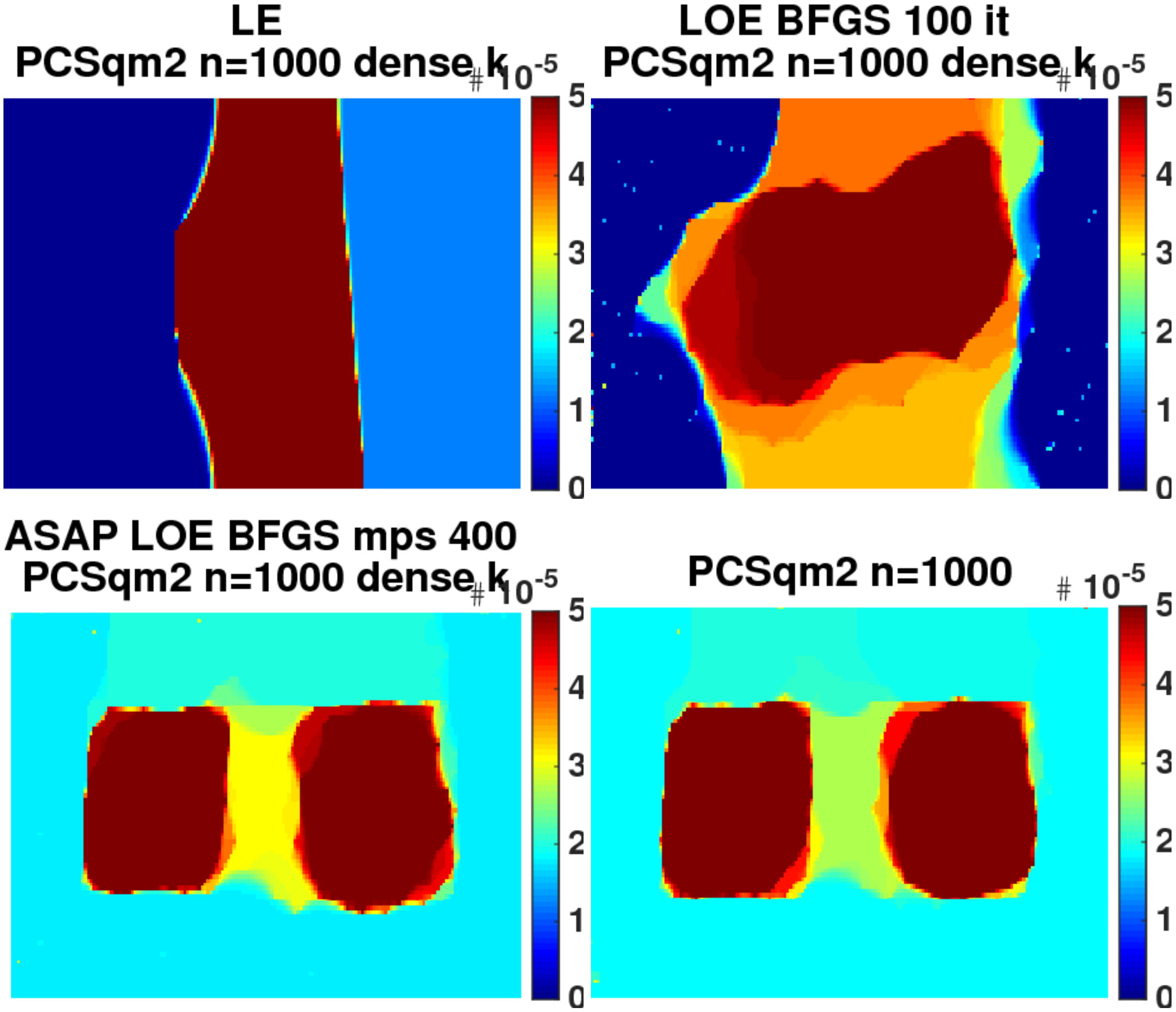


Figure 4.12: TV MPLE applied to example embeddings of PCS $n = 1000$, k dense, and top left : LE, top right : LOE BFGS maxIt=100, bottom left : ASAP LOE BFGS max patch size 400, bottom right : estimated density from ground truth points, see column 2 of Figure 4.7

4.6 Summary and discussion

We have demonstrated that the computational efficiency of LOE for the kNN embedding problem can be significantly improved, while maintaining and often improving spatial and ordinal accuracy in a distributed setting. Our application of the divide-and-conquer ASAP method renders the problem of kNN embedding significantly more tractable, distributing the embedding steps, and using fast spectral methods to combine them. We expect that such improvements will make it possible to use kNN embeddings in a broader range of settings, and that the ASAP framework will be of independent interest to the machine learning community for tackling large geometric embedding problems.

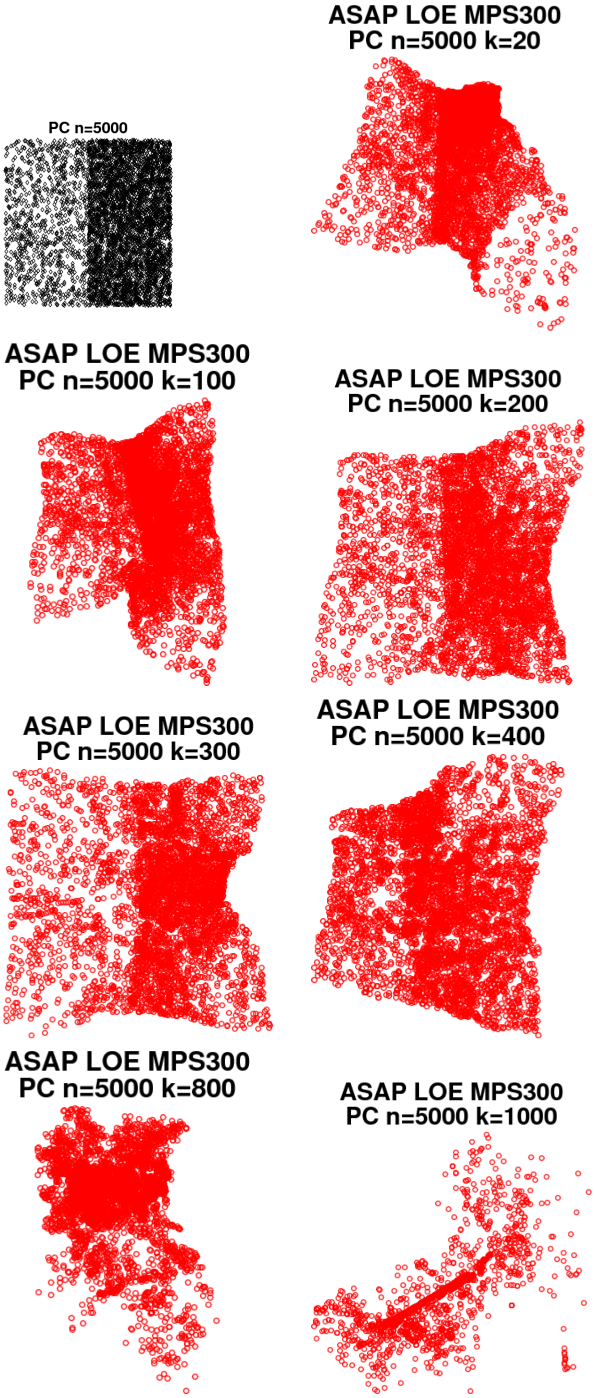


Figure 4.15: ASAP LOE BFGS MPS=300, $n = 5000$, k increasing by 20, Top left : originally sampled points, Remaining plots : recovered embeddings

APPENDIX A

To examine the effect of the non-local regularization term, we compute an alternate formulation of the NL H^1 MPLE problem and derive an inequality that solutions must satisfy. Recall from equation (2.1) that NL H^1 MPLE applied to the event samples $X = \{x_i\}_{i=1}^n$ with parameters $\alpha, \beta \geq 0$ is given by the following optimization.

$$u_{\alpha,\beta,X} := \underset{u \geq 0, \int_\Omega u = 1}{\arg\max} \sum_{i=1}^n \log\left(u(x_i)\right) - \alpha \iint_{\Omega\times\Omega} \left(\nabla_{w,s} u(x,y)\right)^2 dxdy - \frac{\beta}{2}\int_\Omega |\nabla u(x)|^2 dx.$$

For every such X, α, β one can show there exists nonnegative constants C_1, C_2 such that $u_{\alpha,\beta,X}$ is also the solution to a more constrained optimization.

$$\begin{aligned} u_{\alpha,\beta,X} = &\arg\max \sum_{i=1}^n \log\left(u(x_i)\right) \text{ subject to} \\ &\left\{u \geq 0, \int_\Omega u = 1, \iint_{\Omega\times\Omega} \left(\nabla_{w,s} u(x,y)\right)^2 dxdy \leq C_1, \frac{1}{2}\int_\Omega |\nabla u(x)|^2 dx \leq C_2\right\}. \end{aligned} \tag{4.7}$$

It can further be shown that for X and $\beta \geq 0$ fixed, C_1 is a non-increasing function of $\alpha \geq 0$ and for X and $\alpha \geq 0$ fixed, C_2 is a non-increasing function of $\beta \geq 0$.

Any solution of equation (4.7) satisfies $\iint_{\Omega\times\Omega}(\nabla_{w,s}u(x,y))^2 dxdy \leq C_1$, and likewise in the discrete setting we have the following.

$$\sum_{i,j\in\Omega} (u_i - u_j)^2 \frac{w_{ij}}{\sqrt{d_i d_j}} \leq C_1.$$

Thus for some nonnegative discrete function $f : \Omega \times \Omega \to \mathbb{R}^{\geq 0}$ with $\sum_{i,j\in\Omega} f_{ij} \leq C_1$ we have the following.

$$\forall i, j \in \Omega, \quad (u_i - u_j)^2 \leq f_{ij} \frac{\sqrt{d_i d_j}}{w_{ij}}. \tag{4.8}$$

Recalling that in our application, we set the weights w_{ij} to be non-local means applied to a housing image, $g : \Omega \to \mathbb{R}$, we can interpret what this means. Up to some factors constrained by the parameter C_1, the squared difference between the density at pixels i and j is bounded by $\sqrt{d_i d_j}/w_{ij}$. Thus the bound is made restrictive when : d_i and d_j are small, which means the patches of g around pixels i and j are very different from the rest of the image; and when w_{ij} is large, which means the neighborhoods of g around pixels i and j are similar to each other.

It is also worth noting that by constraint, the left-hand side of (4.8) is always smaller than or equal to 1. Thus for the inequality to be nontrivial, we must have $f_{ij} < w_{ij}/\sqrt{d_i d_j}$ for some pair $i, j \in \Omega$. Thus C_1 must be sufficiently small (or α sufficiently large) in order to guarantee that the non-local smoothing will have any effect on u.

APPENDIX B

One of the main questions in the field of rigidity theory asks whether one can uniquely determine (up to rigid transformations, such as translations, rotations, reflections) the coordinates of a set of points $p_1, \ldots, p_n$ given a partial set of distances $d_{ij} = ||p_i - p_j||$ between n points in $\mathbb{R}^d$. To make this dissertation self-contained, this short appendix if a very brief summary of the main definitions and results related to local and global rigidity from the literature (e.g., [39, 68, 74, 106, and references therein]). Readers who are unfamiliar with rigidity theory may use this short Appendix as a glossary. As previously discussed in Section 4.2.1, one of the steps of the divide-and-conquer approach proposed for the kNN-recovery problem relies to testing whether the underlying resulting patches are globally rigid. As observed in our numerical simulations detailed in Figures 4.3,4.4, 4.5, 4.6, 4.7 the final reconstruction is more accurate when we rely on global rigidity as a postprocessing step for the partitions obtained via spectral clustering. The intuition behind our approach is as follows. In the case when distance information is available, testing for global rigidity is a crucial step in making sure that each of the local patches has a unique embedding in its own reference frame, approximatively consistent with the ground truth, up to a rigid transformation. Since in the kNN-recovery problem, we do not have distance information but only ordinal data, thus we are faced with solving even a harder problem, we expect that the global rigidity check will improve the accuracy of the local patch embeddings. One specific example where our current rigidity heuristics improved results was in performing ASAP LOE BFGS with max patch size 300, on $n = 5000$ points drawn from the constant half-plane distribution, letting $k = 18$. In that example, performing the rigidity check and pruning gave a runtime of 107.056 s, an ordinal error of 0.00107096, and 0.0585465 Procrustes error, while skipping the rigidity check and pruning gave a runtime of 192.606 s, an ordinal error of 0.00154208 A error, and 0.175992 Procrustes error.

A *bar and joint framework* in $\mathbb{R}^d$ is defined as an undirected graph $G = (V, E)$ ($|V| = n, |E| = m$) together with a *configuration* p which assigns a point p_i in $\mathbb{R}^d$ to each vertex i of the graph. The edges of the graph correspond to distance constraints, that is, $(i, j) \in E$ if there is a bar of length d_{ij} between points p_i and p_j. We say that a framework $G(p)$ is *locally rigid* if there exists a neighborhood U of $G(p)$ such that $G(p)$ is the only framework in U with the same set of edge lengths, up to rigid transformations. In other words, there is no continuous deformation that preserves the given edge lengths. A configuration is *generic* if the coordinates do not satisfy any non-zero polynomial equation with integer coefficients (or equivalently algebraic coefficients).

Local rigidity in $\mathbb{R}^d$ has been shown to be a generic property, in the sense that either all generic frameworks of the graph G are locally rigid, or none of them are. A consequence of the seminal results of Gluck [64] and Asimow and Roth [5] asserts that the dimension of the null space of the rigidity matrix is the same at every generic point, and hence local rigidity in $\mathbb{R}^d$ is a generic property, meaning that either all generic frameworks of the graph G are locally rigid, or none of them are. With probability one, the rank of the rigidity matrix that corresponds to the unknown true displacement vectors equals the rank of the randomized rigidity matrix. A similar randomized algorithm for generic local rigidity was described in [68, Algorithm 3.2]. *Generic local rigidity* in $\mathbb{R}^d$ can be considered a combinatorial property of the graph G itself, independent of the particular realization. Using this observation, generic local rigidity can therefore be tested efficiently in any dimension using a randomized algorithm [74]: one can just randomize the displacement vectors $p_1, \ldots, p_n$ while ignoring the prescribed distance constraints that they have to satisfy, construct the so called *rigidity matrix* corresponding to the framework of the original graph with the randomized points and check its rank. We use this to make sure the obtained patches are local rigid.

Since local generic rigidity does not imply unique realization of the framework, it is possible that there exist multiple non-congruent realizations that satisfy the prescribed distances (which we do not even have available in the kNN recovery problem) One may consider for example, the $2D$-rigid graph with $n = 4$ vertices and $m = 5$ edges consisting of two triangles that can be folded with respect to their joint edge. We call a framework $G(p)$ *globally rigid* in $\mathbb{R}^d$ if all frameworks $G(q)$ in $\mathbb{R}^d$ which are $G(p)$-equivalent (have all bars the same length as $G(p)$) are congruent to $G(p)$ (i.e., related by a rigid transformation). Hendrickson proved two key necessary conditions for global rigidity of a framework at a generic configuration:

Theorem 4.6.1 (Hendrickson [74])**.** *If a framework $G(p)$, other than a simplex, is globally rigid for a generic configuration p in $\mathbb{R}^d$ then: 1) The graph G is vertex $(d + 1)$-connected; 2) The* *framework $G(p)$ is edge-2-rigid (or, redundantly rigid), in the sense that removing any one edge leaves a graph which is infinitesimally rigid.*

We say that a graph G is *generically globally rigid* in $\mathbb{R}^d$ if $G(p)$ is globally rigid at all generic configurations p [37, 38]. Though it has been conjectured for many years that global rigidity is a generic property, this fact was shown to be true only very recently. The seminal work of [38, 68] proves that global rigidity is a generic property of the graph in each dimension. The conditions of Hendrickson as stated in Theorem 4.6.1 are necessary for generic global rigidity. They are also sufficient on the line, and in the plane [76]. However, by a result of Connelly [37], $K_{5,5}$ in 3-space is generically edge-2-rigid and 5-connected but is not generically globally rigid.

One of the tools used in testing for global rigidity of frameworks relies on the notions on stress matrices, more popular perhaps in the engineering community. A *stress* is defined an assignment of scalars w_{ij} to the edges of the given graph G such that for every node $i \in V$ it holds that

$$\sum_{j\colon (i,j)\in E} \omega_{ij}(p_i - p_j) = 0. \tag{4.9}$$

Alternatively, it can be show that a stress is a vector w in the left null space of the rigidity matrix: $R_G(p)^T w = 0$. A stress vector can be rearranged into an $n \times n$ symmetric matrix Ω, known as the *stress matrix*, such that for $i \neq j$, the (i, j) entry of Ω is $\Omega_{ij} = -\omega_{ij}$, and the diagonal entries for (i, i) are $\Omega_{ii} = \sum_{j\colon j\neq i} \omega_{ij}$. Since all row and column sums are zero, it follows that the all-ones vector $(1 \;\; 1 \;\; \cdots \;\; 1)^T$ is in the null space of Ω as well as each of the coordinate vectors of the configuration p. Therefore, it follows that for generic configurations the rank of the stress matrix is at most $n - (d + 1)$. The following pairs of theorems give sufficient and necessary conditions for generic global rigidity:

Theorem 4.6.2 (Connelly [38])**.** *If p is a generic configuration in $\mathbb{R}^d$, such that there is a stress, where the rank of the associated stress matrix Ω is $n - (d + 1)$, then $G(p)$ is globally rigid in $\mathbb{R}^d$.*

Theorem 4.6.3 (Gortler, Healy, and Thurston [68])**.** *Suppose that p is a generic configuration in $\mathbb{R}^d$, such that $G(p)$ is globally rigid in $\mathbb{R}^d$. Then either $G(p)$ is a simplex or there is a stress where the rank of the associated stress matrix Ω is $n - (d + 1)$.*

Based on the latter theorem, the authors of [68] also provided a randomized polynomial algorithm for checking generic global rigidity of a graph [68, Algorithm 3.3], which we use to test for global rigidity of the patches in the kNN-recovery problem. If a given patch is generically locally rigid then their algorithm picks a random stress vector of the left null space of the rigidity matrix associated to this patch, and converts it into a stress matrix. If the rank of the stress matrix is exactly $n-(d+1)$, then we conclude that the patch is globally rigid, and if the rank is lower, then the respective patch is not globally rigid.

REFERENCES

[1] Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D. J., and Belongie, S. Generalized non-metric multidimensional scaling. In *International Conference on Artificial Intelligence and Statistics* (2007), pp. 11–18.

[2] Ailon, N. Active learning ranking from pairwise preferences with almost optimal query complexity. In *NIPS* (2011), pp. 810–818.

[3] Aldroubi, A., Chen, X., and Powell, A. Stability and robustness of ℓ_q minimization using null space property. In *Proc. 10th Int. Conf. Sampl. Theory Appl.* (Singapore, May 2011).

[4] Antoniadis, A. Wavelet methods in statistics: Some recent developments and their applications. *Stat. Surv. 1* (2007), 16–55.

[5] Asimow, L., and Roth, B. The rigidity of graphs. *Trans. Amer. Math. Soc. 245* (1978), 279–289.

[6] Batty, M., Longley, P., and Fotheringham, S. Urban growth and form: scaling, fractal geometry, and diffusion-limited aggregation. *Environment and planning A 21* (1989), 1447–1472.

[7] Belkin, M., and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 6 (June 2003), 1373–1396.

[8] Belongie, S., Fowlkes, C., Chung, F., and Malik, J. Spectral partitioning with indefinite kernels using the Nyström extension. In *Computer Vision-ECCV 2002*. Springer, 2002, pp. 531–542.

[9] Bernasco, W., and Nieuwbeerta, P. How do residential burglars select target areas? a new approach to the analysis of criminal location choice. *British Journal of Criminology 45*, 3 (2005), 296–315.

[10] Bertozzi, A. L., and Flenner, A. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation 10*, 3 (2012), 1090–1118.

[11] Biswas, P., Aghajan, H., and Ye, Y. Semidefinite programming algorithms for sensor network localization using angle of arrival information. In *Proc. 39th Annu. Asilomar Conf. Signals, Systems, and Computers* (Oct. 2005), pp. 220–224.

[12] Biswas, P., Lian, T. C., Wang, T. C., and Ye, Y. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks 2*, 2 (2006), 188–220.

[13] Biswas, P., Liang, T., Toh, K., Ye, Y., and Wang, T. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering 3*, 4 (2006), 360–371.

[14] Biswas, P., and Ye, Y. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks* (New York, 2004), ACM, pp. 46–54.

[15] Block, R., and Bernasco, W. Finding a serial burglar's home using distance decay and conditional origin–destination patterns: a test of empirical bayes journey-to-crime estimation in the hague. *Journal of Investigative Psychology and Offender Profiling 6*, 3 (2009), 187–211.

[16] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn. 3*, 1 (2011), 1–122.

[17] Boyd, S., and Vandenberghe, L. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2009.

[18] Bruckstein, A. M., Donoho, D. L., and Elad, M. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev. 51*, 1 (2009), 34–81.

[19] Buades, A., Coll, B., and Morel, J.-M. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation 4*, 2 (2005), 490–530.

[20] Cai, T. T., and Zhang, A. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Appl. Comput. Harmon. Anal. 35*, 1 (Jul. 2013), 74–93.

[21] Candès, E., and Tao, T. Decoding by linear programming. *IEEE Trans. Inf. Theory 51*, 12 (Dec. 2005), 4203–4215.

[22] Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory 52*, 2 (Feb. 2006), 489–509.

[23] Chambolle, A., and Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision 40*, 1 (May 2011), 120–145.

[24] Chartrand, R. Exact reconstructions of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett. 14*, 10 (October 2007), 707–710.

[25] Chartrand, R. Nonconvex compressed sensing and error correction. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Honolulu, HI, Apr. 2007).

[26] Chartrand, R. Nonconvex compressive sensing and reconstruction of gradient-sparse images: random vs. tomographic Fourier sampling. In *Proc. IEEE Int. Conf. Image Process.* (San Diego, CA, Oct. 2008).

[27] Chartrand, R. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. In *Proc. IEEE Int. Symp. Biomed. Imaging* (Boston, MA, Jun. 2009).

[28] Chartrand, R. Nonconvex splitting for regularized low-rank + sparse decomposition. *IEEE Trans. Signal Process. 60*, 11 (Nov. 2012), 5810–5819.

[29] Chartrand, R. Generalized shrinkage and penalty functions. In *Proc. IEEE Glob. Conf. Signal Inform. Process.* (Austin, TX, Dec. 2013).

[30] Chartrand, R. Nonconvex compressive sensing for X-ray CT: an algorithm comparison. In *Proc. Asilomar Conf. Signal Syst. Comput.* (Pacific Grove, CA, Nov. 2013).

[31] Chartrand, R. Shrinkage mappings and their induced penalty functions. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Florence, Italy, May 2014).

[32] Chartrand, R., and Staneva, V. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems 24*, 035020 (2008), 1–14.

[33] Chartrand, R., and Wohlberg, B. A nonconvex ADMM algorithm for group sparsity with sparse groups. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Vancouver, Canada, May 2013), pp. 6009–6013.

[34] Chen, X., Xu, F., and Ye, Y. Lower bound theory of nonzero entries in solutions of ℓ_2-ℓ_p minimization. *SIAM J. Sci. Comput. 32*, 5 (2010), 2832–2852.

[35] Chung, F. R. *Spectral graph theory*, vol. 92. AMS Bookstore, 1997.

[36] Cohen, A., Dahmen, W., and DeVore, R. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc. 22*, 1 (Jan. 2009), 211–231.

[37] Connelly, R. On generic global rigidity. *Applied Geometry and Discrete Mathematics 4* (1991), 147–155.

[38] Connelly, R. Generic global rigidity. *Discrete Comput. Geom 33* (2005), 549–563.

[39] Connelly, R., and Whiteley, W. J. Global rigidity: The effect of coning. *Discrete and Computational Geometry* (2009). ISSN 0179-5376 (Print) 1432-0444 (Online).

[40] Cour, T., Benezit, F., and Shi, J. Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), vol. 2, pp. 1124–1131 vol. 2.

[41] Cucuringu, M., Lipman, Y., and Singer, A. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Trans. Sen. Netw. 8*, 3 (Aug. 2012), 19:1–19:42.

[42] Cucuringu, M., Singer, A., and Cowburn, D. Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference 1*, 1 (2012), 21–67.

[43] Cucuringu, M., and Woodworth, J. Ordinal embedding of unweighted kNN graphs via synchronization. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE International Workshop on* (2015), IEEE.

[44] Daubechies, I., Defrise, M., and Mol, C. D. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math. 57*, 11 (Nov. 2004), 1413–1457.

[45] Davies, M. E., and Gribonval, R. Restricted isometry constants where ℓ^p sparse recovery can fail for $0 < p \leq 1$. *IEEE Trans. Inf. Theory 55*, 5 (May 2009), 2203–2214.

[46] Dong, B., and Zhang, Y. An efficient algorithm for ℓ_0 minimization in wavelet frame based image restoration. *J. Sci. Comput. 54*, 2-3 (Feb. 2013), 350–368.

[47] Donoho, D. L. Compressed sensing. *IEEE Trans. Inf. Theory 52*, 4 (Apr. 2006), 1289–1306.

[48] Donoho, D. L., and Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proc. Natl. Acad. Sci. USA 100*, 5 (2003), 2197–2202.

[49] Donoho, D. L., Johnstone, I., and Montanari, A. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. *Information Theory, IEEE Transactions on 59*, 6 (2013), 3396–3433.

[50] Edler, D., and Rosvall, M. The mapequation. `http://www.mapequation.org`. software package, available online.

[51] Eggermont, P. P. B., and LaRiccia, V. N. *Maximum Penalized Likelihood Estimation: Regression*, vol. 2. Springer, 2001.

[52] Elad, M., and Bruckstein, A. M. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Trans. Inf. Theory 48*, 9 (Sep. 2002), 2558–2567.

[53] Foucart, S. A note on guaranteed sparse recovery via ℓ_1-minimization. *Appl. Comput. Harmon. Anal. 29*, 1 (Jul. 2010), 97–103.

[54] Foucart, S. Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, M. Neamtu and L. Schumaker, Eds., vol. 13 of *Springer Proceedings in Mathematics*. Springer, New York, NY, 2012, pp. 65–77.

[55] Foucart, S., and Lai, M.-J. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Appl. Comput. Harmon. Anal. 26*, 3 (May 2009), 395–407.

[56] Fowlkes, C., Belongie, S., Chung, F., and Malik, J. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (2004), 214–225.

[57] Fruchterman, T. M., and Reingold, E. M. Graph drawing by force-directed placement. *Software: Practice and experience 21*, 11 (1991), 1129–1164.

[58] Gabay, D., and Mercier, B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comp. Math. Appl. 2*, 1 (1976), 17–40.

[59] Gao, H.-Y., and Bruce, A. G. Waveshrink with firm shrinkage. *Statistica Sinica 7*, 4 (1997), 855–874.

[60] Gao, Y., and Sun, D. A majorized penalty approach for calibrating rank constrained correlation matrix problems.

[61] Gilboa, G., and Osher, S. J. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation 6*, 2 (2007), 595–630.

[62] Gilboa, G., and Osher, S. J. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation 7*, 3 (2008), 1005–1028.

[63] Glowinski, R., and Marrocco, A. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de Dirichlet non lineares. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle 9* (1975), 41–76.

[64] Gluck, H. Almost all simply connected closed surfaces are rigid. *Geometric Topology, Lecture Notes in Mathematics 438* (1975), 225–239.

[65] Goldstein, T., and Osher, S. The split Bregman method for L1 regularized problems. *SIAM J. Imaging Sci. 2*, 2 (2009), 323–343.

[66] Gonzalez, M., Huang, X., Martinez, D. S. H., Hsieh, C. H., Huang, Y. R., Irvine, B., Short, M. B., and Bertozzi, A. L. A third generation micro-vehicle testbed for cooperative control and sensing strategies. In *ICINCO (2)* (2011), pp. 14–20.

[67] Gorodnitsky, I. F., and Rao, B. D. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on 45*, 3 (1997), 600–616.

[68] Gortler, S. J., Healy, A. D., and Thurston, D. P. Characterizing generic global rigidity. *AMERICAN JOURNAL OF MATHEMATICS 4* (2010), 897.

[69] Grady, L., and Schwartz, E. L. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 3 (2006), 469–475.

[70] Grant, M., and Boyd, S. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110.

[71] Grant, M., and Boyd, S. CVX: MATLAB software for disciplined convex programming, version 1.21, Apr. 2011.

[72] Gribonval, R., and Nielsen, M. Sparse representations in unions of bases. *IEEE Trans. Inf. Theory 49*, 12 (Dec. 2003), 3320–3325.

[73] Gribonval, R., and Nielsen, M. Highly sparse representations from dictionaries are unique and independent of the sparseness measure. *Appl. Comput. Harmon. Anal. 22*, 3 (May 2007), 335–355.

[74] Hendrickson, B. Conditions for unique graph realizations. *SIAM J Comput 21* (1992), 65–84.

[75] Herold, M., Goldstein, N. C., and Clarke, K. C. The spatiotemporal form of urban growth: measurement, analysis and modeling. *Remote sensing of Environment 86*, 3 (2003), 286–302.

[76] Jackson, B., and Jordán, T. Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B 94*, 1 (2005), 1–29.

[77] Jamieson, K. G., and Nowak, R. D. Active ranking using pairwise comparisons. In *NIPS* (2011), vol. 24, pp. 2240–2248.

[78] Jamieson, K. G., and Nowak, R. D. Low-dimensional embedding using adaptively selected ordinal data. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on* (2011), IEEE, pp. 1077–1084.

[79] Kamada, T., and Kawai, S. An algorithm for drawing general undirected graphs. *Information processing letters 31*, 1 (1989), 7–15.

[80] Kennedy, L. W., Caplan, J. M., and Piza, E. Risk clusters, hotspots, and spatial intelligence: risk terrain modeling as an algorithm for police resource allocation strategies. *Journal of Quantitative Criminology 27*, 3 (2011), 339–362.

[81] Kostić, T., and Bertozzi, A. L. Statistical density estimation using threshold dynamics for geometric motion. *Journal of Scientific Computing 54*, 2-3 (2013), 513–530.

[82] Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27.

[83] Kruskal, J. B. Nonmetric multidimensional scaling: a numerical method. *Psychometrika 29*, 2 (1964), 115–129.

[84] Kruskal, J. B., and Wish, M. *Multidimensional scaling*, vol. 11. Sage, 1978.

[85] Lai, M.-J., and Liu, L. Y. The null space property for sparse recovery from multiple measurement vectors. *Appl. Comput. Harmon. Anal. 30*, 3 (May 2011), 402–406.

[86] Lan, Y., Guo, J., Cheng, X., and Liu, T.-Y. Statistical consistency of ranking methods in a rank-differentiable probability space. In *NIPS* (2012), pp. 1241–1249.

[87] Lewis, E., Mohler, G. O., Brantingham, P. J., and Bertozzi, A. L. Self-exciting point process models of civilian deaths in Iraq. *Security Journal 25*, 3 (2011), 244–264.

[88] Liu, H., and Brown, D. E. Criminal incident prediction using a point-pattern-based density model. *International journal of forecasting 19*, 4 (2003), 603–622.

[89] Lou, Y., Zhang, X., Osher, S. J., and Bertozzi, A. L. Image recovery via nonlocal operators. *Journal of Scientific Computing 42*, 2 (2010), 185–197.

[90] Majumdar, A., and Ward, R. K. An algorithm for sparse MRI reconstruction by Schatten p-norm minimization. *Magn. Reson. Imaging 29*, 3 (April 2011), 408–417.

[91] Martinez, D. S. H., Gonzalez, M., Huang, X., Irvine, B., Hsieh, C. H., Huang, Y. R., Short, M. B., and Bertozzi, A. L. An economical testbed for cooperative control and sensing strategies of robotic micro-vehicles. In *Informatics in Control, Automation and Robotics*. Springer, 2013, pp. 65–75.

[92] McFee, B., and Lanckriet, G. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 721–728.

[93] McFee, B., and Lanckriet, G. R. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010), pp. 775–782.

[94] Merkurjev, E., Kostic, T., and Bertozzi, A. L. An MBO scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences 6*, 4 (2013), 1903–1930.

[95] Meyer, R. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *J. Comput. Syst. Sci. 12*, 1 (Feb. 1976), 108–121.

[96] Mohar, B. The Laplacian spectrum of graphs. *Graph theory, combinatorics, and applications 2* (1991), 871–898.

[97] Mohler, G. O., Bertozzi, A. L., Goldstein, T. A., and Osher, S. J. Fast TV regularization for 2d maximum penalized likelihood estimation. *Journal of Computational and Graphical Statistics 20*, 2 (2011), 479–491.

[98] Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. Self-exciting point process modeling of crime. *Journal of the American Statistical Association 106*, 493 (2011).

[99] Muthukrishnan, S. *Data streams: Algorithms and applications*. Now Publishers Inc., Hanover, MA, 2005.

[100] Ouyang, H., and Gray, A. Learning dissimilarities by ranking: from SDP to QP. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 728–735.

[101] Peyré, G., Bougleux, S., and Cohen, L. Non-local regularization of inverse problems. In *Computer Vision–ECCV 2008*. Springer, 2008, pp. 57–68.

[102] Quist, M., and Yona, G. Distributional scaling: An algorithm for structure-preserving embedding of metric and nonmetric spaces. *The Journal of Machine Learning Research 5* (2004), 399–420.

[103] Rockafellar, R. T., and Wets, R. J.-B. *Variational Analysis*. Springer, Berlin, Germany, 1998.

[104] Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J. Core-periphery structure in networks. *SIAM Journal on Applied mathematics 74*, 1 (2014), 167–190.

[105] Rosales, R., and Fung, G. Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 367–373.

[106] Roth, B. Rigid and flexible frameworks. *The American Mathematical Monthly 88* (1981), 6–21.

[107] Saab, R., Chartrand, R., and Özgür Yilmaz. Stable sparse approximations via nonconvex optimization. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Las Vegas, NV, Apr. 2008), pp. 3885–3888.

[108] Sardy, S., and Tseng, P. Density estimation by total variation penalized likelihood driven by the sparsity ℓ_1 information criterion. *Scandinavian Journal of Statistics 37*, 2 (2010), 321–337.

[109] Scott, D. W. *Multivariate density estimation*. Wiley, 1992.

[110] Shamsi, D., Taheri, N., Zhu, Z., and Ye, Y. On Sensor Network Localization Using SDP Relaxation. *ArXiv e-prints* (Oct. 2010).

[111] Shaw, B., and Jebara, T. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 937–944.

[112] Shepard, R. N. The analysis of proximities: Multidimensional scaling with an unknown distance function i. *Psychometrika 27*, 2 (1962), 125–140.

[113] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888–905.

[114] Short, M., D'Orsogna, M., Brantingham, P., and Tita, G. Measuring and modeling repeat and near-repeat burglary effects. *Journal of Quantitative Criminology 25*, 3 (2009), 325–339.

[115] Short, M. B., Brantingham, P. J., Bertozzi, A. L., and Tita, G. E. Dissipation and displacement of hotspots in reaction-diffusion models of crime. *Proceedings of the National Academy of Sciences 107*, 9 (2010), 3961–3965.

[116] Short, M. B., D'Orsogna, M. R., Pasour, V. B., Tita, G. E., Brantingham, P. J., Bertozzi, A. L., and Chayes, L. B. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences 18*, supp01 (2008), 1249–1267.

[117] Sibson, R. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society. Series B (Methodological)* (1978), 234–238.

[118] Silverman, B. W. *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986.

[119] Singer, A. Angular synchronization by eigenvectors and semidefinite programming. *Appl. Comput. Harmon. Anal. 30*, 1 (2011), 20–36.

[120] Smith, L. M., Keegan, M. S., Wittman, T., Mohler, G. O., and Bertozzi, A. L. Improving density estimation by incorporating spatial information. *EURASIP Journal on Advances in Signal Processing 2010* (2010), 7.

[121] Sun, Q. Sparse approximation property and stable recovery of sparse signals from noisy measurements. *IEEE Trans. Signal Process. 59*, 10 (Oct. 2011), 5086–5090.

[122] Sun, Q. Recovery of sparsest signals via ℓ^q-minimization. *Appl. Comput. Harmon. Anal. 32*, 3 (May 2012), 329–341.

[123] Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. T. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033* (2011).

[124] Terada, Y., and Luxburg, U. V. Local ordinal embedding. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (2014), pp. 847–855.

[125] Toh, K., Biswas, P., and Ye, Y. SNLSDP version 0 - a MATLAB software for sensor network localization, October 2008.

[126] Torgerson, W. S. *Theory and methods of scaling.* Wiley, 1958.

[127] Townsley, M., Homel, R., and Chaseling, J. Infectious burglaries. A test of the near repeat hypothesis. *British Journal of Criminology 43*, 3 (2003), 615–633.

[128] Van der Maaten, L., and Weinberger, K. Stochastic triplet embedding. In *IEEE Workshop on Machine Learning for Signal Processing (MLSP)* (2012), pp. 1–6.

[129] Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing 17*, 4 (2007), 395–416.

[130] Von Luxburg, U., and Alamgir, M. Density estimation from unweighted k-nearest neighbor graphs: a roadmap. In *Advances in Neural Information Processing Systems* (2013), pp. 225–233.

[131] von Luxburg, U., et al. Uniqueness of ordinal embedding. In *Proceedings of The 27th Conference on Learning Theory* (2014), pp. 40–67.

[132] Voronin, S., and Chartrand, R. A new generalized thresholding algorithm for inverse problems with sparsity constraints. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Vancouver, Canada, May 2013), pp. 1636–1640.

[133] Wang, X., and Brown, D. E. The spatio-temporal modeling for criminal incidents. *Security Informatics 1*, 1 (2012), 1–17.

[134] Wauthier, F., Jordan, M., and Jojic, N. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning* (2013), pp. 109–117.

[135] Weinberger, K. Q., and Saul, L. K. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI* (2006), vol. 6, pp. 1683–1686.

[136] Wilson, J. Q., and Kelling, G. L. Broken windows. *Atlantic Monthly 249*, 3 (1982), 29–38.

[137] Woodworth, J., and Chartrand, R. Compressed sensing recovery via nonconvex shrinkage penalties. *arXiv preprint arXiv:1504.02923* (2015).

[138] Woodworth, J. T., Mohler, G. O., Bertozzi, A. L., and Brantingham, P. J. Non-local crime density estimation incorporating housing information. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 372*, 2028 (2014), 20130403.

[139] Wu, R., and Chen, D.-R. The improved bounds of restricted isometry constant for recovery via ℓ_p-minimization. *IEEE Trans. Inf. Theory 59*, 9 (Sep. 2013), 6142–6147.

[140] Wu, S.-H., Chuang, K.-T., Chen, C.-M., and Chen, M.-S. Toward the optimal itinerary-based KNN query processing in mobile sensor networks. *Knowledge and Data Engineering, IEEE Transactions on 20*, 12 (2008), 1655–1668.

[141] Yan, D., Huang, L., and Jordan, M. I. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), ACM, pp. 907–916.

[142] Yukawa, M., and Amari, S.-i. ℓ_p-regularized least squares ($0 < p < 1$) and critical path. arXiv preprint 1304.6591, 2013.

[143] Zhang, X., and Chan, T. F. Wavelet inpainting by nonlocal total variation. *Inverse problems and Imaging 4*, 1 (2010), 191–210.

[144] Zhou, D., and Schölkopf, B. A regularization framework for learning from graph data. In *ICML 2004 Workshop on Statistical Relational Learning and its Connections to Other Fields* (2004), Citeseer, p. 132.

[145] Zhu, Z., So, A. M. C., and Ye, Y. Universal rigidity: Towards accurate and efficient localization of wireless networks. In *Proc. IEEE INFOCOM* (2010).