The Dissertation Committee for Joel Aaron Tropp
certifies that this is the approved version of the following dissertation:

Topics in Sparse Approximation

Committee:

Inderjit S. Dhillon, Supervisor

Anna C. Gilbert, Supervisor

E. Ward Cheney

Alan K. Cline

Robert W. Heath Jr.

Topics in Sparse Approximation

by

Joel Aaron Tropp, B.A., B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2004

To my parents and my little sister,
who have supported me
through all my successes and failures.

Topics in Sparse Approximation

Publication No. _____

Joel Aaron Tropp, Ph.D.
The University of Texas at Austin, 2004

Supervisors: Inderjit S. Dhillon
Anna C. Gilbert

Sparse approximation problems request a good approximation of an input signal as a linear combination of elementary signals, yet they stipulate that the approximation may involve only a few of the elementary signals. This class of problems arises throughout applied mathematics, statistics, and electrical engineering, but small theoretical progress has been made over the last fifty years. This dissertation offers four main contributions to the theory of sparse approximation.

The first two contributions concern the analysis of two types of numerical algorithms for sparse approximation: greedy methods and convex relaxation methods. Greedy methods make a sequence of locally optimal choices in an effort to obtain a globally optimal solution. Convex relaxation methods replace the combinatorial sparse approximation problem with a related convex optimization in hope that their solutions will coincide. This work delineates conditions under which greedy methods and convex relaxation methods actually succeed in solving a well-defined sparse approximation problem in part or in full. The conditions for both classes of algorithms are remarkably similar, in spite of the fact that the two analyses differ significantly.

The study of these algorithms yields geometric conditions on the collection of elementary signals which ensure that sparse approximation problems are computationally tractable. One may interpret these conditions as a requirement that the elementary signals should form a good packing of points in projective space. The third contribution of this work is an alternating projection algorithm that can produce good packings of points in projective space. The output of this algorithm frequently matches the best recorded solutions of projective packing problems. It can also address many related packing problems that have never been studied numerically.

Finally, the dissertation develops a novel connection between sparse approximation problems and clustering problems. This perspective shows that many clustering problems from the literature can be viewed as sparse approximation problems where the collection of elementary signals must be learned along with the optimal sparse approximation. This treatment also yields many novel clustering problems, and it leads to a numerical method for solving them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Sparse approximation problems have two defining characteristics:

1. An input signal is approximated by a linear combination of elementary signals. In many modern applications, the elementary signals are drawn from a large, linearly dependent collection.

2. A preference for "sparse" linear combinations is imposed by penalizing nonzero coefficients. The most common penalty is the number of elementary signals that participate in the approximation.

The problem domain must justify the linear model, the choice of elementary signals, and the sparsity criterion.

Sparse approximation has been studied for nearly a century, and it has numerous applications. Temlyakov [104] locates the first example in a 1907 paper of Schmidt [93]. In the 1950s, statisticians launched an extensive investigation of another sparse approximation problem called subset selection [75]. Later, approximation theorists began a systematic study of m-term approximation with respect to orthonormal bases and redundant systems [25, 104]. Over the last decade, the signal processing community—spurred by the work of Coifman et al. [13, 14] and Mallat et al. [72, 23, 22]—has become interested in sparse representations for compression and analysis of audio [52], images

[40], and video [77]. Sparsity criteria also arise in deconvolution [103], signal modeling [87], pre-conditioning [56], machine learning [49], de-noising [10], and regularization [21].

Most sparse approximation problems employ a linear model in which the collection of elementary signals is both linearly dependent and large. These models are often called redundant or overcomplete. Recent research suggests that overcomplete models offer a genuine increase in approximation power [85, 39]. Unfortunately, they also raise a serious challenge. How do we find a good representation of the input signal among the plethora of possibilities? One method is to select a parsimonious or sparse representation. The exact rationale for invoking sparsity may range from engineering to economics to philosophy. Three common justifications:

1. It is sometimes known *a priori* that the input signal can be expressed as a short linear combination of elementary signals that has been contaminated with noise.

2. The approximation may have an associated cost that must be controlled. For example, the computational cost of evaluating the approximation depends on the number of elementary signals that participate. In compression, the goal is to minimize the number of bits required to store the approximation.

3. Some researchers cite Occam's Razor, *"Pluralitas non est ponenda sine necessitate."* Causes must not be multiplied beyond necessity.[1]

[1]Beware! The antiquity of Occam's Razor guarantees neither its accuracy nor its applicability [28].

In short, sparse approximation problems arise in many applications for many reasons. It is therefore surprising how few theoretical results on these problems are available. This dissertation attempts to improve the situation. Let us present a detailed outline of the work, along with a summary of our contributions.

Chapter 2 offers a rigorous introduction to one important class of sparse approximation problems. It describes a general class of linear models, called *dictionaries*, and it shows how to parameterize approximations in these models. This leads to a natural method for measuring the sparsity of an approximation. Then, we discuss four major sparse approximation problems, which request different compromises between the error in making the approximation and the sparsity of the approximation.

When the dictionary is simple enough (more precisely, orthonormal), sparse approximation problems can be solved efficiently with basic algorithms. But the problems become computationally difficult in general because sparsity is a discontinuous, nonconvex function. We review a result from the literature which shows that sparse approximation is NP-hard when the linear model and the input signal are unrestricted [76, 22]. From this discussion, we gain the insight that sparse approximation problems may still be tractable if the dictionary is sufficiently close to orthonormal. A large part of the dissertation can be interpreted as a rigorous justification of this claim.

In Chapter 3, we present some of the basic numerical approaches to sparse approximation. Let us introduce the two most common types of heuristic.

1. Greedy methods make a sequence of locally optimal choices in an effort to produce a good global solution to the approximation problem.

This category includes forward selection procedures (such as matching pursuits), backward elimination, and others [75, 104].

2. The convex relaxation approach replaces the nonconvex sparsity measure with a related convex function to obtain a convex optimization problem. The convex program can be solved in polynomial time with standard software, and one hopes that it will yield a good sparse approximation [105, 10].

In addition, researchers have published a vast array of other heuristic methods. We do not refer to these approaches as "algorithms" because the literature contains virtually no proof that any of these numerical methods can solve a well-defined sparse approximation problem in full or in part.

In Chapter 4, we begin to develop the basic tools that are used to analyze greedy methods and convex relaxation methods. In particular, we introduce the *coherence parameter*, which quantifies how far a dictionary deviates from orthonormal. The coherence parameter can also be interpreted as a measure of how well the elements of a dictionary are dispersed in a certain projective space. We present several common dictionaries that have very low coherence, i.e., are nearly orthonormal. These examples emphasize the important point that these *incoherent* dictionaries may contain far more elements than an orthonormal basis for the same Euclidean space.

We also discuss some other geometric quantities associated with the dictionary. The most important is the *Exact Recovery Coefficient* associated with a sub-collection of elementary signals. This number reflects the difficulty of computing sparse approximations that involve these elementary signals. The Exact Recovery Coefficient has a somewhat complicated definition, but it is

possible to bound it in terms of the coherence parameter. One of our contributions is to show that the Exact Recovery Coefficient provides a natural sufficient condition for both greedy methods and convex relaxation methods to solve certain sparse approximation problems correctly. It is remarkable that the same quantity arises in the analysis of two very different algorithms.

Chapter 5 uses the tools from Chapter 4 to develop sufficient conditions for a particular greedy method to solve several different sparse approximation problems. The proof demonstrates that a locally optimal greedy step can identify elementary signals from the globally optimal sparse approximation of an input signal. This argument requires hypotheses on an Exact Recovery Coefficient. This work is fundamental because it isolates for the first time a sharp condition which ensures that greedy choices are globally optimal. An informal corollary of the argument is that greedy methods for sparse approximation succeed whenever the dictionary is incoherent.

Chapter 6 analyzes the convex relaxations of several sparse approximation problems. Once again, we will see that the Exact Recovery Coefficient plays a fundamental role in determining when convex relaxation succeeds. An informal corollary is that convex relaxation methods also work well when the dictionary is incoherent. Our proofs unify and extend most of the recent results on a convex relaxation method known as Basis Pursuit. Moreover, we close a serious gap in the literature by demonstrating that convex relaxation methods can succeed even when the optimal sparse approximation has a nonzero error.

Chapters 5 and 6 demonstrate that sparse approximation problems are computationally tractable when the dictionary is incoherent. This observation raises the question of how to construct incoherent dictionaries. We may rephrase this question as a geometric feasibility problem: How do we arrange

a fixed number of points in a projective space so that the closest pair of points is at least a specified distance apart?

Chapter 7 develops a numerical approach to this feasibility problem. To solve the problem, we show that it is both necessary and sufficient to construct a matrix that satisfies a spectral constraint and a structural constraint. Our procedure alternately enforces these two constraints in hope of reaching a matrix that satisfies both. This algorithm frequently yields results that are comparable with the best recorded solutions to the feasibility problem. A similar approach can also be used to attack related feasibility problems, many of which have not been studied numerically.

Finally, Chapter 8 presents a new connection between sparse approximation problems and data clustering problems. The basic insight is that clustering is a sparse approximation problem in which the linear model must be learned along with sparse approximations of the input data. We show that clustering can be recast as a low-rank matrix approximation problem with sparsity constraints. By varying the constraints, we can recover many of the clustering problems that have appeared in the literature. Moreover, this perspective yields a simple algorithm that can be modified to approach any one of these clustering problems.

Chapter 2

Sparse Approximation Problems

This chapter makes rigorous the notion of a sparse approximation problem. As discussed in the introduction, these problems request an approximation of a target signal using a linear combination of elementary signals drawn from a large collection. The goal is to achieve some compromise between the error in approximation and the cost in approximation, which is measured as the number of elementary signals that participate in the approximation.

The first section provides the basic definitions and notation that underlie our treatment of sparse approximation. It discusses the ambient space in which the approximation is performed and the means of measuring the approximation error. It gives a description of the linear model, and it explains precisely how we measure the cost of an approximation in this model.

The second section introduces four sparse approximation problems, which manage different tradeoffs between error and cost. It discusses the basic characteristics of these problems and some of the applications in which they arise. These first two sections are essential background for the rest of the dissertation.

The last section reviews some known results on the computational complexity of sparse approximation. We discover that certain linear models lead to sparse approximation problems that can be solved with a time cost that is linear in the size of the input. On the other hand, at least one type of linear model yields a sparse approximation problem that is NP-hard. The contrast

between these two cases leads directly to our major research problem. When and how can sparse approximation problems be solved efficiently?

2.1 Mathematical Setting

This section contains the basic definitions and notations that we use throughout the dissertation. We begin by describing the space from which signals are drawn and how we measure the error in approximating a target signal. Then we present a linear model for signals in this space, and we discuss how the model is parameterized. This leads to a natural measure of the cost of an approximation. With this background, we will be ready to present formal statements of the problems that we consider throughout the dissertation.

2.1.1 Signal Space

We work in the finite-dimensional, complex inner-product space $\mathbb{C}^d$, which is called the *signal space*. Elements of the signal space will generally be referred to as *signals*. The usual Hermitian inner product is written as $\langle \cdot, \cdot \rangle$, and we denote the corresponding Euclidean norm by $\|\cdot\|_2$. Since we are working in an inner-product space, the distance between two signals is the Euclidean norm of their difference. Although we could also study sparse approximation in other normed spaces, we leave this topic for future research.

Given an input signal (also called a *target signal*), an *approximation problem* elicits the nearest signal that satisfies some additional constraint. In classical problems, the approximant is often drawn from a subspace or a convex subset. In constrast, sparse approximation problems involve highly nonlinear constraints that are related to the cost of the approximation.

The decision to work in a finite-dimensional space deserves justification. Although infinite-dimensional spaces model some applications more accurately, computations always involve finite-dimensional approximations. Therefore, our theory corresponds with the numerical problems that one actually solves. Another advantage is that we avoid technical complications that might distract us from the essence of sparse approximation.

2.1.2 The Dictionary

A *dictionary* for the signal space is a finite collection $\mathscr{D}$ of unit-norm elementary signals. The elementary signals in $\mathscr{D}$ will usually be called *atoms*, and each atom is denoted by $\boldsymbol{\varphi}_\omega$, where the parameter ω is drawn from an index set Ω. The indices may have an interpretation, such as the time–frequency or time–scale localization of an atom, or they may simply be labels without an underlying metaphysics. The whole dictionary structure is thus

$$\mathscr{D} \quad = \quad \{\boldsymbol{\varphi}_\omega \ : \ \omega \in \Omega\}\,.$$

The letter N will denote the number of atoms in the dictionary. We have $N = |\mathscr{D}| = |\Omega|$, where $|\cdot|$ denotes the cardinality of a set.

If the dictionary spans the signal space, then we say that the dictionary is *complete* or *total*. In this case, every signal can be approximated with zero error using a linear combination of atoms. If the atoms form a linearly dependent set, then the dictionary is *redundant*. In this case, every signal has an infinite number of best approximations. For a dictionary to be complete, it is necessary that $N \geq d$. For a dictionary to be redundant, it is sufficient that $N > d$. In many modern applications, the dictionary is both complete and redundant.

2.1.3 Coefficient Space

A *representation* of a signal is a linear combination of atoms that equals the signal. Every representation is parameterized by a list of coefficients that we collect into a *coefficient vector*, which formally belongs to $\mathbb{C}^\Omega$. In case this notation is unfamiliar, $\mathbb{C}^\Omega$ is the set of all functions from Ω into $\mathbb{C}$. This set is made into a linear space with the standard definitions of addition and multiplication by scalars. The canonical basis for this space is given by the vectors whose coordinate projections are identically zero, except for a single unit component.

If $\boldsymbol{c}$ is a coefficient vector, its ω-th component will be denoted with a subscript as c_ω or in functional notation as $\boldsymbol{c}(\omega)$. We will alternate freely between these notations, depending on which is more typographically felicitous.

The *support* of a coefficient vector is the set of indices at which it is nonzero:

$$\operatorname{supp}(\boldsymbol{c}) \quad \stackrel{\text{def}}{=} \quad \{\omega \in \Omega \ : \ c_\omega \neq 0\}. \tag{2.1}$$

Suppose that $\Lambda \subset \Omega$. Without notice, we may embed "short" coefficient vectors from $\mathbb{C}^\Lambda$ into $\mathbb{C}^\Omega$ by extending them with zeros. Likewise, we may restrict long coefficient vectors from $\mathbb{C}^\Omega$ to their support. Both transformations will be natural in context.

2.1.4 Sparsity and Diversity

A sparse approximation problem seeks an approximation that can be represented with a low cost. In this dissertation, we will measure the cost of a representation as the number of atoms that participate. To make this idea rigorous, define the *sparsity* of a coefficient vector to be the number of places

where it equals zero. The complementary notion, *diversity*, counts the number of places where the coefficient vector does not equal zero. Diversity is calculated with the ℓ_0 *quasi-norm* $\|\cdot\|_0$, which is defined as

$$\|\boldsymbol{c}\|_0 \stackrel{\text{def}}{=} |\text{supp}(\boldsymbol{c})| . \tag{2.2}$$

For any positive number p, define

$$\|\boldsymbol{c}\|_p \stackrel{\text{def}}{=} \left[\sum\nolimits_{\omega\in\Omega} |c_\omega|^p\right]^{1/p} \tag{2.3}$$

with the convention that $\|\boldsymbol{c}\|_\infty \stackrel{\text{def}}{=} \max_{\omega\in\Omega} |c_\omega|$. As one might expect, there is an intimate connection between the definitions (2.2) and (2.3). Indeed, $\|\boldsymbol{c}\|_0 = \lim_{p\downarrow 0} \|\boldsymbol{c}\|_p^p$. It is well known that the function (2.3) is convex if and only if $1 \le p \le \infty$, in which case it describes the ℓ_p norm.

2.1.5 Other Cost Functions

The technical parts of this dissertation will study sparse approximation problems that compute the cost of a coefficient vector by means of the ℓ_0 quasi-norm. These problems are both difficult and fundamental. Nevertheless, the ℓ_0 quasi-norm is not appropriate for every application, and other cost functions also lead to sparse approximations. It is worth spending a moment here to touch on other possibilities.

To promote sparsity, a cost function ought to have two qualities. It should not charge for zero coefficients, and it should charge proportionately more for small coefficients than for large coefficients. Gribonval and Nielsen have studied a class of functions that have exactly these properties [53]. Suppose that f is a function from $[0, \infty)$ to $[0, \infty)$ for which

1. $f(0) = 0$ and $f(1) = 1$,

2. f is nondecreasing, and

3. the function $c \mapsto f(c)/c$ is nonincreasing on $(0, \infty)$.

Requirement 3 implies that f is sub-additive. That is, $f(b+c) \leq f(b) + f(c)$ for all nonnegative numbers b and c. To see this, just add the inequalities

$$\frac{b\, f(b+c)}{b+c} \leq f(b) \qquad \text{and} \qquad \frac{c\, f(b+c)}{b+c} \leq f(c).$$

Given a function f that satisfies Requirements 1–3, we may define a *cost function* $\text{cost}_f(\cdot)$, which maps coefficient vectors to nonnegative numbers by the formula

$$\text{cost}_f(\boldsymbol{c}) \quad \stackrel{\text{def}}{=} \quad \sum\nolimits_{\omega \in \Omega} f(|c_\omega|).$$

The class of all cost functions is a convex set, and every cost function is sub-additive. For more details, refer to the report [53]. Note that the normalization $f(1) = 1$ was not part of the original definition.

Let us mention a few examples. The ℓ_0 quasi-norm and the ℓ_1 norm are both cost functions. Another important cost function is

$$\text{cost}(\boldsymbol{c}) \quad = \quad \sum\nolimits_{\omega \in \Omega} \log_2 (|c_\omega| + 1).$$

This function is roughly proportional to the number of bits necessary to represent the coefficient vector at a fixed precision [19, Chapter 7]. As such, it is the natural cost function to use for data compression. On the other hand, observe that the squared ℓ_2 norm is not a cost function because the function $c \mapsto c^2$ fails Requirement 3.

An important conceptual point is that the ℓ_1 norm is the distinguished example a convex cost function.

Proposition 2.1. *The ℓ_1 norm is the only cost function that is also convex.*

Proof. Suppose that f is convex and satisfies Requirements 1–3 above. For every nonnegative c and for every t in the interval $(0, 1]$, the convexity of f yields the inequality

$$f(tc) \quad \leq \quad t\, f(c) + (1-t)\, f(0) \quad = \quad t\, f(c).$$

On the other hand, Requirement 3 implies that

$$\frac{f(tc)}{t\, c} \quad \geq \quad \frac{f(c)}{c}.$$

These inequalities together with Requirement 1 deliver the relation

$$f(tc) \quad = \quad t\, f(c) \qquad \text{for all } c \geq 0 \text{ and } t \in [0, 1].$$

Since $f(1) = 1$, we set $c = 1$ to see that $f(t) = t$ for all $t \in [0, 1]$. By the same token, we can also write

$$1 \quad = \quad f(t/t) \quad = \quad t\, f(1/t)$$

to discover that $f(1/t) = 1/t$ for all $t \in (0, 1]$.

In conclusion, $f(c) = c$ for all nonnegative c. Therefore, the associated cost function $\mathrm{cost}_f(\cdot)$ must coincide with the ℓ_1 norm. □

One interpretation of this result is that the ℓ_1 norm is the natural convexification of any cost function.

2.1.6 Synthesis and Analysis Matrices

Fix a dictionary $\mathscr{D} = \{\boldsymbol{\varphi_\omega} : \omega \in \Omega\}$. Although one could use summations to express linear combinations of atoms from $\mathscr{D}$, that notation muddies the water. Instead, let us define a matrix $\boldsymbol{\Phi}$, called the *dictionary synthesis matrix*, that maps coefficient vectors to signals. Formally,

$$\boldsymbol{\Phi} : \mathbb{C}^\Omega \longrightarrow \mathbb{C}^d \qquad \text{by the rule} \qquad \boldsymbol{\Phi} : \boldsymbol{c} \longmapsto \sum\nolimits_{\omega \in \Omega} c_\omega \, \boldsymbol{\varphi}_\omega.$$

The matrix $\boldsymbol{\Phi}$ describes the action of this linear transformation in the canonical bases of the underlying vector spaces. Therefore, the columns of $\boldsymbol{\Phi}$ are the atoms. We will often treat the dictionary and the dictionary synthesis matrix interchangeably.

The conjugate transpose of $\boldsymbol{\Phi}$ is called the *dictionary analysis matrix*, and it maps each signal to a coefficient vector that lists the inner products between signal and atoms.

$$\boldsymbol{\Phi}^* : \mathbb{C}^d \longrightarrow \mathbb{C}^\Omega \qquad \text{by the rule} \qquad (\boldsymbol{\Phi}^* \, \boldsymbol{s})(\omega) \quad = \quad \langle \boldsymbol{s}, \boldsymbol{\varphi}_\omega \rangle \, .$$

If the matrix $\boldsymbol{\Phi}^*$ is expressed with respect to the canonical bases of the underlying vectors spaces, then its rows are atoms, conjugate-transposed.

2.2 Formal Problem Statements

We will consider four basic sparse approximation problems, which manage different compromises between the error in approximation and the cost of representing the approximation. The problems, briefly:

1. Find the sparsest representation of the target signal.

2. Given a target signal, find the sparsest coefficient vector that represents an approximation with a prescribed error tolerance.

3. From all coefficient vectors with a prescribed level of sparsity, find one that yields the best approximation of the target signal.

4. Given a target signal, find a coefficient vector that balances the sparsity and approximation error.

2.2.1 The Sparsest Representation of a Signal

The most basic problem is to find the sparsest representation of a target signal $\boldsymbol{s}$. This question may be phrased as

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{c}\|_0 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\,\boldsymbol{c} = \boldsymbol{s}. \tag{EXACT}$$

Throughout the dissertation, we will always use the label (EXACT) to refer to this mathematical program. Note that the solution of the optimization problem is a coefficient vector, not a signal. If $\boldsymbol{c}_{\text{opt}}$ is a coefficient vector that solves (EXACT), the atoms indexed in $\operatorname{supp}(\boldsymbol{c}_{\text{opt}})$ must be linearly independent, or else some could be discarded to improve the sparsity of the representation.

If the dictionary is complete, then every signal has a representation using d atoms. On the other hand, essentially all signals require fully d atoms on account of the following result.

Proposition 2.2. *If $m < d$, the collection of signals that have a representation using m atoms forms a set of Lebesgue measure zero in $\mathbb{C}^d$.*

Proof. The signals that lie in the span of m fixed atoms form an m-dimensional subspace of $\mathbb{C}^d$, which has measure zero. There are $\binom{N}{m}$ ways to choose m

atoms, so the collection of signals that have a representation over m atoms is a finite union of m-dimensional subspaces. This union has measure zero in $\mathbb{C}^d$. $\square$

In spite of this obvious fact, the bulk of the recent literature on sparse approximation has focused on the case where the input signal has an exact sparse representation [31, 35, 54, 43, 29, 53].

Even though the problem (EXACT) is somewhat academic, it still rewards study. The primary justification is that our analysis of algorithms for other sparse approximation problems ultimately rests on results for (EXACT). Second, the analysis of the simpler problem can provide lower bounds on the computational complexity of more general sparse approximation problems. Finally, even though Proposition 2.2 shows that natural signals are not perfectly sparse, one can imagine applications in which a sparse signal is constructed and transmitted without error. This situation is modeled by (EXACT).

2.2.2 Error-Constrained Approximation

Let us continue with an immediate generalization of (EXACT). Instead of seeking the sparsest exact representation of a signal, we seek the sparsest representation that achieves a prescribed approximation error. This type of challenge arises in numerical analysis, where a common problem is to approximate or interpolate a complicated function using a short linear combination of more elementary functions. The approximation must not commit too great an error. At the same time, one pays for each additional term in the linear combination whenever the approximation is evaluated [76].

To state the problem formally, suppose that $\boldsymbol{s}$ is an arbitrary input signal,

and fix an error tolerance ε. We wish to solve the optimization problem

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{c}\|_0 \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \leq \varepsilon. \tag{2.4}$$

The solution of the problem is a coefficient vector, say $\boldsymbol{c}_{\text{opt}}$. The corresponding approximation of the target signal is given by $\boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}}$. Note that the support of $\boldsymbol{c}_{\text{opt}}$ must index a linearly independent collection of atoms, or else some could be discarded to increase the sparsity of the solution. It should also be clear that (EXACT) arises from (2.4) by setting the tolerance ε to zero.

Another important point is that the solutions of (2.4) will generally form a union of convex sets. Each minimizer will have the same level of sparsity, but they will yield different approximation errors. One may remove some of this multiplicity by considering the more convoluted mathematical program

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{c}\|_0 + \tfrac{1}{2}\,\|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2\,\varepsilon^{-1} \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \leq \varepsilon. \quad \text{(ERROR)}$$

Any minimizer of (ERROR) also solves (2.4), but it produces the smallest approximation error possible at that level of sparsity. Observe that, when ε reaches the norm of the input signal, the unique solution to either (2.4) or (ERROR) is the zero vector. On the other hand, as ε approaches zero, solutions will involve as many atoms as necessary to represent the signal exactly. Proposition 2.2 warns that essentially every signal in $\mathbb{C}^d$ will require d atoms.

2.2.3 Sparsity-Constrained Approximation

Approximation theorists prefer another flavor of the sparse approximation problem, called *m-term approximation*. We refer to this problem as *sparsity-constrained approximation*. In this case, one is asked to provide the best

approximation of a signal using a linear combination of m atoms or fewer from the dictionary. Formally,

$$\min_{\boldsymbol{c} \in \mathbb{C}^{\Omega}} \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \qquad \text{subject to} \qquad \|\boldsymbol{c}\|_0 \;\leq\; m. \tag{SPARSE}$$

Provided that the input signal has no representation using fewer than m atoms, the support of a solution must index a linearly independent collection of atoms. Of course, the optimal approximation error must decline as the number of atoms m increases.

Typically, approximation theorists study this approximation problem in the infinite-dimensional setting. They restrict their attention to functions in some smoothness class, and then they bound the rate at which the optimal approximation error declines as the number of atoms in the approximation increases. See [25, 104] for an introduction to this literature.

2.2.4 The Subset Selection Problem

Statisticians often wish to predict the value of one random variable using a linear combination of other random variables. At the same time, they must negotiate a compromise between the number of variables involved and the mean squared prediction error to avoid overfitting. The problem of determining the correct variables is called *subset selection*, and it was probably the first type of sparse approximation to be studied in depth. As Miller laments, statisticians have made limited theoretical progress due to numerous complications that arise in the stochastic setting [75].

We will consider a deterministic version of subset selection that manages a simple tradeoff between the squared approximation error and the number of atoms that participate. Let $\boldsymbol{s}$ be an arbitrary input signal; we will not assume

that it has any particular structure nor that is is drawn from a probability distribution. Suppose that τ is a threshold that quantifies how much improvement in the approximation error is necessary before we admit an additional term into the approximation. We may state the formal problem

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2^2 \;+\; \tau^2 \;\|\boldsymbol{c}\|_0 \,. \tag{SUBSET}$$

The support of a solution must index a linearly independent collection of atoms. When τ reaches the norm of the input signal, the zero vector is the unique solution of (SUBSET). On the other hand, as τ approaches zero, solutions will involve as many atoms as it takes to represent the signal exactly.

2.3 Computational Complexity

The computational complexity of sparse approximation has not been studied in great detail. It will be valuable, however, to set forth what is known. We will follow the literature by studying the sparsity-constrained problem

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \qquad \text{subject to} \qquad \|\boldsymbol{c}\|_0 \leq m. \tag{SPARSE}$$

In short, we wish to obtain the best Euclidean approximation of the input signal using at most m atoms. If the support of the optimal coefficient vector were known, the problem would fall to the usual least squares methods [51]. Selecting the optimal support, however, is nominally a combinatorial problem. The naïve strategy would sift through all $\binom{N}{m}$ possibilities.

As we will see, it is true that (SPARSE) is NP-hard in general. Nevertheless, it is quite easy to solve (SPARSE) when the dictionary is orthonormal.

2.3.1 Orthonormal Dictionaries

Let $\mathscr{D}$ be an orthonormal dictionary for the signal space $\mathbb{C}^d$, and suppose that $|\mathscr{D}| = N$. Given an input signal $\boldsymbol{s}$, consider the orthogonal series

$$\sum\nolimits_{\omega\in\Omega} \langle \boldsymbol{s}, \boldsymbol{\varphi}_\omega \rangle \, \boldsymbol{\varphi}_\omega.$$

If we sort the terms so that the inner products are nonincreasing in magnitude, then we may truncate the series after m terms to obtain an optimal m-term approximation of the input signal. The coefficients in the representation of this approximation are just the inner products that appear in these m terms.

This procedure could be implemented at a total time cost of $O(d\,N)$, which is how long it takes to compute all the inner products between the signal and the dictionary. In certain cases, the procedure can be implemented even more efficiently. If, for example, the dictionary synthesis matrix is the discrete Fourier transform, then all d inner products between the signal and the dictionary can be calculated in time $O(d \log d)$.

This discussion suggests a powerful (and accurate) heuristic. Sparse approximation is easy when the atoms in the dictionary are nearly orthogonal. Most of this dissertation can be interpreted as a formal justification of this intuition.

2.3.2 General Dictionaries

On the other hand, (SPARSE) is NP-hard in general. This important result was developed independently by Natarajan [76] and Davis et al. [22].

To state their result, let us describe a computational problem called *Exact Cover by 3-Sets*, which is usually abbreviated X3C. An instance of X3C consists of

- a finite universe $\mathscr{U}$ and
- a collection $\mathscr{X}$ of subsets $X_1, \dots, X_N$ such that $|X_n| = 3$ for each n.

The problem asks whether $\mathscr{X}$ contains a disjoint collection subsets whose union equals $\mathscr{U}$.

Proposition 2.3. *Any instance of Exact Cover by 3-Sets is reducible in polynomial time to the sparse approximation problem* (SPARSE).

Proof. The reduction is straightforward. Let the index set $\Omega = \{1, 2, \dots, N\}$. Choose the n-th atom to be the indicator vector of the subset X_n. That is, we set $\boldsymbol{\varphi}_n(u) = 1$ when $u \in X_n$ and $\boldsymbol{\varphi}_n(u) = 0$ otherwise. Select the target signal $\boldsymbol{s}$ to be $\mathbf{e}$, the vector of ones, and choose $m = \frac{1}{3}|\mathscr{U}|$. To determine whether the instance of Exact Cover by 3-Sets has a solution, we claim that it is sufficient to check whether a solution of the corresponding sparse approximation problem achieves a zero error.

Suppose that the instance of X3C has an affirmative solution using subsets indexed by the set Λ. Choose $\boldsymbol{c}_{\text{opt}}$ to be the indicator vector of Λ. That is, $\boldsymbol{c}_{\text{opt}}(n) = 1$ for $n \in \Lambda$ and zero otherwise. It follows immediately that $\boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}} = \mathbf{e}$, so the sparse approximation problem has a solution with zero error.

Conversely, suppose that $\boldsymbol{c}_{\text{opt}}$ is an optimal solution of (SPARSE) and that the corresponding approximation $\boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}}$ equals $\mathbf{e}$. Since $\boldsymbol{c}_{\text{opt}}$ contains no more than $\frac{1}{3}|\mathscr{U}|$ nonzero entries and each column of the synthesis matrix $\boldsymbol{\Phi}$ contains exactly three unit entries, it follows that $\{X_n : n \in \operatorname{supp}(\boldsymbol{c}_{\text{opt}})\}$ must be a disjoint collection of subsets that covers $\mathscr{U}$. □

Exact Cover by 3-Sets is a classic NP-complete problem [45]. We therefore reach the advertised result.

Corollary 2.4 (Complexity of Sparse Approximation). *If the dictionary and target signal are unrestricted, then the sparse approximation problem* (SPARSE) *is NP-hard.*

This proof can be adapted to show that the error-constrained sparse approximation problem (ERROR) and the subset selection problem (SUBSET) are NP-hard in general. Using a corollary of this argument, Davis et al. [22] demonstrated that it is NP-hard just to approximate the solution of (SPARSE).

It is useful to give the proof of the complexity result because it highlights that the difficult case has been manufactured artificially. Therefore, it is not quixotic to study sparse approximation problems that are more limited in scope. Indeed, this dissertation provides explicit sufficient conditions under which certain sparse approximation problems can be solved in polynomial time with simple algorithms.

Chapter 3

Numerical Methods for Sparse Approximation

Over the last fifty years, many different methods have been proposed to solve sparse approximation problems. The two most common approaches are greedy methods and convex relaxation methods.

A greedy method for sparse approximation constructs a sparse approximant one step at a time by selecting the atom most strongly correlated with the residual part of the signal and uses it to update the current approximation. The first section of this chapter presents two of the most prevalent greedy techniques, Matching Pursuit and Orthogonal Matching Pursuit. We discuss some of the basic attributes of these approaches, and we show how they should be modified to address different sparse approximation problems.

Convex relaxation replaces a combinatorial sparse approximation problems with a related convex program. The convex problem can be solved in polynomial time with standard software, and we hope that its solution will resemble the solution of the original sparse approximation problem. The second section presents the convex relaxations of several sparse approximation problems, and it discusses the basic attributes of these relaxations.

Unfortunately, the literature contains almost no theoretical guarantees that these numerical methods actually solve sparse approximation problems. The main contribution of this dissertation is to delineate circumstances in

which greedy methods and convex relaxation methods provably yield nearly optimal solutions to various sparse approximation problems.

In Section 3.3, we mention several other numerical approaches to sparse approximation problems. We will not analyze these other methods.

3.1 Greedy Methods

If the dictionary is orthonormal, we have seen that it is possible to solve (SPARSE) by choosing the atoms whose absolute inner products with the target signal are as large as possible. One way to accomplish this is to choose the atom most strongly correlated with the signal, subtract its contribution from the signal, and iterate. Greedy methods for sparse approximation refine this procedure so that it can be applied to more general dictionaries.

First, we present Matching Pursuit, which is a straightforward extension of the basic algorithm that succeeds for an orthonormal dictionary. Then, we develop Orthogonal Matching Pursuit, which adds a least-squares minimization to improve performance. Since these algorithms are iterative, one must supply a criterion for stopping the iteration. This criterion will depend on what sparse approximation problem we are trying to solve. The section concludes with a short discussion about the history of greedy methods.

3.1.1 Matching Pursuit

Let us begin with a formal statement of the Matching Pursuit (MP) procedure. Let us fix a dictionary $\mathscr{D}$ and a stopping criterion.

Algorithm 3.1 (Matching Pursuit).
INPUT:

- A d-dimensional target signal $\boldsymbol{s}$

OUTPUT:

- Returns a coefficient vector $\boldsymbol{c}$ in $\mathbb{C}^{\Omega}$

PROCEDURE:

1. Initialize the coefficient vector $\boldsymbol{c} \leftarrow \mathbf{0}$, the residual $\boldsymbol{r}_0 \leftarrow \boldsymbol{s}$, and the loop index $t = 1$.

2. Determine an index λ_t for which

$$\lambda_t \quad \in \quad \arg\max_{\omega} \left| \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\omega} \rangle \right| .$$

3. Update the coefficient vector:

$$\boldsymbol{c}(\lambda_t) \quad \leftarrow \quad \boldsymbol{c}(\lambda_t) + \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\lambda_t} \rangle .$$

4. Compute the new residual:

$$\boldsymbol{r}_t \quad \leftarrow \quad \boldsymbol{r}_{t-1} - \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\lambda_t} \rangle \ \boldsymbol{\varphi}_{\lambda_t} .$$

5. Increment the loop counter: $t \leftarrow t + 1$.

6. If the stopping criterion has not been met, return to Step 2.

Let us dissect this algorithm. Step 2 is the greedy selection, which chooses an atom that is most strongly correlated with the residual part of the signal. Note that MP may select the same index many times over when the dictionary is not orthogonal. This repetition occurs because the inner product between

an atom and the residual does not account for the contributions of other atoms to the residual. Step 3 updates the current coefficient vector to account for the effect of the atom λ_t. Step 4 computes a new residual by subtracting a component in the direction of the atom $\boldsymbol{\varphi}_{\lambda_t}$. If the dictionary is complete, it can be shown that the norm of the residual converges to zero as t approaches infinity [72]. At each step, the algorithm implicitly calculates a new approximant of the target signal. This approximant $\boldsymbol{a}_t$ satisfies the relationship

$$\boldsymbol{a}_t \quad = \quad \boldsymbol{s} - \boldsymbol{r}_t.$$

The loop repeats until the stopping criterion is satisfied.

Let us estimate the computational cost of this procedure. The greedy selection in Step 2 nominally involves computing all the inner products between the residual and the dictionary, which generally requires $O(d\,N)$ floating-point operations. For structured dictionaries, it may be possible to perform this calculation more efficiently. Steps 3 and 4 require only $O(d)$ floating-point operations. If the loop executes T times, then the cost of the algorithm is at most $O(d\,T\,N)$.

3.1.2 Orthogonal Matching Pursuit

In this dissertation, we will concentrate on an algorithm called Orthogonal Matching Pursuit (OMP), which adds a least-squares minimization to Algorithm 3.1 to obtain the best approximation over the atoms that have already been chosen. This revision significantly improves the behavior of the procedure.

We continue with a formal statement of the algorithm. Again, let us fix a dictionary and a stopping criterion.

Algorithm 3.2 (Orthogonal Matching Pursuit).

INPUT:

- A d-dimensional target signal $\boldsymbol{s}$

OUTPUT:

- A coefficient vector $\boldsymbol{c}$ in $\mathbb{C}^{\Omega}$

PROCEDURE:

1. Initialize the index set $\Lambda_0 = \emptyset$, the residual $\boldsymbol{r}_0 \leftarrow \boldsymbol{s}$, and the loop index $t \leftarrow 1$.

2. Determine an index λ_t for which

$$\lambda_t \quad \in \quad \arg\max_{\omega} \left| \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\omega} \rangle \right|.$$

3. Update the index set: $\Lambda_t \leftarrow \Lambda_{t-1} \cup \{\lambda_t\}$.

4. Find the solution $\boldsymbol{c}$ of the least-squares problem

$$\min_{\boldsymbol{c} \in \mathbb{C}^{\Lambda_t}} \left\| \boldsymbol{s} - \sum\nolimits_{j=1}^{t} \boldsymbol{c}(\lambda_j)\, \boldsymbol{\varphi}_{\lambda_j} \right\|_2.$$

5. Compute the new residual using the least-squares coefficients $\boldsymbol{c}$:

$$\boldsymbol{r}_t \quad \leftarrow \quad \boldsymbol{s} - \sum\nolimits_{j=1}^{t} \boldsymbol{c}(\lambda_j)\, \boldsymbol{\varphi}_{\lambda_j}.$$

6. Increment the loop counter: $t \leftarrow t + 1$.

7. If the stopping criterion has not been met, return to Step 2.

The anatomy of Orthogonal Matching Pursuit is similar to that of Matching Pursuit. Step 2 selects a new atom using the same greedy selection criterion as MP, and Step 3 adds its index to the list of atoms that have been chosen. The current coefficients are computed in Step 4 by solving a least squares problem, which is the essential difference between MP and OMP, and then Step 5 determines the new residual. The solution of the least-squares problem implicitly determines an approximant $\boldsymbol{a}_t$ of the target signal:

$$\boldsymbol{a}_t \quad = \quad \sum\nolimits_{j=1}^{t} \boldsymbol{c}(\lambda_j)\, \boldsymbol{\varphi}_{\lambda_j}.$$

The loop continues until the stopping criterion is satisfied.

The behavior of OMP differs significantly from that of MP. Observe that OMP maintains a loop invariant:

$$\left\langle \boldsymbol{r}_t, \boldsymbol{\varphi}_{\lambda_j} \right\rangle = 0 \qquad \text{for } j = 1, \dots, t.$$

It follows that Step 2 always selects an atom that is linearly independent from the atoms that have already been chosen. In consequence, the residual must equal zero after d steps. Moreover, the solution to the least-squares problem in Step 4 is always unique.

Let us estimate the time cost of this algorithm. Step 2 can always be implemented in $O(d\,N)$ time by computing all the inner products between the residual and the dictionary. At iteration t, solving the least-squares problem in Step 4 requires only $O(t\,d)$ time because we may build on the solution of the least-squares problem in iteration $(t-1)$. If no additional efficiencies are made, the time cost of the algorithm is $O(d\,T\,(T+N))$, where T is the total number of iterations. Step 2 will usually dominate the cost of this algorithm unless one is able to exploit some structure in the dictionary.

3.1.3 Stopping Criteria

To complete the statement of the two algorithms, we must provide a condition for determining when to halt the iteration. By varying the stopping criterion, we can tailor greedy methods for different sparse approximation problems. Here are three possibilities.

1. One may wait until the norm of the residual $\boldsymbol{r}_t$ equals zero. This criterion is appropriate for the problem of recovering a sparse input signal (EXACT).

2. One may halt the procedure when the norm of the residual $\boldsymbol{r}_t$ declines below a specified threshold. This criterion is appropriate for the error-constrained approximation problem (ERROR).

3. One may halt the procedure after m distinct atoms have been selected. Then the algorithm will return a coefficient vector that is supported on m indices. This method is appropriate for solving the sparsity-constrained approximation problem (SPARSE).

A natural stopping rule for the subset selection problem (SUBSET) is not immediately apparent.

Suppose that the dictionary is orthonormal. It is not hard to check that MP and OMP, equipped with these stopping rules, are both correct algorithms for the respective sparse approximation problems.

3.1.4 A Counterexample for MP

Nevertheless, there is no reason to expect that greedy methods will succeed in general. Indeed, the literature contains examples where MP and OMP

fail catastrophically even though the target signal has a very sparse representation over the dictionary [24, 10].

Suppose that $\mathscr{D}$ is an orthonormal dictionary for $\mathbb{C}^d$, where $d \geq 3$. We adjoin the atom

$$\boldsymbol{\psi} \quad = \quad \alpha \left[\boldsymbol{\varphi}_1 + \boldsymbol{\varphi_2} + \sum_{n=3}^{d} \frac{1}{n-2} \, \boldsymbol{\varphi}_n \right]$$

where α is chosen so that $\boldsymbol{\psi}$ has unit norm. If Matching Pursuit is executed with the input signal $\boldsymbol{s} = \boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2$, then the algorithm will continue forever with an approximation error

$$\|\boldsymbol{s} - \boldsymbol{a}_t\|_2 \quad \asymp \quad t^{-1/2}.$$

Yet the input signal clearly has a two-term representation. One may construct related examples in which Orthogonal Matching Pursuit requires fully d steps to reconstruct a signal that has a two-term representation.

3.1.5 History of Greedy Methods

Greedy algorithms for sparse approximation first appeared in the statistics literature sometime during the 1950s, where they were used for solving subset selection problems. The basic technique, called *forward selection*, is a more elaborate version of Matching Pursuit. Many other variations were subsequently developed including backward elimination and stepwise regression (also known as Efroymsen's algorithm). These algorithms are discussed at length in the monograph [75].

Algorithm 3.1 was invented in 1981 under the cognomen *Projection Pursuit Regression* [38]. This algorithm was introduced to the signal processing

literature by Mallat and Zhang, who renamed it Matching Pursuit [72]. Approximation theorists refer to the *Pure Greedy Algorithm* [104].

Orthogonal Matching Pursuit was developed independently by many researchers. The earliest reference appears to be a 1989 paper of Chen, Billings, and Luo [9]. The first signal processing papers on OMP arrived around 1993 [83, 23].

Statisticians invoke many elaborate stopping rules when they apply greedy methods for subset selection. The most famous is probably to examine the C_p statistic of Mallows [73]. For more details, see the monograph [75].

3.2 Convex Relaxation Methods

A common approach to solving a combinatorial problem is to replace it with a relaxed version that can be solved more efficiently. The presence of the ℓ_0 quasi-norm makes sparse approximation problems combinatorial in nature. Proposition 2.1 shows that the ℓ_1 norm provides a natural convex relaxation of the ℓ_0 quasi-norm, and it suggests that we may be able solve sparse approximation problems by introducing an ℓ_1 norm in place of the ℓ_0 quasi-norm.

This section presents the convex relaxations of several different sparse approximation problems, and it discusses their basic attributes. We conclude with a historical perspective on convex relaxation.

3.2.1 The Sparsest Representation of a Signal

Recall that the problem of determining the sparsest representation of an input signal is

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{c}\|_0 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\,\boldsymbol{c} \;=\; \boldsymbol{s}. \tag{EXACT}$$

According to our heuristic, the natural convex relaxation is

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\,\boldsymbol{b} \;=\; \boldsymbol{s}. \tag{R-EXACT}$$

Note that the convex relaxation is not an algorithm itself but another computational problem that must be solved. Since (R-EXACT) is a convex program, we may use standard mathematical programming software to compute a minimizer in polynomial time [5]. In this work, we will not discuss algorithms for solving convex optimization problems. Rather, we will concentrate on the relationship between the solution of the convex relaxation and the original sparse approximation problem.

Chen, Donoho, and Saunders were the first to propose that (R-EXACT), which they call *Basis Pursuit*, could be used to solve (EXACT). Their paper [10] offers copious numerical evidence that the relaxation succeeds, but it provides no rigorous proof. Subsequently, some theoretical results have been developed. We postpone this discussion until Section 6.1.

3.2.2 Error-Constrained Approximation

Given a signal $\boldsymbol{s}$ and an error tolerance ε, the error-constrained sparse approximation problem is

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{c}\|_0 \;+\; \tfrac{1}{2}\,\|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2\,\varepsilon^{-1} \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \;\leq\; \varepsilon. \tag{ERROR}$$

Its convex relaxation is

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \ \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2 \ \leq \ \delta. \tag{R-ERROR}$$

It might seem that the relaxation ought to include the approximation error in the objective function. This measure is unnecessary because it leads to a very similar optimization problem, as one may discover by inspecting the respective Lagrangian functions. Since (R-ERROR) requests the minimizer of a convex function over a convex set, we may apply standard mathematical programming software to solve it [5].

Observe that, as δ approaches zero, the solution of (R-ERROR) approaches the solution of (R-EXACT). Indeed, the case $\delta = 0$ reduces to the problem (R-EXACT). On the other hand, when δ exceeds the norm of the input signal, the unique solution of (R-ERROR) is the zero vector. In Chapter 6, we will develop theory that illuminates the correct relationship between δ and ε.

The relaxation (R-ERROR) was proposed and studied in [107]. Around the same time, [30] introduced the same relaxation independently. Otherwise, there are no theoretical results on the performance of this convex relaxation.

3.2.3 Subset Selection

Given a signal $\boldsymbol{s}$ and a threshold τ, the subset selection problem is

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2^2 \ + \ \tau^2 \ \|\boldsymbol{c}\|_0 \,. \tag{SUBSET}$$

For reasons that we will discuss below, the convex relaxation that we study has a form slightly different from the original problem. Our relaxation is

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \quad \tfrac{1}{2} \ \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2^2 \ + \ \gamma \ \|\boldsymbol{b}\|_1 \,. \tag{R-SUBSET}$$

Since the objective of (R-SUBSET) is an unconstrained convex function, we can use standard mathematical programming software to find a minimizer [5].

Let us take a moment to understand why the subset selection problem and its convex relaxation have slightly different structures. Suppose for a moment that the dictionary is orthonormal. Then one may solve the subset selection problem (SUBSET) by applying a hard threshold operator (see Figure 3.1) with cutoff τ to each coefficient in the orthogonal expansion of the signal [71]. In effect, one retains every atom whose inner product with the signal is larger than τ and discards the rest. On the other hand, one may solve the convex relaxation (R-SUBSET) by applying a soft threshold operator (see Figure 3.1) with cutoff γ to each coefficient in the orthogonal expansion of the signal [71]. This amounts to retaining every atom whose inner product with the signal is strictly greater than γ and discarding the rest. We see that the form of the relaxation has been adapted so that the parameter still determines the location of the cutoff (when the dictionary is orthonormal). In Chapter 6, we will unveil the correct relationship between γ and τ in the general case.

The reader should be aware that convex programs of the form (R-SUBSET) have been proposed for many different applications. Geophysicists have long used them for deconvolution [103, 92], and the statistics community uses (R-SUBSET) for linear regression problems [105]. Under mild assumptions, it can be shown that support vector machines, which are used for machine learning applications, solve the same optimization problem [49]. Chen, Donoho, and Saunders have applied (R-SUBSET) to de-noise signals [10], and Fuchs has put it forth for several other signal processing problems, e.g., in [41, 42]. Daubechies, Defrise, and De Mol have proposed a related convex program for regularizing linear inverse problems [21]. Most intriguing, perhaps, Olshausen

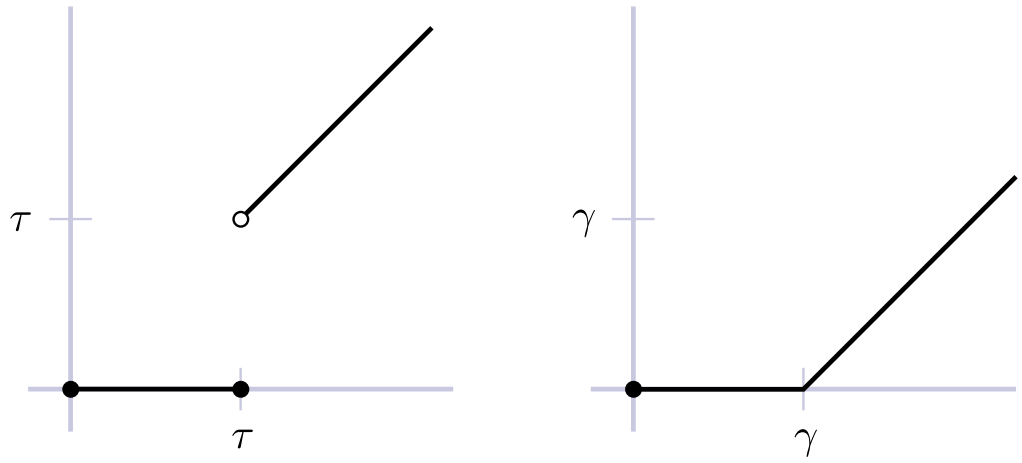


Figure 3.1: HARD AND SOFT THRESHOLDING. At left, the hard thresholding operator with cutoff τ. At right, the soft thresholding operator with cutoff γ.

and Field have argued that the mammalian visual cortex may solve similar minimization problems to produce sparse representations of images [79]. None of the papers we have just mentioned develops any correspondence between the solutions of (SUBSET) and (R-SUBSET).

3.2.4 Sparsity-Constrained Approximation

Convex relaxation does not seem to be an appropriate method for solving (SPARSE) because it provides no control on the number of terms involved in the approximation.

3.2.5 History of Convex Relaxation

The ascendance of convex relaxation for sparse approximation was propelled by two theoretical–technological developments of the last half century. First, the philosophy and methodology of robust statistics—developed by von

Neumann, Tukey and Huber—show that ℓ_1 loss criteria can be applied to defend statistical estimators against outlying data points. Robust estimators qualitatively prefer a few large errors and many tiny errors to the armada of moderate deviations introduced by mean-squared-error criteria. Second, the elevation during the 1950s of linear programming to the level of *technology* and the interior-point revolution of the 1980s have made it both tractable and commonplace to solve the large-scale optimization problems that arise from convex relaxation.

It appears that a 1973 paper of Claerbout and Muir is the crucible in which these reagents were first combined for the express goal of yielding a sparse representation [12]. They write,

> In deconvolving any observed seismic trace, it is rather disappointing to discover that there is a nonzero spike at every point in time regardless of the data sampling rate. One might hope to find spikes only where real geologic discontinuities take place. Perhaps the L_1 norm can be utilized to give a [sparse] output trace....

This idea was subsequently developed in the geophysics literature by [103, 67, 78]. In 1986, Santosa and Symes proposed the convex relaxation (R-SUBSET) as a method for recovering sparse spike trains, and they proved that the method succeeds under moderate restrictions [92].

Around 1990, the work on ℓ_1 criteria in signal processing recycled to the statistics community. Donoho and Johnstone wrote a pathbreaking paper [32] which proved that one could determine a nearly optimal minimax estimate of a smooth function contaminated with noise by solving the convex relaxation (R-SUBSET) where $\boldsymbol{\Phi}$ is an appropriate wavelet basis and γ is related to the

variance of the noise. Slightly later, Tibshirani proposed that (R-SUBSET), which he calls *the Lasso*, could be used to solve subset selection problems in the stochastic setting [105]. From here, it is only a short step to Basis Pursuit and Basis Pursuit de-noising [10].

This history could not be complete without mention of parallel developments in theoretical computer science. It has long been known that some combinatorial problems are intimately bound up with continuous convex programming problems. In particular, the problem of determining the maximum value that an affine function attains at some vertex of a polytope can be solved using a linear program [82]. A major theme in modern computer science is that many other combinatorial problems can be solved approximately by means of a convex relaxation. For example, a celebrated paper of Goemans and Williamson proves that a certain convex program can be used to produce a graph cut whose weight exceeds 87% of the maximum cut [50]. The present work draws deeply on the fundamental idea that a combinatorial problem and its convex relaxation often have closely related solutions.

3.3 Other Methods

There are at least three other approaches to solving sparse approximation problems. For reference, we offer a brief mention of these techniques. We will not discuss them again.

3.3.1 The Brute Force Approach

Brute force methods sift through all potential approximations to find the global optimum. Exhaustive searches quickly become intractable as the problem size grows, and more sophisticated techniques, such as branch-and-

bound, do not accelerate the hunt enough to be practical [75].

3.3.2 The Nonlinear Programming Approach

Some researchers have developed specialized nonlinear programming software that attempts to solve sparse approximation problems directly by such means as interior-point methods [86]. These techniques are only guaranteed to discover a locally optimal solution.

3.3.3 The Bayesian Approach

The Bayesian approach assumes that coefficients in the linear combination are random variables with a "sparse" prior distribution. Sometimes, the elementary signals are also treated as random variables with known prior distributions. Input signals are then used to estimate posterior probabilities, and the most likely models are selected for further attention. There do not seem to be any theoretical results for this paradigm [68, 91, 75].

Chapter 4

Geometry of Sparse Approximation

The goal of this chapter is to bring together all of the linear algebra, functional analysis, and discrete geometry that we will need to analyze sparse approximation algorithms. All the ideas we discuss have natural geometric interpretations. Some concepts will be familiar from elementary linear algebra, while others have arisen directly from the study of sparse approximation. Although this chapter may seem like a diversion, the material is important to develop an intuitive understanding of the subsequent chapters.

Most of the background material in this section may be traced to the usual suspects [61, 63, 51], which will not generally be cited in the text. Other references will be given explicitly, and the contributions of the author will be noted.

4.1 Sub-dictionaries

A linearly independent collection of atoms is called a *sub-dictionary*. If the atoms in a sub-dictionary are indexed by the set Λ, then we define a synthesis matrix $\boldsymbol{\Phi}_\Lambda : \mathbb{C}^\Lambda \to \mathbb{C}^d$ and an analysis matrix $\boldsymbol{\Phi}_\Lambda{}^* : \mathbb{C}^d \to \mathbb{C}^\Lambda$.

$$\boldsymbol{\Phi}_\Lambda : \boldsymbol{c} \longmapsto \sum\nolimits_{\lambda \in \Lambda} c_\lambda \, \boldsymbol{\varphi}_\lambda, \qquad \text{and} \qquad (\boldsymbol{\Phi}_\Lambda{}^* \, \boldsymbol{s})(\lambda) = \langle \boldsymbol{s}, \boldsymbol{\varphi}_\lambda \rangle \quad \text{for } \lambda \in \Lambda.$$

These matrices are entirely analogous with the dictionary synthesis and analysis matrices. We will frequently use the fact that the synthesis matrix $\boldsymbol{\Phi}_\Lambda$

has full column rank.

The *Gram matrix* of the sub-dictionary is given by $\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda$. Observe that the (λ, ω) entry of this matrix is the inner product $\langle \boldsymbol{\varphi}_\omega, \boldsymbol{\varphi}_\lambda \rangle$. Therefore, the Gram matrix is Hermitian, and it has a unit diagonal (since all atoms have unit Euclidean norm). One should interpret the Gram matrix as a table of the correlations between atoms listed by Λ. Note that the Gram matrix of a sub-dictionary is always invertible.

We will encounter two other matrices frequently enough to single them out. The *Moore–Penrose generalized inverse* of the synthesis matrix is denoted by $\boldsymbol{\Phi}_\Lambda{}^\dagger$, and it may be calculated using the formula $\boldsymbol{\Phi}_\Lambda{}^\dagger = (\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \boldsymbol{\Phi}_\Lambda{}^*$. For any signal $\boldsymbol{s}$, the coefficient vector $\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{s}$ synthesizes the best approximation of $\boldsymbol{s}$ using the atoms in Λ. The orthogonal projector that produces this best approximation will be denoted as P_Λ. This projector may be expressed using the generalized inverse: $\mathsf{P}_\Lambda = \boldsymbol{\Phi}_\Lambda \boldsymbol{\Phi}_\Lambda{}^\dagger$. Recall that $\mathsf{P}_\Lambda \boldsymbol{s}$ is always orthogonal to the residual $(\boldsymbol{s} - \mathsf{P}_\Lambda \boldsymbol{s})$.

4.2 Summarizing the Dictionary

To develop simple results for general dictionaries, we need a method for summarizing the behavior of the dictionary. This section describes an attractive approach based on the inner products between atoms. We also exhibit several different dictionaries and compute their summary parameters.

4.2.1 Coherence

The most fundamental quantity associated with a dictionary is the *coherence parameter* μ. It equals the maximum absolute inner product between

two distinct atoms:

$$\mu \quad \stackrel{\text{def}}{=} \quad \max_{j \neq k} \left| \left\langle \boldsymbol{\varphi}_{\omega_j}, \boldsymbol{\varphi}_{\omega_k} \right\rangle \right|.$$

Roughly speaking, this number measures how much two atoms can look alike. Coherence is a blunt instrument since it only reflects the most extreme correlations in the dictionary. Nevertheless, it is easy to calculate, and it captures well the behavior of some dictionaries. Informally, we say that a dictionary is *incoherent* when we judge that μ is small.

It is obvious that every orthonormal basis has coherence zero. A union of two orthonormal bases has coherence no smaller than $d^{-1/2}$ [54]. A dictionary of concatenated orthonormal bases is called a *multi-ONB*. For some d, it is possible to build a multi-ONB that contains d or even $(d+1)$ orthonormal bases yet retains the minimal possible coherence $d^{-1/2}$ [59]. Gilbert, Muthukrishnan, and Strauss have exhibited a method for constructing even larger dictionaries with slightly higher coherence [48]. For general dictionaries, a lower bound on the coherence is

$$\mu \quad \geq \quad \sqrt{\frac{N-d}{d\,(N-1)}}.$$

If each atomic inner product meets this bound, the dictionary is called an *equiangular tight frame*. See [100, 60, 101] for more details. A derivation of this bound appears in Section 7.5.

The coherence parameter of a dictionary was first mentioned as a quantity of heuristic interest in [22], but the first formal treatment appears in [31]. It is also related to an eponymous concept from the geometry of numbers [115].

4.2.2 Example: The Dirac–Fourier Dictionary

Consider the dictionary for $\mathbb{C}^d$ that has synthesis matrix

$$\boldsymbol{\Phi} \quad = \quad \left[\begin{array}{c|c} \mathsf{I}_d & \mathscr{F}_d \end{array} \right],$$

where I_d is the d-dimensional identity matrix and $\mathscr{F}_d$ is the d-dimensional discrete Fourier transform (DFT) matrix. For reference, the (j, k) entry of $\mathscr{F}_d$ is the complex number $\exp\{-2\pi \mathrm{i}\, jk/d\}/\sqrt{d}$.

This dictionary is called the Dirac–Fourier dictionary because it consists of impulses and discrete complex exponentials. In a d-dimensional signal space, it contains $2\,d$ atoms, and it forms a two-ONB because the identity matrix and the DFT matrix are both unitary. It is very easy to check that the coherence μ of the Dirac–Fourier dictionary is $1/\sqrt{d}$. Therefore, it has the smallest coherence possible for a multi-ONB.

4.2.3 Cumulative Coherence

We have introduced the coherence parameter μ because it is easy to calculate, and yet it can be used to bound much more complicated quantities associated with the dictionary. A refinement of the coherence parameter is the *cumulative coherence function*. It measures how much a collection of m atoms can resemble a fixed, distinct atom. Formally,

$$\mu_1(m) \quad \stackrel{\text{def}}{=} \quad \max_{|\Lambda| = m} \max_{\omega \notin \Lambda} \sum\nolimits_{\lambda \in \Lambda} \left| \langle \boldsymbol{\varphi}_\omega, \boldsymbol{\varphi}_\lambda \rangle \right|.$$

We place the convention that $\mu_1(0) = 0$. The subscript on μ_1 serves as a mnemonic that the cumulative coherence is an absolute sum, and it distinguishes the function μ_1 from the number μ. When the cumulative coherence

grows slowly, we say informally that the dictionary is *incoherent* or *quasi-incoherent*.

The cumulative coherence function has an important interpretation in terms of sub-dictionaries. Suppose that Λ indexes m atoms. Then the number $\mu_1(m-1)$ gives an upper bound on the sum of the absolute off-diagonal entries in each row (or column) of the Gram matrix $\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda$. Several other facts about μ_1 follow immediately from the definition.

Proposition 4.1. *The cumulative coherence has the following properties:*

1. *It generalizes the coherence:* $\mu_1(1) = \mu$ *and* $\mu_1(m) \leq m\,\mu$.
2. *Its first differences are nonnegative:*

$$\mu_1(m+1) - \mu_1(m) \quad \geq \quad 0 \qquad \text{for each } m \geq 0.$$

3. *Its second differences are nonpositive:*

$$\mu_1(m+2) - 2\,\mu_1(m+1) + \mu_1(m) \quad \leq \quad 0 \qquad \text{for each } m \geq 0.$$

4. *For an orthonormal basis,* $\mu_1(m) = 0$ *for each nonnegative* m.

The concept of cumulative coherence was developed independently in [29, 106]. In Section 4.10, we will suggest geometric interpretations of both the coherence parameter and the cumulative coherence function.

4.2.4 Example: Double Pulses

For a realistic dictionary where the atoms have analytic definitions, the cumulative coherence function is not too difficult to compute. We begin with

a simple example of a dictionary in the signal space $\mathbb{C}^d$. We will index the components of a signal vector from zero to $(d-1)$.

We construct d atoms, indexed from zero to $(d-1)$. For each index k, we define an atom

$$\boldsymbol{\varphi}_k(t) \quad = \quad \begin{cases} \sqrt{35}/6, & t = k \\ 1/6, & t \equiv k+1 \pmod{d} \\ 0, & \text{otherwise.} \end{cases}$$

Two atoms have a nonzero inner product if and only if their indices are adjacent:

$$\langle \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k \rangle \quad = \quad \begin{cases} 1, & j = k \\ \sqrt{35}/36, & j \equiv k-1 \pmod{d} \\ \sqrt{35}/36, & j \equiv k+1 \pmod{d} \\ 0, & \text{otherwise.} \end{cases}$$

It follows that the cumulative coherence function of this dictionary is

$$\mu_1(m) \quad = \quad \begin{cases} \sqrt{35}/36, & m = 1 \\ \sqrt{35}/18, & m \geq 2. \end{cases}$$

We see that the cumulative coherence is always less than a third, while the quantity $m\,\mu$ grows without bound.

4.2.5 Example: Decaying Atoms

To see a more complicated example, we consider a dictionary of exponentially decaying atoms. To streamline the calculations, we work in the infinite-dimensional Hilbert space ℓ_2 of square-summable, complex-valued sequences.

Fix a parameter $\beta < 1$. For each index $k \geq 0$, define an atom by

$$\boldsymbol{\varphi}_k(t) \quad = \quad \begin{cases} 0, & 0 \leq t < k \\ \beta^{t-k}\sqrt{1-\beta^2}, & k \leq t. \end{cases}$$

It can be shown that the atoms span ℓ_2, so they form a complete dictionary. The absolute inner product between two atoms is

$$|\langle \boldsymbol{\varphi}_k, \boldsymbol{\varphi}_j \rangle| \quad = \quad \beta^{|k-j|}.$$

In particular, each atom has unit norm. It also follows that the coherence of the dictionary equals β.

Here is the calculation of the cumulative coherence function in detail:

$$\begin{aligned} \mu_1(m) &= \max_{|\Lambda|=m} \max_{k \notin \Lambda} \sum_{j \in \Lambda} |\langle \boldsymbol{\varphi}_k, \boldsymbol{\varphi}_j \rangle| \\ &= \max_{|\Lambda|=m} \max_{k \notin \Lambda} \sum_{j \in \Lambda} \beta^{|k-j|}. \end{aligned}$$

The maximum occurs, for example, when $k = \lfloor \frac{m}{2} \rfloor$ and

$$\Lambda = \{0, 1, 2, \dots, \lfloor \tfrac{m}{2} \rfloor - 1, \lfloor \tfrac{m}{2} \rfloor + 1, \dots, m-1, m\}.$$

The exact form of the cumulative coherence function depends on the parity of m. For m even,

$$\mu_1(m) = \frac{2\beta\,(1 - \beta^{m/2})}{1 - \beta}$$

while for m odd,

$$\mu_1(m) = \frac{2\beta\,(1 - \beta^{(m-1)/2})}{1 - \beta} + \beta^{(m+1)/2}.$$

Notice that $\mu_1(m) < 2\beta/(1-\beta)$ for all m. On the other hand, the quantity $m\,\mu$ grows without bound.

4.3 Operator Norms

One of the most useful tools in our satchel is the operator norm. Let us treat the matrix A as a map between two finite-dimensional vector spaces equipped respectively with the ℓ_p and ℓ_q norms. The (p, q) *operator norm* of A measures the factor by which the matrix can increase the length of a vector. It may be calculated with any of the following expressions:

$$\|\mathsf{A}\|_{p,q} \stackrel{\text{def}}{=} \max_{\boldsymbol{z} \neq \mathbf{0}} \frac{\|\mathsf{A}\,\boldsymbol{z}\|_q}{\|\boldsymbol{z}\|_p} = \max_{\|\boldsymbol{z}\|_p = 1} \|\mathsf{A}\,\boldsymbol{z}\|_q = \max_{\|\boldsymbol{z}\|_p \leq 1} \|\mathsf{A}\,\boldsymbol{z}\|_q .$$

In words, the operator norm equals the maximum ℓ_q norm of any point in the image of the ℓ_p unit ball under A.

A quantity related to the operator norm is the restricted minimum

$$\min_{\substack{\boldsymbol{z} \in \mathscr{R}(\mathsf{A}^*) \\ \boldsymbol{z} \neq \mathbf{0}}} \frac{\|\mathsf{A}\,\boldsymbol{z}\|_q}{\|\boldsymbol{z}\|_p} \tag{4.1}$$

where $\mathscr{R}(\cdot)$ denotes the range (i.e., column span) of its argument. The expression (4.1) measures the factor by which the nonsingular part of A can decrease the length of a vector. If the matrix has full row-rank, we can express the minimum in terms of a generalized inverse.

Proposition 4.2. *The following bound holds for every matrix* A*.*

$$\min_{\substack{\boldsymbol{z} \in \mathscr{R}(\mathsf{A}^*) \\ \boldsymbol{z} \neq \mathbf{0}}} \frac{\|\mathsf{A}\,\boldsymbol{z}\|_q}{\|\boldsymbol{z}\|_p} \quge \quad \big\|\mathsf{A}^\dagger\big\|_{q,p}^{-1}. \tag{4.2}$$

If A *has full row-rank, equality holds in* (4.2)*. When* A *is invertible, this result implies*

$$\min_{\boldsymbol{z} \neq \mathbf{0}} \frac{\|\mathsf{A}\,\boldsymbol{z}\|_q}{\|\boldsymbol{z}\|_p} \quad = \quad \big\|\mathsf{A}^{-1}\big\|_{q,p}^{-1}.$$

Proof. First, observe that

$$\left[\min_{\substack{\boldsymbol{z} \in \mathscr{R}(\mathsf{A}^*) \\ \boldsymbol{z} \neq \mathbf{0}}} \frac{\|\mathsf{A}\,\boldsymbol{z}\|_q}{\|\boldsymbol{z}\|_p}\right]^{-1} \quad = \quad \max_{\substack{\boldsymbol{z} \in \mathscr{R}(\mathsf{A}^*) \\ \boldsymbol{z} \neq \mathbf{0}}} \frac{\|\boldsymbol{z}\|_p}{\|\mathsf{A}\,\boldsymbol{z}\|_q}.$$

Next, make the substitution $\boldsymbol{w} = \mathsf{A}\,\boldsymbol{z}$. The matrix $\mathsf{A}^\dagger\mathsf{A}$ is a projector onto the range of A^*, which implies that $\mathsf{A}^\dagger\,\boldsymbol{w} = \boldsymbol{z}$. As the vector $\boldsymbol{z}$ ranges over $\mathscr{R}(\mathsf{A}^*)$, the vector $\boldsymbol{w}$ ranges over all of $\mathscr{R}(\mathsf{A})$ because $\mathscr{R}(\mathsf{A}) = \mathscr{R}(\mathsf{A}\mathsf{A}^*)$. Thus,

$$\begin{aligned}\max_{\substack{\boldsymbol{z} \in \mathscr{R}(\mathsf{A}^*) \\ \boldsymbol{z} \neq \mathbf{0}}} \frac{\|\boldsymbol{z}\|_p}{\|\mathsf{A}\,\boldsymbol{z}\|_q} \quad &= \quad \max_{\substack{\boldsymbol{w} \in \mathscr{R}(\mathsf{A}) \\ \boldsymbol{w} \neq \mathbf{0}}} \frac{\big\|\mathsf{A}^\dagger\,\boldsymbol{w}\big\|_p}{\|\boldsymbol{w}\|_q} \\ &\leq \quad \max_{\boldsymbol{w} \neq \mathbf{0}} \frac{\big\|\mathsf{A}^\dagger\,\boldsymbol{w}\big\|_p}{\|\boldsymbol{w}\|_q}.\end{aligned}$$

When A has full row-rank, the last two maxima are taken over the same set, which converts the inequality to an equality. Complete the proof by identifying the last maximum as $\left\|A^\dagger\right\|_{q,p}$. □

The dual of the finite-dimensional normed linear space $(\mathbb{C}^m, \ell_p)$ is the space $(\mathbb{C}^m, \ell_{p'})$ where $1/p + 1/p' = 1$. In particular, ℓ_1 and ℓ_∞ are dual to each other, while ℓ_2 is self-dual. If the matrix A maps from ℓ_p to ℓ_q, its conjugate transpose A^* should be viewed as a map between the dual spaces $\ell_{q'}$ and $\ell_{p'}$. Under this regime, the operator norm of a matrix always equals the operator norm of its conjugate transpose:

$$\|A\|_{p,q} \quad = \quad \|A^*\|_{q',p'} \,. \tag{4.3}$$

Therefore, any procedure for calculating the norm of a matrix can also be used to calculate the norm of its conjugate transpose.

4.3.1 Calculating Operator Norms

Some basic operator norms can be determined with ease, while others are quite stubborn. The following table describes how to compute the most important ones.

		Co-domain		
		ℓ_1	ℓ_2	ℓ_∞
Domain	ℓ_1	Maximum ℓ_1 norm of a column	Maximum ℓ_2 norm of a column	Maximum absolute entry of matrix
	ℓ_2	NP-hard	Maximum singular value	Maximum ℓ_2 norm of a row
	ℓ_∞	NP-hard	NP-hard	Maximum ℓ_1 norm of a row

The computational complexity of the $(\infty, 1)$ norm is due to Rohn [89]. Using his methods, one can prove that it is also NP-hard to calculate the $(\infty, 2)$ norm. The result for the $(2, 1)$ norm follows from equation (4.3).

4.4 Singular Values

Under any linear map A, the image of the Euclidean unit ball is an ellipsoid. The Euclidean lengths of the semi-axes of this ellipsoid are called the *singular values* of the map. The maximum singular value of A coincides with the $(2, 2)$ operator norm of the matrix. If A has more rows than columns, its singular values may also be defined algebraically as the square roots of the eigenvalues of $\mathsf{A}^*\mathsf{A}$.

Suppose that Λ indexes a collection of m atoms. If m is small enough, then we may develop good bounds on the singular values of $\mathbf{\Phi}_\Lambda$ using the cumulative coherence.

Proposition 4.3. *Suppose that $|\Lambda| = m$. Each singular value σ of the matrix $\mathbf{\Phi}_\Lambda$ satisfies*

$$1 \;-\; \mu_1(m-1) \quad\leq\quad \sigma^2 \quad\leq\quad 1 \;+\; \mu_1(m-1).$$

A version of this proposition was used implicitly by Gilbert, Muthukrishnan, and Strauss in [48], and the current version appears in [110]. The present result first reached print in the article of Donoho and Elad [29].

Proof. Consider the Gram matrix $\mathsf{G} = \mathbf{\Phi}_\Lambda{}^*\mathbf{\Phi}_\Lambda$. The Geršgorin Disc Theorem [61] states that every eigenvalue of G lies in one of the m discs

$$\left\{ z \;:\; |\mathsf{G}(\lambda,\lambda) - z| \;\leq\; \sum\nolimits_{\omega\neq\lambda} |\mathsf{G}(\lambda,\omega)| \right\} \qquad \text{for each index } \lambda \text{ in } \Lambda.$$

The normalization of the atoms implies that $\mathsf{G}(\lambda, \lambda) \equiv 1$. Meanwhile, the sum is bounded above by $\mu_1(m-1)$. The result follows since the eigenvalues of G equal the squared singular values of $\mathbf{\Phi}_\Lambda$. $\square$

When $N \leq d$, it is possible to develop alternate bounds on the singular values of $\mathbf{\Phi}_\Lambda$ using interlacing theorems. The following result specializes Theorem 7.3.9 of [61].

Proposition 4.4. *Suppose that $N \leq d$. Then each singular value σ of the matrix $\mathbf{\Phi}_\Lambda$ satisfies*

$$\sigma_{\min}(\mathbf{\Phi}) \quad \leq \quad \sigma \quad \leq \quad \sigma_{\max}(\mathbf{\Phi}),$$

where $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values of a matrix.

4.5 The Inverse Gram Matrix

Suppose that Λ indexes a sub-dictionary, and let G denote the Gram matrix $\mathbf{\Phi}_\Lambda{}^* \mathbf{\Phi}_\Lambda$. The (∞, ∞) operator norm of G^{-1} will arise in our calculations, so we need to develop a bound on it. Afterward, we will make a connection between this inverse and the dual system of the sub-dictionary.

Proposition 4.5. *Let $m = |\Lambda|$, and suppose that $\mu_1(m-1) < 1$. Then*

$$\left\| \mathsf{G}^{-1} \right\|_{\infty,\infty} \quad = \quad \left\| \mathsf{G}^{-1} \right\|_{1,1} \quad \leq \quad \frac{1}{1 - \mu_1(m-1)}. \tag{4.4}$$

This proposition was established independently in [44, 106]. For comparison, observe that $\|\mathsf{G}\|_{\infty,\infty} \leq 1 + \mu_1(m-1)$.

Proof. First, note that the two operator norms in (4.4) are equal because the inverse Gram matrix is Hermitian. Since the atoms are normalized, the

Gram matrix has a unit diagonal. Therefore, we may split it as the sum of its diagonal and off-diagonal parts: $\mathsf{G} = \mathsf{I}_m + \mathsf{A}$. Each row of the matrix A lists the inner products between a fixed atom and $(m-1)$ other atoms. Therefore, $\|\mathsf{A}\|_{\infty,\infty} \leq \mu_1(m-1)$. Now invert G using a Neumann series:

$$\begin{aligned} \left\|\mathsf{G}^{-1}\right\|_{\infty,\infty} &= \left\|\sum\nolimits_{k=0}^{\infty}(-\mathsf{A})^k\right\|_{\infty,\infty} \\ &\leq \sum\nolimits_{k=0}^{\infty}\|\mathsf{A}\|_{\infty,\infty}^k \\ &= \frac{1}{1-\|\mathsf{A}\|_{\infty,\infty}}. \end{aligned}$$

Introduce the estimate for $\|\mathsf{A}\|_{\infty,\infty}$ to complete the proof. □

The inverse of the Gram matrix has a useful interpretation. The atoms in Λ form a linearly independent set, so there is a unique collection of *dual vectors* $\{\boldsymbol{\psi}_\lambda\}_{\lambda\in\Lambda}$ that has the same linear span as $\{\boldsymbol{\varphi}_\lambda\}_{\lambda\in\Lambda}$ and that satisfies the bi-orthogonal property

$$\langle \boldsymbol{\psi}_\lambda, \boldsymbol{\varphi}_\lambda\rangle = 1 \qquad \text{and} \qquad \langle \boldsymbol{\psi}_\lambda, \boldsymbol{\varphi}_\omega\rangle = 0 \qquad \text{for } \lambda, \omega \text{ in } \Lambda \text{ and } \omega \neq \lambda.$$

That is, each dual vector $\boldsymbol{\psi}_\lambda$ is orthogonal to the atoms with different indices, and it is scaled in this (unique) direction until its inner product with $\boldsymbol{\varphi}_\lambda$ equals one. The definition of the dual system suggests that it somehow inverts the sub-dictionary. Indeed, the dual vectors form the columns of $(\boldsymbol{\Phi}_\Lambda^\dagger)^*$. We may calculate that

$$\begin{aligned} \mathsf{G}^{-1} &= (\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1} \\ &= (\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1}(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1} \quad = \quad \boldsymbol{\Phi}_\Lambda{}^\dagger(\boldsymbol{\Phi}_\Lambda{}^\dagger)^*. \end{aligned}$$

Therefore, the inverse Gram matrix tabulates the inner products between the dual vectors. More information about dual vectors and biorthogonal systems may be located in any book on functional analysis, e.g., [63].

4.6 The Exact Recovery Coefficient

Now we will develop a measure of the similarity between a sub-dictionary and the remaining atoms from the dictionary. Let Λ index a sub-dictionary, and define the quantity

$$\mathrm{ERC}(\Lambda; \mathscr{D}) \quad \stackrel{\mathrm{def}}{=} \quad 1 \; - \; \max_{\omega \notin \Lambda} \left\| \boldsymbol{\Phi}_{\Lambda}{}^{\dagger} \boldsymbol{\varphi}_{\omega} \right\|_1 .$$

The letters "ERC" abbreviate the term *Exact Recovery Coefficient*, so called because $\mathrm{ERC}(\Lambda; \mathscr{D}) > 0$ will turn out to be a sufficient condition for several different algorithms to recover exact superpositions of atoms from Λ. *Nota bene* that every atom in the dictionary makes a critical difference in the value of $\mathrm{ERC}(\Lambda; \mathscr{D})$. Nevertheless, we will almost always omit the dictionary from the notation.

Proposition 4.6. *Suppose that $|\Lambda| \leq m$. A lower bound on the Exact Recovery Coefficient is*

$$\mathrm{ERC}(\Lambda) \quad \geq \quad \frac{1 \; - \; \mu_1(m-1) \; - \; \mu_1(m)}{1 \; - \; \mu_1(m-1)}.$$

It follows that $\mathrm{ERC}(\Lambda) > 0$ *whenever*

$$\mu_1(m-1) \; + \; \mu_1(m) \quad < \quad 1.$$

This argument independently appeared in [44, 106]. For every sub-dictionary of an orthonormal basis, the Exact Recovery Coefficient equals one.

Proof. Begin the calculation by expanding the generalized inverse and applying a norm estimate.

$$\begin{aligned} \max_{\omega \notin \Lambda} \left\| \boldsymbol{\Phi}_{\Lambda}{}^{\dagger} \boldsymbol{\varphi}_{\omega} \right\|_1 \quad &= \quad \max_{\omega \notin \Lambda} \left\| (\boldsymbol{\Phi}_{\Lambda}{}^{*} \, \boldsymbol{\Phi}_{\Lambda})^{-1} \, \boldsymbol{\Phi}_{\Lambda}{}^{*} \boldsymbol{\varphi}_{\omega} \right\|_1 \\ &\leq \quad \left\| (\boldsymbol{\Phi}_{\Lambda}{}^{*} \, \boldsymbol{\Phi}_{\Lambda})^{-1} \right\|_{1,1} \, \max_{\omega \notin \Lambda} \left\| \boldsymbol{\Phi}_{\Lambda}{}^{*} \boldsymbol{\varphi}_{\omega} \right\|_1 . \end{aligned}$$

For the first term, Proposition 4.5 provides an upper bound of $[1-\mu_1(m-1)]^{-1}$. An estimate of the second term is

$$\max_{\omega \notin \Lambda} \|\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\varphi}_\omega\|_1 \quad = \quad \max_{\omega \notin \Lambda} \sum\nolimits_{\lambda \in \Lambda} |\langle \boldsymbol{\varphi}_\omega, \boldsymbol{\varphi}_\lambda \rangle| \quad \leq \quad \mu_1(m).$$

Combine the inequalities to prove the result. □

Now let us turn to the geometric interpretation of the Exact Recovery Coefficient. Form the collection of signals

$$\mathscr{A}_1(\Lambda; \mathscr{D}) \quad \stackrel{\text{def}}{=} \quad \left\{ \boldsymbol{\Phi}_\Lambda \boldsymbol{b} \ : \ \boldsymbol{b} \in \mathbb{C}^\Lambda \text{ and } \|\boldsymbol{b}\|_1 \leq 1 \right\}.$$

This definition is adapted from the approximation theory literature [25, 104]. The set $\mathscr{A}_1(\Lambda)$ might be called the *antipodal convex hull* of the sub-dictionary because it is the smallest convex set that contains $z\,\boldsymbol{\varphi}_\omega$ for every unimodular complex number z and every index ω. See Figure 4.1 for an illustration.

Recall that $\mathsf{P}_\Lambda \boldsymbol{\varphi}_\omega = \boldsymbol{\Phi}_\Lambda \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega$ gives the orthogonal projection of the atom $\boldsymbol{\varphi}_\omega$ onto the span of the atoms indexed by Λ. Therefore, the coefficient vector $\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega$ can be used to synthesize this projection. We conclude that the quantity $1 - \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega \right\|_1$ measures how far the projected atom $\mathsf{P}_\Lambda \boldsymbol{\varphi}_\omega$ lies from the boundary of $\mathscr{A}_1(\Lambda)$. If every projected atom lies well within the antipodal convex hull, then it is possible to recover superpositions of atoms from Λ. The intuition is that the coefficient associated with an atom outside Λ must be quite large to represent anything in the span of the sub-dictionary. Figure 4.1 exhibits the geometry.

4.6.1 Structured Dictionaries

When the dictionary has additional structure, it is possible to refine our sufficient conditions for $\mathrm{ERC}(\Lambda) > 0$. In particular, we can develop a sharper

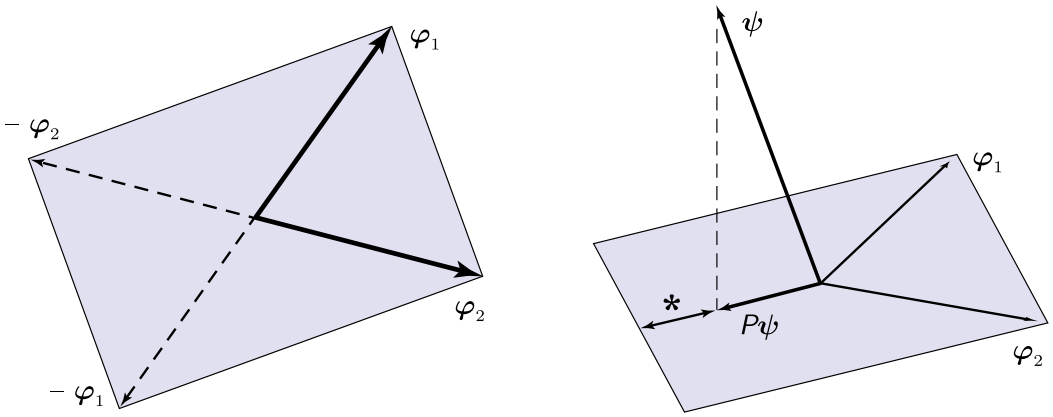


Figure 4.1: The Exact Recovery Coefficient. At left, we have shaded the antipodal convex hull of the two atoms $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$. At right, the asterisk $(*)$ indicates the distance that the projection of the atom $\boldsymbol{\psi}$ lies from the edge of the antipodal convex hull (in signal space). The Exact Recovery Coefficient bounds the corresponding distance in coefficient space.

result for multi-ONBs. The proof involves a difficult calculation, which the casual reader may wish to avoid.

Theorem 4.7. *Suppose that the dictionary consists of J concatenated orthonormal bases with overall coherence μ, and assume that Λ indexes p_j atoms from the j-th basis, $j = 1, \ldots, J$. Without loss of generality, assume that $0 < p_1 \leq p_2 \leq \cdots \leq p_J$. Then* $\mathrm{ERC}(\Lambda) > 0$ *holds whenever*

$$\sum_{j=2}^{J} \frac{\mu p_j}{1 + \mu p_j} \quad < \quad \frac{1}{2\,(1 + \mu p_1)}.$$

Proof. Permute the columns of the synthesis matrix $\boldsymbol{\Phi}_\Lambda$ so that

$$\boldsymbol{\Phi}_\Lambda \quad = \quad \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \ldots & \boldsymbol{\Phi}_J \end{bmatrix},$$

where the p_j columns of submatrix $\boldsymbol{\Phi}_j$ are the atoms from the j-th basis. Suppose that there are a total of m atoms. We seek a good upper bound for $\left\| {\boldsymbol{\Phi}_\Lambda}^\dagger \boldsymbol{\psi} \right\|_1$, where $\boldsymbol{\psi}$ is an atom not indexed in Λ. We will develop this matrix–vector product explicitly under a worst-case assumption on the size of the matrix and vector entries.

The generalized inverse can be expanded as ${\boldsymbol{\Phi}_\Lambda}^\dagger = \left({\boldsymbol{\Phi}_\Lambda}^* \boldsymbol{\Phi}_\Lambda\right)^{-1} {\boldsymbol{\Phi}_\Lambda}^*$. Our first goal is to develop a bound on the entries of the inverse Gram matrix. The Gram matrix ${\boldsymbol{\Phi}_\Lambda}^* \boldsymbol{\Phi}_\Lambda$ has the block form

$$G \quad \stackrel{\text{def}}{=} \quad \left[\begin{array}{c|c|c|c} \mathsf{I}_{p_1} & -A_{12} & \ldots & -A_{1J} \\ \hline -A_{21} & \mathsf{I}_{p_2} & \ldots & -A_{2J} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline -A_{J1} & -A_{J2} & \ldots & \mathsf{I}_{p_J} \end{array}\right] \quad \stackrel{\text{def}}{=} \quad \mathsf{I}_m - A,$$

where the entries of A are bounded in magnitude by μ. Using $|\cdot|$ to denote the entrywise absolute value of a matrix, we have the entrywise inequality

$$\left|G^{-1}\right| \quad = \quad \left|\mathsf{I}_m + \sum\nolimits_{k=1}^{\infty} A^k\right| \quad \leq \quad \mathsf{I}_m + \sum\nolimits_{k=1}^{\infty} |A|^k.$$

Therefore, we are at liberty in our estimates to assume that every nonzero entry of A equals μ. To proceed, creatively rewrite the Gram matrix as

$$G \quad = \quad (\mathsf{I}_m + \mu B) - (A + \mu B),$$

where B is the block matrix

$$B \quad \stackrel{\text{def}}{=} \quad \left[\begin{array}{c|c|c|c} 1_{p_1} & 0 & \dots & 0 \\ \hline 0 & 1_{p_2} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & 1_{p_J} \end{array}\right].$$

We use 1_p to indicate the $p \times p$ matrix of ones. By the foregoing, we have the entrywise bound

$$\left|G^{-1}\right| \quad \leq \quad \left((\mathsf{I}_m + \mu B) - \mu 1_m\right)^{-1},$$

which yields

$$\left|G^{-1}\right| \quad \leq \quad \left(\mathsf{I}_m - \mu \left(\mathsf{I}_m + \mu B\right)^{-1} 1_m\right)^{-1} \left(\mathsf{I}_m + \mu B\right)^{-1}. \tag{4.5}$$

Now, we work out the inverses from the right-hand side of (4.5). Using Neumann series, compute that

$$\left(\mathsf{I}_m + \mu B\right)^{-1} = \left[\begin{array}{c|c|c|c} \mathsf{I}_{p_1} - \frac{\mu}{1+\mu p_1} 1_{p_1} & 0 & \dots & 0 \\ \hline 0 & \mathsf{I}_{p_2} - \frac{\mu}{1+\mu p_2} 1_{p_2} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & \mathsf{I}_{p_J} - \frac{\mu}{1+\mu p_J} 1_{p_J} \end{array}\right]. \tag{4.6}$$

Meanwhile, the series development of the other inverse in (4.5) is

$$\left(\mathsf{I}_m - \mu \left(\mathsf{I}_m + \mu B\right)^{-1} 1_m\right)^{-1} \quad = \quad \mathsf{I}_m + \sum_{k=1}^{\infty} \left(\mu \left(\mathsf{I}_m + \mu B\right)^{-1} 1_m\right)^k. \tag{4.7}$$

In this proof, we will use $\mathbf{e}_p$ to denote the p-dimensional column vector whose entries all equal one. Then (4.6) allows us to calculate the product

$$\mu\,(\mathsf{I}_m + \mu B)^{-1}\mathsf{1}_m \quad = \quad \begin{bmatrix} \frac{\mu}{1+\mu p_1}\,\mathbf{e}_{p_1} \\ \hline \frac{\mu}{1+\mu p_2}\,\mathbf{e}_{p_2} \\ \hline \vdots \\ \hline \frac{\mu}{1+\mu p_J}\,\mathbf{e}_{p_J} \end{bmatrix} \mathbf{e}_m^T \quad \stackrel{\text{def}}{=} \quad \boldsymbol{v}\,\mathbf{e}_m^T.$$

It is easy to see that

$$\mathbf{e}_m^T\,\boldsymbol{v} \quad = \quad \sum_{j=1}^{J} \frac{\mu p_j}{1+\mu p_j}.$$

On account of the last two equations, the series in (4.7) collapses.

$$\sum_{k=1}^{\infty} \left(\boldsymbol{v}\,\mathbf{e}_m^T\right)^k \quad = \quad \left(\boldsymbol{v}\,\mathbf{e}_m^T\right) \sum_{k=1}^{\infty} \left(\mathbf{e}_m^T\,\boldsymbol{v}\right)^{k-1} \quad = \quad \frac{1}{1 - \sum_{j=1}^{J} \frac{\mu p_j}{1+\mu p_j}}\,\boldsymbol{v}\,\mathbf{e}_m^T.$$

We reach the bound

$$\left(\mathsf{I}_m - \mu\,(\mathsf{I}_m + \mu B)^{-1}\,\mathsf{1}_m\right)^{-1} \quad \leq \quad \mathsf{I}_m + \frac{1}{1 - \sum_{j=1}^{J} \frac{\mu p_j}{1+\mu p_j}}\,\boldsymbol{v}\,\mathbf{e}_m^T. \tag{4.8}$$

At last, we are prepared to develop a bound on the product that ultimately concerns us. Inequality (4.5) shows that

$$\begin{aligned} \left|\boldsymbol{\Phi}_\Lambda{}^\dagger\,\boldsymbol{\psi}\right| \quad &= \quad \left|(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1}\,\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\psi}\right| \\ &\leq \quad \left(\mathsf{I}_m - \mu\,(\mathsf{I}_m + \mu B)^{-1}\,\mathsf{1}_m\right)^{-1} (\mathsf{I}_m + \mu B)^{-1}\,\left|\boldsymbol{\Phi}_\Lambda{}^*\,\boldsymbol{\psi}\right|. \end{aligned} \tag{4.9}$$

We will work through the terms from right to left. Assume that the vector $\boldsymbol{\psi}$ is drawn from basis number Z. So

$$\left|\boldsymbol{\Phi}_\Lambda{}^*\,\boldsymbol{\psi}\right| \quad \leq \quad \left[\ \mu\,\mathbf{e}_{p_1}^T \ \middle|\ \ldots\ \middle|\ \mathbf{0}_{p_Z}^T \ \middle|\ \ldots\ \middle|\ \mu\,\mathbf{e}_{p_J}^T\ \right]^T. \tag{4.10}$$

Equations (4.6) and (4.10) imply

$$(\mathsf{I}_m + \mu B)^{-1}\,\left|\boldsymbol{\Phi}_\Lambda{}^*\,\boldsymbol{\psi}\right| \quad \leq \quad \left[\ \tfrac{\mu}{1+\mu p_1}\mathbf{e}_{p_1}^T \ \middle|\ \ldots\ \middle|\ \mathbf{0}_{p_Z}^T \ \middle|\ \ldots\ \middle|\ \tfrac{\mu}{1+\mu p_J}\mathbf{e}_{p_J}^T\ \right]^T. \tag{4.11}$$

Introducing (4.8) and (4.11) into (4.9) yields

$$\left|\boldsymbol{\Phi}_\Lambda{}^\dagger\,\boldsymbol{\psi}\right| \quad \leq \quad \begin{bmatrix} \frac{\mu}{1+\mu p_1}\mathbf{e}_{p_1} \\ \hline \vdots \\ \hline \mathbf{0}_{p_Z} \\ \hline \vdots \\ \hline \frac{\mu}{1+\mu p_J}\mathbf{e}_{p_J} \end{bmatrix} + \frac{\sum_{j\neq Z}\frac{\mu p_j}{1+\mu p_j}}{1-\sum_{j=1}^{J}\frac{\mu p_j}{1+\mu p_j}} \begin{bmatrix} \frac{\mu}{1+\mu p_1}\mathbf{e}_{p_1} \\ \hline \vdots \\ \hline \frac{\mu}{1+\mu p_Z}\mathbf{e}_{p_Z} \\ \hline \vdots \\ \hline \frac{\mu}{1+\mu p_J}\mathbf{e}_{p_J} \end{bmatrix}. \quad (4.12)$$

Finally, apply the ℓ_1 norm to inequality (4.12) to reach

$$\left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\,\boldsymbol{\psi}\right\|_1 \quad \leq \quad \frac{\sum_{j\neq Z}\frac{\mu p_j}{1+\mu p_j}}{1-\sum_{j=1}^{J}\frac{\mu p_j}{1+\mu p_j}}. \quad (4.13)$$

Since the function $r \mapsto \frac{r}{1+r}$ is increasing, the bound (4.13) is weakest when $Z = 1$. We conclude that the inequality $\max_{\boldsymbol{\psi}} \left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\boldsymbol{\psi}\right\|_1 < 1$ holds whenever

$$\sum_{j=2}^{J}\frac{\mu p_j}{1+\mu p_j} \quad < \quad \frac{1}{2\,(1+\mu p_1)}.$$

□

We may specialize this theorem to provide a result for two-ONBs.

Corollary 4.8. *Suppose that the dictionary consists of two orthonormal bases with overall coherence μ, and suppose that Λ indexes p atoms from the first basis and q atoms from the second basis, where $p \leq q$. Then* $\text{ERC}(\Lambda) > 0$ *whenever*

$$2\,\mu^2\,pq + \mu\,q \quad < \quad 1.$$

It is worth mentioning that this corollary can be established directly using a much prettier argument. In this special case, the matrix G^{-1} can be calculated elegantly by expanding it in a Neumann series and inspecting the even and

odd powers separately. This approach does not seem to extend to the general case.

We can also provide a result that depends only on the number of atoms in Λ, rather than their disbursement among the bases.

Corollary 4.9. *Suppose that the dictionary consists of J concatenated orthonormal bases with overall coherence μ. The condition* $\operatorname{ERC}(\Lambda) > 0$ *holds for every index set Λ with cardinality m provided that*

$$m \quad < \quad \left(\sqrt{2} - 1 + \frac{1}{2\,(J-1)} \right) \mu^{-1}.$$

Proof sketch. Minimize $\sum p_j$ subject to the constraints that $p_j \geq 0$ for each j and that

$$\sum_{j=2}^{J} \frac{\mu p_j}{1 + \mu p_j} \quad \geq \quad \frac{1}{2\,(1 + \mu p_1)}.$$

□

For the details of the calculation, see [54]. Strictly speaking, [54] proves a qualitatively different result, because its authors have not identified the Exact Recovery Coefficient as a quantity of fundamental interest. Rather, they prove that the condition in Corollary 4.9 is sufficient to invoke the condition in Theorem 4.7.

4.7 Uniqueness of Sparse Representations

A representation of a signal is said to be *unique* if every other representation of the signal requires strictly more atoms. The terminology is due to the fact that a representation of a signal $\boldsymbol{s}$ is unique if and only if that representation is the unique solution of (EXACT) with input $\boldsymbol{s}$. In an incoherent dictionary, every sufficiently sparse representation is unique.

Proposition 4.10 (Donoho–Elad [29], Gribonval–Nielsen [54]). *Suppose that $\mu_1(m-1) < 1$. Then every representation that involves $\frac{1}{2}\,m$ atoms or fewer is unique. In particular, $m < \frac{1}{2}\,(\mu^{-1}+1)$ is a sufficient condition for all m-term representations to be unique.*

Proof. Suppose that the sparsest representation of a signal involves k atoms, where $k \leq \frac{1}{2}\,m$. Suppose that the signal has a different representation using k atoms. Then the two representations together use no more than m atoms. Subtracting one representation from the other, we obtain a representation of the zero vector that uses no more than m atoms. But every collection of m atoms is linearly independent on account of Proposition 4.3.

The second claim of the proposition follows when we introduce the bound $\mu_1(m-1) \leq (m-1)\,\mu$ into the inequality $\mu_1(m-1) < 1$. □

When the dictionary has more structure, it is possible to develop better conditions that guarantee uniqueness.

Proposition 4.11 (Gribonval–Nielsen [54]). *Suppose that the dictionary consists of J concatenated orthonormal bases with total coherence μ. A sufficient condition for every m-term representation to be unique is that*

$$m \quad < \quad \frac{1}{2}\left(1 + \frac{1}{J-1}\right)\mu^{-1}.$$

4.8 Projective Spaces

Projective spaces provide the correct setting for understanding many geometric properties of a dictionary. To develop this concept, we begin with an equivalence relation on the collection of d-dimensional complex vectors:

$$\boldsymbol{w} \equiv \boldsymbol{z} \qquad \Longleftrightarrow \qquad \boldsymbol{w} = \zeta\,\boldsymbol{z} \quad \text{for } \zeta \text{ in } \mathbb{C}^{\times}.$$

(We denote by $\mathbb{C}^\times$ the set of nonzero complex numbers.) Under this equivalence relation, every nonzero vector is identified with the one-dimensional subspace spanned by that vector. The zero vector lies in a class by itself. For our purposes, the $(d-1)$-dimensional *complex projective space* will be defined as the collection of nonzero, d-dimensional complex vectors, modulo this equivalence relation:

$$\mathbb{P}^{d-1}(\mathbb{C}) \quad \stackrel{\text{def}}{=} \quad \frac{\mathbb{C}^d \setminus \{\mathbf{0}\}}{\mathbb{C}^\times}.$$

In words, $\mathbb{P}^{d-1}(\mathbb{C})$ is the set of one-dimensional subspaces of $\mathbb{C}^d$. The real projective space $\mathbb{P}^{d-1}(\mathbb{R})$ is defined in much the same way, and it may be viewed as the collection of all lines through the origin of $\mathbb{R}^d$. On analogy, we will refer to the elements of a complex projective space as *lines*.

The natural metric for $\mathbb{P}^{d-1}(\mathbb{C})$ is the acute angle between two lines—or what is equivalent—the sine of the acute angle. Therefore, the projective distance between two d-dimensional vectors $\boldsymbol{z}$ and $\boldsymbol{w}$ will be calculated as

$$\operatorname{dist}(\boldsymbol{z}, \boldsymbol{w}) \quad \stackrel{\text{def}}{=} \quad \left[1 \; - \; \left(\frac{|\langle \boldsymbol{z}, \boldsymbol{w} \rangle|}{\|\boldsymbol{z}\|_2 \; \|\boldsymbol{w}\|_2}\right)^2\right]^{1/2}. \tag{4.14}$$

In particular, if both vectors have unit norm,

$$\operatorname{dist}(\boldsymbol{z}, \boldsymbol{w}) \quad = \quad \sqrt{1 \; - \; |\langle \boldsymbol{z}, \boldsymbol{w} \rangle|^2}.$$

The distance between two lines ranges between zero and one. Equipped with this distance, $\mathbb{P}^{d-1}(\mathbb{C})$ forms a compact metric space [15].

4.9 Minimum Distance, Maximum Correlation

We view a dictionary as a finite set of lines in the projective space $\mathbb{P}^{d-1}(\mathbb{C})$. Given an arbitary nonzero signal $\boldsymbol{s}$, we will calculate the *minimum distance*

from the signal to the dictionary as

$$\min_{\omega \in \Omega} \operatorname{dist}(\boldsymbol{s}, \boldsymbol{\varphi}_\omega).$$

A complementary notion is the *maximum correlation* of the signal with the dictionary.

$$\operatorname{maxcor}(\boldsymbol{s}) \quad \stackrel{\text{def}}{=} \quad \max_{\omega \in \Omega} \frac{|\langle \boldsymbol{s}, \boldsymbol{\varphi}_\omega \rangle|}{\|\boldsymbol{s}\|_2} \quad = \quad \frac{\|\boldsymbol{\Phi}^* \boldsymbol{s}\|_\infty}{\|\boldsymbol{s}\|_2}.$$

Since the atoms are normalized, $0 \le \operatorname{maxcor}(\boldsymbol{s}) \le 1$. The relationship between the minimum distance and the maximum correlation is the following.

$$\min_{\omega \in \Omega} \operatorname{dist}(\boldsymbol{s}, \boldsymbol{\varphi}_\omega) \quad = \quad \sqrt{1 - \operatorname{maxcor}(\boldsymbol{s})^2}. \tag{4.15}$$

4.10 Packing Radii

We will be interested in several extremal properties of the dictionary that are easiest to understand in a general setting. Let $\mathbb{X}$ be a compact metric space with metric $\operatorname{dist}_{\mathbb{X}}$, and choose $Y = \{y_k\}$ to be a discrete set of points in $\mathbb{X}$. The *packing radius* of the set Y is defined as

$$\operatorname{pack}_{\mathbb{X}}(Y) \quad \stackrel{\text{def}}{=} \quad \min_{j \neq k} \operatorname{dist}_{\mathbb{X}}(y_j, y_k).$$

In words, the packing radius is the size of the largest open ball that can be centered at any point of Y without encompassing any other point of Y. An *optimal packing* of N points in $\mathbb{X}$ is a set Y_{opt} of cardinality N that has maximal packing radius, i.e., Y_{opt} solves the mathematical program

$$\max_{|Y| = N} \operatorname{pack}_{\mathbb{X}}(Y).$$

It is generally quite difficult to produce an optimal packing or even to check whether a collection of points gives an optimal packing. Figure 4.2 illustrates

packing in the unit square, and Figure 4.3 examines the situation in a projective space. The standard reference on packing is the *magnum opus* of Conway and Sloane [16].

The packing radius of the dictionary in the projective space $\mathbb{P}^{d-1}(\mathbb{C})$ is given by

$$\operatorname{pack}(\mathscr{D}) \quad \stackrel{\text{def}}{=} \quad \min_{\lambda \neq \omega} \operatorname{dist}(\boldsymbol{\varphi}_\lambda, \boldsymbol{\varphi}_\omega).$$

That is, the packing radius measures the minimum distance between any pair of distinct atoms. The coherence parameter is intimately related to this packing radius. Indeed,

$$\mu \quad = \quad \sqrt{1 - \operatorname{pack}(\mathscr{D})^2}.$$

Therefore, the dictionary provides a good packing if and only if the coherence is small. It is easily seen that the orthonormal bases for $\mathbb{C}^d$ give the only optimal packings of d points in $\mathbb{P}^{d-1}(\mathbb{C})$. For general d and N, it is quite difficult to construct minimally coherent dictionaries. See Chapter 7 for a numerical approach to this problem.

The geometric interpretation of the cumulative coherence μ_1 is not as straightforward. One may imagine centering a collection of m open balls of nondecreasing radii at a fixed atom. The k-th ball is chosen so that it contains no more than $(k-1)$ other atoms. Roughly, the cumulative coherence $\mu_1(m)$ is complementary to the maximum total radius of a nested collection of m balls that can be centered at any atom and still retain this property.

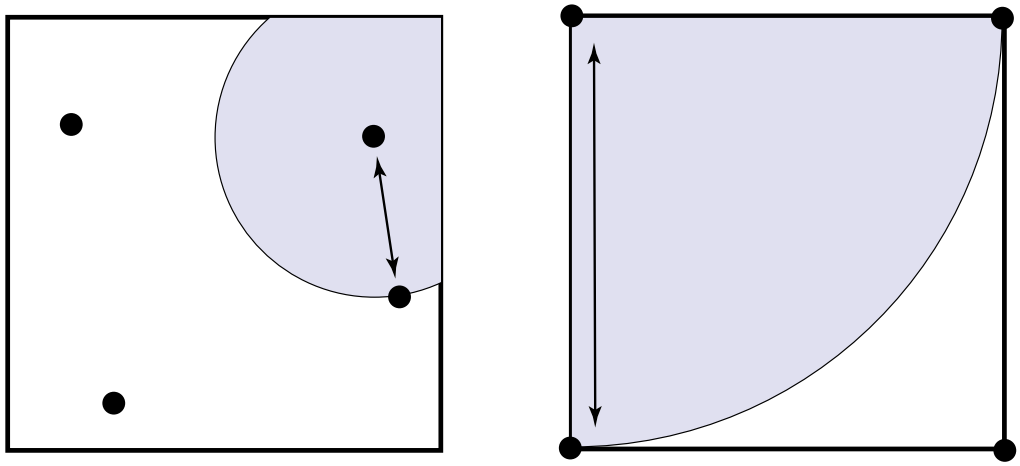


Figure 4.2: Packing example, Euclidean unit square. At left, the arrow indicates the packing radius of four points in the Euclidean unit square. At right, an *optimal* packing of four points in the unit square.

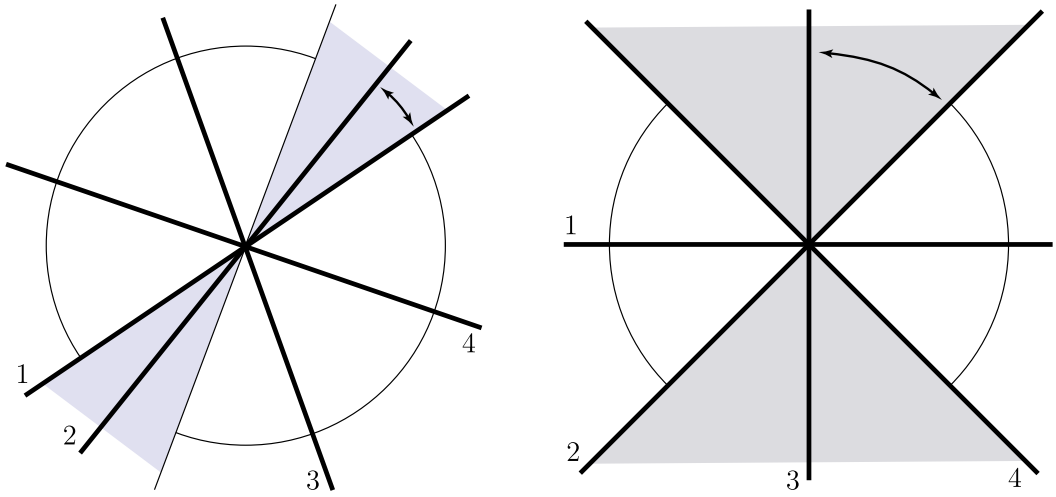


Figure 4.3: Packing example, real projective space. At left, the arrow indicates the packing radius of a collection of numbered lines, considered as elements of $\mathbb{P}^1(\mathbb{R})$. Note that, in this projective space, a "ball" becomes a doubly-infinite cone. At right, an *optimal* packing of four lines in $\mathbb{P}^1(\mathbb{R})$.

4.11 Covering Radii

Let us return to our compact metric space $\mathbb{X}$, from which we select a finite set of points $Y = \{y_k\}$. The *covering radius* of the set Y is defined as

$$\operatorname{cover}_{\mathbb{X}}(Y) \quad \stackrel{\text{def}}{=} \quad \max_{x \in \mathbb{X}} \min_{k} \operatorname{dist}_{\mathbb{X}}(x, y_k).$$

In words, the covering radius is the size of the largest open ball that can be centered at some point of $\mathbb{X}$ without encompassing a point of Y. An *optimal covering* with N points is a set Y_{opt} of cardinality N that has minimal covering radius. That is, Y_{opt} solves the mathematical program

$$\min_{|Y| = N} \operatorname{cover}_{\mathbb{X}}(Y).$$

N. J. A. Sloane has advanced the "meta-theorem" that optimal coverings are more regular than optimal packings [94]. It is often extremely difficult to compute the covering radius of an ensemble of points, let alone to produce an optimal covering. See Figure 4.4 for an example of covering in the Euclidean unit square; Figure 4.5 demonstrates covering of a projective space. Conway and Sloane's book is also an important reference on covering [16].

In our projective space $\mathbb{P}^{d-1}(\mathbb{C})$, the covering radius of the dictionary is given by the formula

$$\operatorname{cover}(\mathscr{D}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{s} \neq \mathbf{0}} \min_{\omega \in \Omega} \operatorname{dist}(\boldsymbol{s}, \boldsymbol{\varphi}_\omega).$$

It follows from the relation (4.15) that the covering radius is attained at a signal (called a *deep hole*) whose maximum correlation with the dictionary is smallest:

$$\operatorname{cover}(\mathscr{D}) \quad = \quad \max_{\boldsymbol{s} \neq \mathbf{0}} \sqrt{1 - \operatorname{maxcor}(\boldsymbol{s})^2}.$$

We will be most interested in how well a sub-dictionary covers its span. To that end, define

$$\operatorname{cover}(\Lambda; \mathscr{D}) \quad \stackrel{\text{def}}{=} \quad \max_{\substack{\boldsymbol{s} \in \mathscr{R}(\boldsymbol{\Phi}_\Lambda) \\ \boldsymbol{s} \neq \mathbf{0}}} \min_{\lambda \in \Lambda} \operatorname{dist}(\boldsymbol{s}, \boldsymbol{\varphi}_\lambda). \tag{4.16}$$

Without the range restriction on $\boldsymbol{s}$, the covering radius of the sub-dictionary would be one unless the atoms in Λ spanned the entire signal space.

Proposition 4.12. *The covering radius of a sub-dictionary satisfies the identity*

$$\operatorname{cover}(\Lambda)^2 \quad = \quad 1 \; - \; \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1}^{-2}.$$

Proof. Begin with (4.16), and apply the definition (4.14) of projective distance to see that

$$\begin{aligned} \operatorname{cover}(\Lambda)^2 \quad &= \quad 1 \; - \; \min_{\substack{\boldsymbol{s} \in \mathscr{R}(\boldsymbol{\Phi}_\Lambda) \\ \boldsymbol{s} \neq \mathbf{0}}} \; \max_{\lambda \in \Lambda} \; \frac{|\langle \boldsymbol{s}, \boldsymbol{\varphi}_\lambda \rangle|^2}{\|\boldsymbol{s}\|_2^2} \\ &= \quad 1 \; - \; \min_{\substack{\boldsymbol{s} \in \mathscr{R}(\boldsymbol{\Phi}_\Lambda) \\ \boldsymbol{s} \neq \mathbf{0}}} \; \frac{\|\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{s}\|_\infty^2}{\|\boldsymbol{s}\|_2^2}. \end{aligned}$$

Since the atoms indexed by Λ form a linearly independent set, $\boldsymbol{\Phi}_\Lambda{}^*$ has full row-rank. It follows from Proposition 4.2 that the minimum equals $\left\| (\boldsymbol{\Phi}_\Lambda{}^\dagger)^* \right\|_{\infty,2}^{-2}$. Apply the identity (4.3) to switch from the $(\infty, 2)$ norm to the $(2, 1)$ norm. $\square$

One interpretation of this proposition is that $\left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1}$ gives the secant of the largest acute angle between a vector in the span of the sub-dictionary and the closest atom from the sub-dictionary. We also learn that calculating $\operatorname{cover}(\Lambda)$ is likely to be NP-hard.

Using the cumulative coherence function, we can develop reasonable estimates for $\operatorname{cover}(\Lambda)$. This result will show that sub-dictionaries of incoherent dictionaries form good coverings.

Proposition 4.13. *Suppose that Λ lists m linearly independent atoms and that $\mu_1(m-1) < 1$. Then*

$$\begin{aligned} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1} &\leq \left[\frac{m}{1 \ - \ \mu_1(m-1)} \right]^{1/2}, \qquad \textit{and therefore} \\ \operatorname{cover}(\Lambda) &\leq \left[1 \ - \ \frac{1 \ - \ \mu_1(m-1)}{m} \right]^{1/2}. \end{aligned}$$

Proof. Write the definition of the operator norm, and estimate the ℓ_1 norm with the ℓ_2 norm:

$$\begin{aligned} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1} &= \max_{\|\boldsymbol{s}\|_2 = 1} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{s} \right\|_1 \\ &\leq \sqrt{m} \max_{\|\boldsymbol{s}\|_2 = 1} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{s} \right\|_2 \quad = \quad \sqrt{m} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,2}. \end{aligned}$$

The $(2,2)$ operator norm of $\boldsymbol{\Phi}_\Lambda{}^\dagger$ is the reciprocal of the minimum singular value of $\boldsymbol{\Phi}_\Lambda$. To complete the proof, apply the lower bound on this singular value given by Proposition 4.3. □

We can develop a second version of Proposition 4.13 by estimating the minimum singular value with Proposition 4.4.

Proposition 4.14. *Assume that $N \leq d$, and suppose that Λ lists m atoms. Then*

$$\left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1} \quad \leq \quad \sqrt{m} \ / \ \sigma_{\min}(\boldsymbol{\Phi}).$$

A separate argument (which we omit) establishes that the covering radius of m vectors strictly exceeds $\sqrt{1 - 1/m}$ unless the vectors are orthonormal. It follows that orthonormal bases give the only optimal coverings of $\mathbb{P}^{d-1}(\mathbb{C})$ using d points. This result also provides an intuition why balls in infinite-dimensional Hilbert spaces cannot be compact: a collection of m vectors must cover its span worse and worse as m increases.

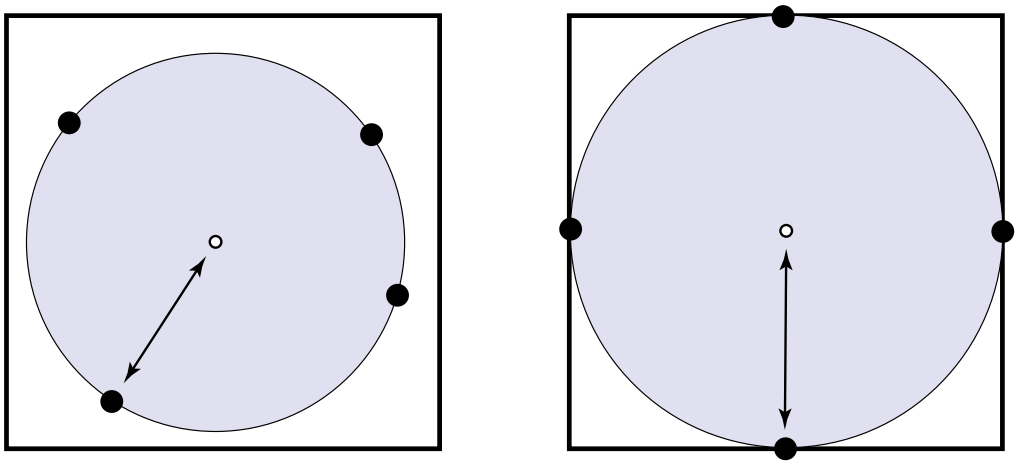


Figure 4.4: Covering example, Euclidean unit square. At left, the arrow indicates the covering radius of four points in the Euclidean unit square, and the open circle marks a deep hole, a point at which the covering radius is attained. At right, an *optimal* covering of the unit square with four points.

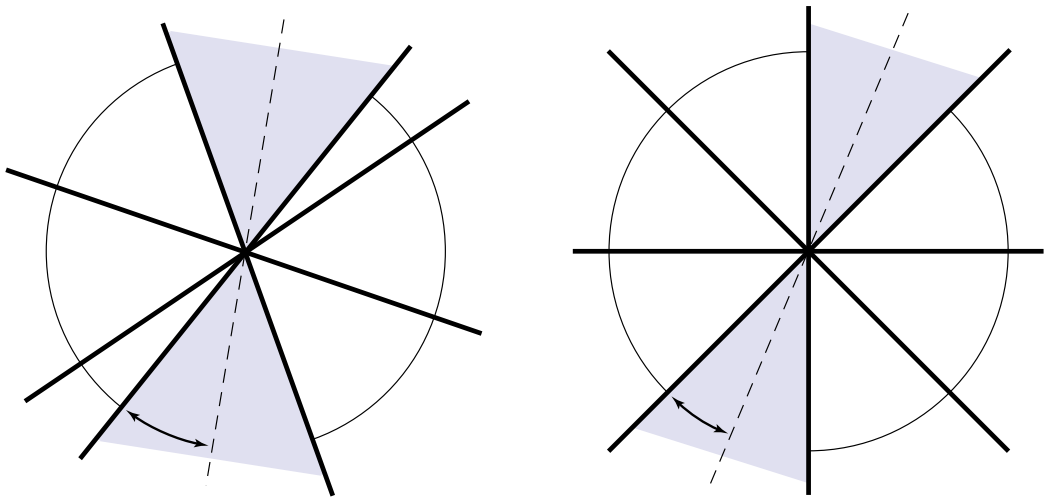


Figure 4.5: Covering example, real projective space. At left, the covering radius of four lines in the real projective space $\mathbb{P}^1(\mathbb{R})$. Dashes mark a deep hole, a line at which the covering radius is attained. At right, an *optimal* covering of $\mathbb{P}^1(\mathbb{R})$ with four lines.

4.12 Quantization

Now let us consider a probability space $(\mathbb{X}, \Sigma, \mathrm{d}x)$ in which metric balls are Σ-measurable, and choose another finite set of points $Y = \{y_k\}$. In this setting, $\min_k \operatorname{dist}(x, y_k)$ is called the *quantization error* for the point x. The expected quantization error is the usual measure of how well Y represents the distribution $\mathrm{d}x$. It is computed with the integral

$$\operatorname{quant}(Y) \quad \stackrel{\text{def}}{=} \quad \int \min_k \operatorname{dist}_{\mathbb{X}}(x, y_k) \, \mathrm{d}x.$$

The relationship $\operatorname{quant}(Y) \leq \operatorname{cover}(Y)$ is always in force. An N-point *optimal codebook* for quantizing $\mathrm{d}x$ is a set that solves the mathematical program

$$\min_{|Y| = N} \operatorname{quant}(Y).$$

Optimal quantization is a difficult problem, and it has been studied extensively [47]. Heuristic methods are available for constructing good codebooks [69, 20, 90].

Suppose that we define a probability measure $\mathrm{d}\boldsymbol{\nu}$ on the projective space $\mathbb{P}^{d-1}(\mathbb{C})$. The expected error in quantizing $\mathrm{d}\boldsymbol{\nu}$ with the dictionary is defined as

$$\operatorname{quant}(\mathscr{D}) \quad \stackrel{\text{def}}{=} \quad \int \min_{\omega} \operatorname{dist}(\boldsymbol{\nu}, \boldsymbol{\varphi}_\omega) \, \mathrm{d}\boldsymbol{\nu} \quad = \quad \int \sqrt{1 - \operatorname{maxcor}(\boldsymbol{\nu})^2} \, \mathrm{d}\boldsymbol{\nu}.$$

Now imagine that we are trying to recover a short linear combination of atoms that has been contaminated with additive noise whose *direction* is distributed according to $\mathrm{d}\boldsymbol{\nu}$. In this situation, the best dictionaries for sparse approximation do a *horrible* job quantizing the direction of the noise. As a result, it is highly likely that the signal can be recovered, even if the noise has significant magnitude.

Chapter 5

Analysis of Greedy Methods

Conventional wisdom has been ambivalent about the application of greedy algorithms for sparse approximation. On the one hand, it is well known that greedy methods generally produce a sequence of approximations that converge to the input signal (provided that the dictionary is complete) [104]. On the other hand, the literature contains several dramatic examples where a greedy method catastrophically fails to recover the optimal sparse representation of a signal. Indeed, there is a dictionary and a signal with a two-term representation over that dictionary for which Matching Pursuit chooses every incorrect atom before it ever selects an optimal atom [24, 10].

Nevertheless, a recent result of Gilbert, Muthukrishnan, and Strauss for Orthogonal Matching Pursuit suggests that greedy methods deserve reconsideration. These authors proved that OMP can identify an m-term representation of an arbitrary signal that achieves an error within a constant factor of the optimal m-term error, provided that the dictionary is incoherent and m is sufficiently small. In particular, OMP solves (EXACT) correctly for every signal with a sufficiently sparse representation. Another way of phrasing their result is that OMP is an approximation algorithm for (SPARSE) over an incoherent dictionary [48]. Their paper gives a significant new insight into the qualitative behavior of OMP.

This chapter contains a new analysis of Orthogonal Matching Pursuit

that delivers a more precise insight into its qualitative and quantitative performance. We will see that OMP offers provably good performance for (EXACT), (ERROR), and (SPARSE) in a wide class of dictionaries, which includes incoherent dictionaries. The results depend significantly on the geometric properties of the dictionary. The Exact Recovery Coefficient plays a starring role.

The first section considers the performance of Orthogonal Matching Pursuit for (EXACT), the problem of determining the sparsest representation of an input signal. We develop a condition which ensures that a greedy selection identifies all the atoms from the optimal sparse representation of the signal and no others. Then we show how to use the cumulative coherence function of the dictionary to check when this condition is in force. Finally, we demonstrate that the condition cannot be improved.

The second section extends the analysis of the first section to cover signals that do not have a sparse representation. In the third and fourth sections, we apply this result to (ERROR) and (SPARSE). In particular, we prove that OMP can produce a representation that uses only atoms from an optimal solution to (ERROR) and achieves an approximation error within a constant factor of the desired error. We also prove that OMP can produce an m-term representation that achieves an error within a constant factor of the optimal m-term error. Using the cumulative coherence function, we provide several explicit bounds on these constants.

Most of the material in this chapter is slated to appear in *IEEE Transactions on Information Theory*. It is drawn from the work [106], which is copyright 2004 by the IEEE, and it is reused with permission.

5.1 The Sparsest Representation of a Signal

We begin with the problem of recovering the sparsest representation of a target signal. For an input signal $\boldsymbol{s}$, the formal statement of the problem is

$$\min_{\boldsymbol{c} \in \mathbb{C}^{\Omega}} \quad \|\boldsymbol{c}\|_0 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\,\boldsymbol{c} = \boldsymbol{s}. \tag{EXACT}$$

Suppose that the input signal $\boldsymbol{s}$ can be written as a linear combination of m atoms and no fewer. Therefore,

$$\boldsymbol{s} \quad = \quad \sum\nolimits_{\lambda \in \Lambda_{\text{opt}}} c_\lambda\, \boldsymbol{\varphi}_\lambda$$

where Λ_{opt} is a subset of Ω with cardinality m. Note that the atoms in Λ_{opt} are linearly independent and that the coefficients c_λ are nonzero. Otherwise, the signal has a representation using fewer than m atoms.

Let $\boldsymbol{\Phi}_{\text{opt}}$ denote the synthesis matrix associated with the sub-dictionary Λ_{opt}. Then the signal can also be expressed as

$$\boldsymbol{s} \quad = \quad \boldsymbol{\Phi}_{\text{opt}}\, \boldsymbol{c}_{\text{opt}}$$

where $\boldsymbol{c}_{\text{opt}} \in \mathbb{C}^{\Lambda_{\text{opt}}}$. Since the optimal atoms are linearly independent, $\boldsymbol{\Phi}_{\text{opt}}$ has full column-rank. Define a second matrix $\boldsymbol{\Psi}_{\text{opt}}$ whose columns are the $(N - m)$ atoms indexed by $\Omega \setminus \Lambda_{\text{opt}}$. Thus $\boldsymbol{\Psi}_{\text{opt}}$ contains the atoms that *do not* participate in the optimal representation.

5.1.1 Greedy Selection of Atoms

The fundamental step in a greedy algorithm is the greedy selection of an atom. Our goal is to develop a condition which ensures that this greedy choice identifies an atom from the optimal index set Λ_{opt}.

Suppose that $\boldsymbol{r}$ is a signal. Recall that greedy selection determines an index λ that satisfies

$$\lambda \quad \in \quad \arg\max_{\omega \in \Omega} \; |\langle \boldsymbol{r}, \boldsymbol{\varphi}_\omega \rangle| \,.$$

Now observe that the vector $\boldsymbol{\Phi}_{\text{opt}}{}^* \boldsymbol{r}$ lists the inner products between the vector $\boldsymbol{r}$ and the optimal atoms. So the expression $\| \boldsymbol{\Phi}_{\text{opt}}{}^* \boldsymbol{r} \|_\infty$ gives the largest magnitude attained among these inner products. Similarly, $\| \boldsymbol{\Psi}_{\text{opt}}{}^* \boldsymbol{r} \|_\infty$ expresses the largest absolute inner product between $\boldsymbol{r}$ and any nonoptimal atom. In consequence, to see whether the largest absolute inner product occurs at an optimal atom, we just need to examine the *greedy selection ratio*

$$\rho(\boldsymbol{r}) \quad \stackrel{\text{def}}{=} \quad \frac{\| \boldsymbol{\Psi}_{\text{opt}}{}^* \boldsymbol{r} \|_\infty}{\| \boldsymbol{\Phi}_{\text{opt}}{}^* \boldsymbol{r} \|_\infty}. \tag{5.1}$$

We see that a greedy choice[1] will recover an optimal atom if and only if $\rho(\boldsymbol{r}) < 1$. The following lemma provides a sufficient condition for the recovery of an optimal atom.

Lemma 5.1 (Greedy Selection). *Suppose that $\boldsymbol{r}$ lies in the column span of $\boldsymbol{\Phi}_{\text{opt}}$. A sufficient condition for $\rho(\boldsymbol{r}) < 1$ is that*

$$\text{ERC}(\Lambda_{\text{opt}}) \quad > \quad 0.$$

Recall that the Exact Recovery Coefficient of Λ_{opt} is defined as

$$\text{ERC}(\Lambda_{\text{opt}}) \quad \stackrel{\text{def}}{=} \quad 1 \; - \; \max_{\boldsymbol{\psi}} \left\| \boldsymbol{\Phi}_{\text{opt}}{}^\dagger \boldsymbol{\psi} \right\|_1 ,$$

where the maximum occurs over the columns of $\boldsymbol{\Psi}_{\text{opt}}$, the nonoptimal atoms.

[1] In case that $\rho(\boldsymbol{r}) = 1$, an optimal atom and a nonoptimal atom both attain the maximal inner product. The algorithm has no provision for determining which one to select. In the sequel, we make the pessimistic assumption that a greedy procedure never chooses an optimal atom when a nonoptimal atom also satisfies the selection criterion. This convention forces greedy techniques to fail for borderline cases, which is appropriate for analyzing algorithmic correctness.

Proof. Notice that the greedy selection ratio (5.1) bears a suspicious resemblance to an induced matrix norm. Before we can apply the usual norm bound, the term $\boldsymbol{\Phi}_{\text{opt}}{}^{*}\,\boldsymbol{r}$ must appear in the numerator. To that end, remember that $\boldsymbol{\Phi}_{\text{opt}}\,\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}$ is an orthogonal projector onto the column span of $\boldsymbol{\Phi}_{\text{opt}}$. This projector is conjugate symmetric, so we have

$$(\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger})^{*}\,\boldsymbol{\Phi}_{\text{opt}}{}^{*}\,\boldsymbol{r} \qu= \quad \boldsymbol{\Phi}_{\text{opt}}\,\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\,\boldsymbol{r} \quad = \quad \boldsymbol{r}.$$

Therefore, we may calculate that

$$\begin{aligned}
\rho(\boldsymbol{r}) &= \frac{\|\boldsymbol{\Psi}_{\text{opt}}{}^{*}\,\boldsymbol{r}\|_{\infty}}{\|\boldsymbol{\Phi}_{\text{opt}}{}^{*}\,\boldsymbol{r}\|_{\infty}} \\
&= \frac{\left\|\boldsymbol{\Psi}_{\text{opt}}{}^{*}(\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger})^{*}\boldsymbol{\Phi}_{\text{opt}}{}^{*}\,\boldsymbol{r}\right\|_{\infty}}{\|\boldsymbol{\Phi}_{\text{opt}}{}^{*}\,\boldsymbol{r}\|_{\infty}} \\
&\leq \left\|\boldsymbol{\Psi}_{\text{opt}}{}^{*}(\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger})^{*}\right\|_{\infty,\infty}.
\end{aligned}$$

Since $\|\cdot\|_{\infty,\infty}$ equals the maximum absolute *row* sum of its argument and $\|\cdot\|_{1,1}$ equals the maximum absolute *column* sum of its argument, we take a conjugate transpose and switch norms. Continuing the calculation,

$$\begin{aligned}
\rho(\boldsymbol{r}) &\leq \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\,\boldsymbol{\Psi}_{\text{opt}}\right\|_{1,1} \\
&= \max_{\boldsymbol{\psi}} \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\,\boldsymbol{\psi}\right\|_{1}
\end{aligned}$$

where the maximization occurs over the columns of $\boldsymbol{\Psi}_{\text{opt}}$, the nonoptimal atoms. □

This theorem is simple enough to state and prove, but it represents a fundamental advance in the study of greedy algorithms for sparse approximation. The important idea that OMP can recover optimal atoms in special cases first appeared in the paper of Gilbert et al. [48].

5.1.2 The Exact Recovery Theorem

Using the Greedy Selection Lemma, it is easy to provide sufficient conditions for greedy algorithms to recover the sparsest representation of an input signal.

Theorem 5.2 (Exact Recovery for OMP). *Suppose that* $\boldsymbol{c}_{\text{opt}}$ *is the sparsest representation of the input signal, and set* $\Lambda_{\text{opt}} = \text{supp}(\boldsymbol{c}_{\text{opt}})$. *A sufficient condition for Orthogonal Matching Pursuit to recover* $\boldsymbol{c}_{\text{opt}}$ *after* $|\Lambda_{\text{opt}}|$ *steps is that*

$$\text{ERC}(\Lambda_{\text{opt}}) \quad > \quad 0.$$

Proof. Suppose that, after t iterations, Orthogonal Matching Pursuit has selected t optimal atoms. At the beginning of the first iteration, this hypothesis is satisfied trivially. We will develop a condition to guarantee that the atom selected in the $(t+1)$-st iteration is also optimal.

Denote by $\boldsymbol{r}_t$ the residual signal at the beginning of the $(t+1)$-st iteration. Orthogonal Matching Pursuit selects another optimal atom if and only if the greedy selection ratio $\rho(\boldsymbol{r}_t)$ is less than one. The statement of the algorithm shows that the residual $\boldsymbol{r}_t$ equals the target signal $\boldsymbol{s}$ minus a linear combination of the t atoms that have already been chosen. Since the signal itself is a linear combination of the atoms indexed by Λ_{opt}, the induction hypothesis shows that the residual $\boldsymbol{r}_t$ lies in column span of $\boldsymbol{\Phi}_{\text{opt}}$. The Greedy Selection Lemma proves that $\rho(\boldsymbol{r}_t) < 1$ whenever $\text{ERC}(\Lambda_{\text{opt}}) > 0$. Therefore, the condition $\text{ERC}(\Lambda_{\text{opt}}) > 0$ ensures that Orthogonal Matching Pursuit selects an optimal atom in the $(t+1)$-st step.

Since Orthogonal Matching Pursuit always chooses a new atom, it follows that m steps of OMP will identify all m atoms that make up the sparsest

representation of $\boldsymbol{s}$. Thus, the coefficient vector returned by OMP synthesizes the target signal perfectly. □

Theorem 5.7 of the sequel shows that Theorem 5.2 is essentially the best possible for OMP. Incredibly, the same condition also implies that the convex relaxation (R-EXACT) will recover the sparsest representation of the input signal, which we prove in Section 6.1.

Gribonval and Nielsen recently observed [55] that the proof here also applies to Matching Pursuit, Algorithm 3.1. Since Matching Pursuit need not select a new atom at each step, the results are somewhat weaker.

Corollary 5.3 (Gribonval–Nielsen [55]). *Suppose that the target signal can be represented using the atoms listed in* Λ_{opt}. *If* $\mathrm{ERC}(\Lambda_{\mathrm{opt}}) > 0$, *then Matching Pursuit selects an index from* Λ_{opt} *during every iteration.*

5.1.3 Coherence Estimates

Since we are unlikely to know the optimal atoms *a priori*, Theorem 5.2 may initially seem useless. But for many dictionaries, the condition $\mathrm{ERC}(\Lambda) > 0$ holds for every index set Λ whose cardinality is sufficiently small. Combining Theorem 5.2 with Proposition 4.6, we reach a more applicable corollary.

Corollary 5.4. *Suppose that*

$$\mu_1(m-1) + \mu_1(m) \quad < \quad 1.$$

Then Orthogonal Matching Pursuit solves (EXACT) *for every input signal that has an* m*-term representation over the dictionary. Moreover, all* m*-term representations are unique.*

Proof. According to Proposition 4.6, the condition $\mu_1(m-1) + \mu_1(m) < 1$ guarantees that $\mathrm{ERC}(\Lambda) > 0$ for every index set Λ of cardinality m. If the sparsest representation of an input signal has a representation over such an index set Λ, then Theorem 5.2 guarantees that OMP will solve (EXACT) for this input signal.

Suppose that some m-term representation were not unique. Then the foregoing paragraph proves that OMP would simultaneously recover two distinct m-term representations of the signal. This is absurd. □

One interpretation of this result is that OMP can recover signals that have sparse representations over incoherent dictionaries. Since $\mu_1(m) \leq m\,\mu$, we can provide a result phrased in terms of the coherence parameter.

Corollary 5.5. *Suppose that*

$$m \quad < \quad \tfrac{1}{2}\,(\mu^{-1} + 1).$$

Then Orthogonal Matching Pursuit solves (EXACT) *for every input signal that has an m-term representation over the dictionary. Moreover, all m-term representations are unique.*

In case that the dictionary has more structure, we can prove better sufficient conditions. For example, Corollary 4.9 yields the following result when one uses OMP with a multi-ONB dictionary.

Corollary 5.6. *Suppose that the dictionary consists of J concatenated orthonormal bases with overall coherence μ, and assume that*

$$m \quad < \quad \left(\sqrt{2} - 1 + \frac{1}{2\,(J-1)}\right) \mu^{-1}.$$

Then Orthogonal Matching Pursuit solves (EXACT) *for every input signal that has an m-term representation over the dictionary. Moreover, all m-term representations are unique.*

Compare these results with the results for the convex relaxation (R-EXACT) that appear in Section 6.1.

5.1.4 Is the ERC Necessary?

One may ask whether Theorem 5.2 also provides a necessary condition for Orthogonal Matching Pursuit to succeed. The answer is a qualified affirmative, as this partial converse proves.

Theorem 5.7 (Exact Recovery Converse). *Let Λ_{opt} index a sub-dictionary for which $\mathrm{ERC}(\Lambda_{\mathrm{opt}}) \leq 0$, and assume that representations over Λ_{opt} are unique. Then there is an input signal that has a representation over Λ_{opt} for which Orthogonal Matching Pursuit fails to solve* (EXACT).

Proof. The argument basically reverses the proof of Theorem 5.2, but we must check that the inequalities in that argument can all hold with equality.

By the uniqueness of sparse representations over Λ_{opt}, every signal that has a representation over Λ_{opt} induces the same two matrices $\boldsymbol{\Phi}_{\mathrm{opt}}$ and $\boldsymbol{\Psi}_{\mathrm{opt}}$. Since $\mathrm{ERC}(\Lambda_{\mathrm{opt}}) \leq 0$, it follows that $\left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{\Psi}_{\mathrm{opt}} \right\|_{1,1} \geq 1$. Choose $\boldsymbol{b}_{\mathrm{bad}}$ from $\mathbb{C}^{\Lambda_{\mathrm{opt}}}$ so that

$$\frac{\left\| \boldsymbol{\Psi}_{\mathrm{opt}}{}^{*} (\boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger})^{*}\, \boldsymbol{b}_{\mathrm{bad}} \right\|_{\infty}}{\left\| \boldsymbol{b}_{\mathrm{bad}} \right\|_{\infty}} \quad = \quad \left\| \boldsymbol{\Psi}_{\mathrm{opt}}^{*} (\boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger})^{*} \right\|_{\infty,\infty}.$$

Therefore,

$$\frac{\left\| \boldsymbol{\Psi}_{\mathrm{opt}}{}^{*} (\boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger})^{*}\, \boldsymbol{b}_{\mathrm{bad}} \right\|_{\infty}}{\left\| \boldsymbol{b}_{\mathrm{bad}} \right\|_{\infty}} \quad \geq \quad 1.$$

The matrix $\boldsymbol{\Phi}_{\text{opt}}$ has full column rank, so the column span of $\boldsymbol{\Phi}_{\text{opt}}$ contains a signal $\boldsymbol{s}_{\text{bad}}$ for which ${\boldsymbol{\Phi}_{\text{opt}}}^* \, \boldsymbol{s}_{\text{bad}} = \boldsymbol{b}_{\text{bad}}$. We conclude that

$$\rho(\boldsymbol{s}_{\text{bad}}) \quad = \quad \frac{\| {\boldsymbol{\Psi}_{\text{opt}}}^* \, \boldsymbol{s}_{\text{bad}} \|_\infty}{\| {\boldsymbol{\Phi}_{\text{opt}}}^* \, \boldsymbol{s}_{\text{bad}} \|_\infty} \quad \geq \quad 1.$$

Therefore, if we run Orthogonal Matching Pursuit with $\boldsymbol{s}_{\text{bad}}$ as input, the procedure chooses a nonoptimal atom in the first step. Since the representation of $\boldsymbol{s}_{\text{bad}}$ over Λ_{opt} is unique, this initial incorrect selection damns OMP from obtaining the sparsest representation of $\boldsymbol{s}_{\text{bad}}$. $\square$

5.2 Identifying Atoms from an Approximation

The analysis in the last section can be adapted to show that greedy algorithms may still be effective even when the input signal does not have a sparse representation. The argument in the proof of the Greedy Selection Lemma requires that the vector $\boldsymbol{r}$ lie in the column span of the optimal atoms. Unfortunately, this premise does not hold unless the signal can be represented completely with the optimal atoms. We need to develop a condition that pertains to arbitrary input signals. Therefore, we will study the performance of greedy selection when applied to a general signal minus a linear combination of atoms from a sub-dictionary Λ.

Fix an arbitrary signal $\boldsymbol{s}$. Let Λ index a sub-dictionary, and assume that $\text{ERC}(\Lambda) > 0$. Define $\boldsymbol{\Phi}_\Lambda$ to be the matrix whose columns are the atoms indexed by Λ, and let $\boldsymbol{\Psi}_\Lambda$ denote the matrix whose columns are the atoms indexed by $\Omega \setminus \Lambda$. The best approximation of the input signal over the atoms in Λ can be written as $\boldsymbol{a}_\Lambda = P_\Lambda \, \boldsymbol{s}$, where P_Λ is the orthogonal projector onto the column span of $\boldsymbol{\Phi}_\Lambda$.

Lemma 5.8 (General Recovery). *Suppose that $\boldsymbol{a}$ is a vector from the column span of $\boldsymbol{\Phi}_\Lambda$. A sufficient condition for $\rho(\boldsymbol{s}-\boldsymbol{a}) < 1$ is that*

$$\|\boldsymbol{s}-\boldsymbol{a}\|_2 \quad > \quad \left[1 \;+\; \left(\frac{\operatorname{maxcor}(\boldsymbol{s}-\boldsymbol{a}_\Lambda)\;\left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\operatorname{ERC}(\Lambda)}\right)^2\right]^{1/2} \|\boldsymbol{s}-\boldsymbol{a}_\Lambda\|_2 \,. \quad (5.2)$$

In words, the lemma says that greedy selection will choose an atom from Λ provided that the vector $\boldsymbol{a}$ compares unfavorably with the best approximation of the signal over Λ. For reference, the maximum correlation between a signal $\boldsymbol{r}$ and the dictionary is defined as

$$\operatorname{maxcor}(\boldsymbol{r}) \quad \stackrel{\text{def}}{=} \quad \frac{\|\boldsymbol{\Phi}^*\,\boldsymbol{r}\|_\infty}{\|\boldsymbol{r}\|_2}.$$

Proof. We may divide the ratio into two pieces, which we bound separately.

$$\begin{aligned}
\rho(\boldsymbol{s}-\boldsymbol{a}) \quad &= \quad \frac{\|\boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a})\|_\infty}{\|\boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a})\|_\infty} \\
&= \quad \frac{\|\boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a}_\Lambda) \;+\; \boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty}{\|\boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a}_\Lambda) \;+\; \boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty} \\
&\le \quad \frac{\|\boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a}_\Lambda)\|_\infty}{\|\boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty} \;+\; \frac{\|\boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty}{\|\boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty} \\
&\stackrel{\text{def}}{=} \quad \rho_{\text{err}} \;+\; \rho_{\text{opt}}.
\end{aligned} \quad (5.3)$$

The term $\boldsymbol{\Phi}_\Lambda{}^*\,(\boldsymbol{s}-\boldsymbol{a}_\Lambda)$ has vanished from the denominator since $(\boldsymbol{s}-\boldsymbol{a}_\Lambda)$ is orthogonal to the column span of $\boldsymbol{\Phi}_\Lambda$.

Since $(\boldsymbol{a}_\Lambda-\boldsymbol{a})$ lies in the column span of $\boldsymbol{\Phi}_\Lambda$, the Greedy Selection Lemma shows that

$$\rho_{\text{opt}} \quad \le \quad 1-\operatorname{ERC}(\Lambda). \quad (5.4)$$

Meanwhile, Proposition 4.2 yields the bound

$$\rho_{\text{err}} \quad \le \quad \frac{\|\boldsymbol{\Psi}_\Lambda{}^*\,(\boldsymbol{a}_\Lambda-\boldsymbol{a})\|_\infty}{\left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}^{-1}\,\|\boldsymbol{a}_\Lambda-\boldsymbol{a}\|_2}.$$

Since $(\boldsymbol{s} - \boldsymbol{a}_\Lambda)$ is orthogonal to the columns of $\boldsymbol{\Phi}_\Lambda$, we may rewrite the numerator of this fraction to obtain

$$\rho_{\text{err}} \quad \leq \quad \frac{\|\boldsymbol{\Phi}^* (\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\|\boldsymbol{a}_\Lambda - \boldsymbol{a}\|_2}. \tag{5.5}$$

Combining equations (5.3), (5.4), and (5.5), we discover that $\rho(\boldsymbol{s} - \boldsymbol{a}) < 1$ whenever

$$\frac{\|\boldsymbol{\Phi}^* (\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\|\boldsymbol{a}_\Lambda - \boldsymbol{a}\|_2} \quad < \quad \text{ERC}(\Lambda).$$

Rearranging this relation yields

$$\|\boldsymbol{\Phi}^* (\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \frac{\left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\text{ERC}(\Lambda)} \quad < \quad \|\boldsymbol{a}_\Lambda - \boldsymbol{a}\|_2 .$$

Square the inequality; add $\|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2^2$ to both sides; and apply the Pythagorean Theorem to the right-hand side to obtain

$$\|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2^2 \; + \; \left(\|\boldsymbol{\Phi}^* (\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \frac{\left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\text{ERC}(\Lambda)} \right)^2 \quad < \quad \|\boldsymbol{s} - \boldsymbol{a}\|_2^2 .$$

Factor the term $\|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2^2$ out from the left-hand side, and identify the maximum correlation of $(\boldsymbol{s} - \boldsymbol{a}_\Lambda)$ with the dictionary. We reach

$$\left[1 \; + \; \left(\frac{\text{maxcor}(\boldsymbol{s} - \boldsymbol{a}_\Lambda) \; \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger\right\|_{2,1}}{\text{ERC}(\Lambda)} \right)^2 \right] \|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2^2 \quad < \quad \|\boldsymbol{s} - \boldsymbol{a}\|_2^2 .$$

Take square roots to complete the argument. □

5.3 Error-Constrained Sparse Approximation

The General Recovery Lemma has an immediate application to the error-constrained sparse approximation problem. For an input signal $\boldsymbol{s}$ and an error

tolerance ε, the goal is to solve

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \ \|\boldsymbol{c}\|_0 \ + \ \tfrac{1}{2} \ \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \ \varepsilon^{-1}$$

$$\text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \ \leq \ \varepsilon. \quad \text{(ERROR)}$$

Suppose that the vector $\boldsymbol{c}_{\text{opt}}$ solves this mathematical program, and let $\boldsymbol{a}_{\text{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}}$ be the corresponding approximation of the target signal. Set $\Lambda_{\text{opt}} = \text{supp}(\boldsymbol{c}_{\text{opt}})$, and assume that $\text{ERC}(\Lambda_{\text{opt}}) > 0$.

Theorem 5.9. *Suppose that we halt Orthogonal Matching Pursuit as soon as the norm of the residual $\boldsymbol{r}_t$ satisfies the inequality*

$$\|\boldsymbol{r}_t\|_2 \quad \leq \quad \left[1 \ + \ \left(\frac{\text{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}) \ \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\right\|_{2,1}}{\text{ERC}(\Lambda_{\text{opt}})}\right)^2\right]^{1/2} \ \varepsilon.$$

Then every atom that OMP has chosen must be listed in Λ_{opt}.

In words, OMP can always compute a representation of the signal that uses only atoms from an optimal solution of (ERROR) and achieves an error only a constant factor worse. We will discuss this constant in more detail after we prove the theorem.

Proof. Suppose that, after the t-th iteration, Orthogonal Matching Pursuit has selected t atoms from Λ_{opt}. At the beginning of the first iteration, this hypothesis is satisfied trivially. We will develop a condition to guarantee that the atom selected in the $(t+1)$-st iteration is also optimal.

Denote by $\boldsymbol{r}_t$ the residual signal at the beginning of the $(t+1)$-st iteration. Orthogonal Matching Pursuit selects another optimal atom if and only if $\rho(\boldsymbol{r}_t) < 1$. The statement of the algorithm shows that the residual $\boldsymbol{r}_t$ equals

the target signal $\boldsymbol{s}$ minus a linear combination of the t atoms that have already been chosen. This linear combination lies in the column span of $\boldsymbol{\Phi}_{\text{opt}}$ on account of the induction hypothesis. Therefore, the General Recovery Lemma proves that OMP will select another atom from Λ_{opt} so long as the norm of the residual satisfies

$$\|\boldsymbol{r}_t\|_2 \quad > \quad \left[1 \; + \; \left(\frac{\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}) \; \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\right\|_{2,1}}{\operatorname{ERC}(\Lambda_{\text{opt}})}\right)^2\right]^{1/2} \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2 \,.$$

If we halt the algorithm as soon as this condition fails, then every atom we have chosen must be listed in Λ_{opt}, and we simultaneously obtain an upper bound on the norm of the residual or—what is the same—the approximation error. Since the approximation $\boldsymbol{a}_{\text{opt}}$ falls from a solution of (ERROR) with parameter ε, if follows that $\|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2 \leq \varepsilon$. □

The bracketed constant in Theorem 5.9 depends on a number of factors. In particular, the Exact Recovery Coefficient of Λ_{opt} must be positive, and the error bound improves significantly when $\operatorname{ERC}(\Lambda_{\text{opt}})$ is close to one. On account of the $(2, 1)$ matrix norm, the error bound increases when the covering radius of the sub-dictionary increases. The error also improves when the optimal residual is badly correlated with the dictionary. Return to Chapter 4 for a more detailed discussion of these quantities.

A practical question is to estimate the size of the bracketed constant. Suppose that the signal has a good approximation from a small sub-dictionary Λ_{opt} with a moderate Exact Recovery Coefficient. If the dictionary is incoherent, one expects that the maximum correlation of the residual with the dictionary is on the order of $d^{-1/2}$ and that the $(2, 1)$ matrix norm is on the order of $\sqrt{|\Lambda_{\text{opt}}|}$. Under these assumptions, the constant may well be less than $\sqrt{2}$.

An interesting way to appreciate this theorem is to apply it to an orthonormal dictionary. In this case, the Exact Recovery Coefficient and the $(2, 1)$ norm are both equal to one. The maximum correlation of the residual with the dictionary is never greater than one. Therefore, Theorem 5.9 implies that we should halt the algorithm as soon as the approximation error satisfies

$$\|\boldsymbol{r}_t\|_2 \quad \leq \quad \varepsilon\sqrt{2}.$$

It turns out that this bound cannot be improved.

Consider the signal

$$\boldsymbol{s} \quad = \quad \varepsilon\,\boldsymbol{\varphi}_\lambda \;+\; \varepsilon\,\boldsymbol{\varphi}_\xi,$$

which has norm $\varepsilon\sqrt{2}$ because the dictionary is orthonormal. The theorem recommends that we stop the algorithm before it starts and return a zero approximation. Can this advice be correct? In fact, it is. Note that the representation $\varepsilon\,\boldsymbol{\varphi}_\lambda$ gives an optimal ε-approximant of $\boldsymbol{s}$ that is supported on the sub-dictionary $\Lambda_{\mathrm{opt}} = \{\lambda\}$. But if we apply OMP to the signal $\boldsymbol{s}$, the first greedy selection could choose either λ or ξ. To prevent the algorithm from choosing the atom ξ, which does not belong to Λ_{opt}, we cannot even take one greedy step.

5.3.1 Coherence Estimates

To make Theorem 5.9 practical, we need some method for determining the stopping criterion *a priori*. Propositions 4.6 and 4.13 allow us to provide a result in terms of the cumulative coherence function.

Corollary 5.10. *Suppose that Λ_{opt} contains no more than m indices, and let us halt Orthogonal Matching Pursuit as soon as the norm of the residual*

satisfies

$$\|\boldsymbol{r}_t\|_2 \quad \leq \quad \left[1 \; + \; \frac{m\,[1 - \mu_1(m-1)]}{[1 - \mu_1(m-1) - \mu_1(m)]^2}\,\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}})^2\right]^{1/2} \varepsilon.$$

Then every atom that the algorithm has chosen must be listed in Λ_{opt}.

If no estimate is available for the maximum correlation of the residual with the dictionary, we may simply bound it above by one.

Using the properties of the cumulative coherence function, we can develop a simpler version of the last corollary.

Corollary 5.11. *Assume that* Λ_{opt} *contains no more than* m *indices, and suppose that* $\mu_1(m) < \frac{1}{3}$. *If we halt Orthogonal Matching Pursuit as soon as the norm of the residual satisfies*

$$\|\boldsymbol{r}_t\|_2 \quad \leq \quad \varepsilon \sqrt{1 \; + \; 6\,m\,\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}})^2},$$

then every atom that the algorithm has chosen must be listed in Λ_{opt}.

We will develop comparable results for the convex relaxation (R-ERROR) in Section 6.4.

5.4 Sparsity-Constrained Approximation

The General Recovery Lemma applies equally to the sparsity-constrained approximation problem. For an input signal $\boldsymbol{s}$ and a number m, the goal is to solve

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \qquad \text{subject to} \qquad \|\boldsymbol{c}\|_0 \leq m. \tag{SPARSE}$$

Suppose that the vector $\boldsymbol{c}_{\mathrm{opt}}$ solves the mathematical program, and let $\boldsymbol{a}_{\mathrm{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\mathrm{opt}}$ be the corresponding approximation of the input signal. Set $\Lambda_{\mathrm{opt}} = \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, and assume that $\operatorname{ERC}(\Lambda_{\mathrm{opt}}) > 0$. Note that $|\Lambda_{\mathrm{opt}}| \leq m$.

Theorem 5.12. *After m iterations, Orthogonal Matching Pursuit constructs an approximation $\boldsymbol{a}_m$ that satisfies the error bound*

$$\|\boldsymbol{s} - \boldsymbol{a}_m\|_2 \quad \leq \quad \left[1 \; + \; \left(\frac{\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}) \; \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\right\|_{2,1}}{\operatorname{ERC}(\Lambda_{\text{opt}})}\right)^2\right]^{1/2} \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2 \,.$$

Proof. Imagine that the condition (5.2) fails at iteration $(T + 1)$. Then, we have an upper bound on the T-term approximation error as a function of the optimal m-term approximation error. If we continue to apply OMP even after t exceeds T, the approximation error will only continue to decrease. If (5.2) holds in every iteration, then OMP identifies all m atoms in Λ_{opt}. Therefore, $\boldsymbol{a}_m = \boldsymbol{a}_{\text{opt}}$, so the theorem holds. □

Although OMP may not recover an optimal approximant $\boldsymbol{a}_{\text{opt}}$, it always constructs an approximant whose error lies within a constant factor of optimal. One might argue that the algorithm has the potential to inflate a moderate error into a large error. But a moderate error indicates that the signal does not have a good sparse representation over the dictionary, and so sparse approximation may not be an appropriate tool. In practice, if it is easy to find a nearly optimal solution, there is no reason to waste a lot of time and resources to reach the *ne plus ultra*. As it is said, "The best is the enemy of the good."

Suppose that the dictionary is orthonormal. It is not hard to show that OMP always produces m-term approximations that achieve the optimal m-term error. This theorem provides a somewhat weaker bound on the performance of the algorithm. At present, we do not know how to address this shortcoming.

5.4.1 Coherence Estimates

Placing a restriction on the cumulative coherence function leads to a much simpler statement of the result.

Corollary 5.13. *Orthogonal Matching Pursuit generates m-term approximants that satisfy the error bound*

$$\begin{aligned}\|\boldsymbol{s} - \boldsymbol{a}_m\|_2 \quad &\leq \\ &\left[1 \;+\; \frac{m\,[1-\mu_1(m-1)]}{[1-\mu_1(m-1)-\mu_1(m)]^2}\,\operatorname{maxcor}(\boldsymbol{s}-\boldsymbol{a}_{\mathrm{opt}})^2\right]^{1/2} \|\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}\|_2 .\end{aligned}$$

If the maximum correlation of the residual with the dictionary is unknown, we may bound it above by one.

We may develop a simpler version of this last corollary by positing a specific bound on the cumulative coherence function.

Corollary 5.14. *Assume that $\mu_1(m) \leq \frac{1}{3}$. Then OMP generates m-term approximants that satisfy*

$$\|\boldsymbol{s} - \boldsymbol{a}_m\|_2 \quad \leq \quad \sqrt{1 + 6\,m\,\operatorname{maxcor}(\boldsymbol{s}-\boldsymbol{a}_{\mathrm{opt}})^2}\;\|\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}\|_2 . \tag{5.6}$$

The hypothesis is in force whenever $m \leq \frac{1}{3}\,\mu^{-1}$.

5.5 Comparison with Previous Work

The literature contains a few results which prove that greedy methods can provide nearly optimal solutions to sparse approximation problems. Let us spend a moment to discuss how this work compares with the results in this chapter.

The most relevant work is due to Gilbert, Muthukrishnan, and Strauss [48]. They proved a theorem on the performance of OMP for (SPARSE) that is qualitatively similar to Corollary 5.14.

Theorem 5.15 (Gilbert–Muthukrishnan–Strauss). *Suppose that μ is the coherence parameter of the dictionary, and assume that $m < \frac{1}{8\sqrt{2}}\,\mu^{-1} - 1$. For an arbitrary input signal $\boldsymbol{s}$, Orthogonal Matching Pursuit generates an m-term approximant $\boldsymbol{a}_m$ that satisfies*

$$\|\boldsymbol{s} - \boldsymbol{a}_m\|_2 \quad \leq \quad 8\sqrt{m}\,\|\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}\|_2\,,$$

where $\boldsymbol{a}_{\mathrm{opt}}$ is the optimal m-term approximation of the signal.

There are a few differences between their result and Corollary 5.14. First, the present work takes advantage of the cumulative coherence function to provide sharper bounds for general dictionaries. Even when we phrase our results in terms of the coherence parameter, our bounds on m and the error constant are much better. Second, the techniques used to prove the theorems differ significantly.

The rest of the literature is not directly comparable to our work, but we will attempt to describe its flavor. For many years, the only positive theoretical results on greedy algorithms were convergence theorems. A typical convergence theorem states that that the norm of the residual $\boldsymbol{r}_t$ declines to zero as the number of iterations t approaches infinity. More sophisticated arguments provide bounds on the rate of convergence [104]. This theory is not really relevant to the sparse approximation problems that we stated in Chapter 2.

Natarajan provided the first proof that a greedy algorithm can approximately solve a well-defined computational problem. He studied a method

called *forward selection*. This algorithm is similar to OMP, but after choosing a new atom, it orthogonalizes the entire dictionary against that atom.

Theorem 5.16 (Natarajan). *Assume that $N \leq d$ and that the entire dictionary is linearly independent. Given an input signal $\boldsymbol{s}$, suppose that one uses the forward selection method to compute an approximation with error tolerance δ. The number of atoms chosen is no greater than*

$$\left\lceil 18\, m \, \left\| \boldsymbol{\Phi}^{\dagger} \right\|_{2,2}^{2} \ln \frac{\|\boldsymbol{s}\|_2}{\delta} \right\rceil,$$

where m is the number of atoms that participate in a solution of (ERROR) *with tolerance $\varepsilon = \frac{1}{2}\,\delta$.*

Note that Natarajan's paper is missing the crucial hypotheses that $N \leq d$ and that the dictionary is linearly independent. With a little effort, one may adapt his analysis to Orthogonal Matching Pursuit, and reduce the factor of 18 down to 8.

Couvreur and Bresler later obtained a qualitative result for another greedy algorithm, the backward elimination procedure [18]. Backward elimination begins by approximating the signal with all the atoms from the dictionary. Then it removes whatever atom contributes the least to the approximation and iterates until the number of atoms is sufficiently small.

Theorem 5.17 (Couvreur–Bresler). *Assume that $N \leq d$ and that the entire dictionary is linearly independent. The backward elimination procedure can solve* (SPARSE) *for every input signal that is sufficiently close to an exact superposition of atoms.*

The definition of "sufficiently close" depends on which superposition of atoms we are trying to recover, and no quantitative estimates are presently available.

Chapter 6

Analysis of Convex Relaxation Methods

Convex relaxation has been applied for over three decades to recover sparse representations of signals, and there is extensive empirical evidence that it often succeeds [103, 67, 78, 92, 8, 10, 98]. Although early theoretical results are beautiful, most lack practical value because they assume that the input signal admits a sparse approximation with zero error [31, 35, 54, 43, 29, 106, 53]. Other results are valid only in very proscribed settings [92, 33]. Unhappily, the theory of sparse approximation has been marred by the lack of a proof that convex relaxation can determine an optimal sparse approximation of a *general* input signal with respect to a *general* dictionary. This chapter proves that convex relaxation yields provably good solutions to the sparse approximation problems (EXACT), (SUBSET), and (ERROR).

In the first section, we study the convex relaxation (R-EXACT), which is also known as *Basis Pursuit*. We prove that the Exact Recovery Coefficient gives a sufficient condition for (R-EXACT) to determine the optimal sparse representation of a sparse signal. This result unifies and extends most of the recent literature on (R-EXACT) [31, 35, 54, 29]. This analysis will play a minor role in the analysis of the other convex relaxations.

In the second section, we develop the basic tools that we need for the study of the relaxations (R-SUBSET) and (R-ERROR). Both relaxations lead to similar Lagrangian functions. The major result is a condition which ensures

that any coefficient vector minimizing the Lagrangian function must be supported on a prescribed index set. This condition also depends on the Exact Recovery Coefficient, and we offer a plausible argument that the condition cannot be improved.

The third section applies the fundamental lemmata to the subset selection problem. Let us give the tenor of our major result. If the dictionary is incoherent and the threshold parameters are correctly chosen, then the solution to (R-SUBSET) identifies every significant atom from the solution to (SUBSET) and no others. To our knowledge, this type of result is unprecedented in the literature.

The fourth section applies the basic tools to the erro-constrained sparse approximation problem. Our major theorem proves that, under appropriate conditions, the solution to the relaxation (R-ERROR) for a given δ is at least as sparse as a solution to the sparse approximation problem (ERROR) for a smaller value of ε. We compare our theory against some result results of Donoho, Elad, and Temlyakov [30].

Most of the material in Section 6.1 is slated to appear in *IEEE Transactions on Information Theory*. It is drawn from the work [106], which is copyright 2004 by the IEEE, and it is reused with permission.

6.1 The Sparsest Representation of a Signal

We begin with the problem of recovering the sparsest representation of an input signal $\boldsymbol{s}$. For reference, the problem statement is

$$\min_{\boldsymbol{c} \in \mathbb{C}^{\Omega}} \|\boldsymbol{c}\|_0 \qquad \text{subject to} \qquad \boldsymbol{s} = \boldsymbol{\Phi}\,\boldsymbol{c}. \tag{EXACT}$$

Its convex relaxation is

$$\min_{\boldsymbol{b} \in \mathbb{C}^{\Omega}} \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \boldsymbol{s} = \boldsymbol{\Phi}\,\boldsymbol{b}. \tag{R-EXACT}$$

(R-EXACT) is sometimes known as Basis Pursuit [10].

Suppose that the coefficient vector $\boldsymbol{c}_{\text{opt}}$ is a solution of (EXACT). Assume that $\boldsymbol{c}_{\text{opt}}$ is supported on the index set Λ_{opt}, and denote the associated synthesis matrix by $\boldsymbol{\Phi}_{\text{opt}}$.

Theorem 6.1 (Relaxed Sparse Representation). *The coefficient vector $\boldsymbol{c}_{\text{opt}}$ is the unique solution of relaxation* (R-EXACT) *whenever*

$$\text{ERC}(\Lambda_{\text{opt}}) \quad > \quad 0.$$

For reference, the Exact Recovery Coefficient of Λ_{opt} is defined as

$$\text{ERC}(\Lambda_{\text{opt}}) \quad \stackrel{\text{def}}{=} \quad 1 \;-\; \max_{\psi} \; \left\| \boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\, \boldsymbol{\psi} \right\|_1,$$

where the maximization occurs over all nonoptimal atoms. Our proof requires a simple lemma about ℓ_1 norms.

Lemma 6.2. *Suppose that $\boldsymbol{v}$ is a vector with nonzero components and that A is a matrix whose columns do not have identical ℓ_1 norms. Then $\|\mathsf{A}\,\boldsymbol{v}\|_1 < \|\mathsf{A}\|_{1,1}\, \|\boldsymbol{v}\|_1$.*

Proof. Calculate that

$$\begin{aligned} \|\mathsf{A}\,\boldsymbol{v}\|_1 \quad &\leq \quad \sum_{j,k} |\mathsf{A}_{jk}|\; |\boldsymbol{v}_k| \quad = \quad \sum_{k} \|\mathsf{A}_k\|_1\; |\boldsymbol{v}_k| \\ &< \quad \max_{k} \|\mathsf{A}_k\|_1 \sum_{k} |\boldsymbol{v}_k| \quad = \quad \|\mathsf{A}\|_{1,1}\; \|\boldsymbol{v}\|_1 . \end{aligned}$$

□

We move on to the demonstration of the theorem.

Proof of Theorem 6.1. Assume that $\mathrm{ERC}(\Lambda_{\mathrm{opt}}) > 0$. Suppose that the target signal $\boldsymbol{s}$ has a representation over a sub-dictionary Λ_{alt} that is different from Λ_{opt}. Therefore, we may write $\boldsymbol{s} = \boldsymbol{\Phi}_{\mathrm{alt}}\, \boldsymbol{b}_{\mathrm{alt}}$ for some coefficient vector $\boldsymbol{b}_{\mathrm{alt}}$ in $\mathbb{C}^{\Lambda_{\mathrm{alt}}}$. Without loss of generality, assume that $\boldsymbol{b}_{\mathrm{alt}}$ contains no zero component. We will prove that $\|\boldsymbol{c}_{\mathrm{opt}}\|_1 < \|\boldsymbol{b}_{\mathrm{alt}}\|_1$.

Since Λ_{opt} provides a maximally sparse representation of $\boldsymbol{s}$, it follows that Λ_{alt} must contain at least one index ξ that does not appear in Λ_{opt}. Using the fact that $1 - \mathrm{ERC}(\Lambda_{\mathrm{opt}}) < 1$, we have

$$\left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{\varphi}_{\xi} \right\|_1 \quad < \quad 1.$$

Since $\left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{\varphi}_{\lambda} \right\|_1 = 1$ for each λ in Λ_{opt}, we have

$$\left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{\varphi}_{\omega} \right\|_1 \quad \leq \quad 1$$

for every index ω belonging to Ω.

The matrix $\boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger} \boldsymbol{\Phi}_{\mathrm{opt}}$ equals the identity, so we may rewrite the ℓ_1 norm of $\boldsymbol{c}_{\mathrm{opt}}$ as follows.

$$\begin{aligned} \|\boldsymbol{c}_{\mathrm{opt}}\|_1 &= \left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger} \boldsymbol{\Phi}_{\mathrm{opt}}\, \boldsymbol{c}_{\mathrm{opt}} \right\|_1 \\ &= \left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{s} \right\|_1 \\ &= \left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger} \boldsymbol{\Phi}_{\mathrm{alt}}\, \boldsymbol{b}_{\mathrm{alt}} \right\|_1 . \end{aligned}$$

Next, assume that the columns of $\boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger} \boldsymbol{\Phi}_{\mathrm{alt}}$ do not have identical ℓ_1 norms. Then the lemma furnishes the following bound:

$$\begin{aligned} \|\boldsymbol{c}_{\mathrm{opt}}\|_1 &< \left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger} \boldsymbol{\Phi}_{\mathrm{alt}} \right\|_{1,1} \|\boldsymbol{b}_{\mathrm{alt}}\|_1 \\ &= \max_{\omega \in \Lambda_{\mathrm{alt}}} \left\| \boldsymbol{\Phi}_{\mathrm{opt}}{}^{\dagger}\, \boldsymbol{\varphi}_{\omega} \right\|_1 \|\boldsymbol{b}_{\mathrm{alt}}\|_1 \\ &\leq \|\boldsymbol{b}_{\mathrm{alt}}\|_1 . \end{aligned}$$

On the other hand, suppose that the columns of $\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\boldsymbol{\Phi}_{\text{alt}}$ do have identical ℓ_1 norms. Then every one of these norms equals $\left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\boldsymbol{\varphi}_{\xi}\right\|_1$, which is strictly less than one. Therefore, we may calculate that

$$\begin{aligned}\|\boldsymbol{c}_{\text{opt}}\|_1 &\leq \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\boldsymbol{\Phi}_{\text{alt}}\right\|_{1,1} \|\boldsymbol{b}_{\text{alt}}\|_1 \\ &= \left\|\boldsymbol{\Phi}_{\text{opt}}{}^{\dagger}\,\boldsymbol{\varphi}_{\xi}\right\|_1 \|\boldsymbol{b}_{\text{alt}}\|_1 \\ &< \|\boldsymbol{b}_{\text{alt}}\|_1 .\end{aligned}$$

In words, any set of nonoptimal coefficients for representing the signal has strictly larger ℓ_1 norm than the optimal coefficients. Therefore, the solution of (R-EXACT) is unique, and it coincides with $\boldsymbol{c}_{\text{opt}}$. □

Theorem 5.2 shows that an identical condition is sufficient for Orthogonal Matching Pursuit to recover the optimal solution of (EXACT).

6.1.1 Coherence Estimates

Theorem 6.1 would not be very useful without a method for checking when the sufficient condition holds. Using Proposition 4.6 to bound the Exact Recovery Coefficient, we obtain the following corollary.

Corollary 6.3. *Suppose that*

$$\mu_1(m-1) + \mu_1(m) \quad < \quad 1.$$

Then the convex relaxation (R-EXACT) *solves the sparse approximation problem* (EXACT) *for every input signal that has an m-term representation over the dictionary.*

This result implies a weaker corollary that contains most of the results for (R-EXACT) from the recent literature.

Corollary 6.4 (Donoho–Elad [29], Gribonval–Nielsen [54]). *Suppose that*

$$m < \tfrac{1}{2}\left(\mu^{-1} + 1\right) \qquad \textit{or} \tag{6.1}$$

$$\mu_1(m) < \tfrac{1}{2}. \tag{6.2}$$

Then the solution of the relaxation (R-EXACT) *is identical with the solution of the sparse approximation problem* (EXACT) *for every input signal that has an m-term representation over the dictionary.*

The bound (6.1) appears in both [29, 54] as a sufficient condition for Basis Pursuit to recover sparse signals. The bound (6.2) also appears in [29].

In case that the dictionary has more structure, we can prove better sufficient conditions. For example, Corollary 4.9 yields the following result when the dictionary is a multi-ONB dictionary.

Corollary 6.5 (Gribonval–Nielsen [54]). *Suppose that the dictionary consists of J concatenated orthonormal bases with overall coherence μ, and assume that*

$$m < \left(\sqrt{2} - 1 + \frac{1}{2\,(J-1)}\right) \mu^{-1}.$$

Then (R-EXACT) *solves* (EXACT) *for every input signal that has an m-term representation over the dictionary. Moreover, all m-term representations are unique.*

Specializing this corollary to the case $J = 2$ yields the major theorem of [35].

Compare these results with the results for OMP that appear in Section 5.1.

6.2 Fundamental Lemmata

This section forges the basic tools we need to study the other convex relaxation methods. These convex relaxations all give rise to the Lagrangian function

$$L(\boldsymbol{b}; \gamma, \boldsymbol{s}) \quad \stackrel{\text{def}}{=} \quad \tfrac{1}{2} \, \|\boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b}\|_2^2 \; + \; \gamma \, \|\boldsymbol{b}\|_1 \, . \tag{6.3}$$

Therefore, we must develop a detailed understanding of the minimizers of this function.

Throughout this section, we will use the following notations. Fix the input signal $\boldsymbol{s}$ and the parameter γ so that the function L depends only on the coefficient vector $\boldsymbol{b}$. Let Λ index a sub-dictionary, which will be used to approximate the input signal. The best approximation of the input signal over the atoms in Λ can be written as $\boldsymbol{a}_\Lambda = P_\Lambda \, \boldsymbol{s}$. The coefficient vector $\boldsymbol{c}_\Lambda = {\boldsymbol{\Phi}_\Lambda}^\dagger \, \boldsymbol{s}$ may be used to synthesize $\boldsymbol{a}_\Lambda$. Although $\boldsymbol{c}_\Lambda$ lies in $\mathbb{C}^\Lambda$, we often extend it to $\mathbb{C}^\Omega$ by padding it with zeros.

6.2.1 The Correlation Condition Lemma

Suppose that the atoms outside Λ have small inner products (i.e., are weakly correlated) with the residual signal $(\boldsymbol{s} - \boldsymbol{a}_\Lambda)$. (The atoms inside Λ are orthogonal to the residual.) The following lemma shows that any coefficient vector which minimizes the objective function (6.3) must be supported inside Λ. This result is the soul of the analysis.

Lemma 6.6 (Correlation Condition). *Suppose that the maximum inner product between the residual signal and any atom fulfills the condition*

$$\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \quad \leq \quad \gamma \, \mathrm{ERC}(\Lambda). \tag{6.4}$$

Then any coefficient vector $\boldsymbol{b}_\star$ *that minimizes the function* (6.3) *must satisfy* $\operatorname{supp}(\boldsymbol{b}_\star) \subset \Lambda$.

Some remarks and corollaries are in order. First, observe that the lemma is worthless unless $\operatorname{ERC}(\Lambda)$ is positive. Return to Section 4.6 for a geometric description of this condition. Next, if the index set does satisfy $\operatorname{ERC}(\Lambda) > 0$, the sufficient condition (6.4) will always hold if γ is large enough. But if γ is too large, then $\boldsymbol{b}_\star = \mathbf{0}$. Third, the lemma is indifferent to the choice of Λ, so long as the inequality (6.4) holds for the residual $(\boldsymbol{s} - \boldsymbol{a}_\Lambda)$. Therefore, the support of $\boldsymbol{b}_\star$ is actually contained in the intersection of all such index sets. Fourth, we write the left-hand side of (6.4) as

$$\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty \quad = \quad \operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\ \|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2 .$$

It follows that the result is strongest when the magnitude of the residual and its maximum correlation with the dictionary are both small. Since the maximum correlation never exceeds one, we obtain a (much weaker) result that depends only on the magnitude of the residual.

Corollary 6.7. *Suppose that the approximation error satisfies*

$$\|\boldsymbol{s} - \boldsymbol{a}_\Lambda\|_2 \quad \leq \quad \gamma\ \operatorname{ERC}(\Lambda).$$

Then any coefficient vector $\boldsymbol{b}_\star$ *that minimizes the function* (6.3) *must be supported inside* Λ.

By normalizing the input signal, we may also obtain a result that depends only on the maximum correlation.

Corollary 6.8. *Suppose that the maximum correlation between the residual signal and the dictionary satisfies the bound*

$$\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_\Lambda) \quad \leq \quad \gamma \operatorname{ERC}(\Lambda).$$

If we scale the input signal to have unit norm and then compute a minimizer $\boldsymbol{b}_\star$ *of the function* (6.3)*, it follows that* $\boldsymbol{b}_\star$ *must be supported inside* Λ*.*

If the input signal can be expressed as an exact superposition of the atoms in the sub-dictionary, we reach another interesting corollary.

Corollary 6.9. *Suppose that the input signal has a representation using the atoms in* Λ *and that* $\operatorname{ERC}(\Lambda) > 0$*. For every positive* γ*, every coefficient vector that minimizes* (6.3) *must be supported inside* Λ*.*

One might wonder whether the condition $\operatorname{ERC}(\Lambda) > 0$ is really necessary to prove these results. The answer is a qualified affirmative. Section 6.2.4 offers a partial converse of Corollary 6.9.

6.2.2 Proof of Correlation Condition Lemma

A simple but powerful geometric idea underlies the proof of the lemma. The expense of using indices outside Λ is not compensated by the improvement in the approximation error. Therefore, any approximation involving other atoms can be projected onto the atoms in Λ to reduce the value of the objective function.

Proof of Lemma 6.6. We will be studying minimizers of the objective function

$$L(\boldsymbol{b}) \quad \stackrel{\text{def}}{=} \quad \tfrac{1}{2} \left\| \boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b} \right\|_2^2 \, + \, \gamma \left\| \boldsymbol{b} \right\|_1 . \tag{6.5}$$

Assume that there is a coefficient vector $\boldsymbol{b}_\star$ which minimizes (6.5) even though $\boldsymbol{b}_\star$ uses an index outside of Λ. That is, $\operatorname{supp}(\boldsymbol{b}_\star) \setminus \Lambda$ is non-empty. We will compare $\boldsymbol{b}_\star$ against the projected coefficient vector $\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star$, which is supported on Λ. Since $\boldsymbol{b}_\star$ minimizes the objective function, the inequality $L(\boldsymbol{b}_\star) \le L(\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star)$ must hold. Rearranging the terms of this relation gives

$$2\gamma \left[\|\boldsymbol{b}_\star\|_1 - \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star\right\|_1 \right] \quad \le \quad \|\boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\, \boldsymbol{b}_\star\|_2^2 - \|\boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b}_\star\|_2^2. \tag{6.6}$$

We will provide a lower bound on the left-hand side of (6.6) and an upper bound on the right-hand side. The correlation condition (6.4) will reverse the consequent inequality.

First, we claim that $\boldsymbol{\Phi}\, \boldsymbol{b}_\star$ does not lie in the range of $\boldsymbol{\Phi}_\Lambda$. Suppose that it did. Then $\boldsymbol{b}_\star$ and $\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star$ synthesize the same signal, and so the right-hand side of (6.6) equals zero. At the same time, Theorem 6.1 shows that the coefficient vector $\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star$ is the unique solution of the convex program

$$\min_{\boldsymbol{b}} \quad \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\, \boldsymbol{b} = \boldsymbol{\Phi}\, \boldsymbol{b}_\star.$$

In consequence, the left-hand side of (6.6) is positive. This combination of events is impossible.

Now we will develop a lower bound on the left-hand side of (6.6). To accomplish this, let us split the coefficient vector into two parts: $\boldsymbol{b}_\star = \boldsymbol{b}_\Lambda + \boldsymbol{b}_{\text{bad}}$. The first vector $\boldsymbol{b}_\Lambda$ contains the components with indices in Λ. The second vector contains the (undesirable) remaining components—those from $\Omega \setminus \Lambda$. This splitting yields the identity

$$\|\boldsymbol{b}_\star\|_1 - \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star\right\|_1 = \|\boldsymbol{b}_\Lambda\|_1 + \|\boldsymbol{b}_{\text{bad}}\|_1 - \left\|\boldsymbol{b}_\Lambda + \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_{\text{bad}}\right\|_1.$$

The upper triangle inequality allows us to cancel the terms involving $\boldsymbol{b}_\Lambda$:

$$\|\boldsymbol{b}_\star\|_1 - \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star\right\|_1 \quad \ge \quad \|\boldsymbol{b}_{\text{bad}}\|_1 - \left\|\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_{\text{bad}}\right\|_1. \tag{6.7}$$

Let us focus on the second term from the right-hand side of (6.7). To be rigorous, we temporarily define a diagonal matrix $\mathbf{R}_{\text{bad}}$ that zeros the components of a coefficient vector indexed by Λ and acts as the identity on the rest. Then

$$\begin{aligned} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_{\text{bad}} \right\|_1 &= \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi} \mathbf{R}_{\text{bad}}\, \boldsymbol{b}_{\text{bad}} \right\|_1 \\ &\leq \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi} \mathbf{R}_{\text{bad}} \right\|_{1,1} \|\boldsymbol{b}_{\text{bad}}\|_1 \\ &= \max_{\omega \notin \Lambda} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega \right\|_1 \|\boldsymbol{b}_{\text{bad}}\|_1 . \end{aligned}$$

Re-introducing this expression into (6.7) gives

$$\|\boldsymbol{b}_\star\|_1 \;-\; \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star \right\|_1 \quad\geq\quad \left[1 \;-\; \max_{\omega \notin \Lambda} \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega \right\|_1 \right] \|\boldsymbol{b}_{\text{bad}}\|_1 .$$

We identify the bracketed quantity as the Exact Recovery Coefficient of Λ to reach the lower bound

$$\|\boldsymbol{b}_\star\|_1 \;-\; \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\Phi}\, \boldsymbol{b}_\star \right\|_1 \quad\geq\quad \text{ERC}(\Lambda)\, \|\boldsymbol{b}_{\text{bad}}\|_1 . \tag{6.8}$$

Next, we need to provide an upper bound on the right-hand side of (6.6). Apply the Law of Cosines to the triangle formed by the signals $\boldsymbol{s}$ and $\boldsymbol{\Phi}\, \boldsymbol{b}_\star$ and $\mathbf{P}_\Lambda \boldsymbol{\Phi}\, \boldsymbol{b}_\star$ to obtain the identity

$$\begin{aligned} &\|\boldsymbol{s} - \mathbf{P}_\Lambda \boldsymbol{\Phi}\, \boldsymbol{b}_\star\|_2^2 \;-\; \|\boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b}_\star\|_2^2 \quad = \\ &\qquad 2\, \text{Re}\, \langle (\mathbf{I}_d - \mathbf{P}_\Lambda) \boldsymbol{\Phi}\, \boldsymbol{b}_\star, \boldsymbol{s} - \mathbf{P}_\Lambda \boldsymbol{\Phi}\, \boldsymbol{b}_\star \rangle \;-\; \|(\mathbf{I}_d - \mathbf{P}_\Lambda) \boldsymbol{\Phi}\, \boldsymbol{b}_\star\|_2^2 . \end{aligned} \tag{6.9}$$

Alternately, one might expand the squared norms on the right-hand side of (6.6) and simplify the consequent monstrosity. Since the matrix $(\mathbf{I}_d - \mathbf{P}_\Lambda)$ annihilates the atoms listed by Λ, it follows that

$$(\mathbf{I}_d - \mathbf{P}_\Lambda) \boldsymbol{\Phi}\, \boldsymbol{b}_\star \quad=\quad (\mathbf{I}_d - \mathbf{P}_\Lambda) \boldsymbol{\Phi}\, \boldsymbol{b}_{\text{bad}}. \tag{6.10}$$

Using (6.10) and the fact that $(\mathsf{I}_d - P_\Lambda)$ is an orthogonal projector, we may manipulate the inner product in (6.9):

$$\begin{aligned}\langle (\mathsf{I}_d - P_\Lambda)\boldsymbol{\Phi}\,\boldsymbol{b}_\star, \boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\,\boldsymbol{b}_\star\rangle &= \langle (\mathsf{I}_d - P_\Lambda)\boldsymbol{\Phi}\,\boldsymbol{b}_{\text{bad}}, \boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\,\boldsymbol{b}_\star\rangle \\ &= \langle \boldsymbol{b}_{\text{bad}}, \boldsymbol{\Phi}^*(\mathsf{I}_d - P_\Lambda)(\boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\,\boldsymbol{b}_\star)\rangle \\ &= \langle \boldsymbol{b}_{\text{bad}}, \boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\rangle\,.\end{aligned}$$

Substitute (6.10) and the expression for the inner product into equation (6.9) to obtain the identity

$$\begin{aligned}\|\boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\,\boldsymbol{b}_\star\|_2^2 \;-\; \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}_\star\|_2^2 \;=& \\ 2\,\operatorname{Re}\langle \boldsymbol{b}_{\text{bad}}, \boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\rangle \;-\; \|(\mathsf{I}_d - P_\Lambda)\boldsymbol{\Phi}\,\boldsymbol{b}_{\text{bad}}\|_2^2\,.&\end{aligned}$$

Since $\boldsymbol{\Phi}\,\boldsymbol{b}_\star$ does not lie in the range of $\boldsymbol{\Phi}_\Lambda$, neither does $\boldsymbol{\Phi}\,\boldsymbol{b}_{\text{bad}}$. So the second term on the right-hand side is strictly positive. Nevertheless, this term is negligible in comparison with the first term whenever $\boldsymbol{b}_{\text{bad}}$ is small. Therefore, we simply discard the second term, and we apply Hölder's Inequality to reach the upper bound

$$\|\boldsymbol{s} - P_\Lambda \boldsymbol{\Phi}\,\boldsymbol{b}_\star\|_2^2 \;-\; \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}_\star\|_2^2 \quad<\quad 2\,\|\boldsymbol{b}_{\text{bad}}\|_1\,\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty\,. \tag{6.11}$$

To proceed, we combine the bounds (6.8) and (6.11) into the inequality (6.6) to discover that

$$\gamma\,\operatorname{ERC}(\Lambda)\,\|\boldsymbol{b}_{\text{bad}}\|_1 \quad<\quad \|\boldsymbol{b}_{\text{bad}}\|_1\,\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty\,.$$

We have assumed that the support of the coefficient vector $\boldsymbol{b}_\star$ contains at least one index outside Λ, so the vector $\boldsymbol{b}_{\text{bad}}$ cannot be null. Therefore, we may divide the most recent inequality by $\|\boldsymbol{b}_{\text{bad}}\|_1$ to conclude that

$$\gamma\,\operatorname{ERC}(\Lambda) \quad<\quad \|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_\Lambda)\|_\infty\,.$$

If this inequality fails, then we must discard our hypothesis that the minimizer $\boldsymbol{b}_\star$ involves an index outside Λ. $\square$

6.2.3 Restricted Minimizers

The sufficient condition of Lemma 6.6 guarantees that the minimizer of (6.3) is supported on the index set Λ. Unfortunately, this does not even rule out the zero vector as a possible minimizer. Therefore, we must develop bounds on how much the minimizing coefficient vector can vary from the desired coefficient vector $\boldsymbol{c}_\Lambda$.

The argument involves standard techniques from convex analysis, but the complex setting demands an artifice. In this subsection only, we will decompose complex vectors into independent real and imaginary parts, i.e. $\boldsymbol{z} = \boldsymbol{x} + \mathrm{i}\,\boldsymbol{y}$. Then we may define the *complex gradient* of a real-valued function $f : \mathbb{C}^m \to \mathbb{R}$ as the vector

$$\nabla f \quad \stackrel{\text{def}}{=} \quad \nabla_{\boldsymbol{x}} f \;+\; \mathrm{i}\,\nabla_{\boldsymbol{y}} f.$$

Here, $\nabla_{\boldsymbol{x}}$ indicates the (real) derivative of f taken with respect to the real variables while fixing the imaginary variables; the definition of $\nabla_{\boldsymbol{y}}$ is similar. If f does not depend on the imaginary variables, then ∇f reduces to the usual real gradient. One may wish to read the article [113] for a more elegant treatment of complex gradients.

In the same spirit, the *complex subdifferential* of a convex function $f : \mathbb{C}^m \to \mathbb{R}$ at a complex vector $\boldsymbol{z}$ may be defined as

$$\partial f(\boldsymbol{z}) \quad \stackrel{\text{def}}{=} \quad \{\, \boldsymbol{g} \in \mathbb{C}^m \;:\; f(\boldsymbol{w}) \geq f(\boldsymbol{z}) + \operatorname{Re}\langle \boldsymbol{g}, \boldsymbol{w} - \boldsymbol{z} \rangle \quad \text{for all } \boldsymbol{w} \in \mathbb{C}^m \,\}.$$

The vectors contained in the subdifferential are called *subgradients*, and they provide affine lower bounds on the function. If the function has a complex

gradient at a point, the complex gradient gives the unique subgradient there. In addition, the complex subdifferential is additive, viz., $\partial(f_1 + f_2)(\boldsymbol{z}) = \partial f_1(\boldsymbol{z}) + \partial f_2(\boldsymbol{z})$. It is straightforward to verify that the complex subdifferential satisfies all the properties of real subdifferentials [88].

Lemma 6.10. *Suppose that the vector $\boldsymbol{b}_\star$ minimizes the objective function (6.3) over all coefficient vectors supported on Λ. A necessary and sufficient condition on such a minimizer is that*

$$\boldsymbol{c}_\Lambda - \boldsymbol{b}_\star \quad = \quad \gamma \, (\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \, \boldsymbol{g} \tag{6.12}$$

where the vector $\boldsymbol{g}$ is drawn from $\partial \, \|\boldsymbol{b}_\star\|_1$. Moreover, the minimizer is unique.

Fuchs developed this necessary and sufficient condition in the real setting using essentially the same method [44].

Proof. Apply the Pythagorean Theorem to (6.3) to see that minimizing L over coefficient vectors supported on Λ is equivalent to minimizing the function

$$F(\boldsymbol{b}) \quad \stackrel{\text{def}}{=} \quad \tfrac{1}{2} \, \|\boldsymbol{a}_\Lambda - \boldsymbol{\Phi}_\Lambda \, \boldsymbol{b}\|_2^2 \; + \; \gamma \, \|\boldsymbol{b}\|_1 \tag{6.13}$$

over coefficient vectors from $\mathbb{C}^\Lambda$. Recall that the atoms indexed by Λ form a linearly independent collection, so $\boldsymbol{\Phi}_\Lambda$ has full column rank. It follows that the quadratic term in (6.13) is strictly convex, and so the whole function F must also be strictly convex. Therefore, its minimizer is unique.

The function F is convex and unconstrained, so $\mathbf{0} \in \partial F(\boldsymbol{b}_\star)$ is a necessary and sufficient condition for the coefficient vector $\boldsymbol{b}_\star$ to minimize F. The complex gradient of the first term of F equals $(\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda) \, \boldsymbol{b}_\star - \boldsymbol{\Phi}_\Lambda{}^* \, \boldsymbol{a}_\Lambda$. From the additivity of subdifferentials, it follows that

$$(\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda) \, \boldsymbol{b}_\star \; - \; \boldsymbol{\Phi}_\Lambda{}^* \, \boldsymbol{a}_\Lambda \; + \; \gamma \, \boldsymbol{g} \quad = \quad \mathbf{0}$$

for some vector $\boldsymbol{g}$ drawn from the subdifferential $\partial \|\boldsymbol{b}_\star\|_1$. Since the atoms indexed by Λ are linearly independent, we may pre-multiply this relation by $(\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1}$ to reach

$$\boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{a}_\Lambda - \boldsymbol{b}_\star \quad = \quad \gamma \, (\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \boldsymbol{g}.$$

Apply the fact that $\boldsymbol{c}_\Lambda = \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{a}_\Lambda$ to reach the conclusion. □

Now we identify the subdifferential of the ℓ_1 norm. To that end, define the signum function as

$$\operatorname{sgn}(r \, \mathrm{e}^{\mathrm{i}\,\theta}) \quad \stackrel{\text{def}}{=} \quad \begin{cases} \mathrm{e}^{\mathrm{i}\,\theta} & \text{for } r > 0 \\ 0 & \text{for } r = 0. \end{cases}$$

One may extend the signum function to vectors by applying it to each component.

Proposition 6.11. *Let $\boldsymbol{z}$ be a complex vector. The complex vector $\boldsymbol{g}$ lies in the complex subdifferential $\partial \|\boldsymbol{z}\|_1$ if and only if*

- $|g_k| \leq 1$ *whenever* $z_k = 0$, *and*
- $g_k = \operatorname{sgn} z_k$ *whenever* $z_k \neq 0$.

In particular, $\|\boldsymbol{g}\|_\infty = 1$ *unless* $\boldsymbol{z} = \boldsymbol{0}$, *in which case* $\|\boldsymbol{g}\|_\infty \leq 1$.

Proof. The complex subdifferential of the absolute value function is given by

$$\partial \, |z| \quad = \quad \begin{cases} \{\operatorname{sgn} z\} & \text{for } z \neq 0 \\ \{z : |z| \leq 1\} & \text{for } z = 0. \end{cases}$$

This fact is manifest from the geometry: the absolute value function defines a cone that emanates from the origin of the complex plane and that has sides with *unit slope* in the radial direction.

Suppose that $\boldsymbol{g}$ is a subgradient of the ℓ_1 norm at the complex vector $\boldsymbol{z}$. Choose a component k of $\boldsymbol{z}$, and select a complex vector $\boldsymbol{w}$ that equals $\boldsymbol{z}$ in every other component. For the distinguished index, we have

$$|w_k| \quad \geq \quad |z_k| \;+\; \operatorname{Re}\left[\, g_k \,(w_k - z_k)^* \,\right] \qquad \text{for all } w_k \in \mathbb{C}. \tag{6.14}$$

In words, each component of $\boldsymbol{g}$ must be a subgradient of the absolute value function at the corresponding component of $\boldsymbol{z}$. On the other hand, assume that each component of $\boldsymbol{g}$ is a subgradient of the absolute value function at the corresponding component of $\boldsymbol{z}$. Then the inequality (6.14) holds for each index k. Sum these inequalities over k to see that

$$\|\boldsymbol{w}\|_1 \quad \geq \quad \|\boldsymbol{z}\|_1 \;+\; \operatorname{Re}\langle \boldsymbol{g}, \boldsymbol{w} - \boldsymbol{z} \rangle \qquad \text{for all complex } \boldsymbol{w}.$$

In words, $\boldsymbol{g}$ is a subgradient of the ℓ_1 norm at $\boldsymbol{z}$. □

At last, we may develop bounds on how much a solution to the restricted problem varies from the desired solution $\boldsymbol{c}_\Lambda$.

Corollary 6.12 (Upper Bounds). *Suppose that the vector $\boldsymbol{b}_\star$ minimizes the function* (6.3) *over all coefficient vectors supported on Λ. The following bounds are in force:*

$$\|\boldsymbol{c}_\Lambda - \boldsymbol{b}_\star\|_\infty \quad \leq \quad \gamma \, \left\| (\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \right\|_{\infty,\infty} \tag{6.15}$$

$$\|\boldsymbol{\Phi}_\Lambda \,(\boldsymbol{c}_\Lambda - \boldsymbol{b}_\star)\|_2 \quad \leq \quad \gamma \, \left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \right\|_{2,1}. \tag{6.16}$$

Proof. We begin with the necessary and sufficient condition

$$\boldsymbol{c}_\Lambda - \boldsymbol{b}_\star \quad = \quad \gamma \,(\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \, \boldsymbol{g} \tag{6.17}$$

where $\boldsymbol{g} \in \partial \|\boldsymbol{b}_\star\|_1$. To obtain (6.15), we take the ℓ_∞ norm of (6.17) and apply the usual estimate:

$$\|\boldsymbol{b}_\star - \boldsymbol{c}_\Lambda\|_\infty \quad = \quad \gamma \left\|(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1}\boldsymbol{g}\right\|_\infty \quad \le \quad \gamma \left\|(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1}\right\|_{\infty,\infty} \|\boldsymbol{g}\|_\infty .$$

Proposition 6.11 shows that $\|\boldsymbol{g}\|_\infty \le 1$, which proves the result.

To develop the second bound (6.16), we pre-multiply (6.17) by the matrix $\boldsymbol{\Phi}_\Lambda$ and compute the Euclidean norm:

$$\|\boldsymbol{\Phi}_\Lambda(\boldsymbol{b}_\star - \boldsymbol{c}_\Lambda)\|_2 \quad = \quad \gamma \left\|(\boldsymbol{\Phi}_\Lambda{}^\dagger)^*\boldsymbol{g}\right\|_2 \quad \le \quad \gamma \left\|(\boldsymbol{\Phi}_\Lambda{}^\dagger)^*\right\|_{\infty,2} \|\boldsymbol{g}\|_\infty .$$

As before, $\|\boldsymbol{g}\|_\infty \le 1$. Finally, we apply the identity (4.3) to switch from the $(\infty, 2)$ operator norm to the $(2, 1)$ operator norm. □

For the record, we present a lower bound.

Corollary 6.13 (Lower Bounds). *Suppose that the vector $\boldsymbol{b}_\star$ minimizes the function* (6.3) *over all coefficient vectors supported on Λ. For every index λ in* $\operatorname{supp}(\boldsymbol{b}_\star)$*,*

$$|\boldsymbol{c}_{\text{opt}}(\lambda) \;-\; \boldsymbol{b}_\star(\lambda)| \quad \ge \quad \gamma \left[2 \;-\; \left\|(\boldsymbol{\Phi}_\Lambda{}^*\boldsymbol{\Phi}_\Lambda)^{-1}\right\|_{\infty,\infty}\right].$$

We will not use this result, so we leave its proof as an exercise for the reader.

6.2.4 Is the ERC Necessary?

Let Λ index a sub-dictionary for which $\text{ERC}(\Lambda) > 0$, and suppose that the input signal can be written as a superposition of atoms from Λ. It follows from the Correlation Condition Lemma and from Corollary 6.12 that, for all sufficiently small γ, the minimizer of the function (6.3) has support equal to Λ. The following theorem shows that this type of result cannot generally hold if $\text{ERC}(\Lambda) < 0$.

Theorem 6.14. *Suppose that* $\mathrm{ERC}(\Lambda) < 0$. *Then we may construct an input signal that has an exact representation using the atoms in* Λ *and yet the minimizer of the function* (6.3) *is* not *supported on* Λ *when* γ *is small.*

Proof. Since $\mathrm{ERC}(\Lambda) < 0$, there must exist an atom $\boldsymbol{\varphi}_\omega$ for which $\left\| \boldsymbol{\Phi}_\Lambda{}^\dagger \, \boldsymbol{\varphi}_\omega \right\|_1 > 1$ even though $\omega \notin \Lambda$. Perversely, we select the input signal to be $\boldsymbol{s} = \mathsf{P}_\Lambda \, \boldsymbol{\varphi}_\omega$. To synthesize $\boldsymbol{s}$ using the atoms in Λ, we use the coefficient vector $\boldsymbol{c}_\Lambda = \boldsymbol{\Phi}_\Lambda{}^\dagger \boldsymbol{\varphi}_\omega$.

According to Corollary 6.12, the minimizer $\boldsymbol{b}_\star$ of the function

$$L(\boldsymbol{b}) \quad = \quad \tfrac{1}{2} \, \|\boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b}\|_2^2 \; + \; \gamma \, \|\boldsymbol{b}\|_1$$

over all coefficient vectors supported on Λ must satisfy

$$\|\boldsymbol{c}_\Lambda - \boldsymbol{b}_\star\|_\infty \leq \gamma \, \left\| (\boldsymbol{\Phi}_\Lambda{}^* \boldsymbol{\Phi}_\Lambda)^{-1} \right\|_{\infty,\infty}.$$

Since $\|\boldsymbol{c}_\Lambda\|_1 > 1$ by construction, we may choose γ small enough that the bound $\|\boldsymbol{b}_\star\|_1 > 1$ is also in force. Define the corresponding approximation $\boldsymbol{a}_\star = \boldsymbol{\Phi}\, \boldsymbol{b}_\star$.

Now we construct a parameterized coefficient vector

$$\boldsymbol{b}(t) \quad \stackrel{\text{def}}{=} \quad (1-t)\, \boldsymbol{b}_\star \; + \; t\, \mathbf{e}_\omega \qquad \text{for } t \text{ in } [0,1].$$

We have used $\mathbf{e}_\omega$ to denote the ω-th canonical basis vector in $\mathbb{C}^\Omega$. For positive t, it is clear that the support of $\boldsymbol{b}(t)$ is not contained in Λ. We will prove that $L(\boldsymbol{b}(t)) < L(\boldsymbol{b}_\star)$ for small, positive t. Since $\boldsymbol{b}_\star$ minimizes L over all coefficient vectors supported on Λ, no global minimizer of L can be supported on Λ.

To proceed, calculate that

$$\begin{aligned} L(\boldsymbol{b}(t)) \quad &= \quad \tfrac{1}{2} \, \|(\boldsymbol{s} - \boldsymbol{a}_\star) \; + \; t\,(\boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega)\|_2^2 \; + \; \gamma \, \|(1-t)\, \boldsymbol{b}_\star \; + \; t\, \mathbf{e}_\omega\|_1 \\ &= \quad \tfrac{1}{2} \, \|\boldsymbol{s} - \boldsymbol{a}_\star\|_2^2 \; + \; t \, \operatorname{Re} \langle \boldsymbol{s} - \boldsymbol{a}_\star, \boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega \rangle \\ &\qquad\qquad + \; \tfrac{1}{2} t^2 \, \|\boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega\|_2^2 \; + \; \gamma\,(1-t)\, \|\boldsymbol{b}_\star\|_1 \; - \; t\gamma. \end{aligned}$$

Differentiate this expression with respect to t and evaluate the derivative at $t = 0$:

$$\left.\frac{\mathrm{d}L(\boldsymbol{b}(t))}{\mathrm{d}t}\right|_{t=0} = \operatorname{Re}\langle \boldsymbol{s} - \boldsymbol{a}_\star, \boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega \rangle + \gamma\,(1 - \|\boldsymbol{b}_\star\|_1).$$

By construction of $\boldsymbol{b}_\star$, the second term is negative. The first term is non-positive because

$$\begin{aligned}\langle \boldsymbol{s} - \boldsymbol{a}_\star, \boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega \rangle &= \langle P_\Lambda(\boldsymbol{s} - \boldsymbol{a}_\star), \boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega \rangle \\ &= \langle \boldsymbol{s} - \boldsymbol{a}_\star, P_\Lambda(\boldsymbol{a}_\star - \boldsymbol{\varphi}_\omega) \rangle \\ &= \langle \boldsymbol{s} - \boldsymbol{a}_\star, \boldsymbol{a}_\star - \boldsymbol{s} \rangle \\ &= -\,\|\boldsymbol{s} - \boldsymbol{a}_\star\|_2^2\,.\end{aligned}$$

Therefore, the derivative is negative, and $L(\boldsymbol{b}(t)) < L(\boldsymbol{b}(0))$ for small, positive t. Since $\boldsymbol{b}(0) = \boldsymbol{b}_\star$, the proof is complete. □

6.3 Subset Selection

We are now prepared to tackle the convex relaxation of the subset selection problem. For a fixed signal $\boldsymbol{s}$ and a threshold τ, we must solve

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2^2 + \tau^2\,\|\boldsymbol{c}\|_0\,. \tag{SUBSET}$$

Let us state a proposition that describes the solutions of the subset selection problem. This result will help us compare the behavior of convex relaxation against the behavior of the original problem.

Proposition 6.15. *Fix an input signal $\boldsymbol{s}$, and choose a threshold τ. Suppose that the coefficient vector $\boldsymbol{c}_{\mathrm{opt}}$ solves* (SUBSET)*, and set $\boldsymbol{a}_{\mathrm{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\mathrm{opt}}$.*

- *For each $\lambda \in \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, we have $|\boldsymbol{c}_{\mathrm{opt}}(\lambda)| \geq \tau$.*

- *For each* $\omega \notin \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$*, we have* $|\langle \boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}, \boldsymbol{\varphi}_\omega \rangle| \leq \tau$*.*

For continuity, we postpone the proof until Section 6.3.3.

The natural convex relaxation of (SUBSET) is

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \quad \tfrac{1}{2} \, \|\boldsymbol{s} - \boldsymbol{\Phi}\, \boldsymbol{b}\|_2^2 \, + \, \gamma \, \|\boldsymbol{b}\|_1 \, . \tag{R-SUBSET}$$

Our theory will supply the correct relationship between γ and τ. One should not expect to solve the subset selection problem directly by means of convex relaxation because the ℓ_1 penalty has the effect of shrinking the optimal coefficients. Statisticians have exploited this property to improve the variance of their estimators [32, 105]. In the present setting, it is a nuisance. Our hope is that the coefficient vector which solves the convex relaxation has the same *support* as the optimal coefficient vector. Then we may solve the original subset selection problem by projecting the signal onto the atoms indexed by the support.

6.3.1 Main Theorem

If the dictionary is sufficiently incoherent and the threshold parameters are correctly chosen, then we can prove that convex relaxation identifies every significant atom from the solution to the subset selection problem and no others. This type of result is unprecedented in the literature.

To simplify the statement of the results, we will extract some of the hypotheses. Fix an input signal $\boldsymbol{s}$, and choose a threshold parameter τ. Suppose that the coefficient vector $\boldsymbol{c}_{\mathrm{opt}}$ is a solution to the subset selection problem (SUBSET) with threshold τ and that $\boldsymbol{a}_{\mathrm{opt}} = \boldsymbol{\Phi}\, \boldsymbol{c}_{\mathrm{opt}}$ is the corresponding approximation of the signal. Let $\Lambda_{\mathrm{opt}} = \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, and define the corresponding synthesis matrix $\boldsymbol{\Phi}_{\mathrm{opt}}$. Assume, moreover, that $\operatorname{ERC}(\Lambda_{\mathrm{opt}}) > 0$.

Theorem 6.16 (Relaxed Subset Selection). *Suppose that the coefficient vector $\boldsymbol{b}_\star$ solves the convex relaxation* (R-SUBSET) *with threshold*

$$\gamma = \tau \,/\, \mathrm{ERC}(\Lambda_{\mathrm{opt}}).$$

Then it follows that

- *the relaxation never selects a nonoptimal atom because*

$$\mathrm{supp}(\boldsymbol{b}_\star) \subset \mathrm{supp}(\boldsymbol{c}_{\mathrm{opt}});$$

- *the solution of the relaxation is nearly optimal since*

$$\|\boldsymbol{b}_\star - \boldsymbol{c}_{\mathrm{opt}}\|_\infty \quad \leq \quad \frac{\|(\boldsymbol{\Phi}_{\mathrm{opt}}{}^* \boldsymbol{\Phi}_{\mathrm{opt}})^{-1}\|_{\infty,\infty}}{\mathrm{ERC}(\Lambda_{\mathrm{opt}})}\ \tau;$$

- *in particular,* $\mathrm{supp}(\boldsymbol{b}_\star)$ *contains every index* λ *for which*

$$|\boldsymbol{c}_{\mathrm{opt}}(\lambda)| \quad > \quad \frac{\|(\boldsymbol{\Phi}_{\mathrm{opt}}{}^* \boldsymbol{\Phi}_{\mathrm{opt}})^{-1}\|_{\infty,\infty}}{\mathrm{ERC}(\Lambda_{\mathrm{opt}})}\ \tau;$$

- *the solution of the convex relaxation is unique.*

We postpone the proof to Section 6.3.3 so that we may discuss the consequences of the theorem. On account of Proposition 6.15, every nonzero coefficient in $\boldsymbol{c}_{\mathrm{opt}}$ has a magnitude of at least τ. Therefore, convex relaxation will not miss a coefficient unless it barely reaches the threshold τ. Observe that the result depends on the Exact Recovery Coefficient of the optimal sub-dictionary Λ_{opt}, so the non-optimal atoms must not resemble the optimal atoms too strongly. The theorem also prefers that the dual system of the sub-dictionary exhibit small pairwise inner products. From the discussion in Chapter 4, we see that

convex relaxation performs best when the dictionary is a good packing of lines in projective space.

As a reality check, let us apply Theorem 6.16 to the case where dictionary is orthonormal. Every sub-dictionary has an Exact Recovery Coefficient of one. In addition, the inverse Gram matrix equals the identity, so its (∞, ∞) norm is one. Therefore, the theorem advises that we solve the convex relaxation with $\gamma = \tau$, and it states that the solution will involve just those atoms whose optimal coefficients are strictly larger than τ. In other words, the theorem describes the behavior of the soft-thresholding operator with cutoff τ. See Section 3.2.3 for some related comments.

6.3.2 Coherence Estimates

Using the cumulative coherence function, we may develop versions of the theorem that depend only on the size of the optimal index set.

Corollary 6.17. *Suppose the coefficient vector $\boldsymbol{b}_\star$ solves the convex relaxation* (R-SUBSET) *with threshold*

$$\gamma \quad = \quad \frac{1 \ - \ \mu_1(m-1)}{1 \ - \ \mu_1(m-1) \ - \ \mu_1(m)} \ \tau.$$

Then $\operatorname{supp}(\boldsymbol{b}_\star)$ *is contained in* $\operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$*, and*

$$\|\boldsymbol{b}_\star - \boldsymbol{c}_{\mathrm{opt}}\|_\infty \quad \leq \quad \frac{\tau}{1 \ - \ \mu_1(m-1) \ - \ \mu_1(m)}.$$

This result follows immediately from the estimates in Propositions 4.5 and 4.6. By positing a specific bound for $\mu_1(m)$, we may develop a more quantitative result.

Corollary 6.18. *Assume that that the support of $\boldsymbol{c}_{\mathrm{opt}}$ indexes m atoms or fewer, where $\mu_1(m) \leq \frac{1}{3}$, and suppose that the coefficient vector $\boldsymbol{b}_\star$ solves*

the convex relaxation (R-SUBSET) *with threshold* $\gamma = 2\,\tau$. *It follows that* $\operatorname{supp}(\boldsymbol{b}_\star) \subset \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$ *and that* $\|\boldsymbol{b}_\star - \boldsymbol{c}_{\mathrm{opt}}\|_\infty \leq 3\,\tau$.

Of course, smaller bounds will give better conclusions. Finally, note that, for signals with a sparse representation, the theorem reduces to the known result that convex relaxation can recover all the atoms in the signal.

Corollary 6.19 (Fuchs [44]). *Suppose that* $\mathrm{ERC}(\Lambda_{\mathrm{opt}}) > 0$, *and assume that the input signal can be expressed exactly with the atoms in* Λ_{opt}. *For sufficiently small* γ, *the solution to the convex relaxation* (R-SUBSET) *has support equal to* Λ_{opt}.

6.3.3 Proof of Main Theorem

We begin with a proof of Proposition 6.15. This result also plays a role in the proof of the main theorem.

Proof of Proposition 6.15. For a given threshold τ and input signal $\boldsymbol{s}$, suppose that the coefficient vector $\boldsymbol{c}_{\mathrm{opt}}$ is a solution of the subset selection problem (SUBSET). Let $\boldsymbol{a}_{\mathrm{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\mathrm{opt}}$.

Take an index ω outside $\operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, and let $\mathsf{P}_{\mathrm{opt}}$ denote the orthogonal projector onto the atoms listed in $\operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$. Adding the atom $\boldsymbol{\varphi}_\omega$ to the approximation would diminish the squared error by exactly

$$\frac{|\langle \boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}, \boldsymbol{\varphi}_\omega \rangle|^2}{\|\mathsf{P}_{\mathrm{opt}}\,\boldsymbol{\varphi}_\omega\|_2^2}. \tag{6.18}$$

This quantity must be less than or equal to τ^2, or else we could immediately construct a solution to the subset selection problem that is strictly better than $\boldsymbol{c}_{\mathrm{opt}}$. Every atom has unit Euclidean norm, and projections can only attenuate the Euclidean norm. It follows that $\|\mathsf{P}_{\mathrm{opt}}\,\boldsymbol{\varphi}_\omega\|_2^2 \leq 1$, and $|\langle \boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}, \boldsymbol{\varphi}_\omega \rangle| \leq \tau$.

Choose an index λ inside $\operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, and let P denote the orthogonal projector onto the span of the atoms listed by $\operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}}) \setminus \{\lambda\}$. Removing the atom $\boldsymbol{\varphi}_\lambda$ from the approximation would increase the squared error by exactly

$$|\boldsymbol{c}_{\mathrm{opt}}(\lambda)|^2 \, \|(\mathsf{I} - \mathsf{P})\boldsymbol{\varphi}_\lambda\|_2^2 \,.$$

This quantity must be at least τ^2. Since $(\mathsf{I} - \mathsf{P})$ is an orthogonal projector, $\|(\mathsf{I} - \mathsf{P})\boldsymbol{\varphi}_\lambda\|_2^2 \leq 1$. We conclude that $|\boldsymbol{c}_{\mathrm{opt}}(\lambda)| \geq \tau$. □

Now we turn our attention to the proof of Theorem 6.16. This result involves a straightforward application of the fundamental lemmata.

Proof of Theorem 6.16. Suppose that the coefficient vector $\boldsymbol{c}_{\mathrm{opt}}$ is a solution to the subset selection problem (SUBSET) with threshold parameter τ. The associated approximation of the signal is $\boldsymbol{a}_{\mathrm{opt}} = \boldsymbol{\Phi}\, \boldsymbol{c}_{\mathrm{opt}}$. Define $\Lambda_{\mathrm{opt}} = \operatorname{supp}(\boldsymbol{c}_{\mathrm{opt}})$, and denote the corresponding synthesis matrix by $\boldsymbol{\Phi}_{\mathrm{opt}}$.

Let us develop an upper bound on the inner product between any atom and the residual vector $(\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}})$. First, note that every atom indexed by Λ_{opt} has a zero inner product with the residual since $\boldsymbol{a}_{\mathrm{opt}}$ is the best approximation of $\boldsymbol{s}$ using the atoms in Λ_{opt}. Choose $\omega \notin \Lambda_{\mathrm{opt}}$. Then Proposition 6.15 shows that $|\langle \boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}}, \boldsymbol{\varphi}_\omega \rangle| \leq \tau$. This relation holds for all $\omega \in \Omega$, and so

$$\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}})\|_\infty \quad \leq \quad \tau. \tag{6.19}$$

The Correlation Condition Lemma states that, for any threshold γ satisfying

$$\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_{\mathrm{opt}})\|_\infty \quad \leq \quad \gamma \, \mathrm{ERC}(\Lambda_{\mathrm{opt}}),$$

the solution $\boldsymbol{b}_\star$ to the convex relaxation (R-SUBSET)is supported on Λ_{opt}. Using the inequality (6.19), we determine that the choice

$$\gamma \quad = \quad \frac{\tau}{\text{ERC}(\Lambda_{\text{opt}})}$$

is sufficient to ensure that $\text{supp}(\boldsymbol{b}_\star) \subset \Lambda_{\text{opt}}$. The uniqueness of the minimizer $\boldsymbol{b}_\star$ follows from Lemma 6.10 since Λ_{opt} indexes a linearly independent collection of atoms. From Corollary 6.12, we obtain the upper bound

$$\|\boldsymbol{c}_{\text{opt}} - \boldsymbol{b}_\star\|_\infty \quad \leq \quad \gamma \, \left\| (\boldsymbol{\Phi}_{\text{opt}}{}^* \boldsymbol{\Phi}_{\text{opt}})^{-1} \right\|_{\infty,\infty}.$$

For any index λ at which $|\boldsymbol{c}_{\text{opt}}(\lambda)| > \gamma \, \|(\boldsymbol{\Phi}_{\text{opt}}{}^* \boldsymbol{\Phi}_{\text{opt}})^{-1}\|_{\infty,\infty}$, it follows that the corresponding coefficient $\boldsymbol{b}_\star(\lambda)$ must be nonzero. □

6.4 Error-Constrained Sparse Approximation

Let $\boldsymbol{s}$ be an input signal, and choose an error tolerance ε. In this section, we consider the error-constrained sparse approximation problem

$$\min_{\boldsymbol{c} \in \mathbb{C}^\Omega} \|\boldsymbol{c}\|_0 + \tfrac{1}{2} \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \, \varepsilon^{-1} \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{c}\|_2 \leq \varepsilon. \quad \text{(ERROR)}$$

We attempt to produce an error-constrained sparse approximation by using the convex relaxation

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2 \leq \delta. \qquad \text{(R-ERROR)}$$

Our theory will supply the correct relationship between δ and ε. Note that if $\delta \geq \|\boldsymbol{s}\|_2$ then the optimal solution to the convex program is the zero vector. When δ is smaller, the approximation error is always equal to δ. Therefore, the minimizer of the convex program will rarely solve the sparse approximation problem (ERROR). To improve the approximation obtained by relaxation, one should project the signal onto the atoms indexed by the support of the minimal coefficient vector.

6.4.1 Main Theorem

Our major theorem proves that, under appropriate conditions, the solution to the relaxation (R-ERROR) for a given δ is at least as sparse as a solution to the sparse approximation problem (ERROR) for a smaller value of ε.

To make the statement of the result more transparent, let us extract some of the hypotheses. Fix an input signal $\boldsymbol{s}$, and choose an error tolerance ε. Suppose that the coefficient vector $\boldsymbol{c}_{\text{opt}}$ solves the sparse approximation problem (ERROR) with tolerance ε, and let the corresponding approximation of the signal be $\boldsymbol{a}_{\text{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}}$. Define the optimal index set $\Lambda_{\text{opt}} = \text{supp}(\boldsymbol{c}_{\text{opt}})$, and let $\boldsymbol{\Phi}_{\text{opt}}$ be the associated synthesis matrix. Assume, moreover, that $\text{ERC}(\Lambda_{\text{opt}}) > 0$.

Theorem 6.20 (Relaxed Sparse Approximation). *Suppose that the coefficient vector $\boldsymbol{b}_\star$ solves the convex relaxation* (R-ERROR) *with an error tolerance*

$$\delta \quad \geq \quad \left[1 \; + \; \left(\frac{\text{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}) \; \left\| \boldsymbol{\Phi}_{\text{opt}}{}^{\dagger} \right\|_{2,1}}{\text{ERC}(\Lambda_{\text{opt}})} \right)^2 \right]^{1/2} \varepsilon. \tag{6.20}$$

Then it follows that

- *this solution is at least as sparse as $\boldsymbol{c}_{\text{opt}}$ since $\text{supp}(\boldsymbol{b}_\star) \subset \text{supp}(\boldsymbol{c}_{\text{opt}})$;*
- *yet $\boldsymbol{b}_\star$ is no sparser than a solution of the sparse approximation problem with tolerance δ;*
- *the coefficient vector $\boldsymbol{b}_\star$ is nearly optimal since*

$$\|\boldsymbol{b}_\star - \boldsymbol{c}_{\text{opt}}\|_2 \leq \delta \; \left\| \boldsymbol{\Phi}_{\text{opt}}{}^{\dagger} \right\|_{2,2};$$

- *the relaxation has no other solution.*

As usual, we postpone the proof until we have completed the commentary. Notice that the result depends strongly on several geometric properties of the dictionary. First, the dependence on the Exact Recovery Coefficient shows that non-optimal atoms must not resemble the optimal atoms too strongly. Second, the presence of the $(2, 1)$ operator norm shows that the optimal atoms should cover their span well. Third, the optimal solution is easiest to recover when the residual left over after approximation is badly correlated with the dictionary. Chapter 4 treats these factors in more detail.

If the dictionary is an orthonormal basis, Theorem 6.20 is the best possible result of its type. To see this, observe that the bound on the error tolerance is the same as the bound in Theorem 5.9. Therefore, the signal that we constructed in Section 5.3 to confuse Orthogonal Matching Pursuit will also confuse (R-ERROR).

6.4.2 Coherence Estimates

Using the cumulative coherence function, we can develop results that do not depend on the specific index set Λ_{opt} at all.

Corollary 6.21. *Fix an input signal, and choose an error tolerance ε. Suppose that $\boldsymbol{c}_{\text{opt}}$ solves the sparse approximation problem* (ERROR) *with tolerance ε and that* $\operatorname{supp}(\boldsymbol{c}_{\text{opt}})$ *contains m indices. Assume that the incoherence condition $\mu_1(m-1) + \mu_1(m) < 1$ holds, and pick the parameter δ so that*

$$\delta \quad \geq \quad \left[1 \; + \; \frac{m\,[1 - \mu_1(m-1)]}{[1 - \mu_1(m-1) - \mu_1(m)]^2}\, \operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}})^2\right]^{1/2} \varepsilon.$$

It follows that

- *the* unique *solution $\boldsymbol{b}_\star$ to the convex relaxation* (R-ERROR) *with error tolerance δ is supported inside* $\operatorname{supp}(\boldsymbol{c}_{\text{opt}})$*;*

- *the coefficient vector* $\boldsymbol{b}_\star$ *is nearly optimal since*

$$\|\boldsymbol{b}_\star - \boldsymbol{c}_{\text{opt}}\|_2 \leq \delta \,/\, \sqrt{1 - \mu_1(m-1)};$$

- *yet* $\boldsymbol{b}_\star$ *is no sparser than a solution of the sparse approximation problem with tolerance* δ*.*

If we do not have any prior knowledge about the maximum correlation between the optimal residual and the dictionary, then we may bound it above by one.

If we posit a bound on the cumulative coherence, we may reach a more quantitative result.

Corollary 6.22. *Fix an input signal, and choose an error tolerance* ε*. Suppose that an optimal solution to* (ERROR) *with tolerance* ε *requires* m *atoms or fewer, where* m *satisfies the incoherence condition* $\mu_1(m) \leq \frac{1}{3}$*. Select* $\delta \geq \varepsilon \sqrt{1 + 6\,m\,\operatorname{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}})^2}$*. It follows that the unique solution to the convex relaxation* (R-ERROR) *with tolerance* δ *involves a subset of the optimal atoms. This solution diverges in Euclidean norm from the optimal coefficient vector by no more than* $\delta\sqrt{3/2}$*. Moreover, it is no sparser than a solution of the sparse approximation problem with tolerance* δ*.*

6.4.3 Comparison with Other Work

Related results have very recently been provided in [30]. Their first result bounds how far the solution of (R-ERROR) may lie from the solution of (ERROR). Assume that the dictionary has coherence μ, and fix an input signal $\boldsymbol{s}$.

Theorem 6.23 (Donoho–Elad–Temlyakov). *Suppose that* $\boldsymbol{c}_{\text{opt}}$ *is an optimal solution of* (ERROR) *with tolerance* ε*, and suppose that* $\boldsymbol{b}_\star$ *is the optimal*

solution of (R-ERROR) *with tolerance* δ. *Assume that*

$$m \quad \stackrel{\text{def}}{=} \quad |\operatorname{supp}(\boldsymbol{c}_{\text{opt}})| \quad \leq \quad \tfrac{1}{4}(\mu^{-1} + 1).$$

Then the solution of (ERROR) *is unique, and we have the bound*

$$\|\boldsymbol{c}_{\text{opt}} - \boldsymbol{b}_\star\|_2^2 \quad \leq \quad \frac{(\varepsilon + \delta)^2}{1 \;-\; (4\,m - 1)\,\mu}.$$

Their result holds for all values of ε and δ, which gives a stability bound for the solution of the relaxation that depends only on the solution of (ERROR) being sufficiently sparse. Meanwhile, our results require that δ be somewhat larger than ε.

It is a little difficult to compare their result with Corollary 6.22 because the hypotheses are somewhat different. Nevertheless, if we choose $m = \frac{1}{4}(\mu^{-1} + 1)$, their theorem gives the useless upper bound of infinity on the deviation between the coefficient vectors. Provided that $\mu \leq \frac{1}{3}$ and that δ is sufficiently large in comparison with ε, Corollary 6.22 yields a finite upper bound on the deviation.

The second relevant result from [30] is a theorem that identifies the support of a solution to (R-ERROR).

Theorem 6.24 (Donoho–Elad–Temlyakov). *Suppose that an optimal solution of* (ERROR) *with tolerance* ε *requires* m *atoms or fewer, where* $m < \frac{1}{2}\,\mu^{-1}$, *and define* $\beta = m\,\mu$. *Then the solution of the relaxation* (R-ERROR) *with tolerance*

$$\delta \quad = \quad \frac{\varepsilon\,\sqrt{m}\,\sqrt{1 - \beta}}{1 - 2\,\beta}$$

involves a subset of the optimal atoms.

This theorem is directly comparable with Corollary 6.21 once we bound the cumulative coherence function in terms of the coherence parameter. Our result advises that we select

$$\delta \quad = \quad \varepsilon \sqrt{1 \ + \ m \, \frac{1 - \beta}{(1 - 2\,\beta)^2}}$$

when performing the relaxation. The theorem of Donoho et al., therefore, is slightly better than the corollary we have matched it against.

Unfortunately, their work mixes up all the different factors that play a role in the sparse approximation problem. Meanwhile, we have identified several different geometric quantities that determine when the relaxation succeeds. Another shortcoming of their approach is that all their results are stated in terms of the coherence parameter. Our results are much more general. We only use the coherence parameter to check when our conditions are actually in force.

6.4.4 Proof of the Main Theorem

Now we prove the result. The argument is rather more difficult than that of Theorem 6.16 because it involves Karush–Kuhn–Tucker conditions.

Proof of Theorem 6.20. Let $\boldsymbol{s}$ be an input signal, and suppose that $\boldsymbol{c}_{\text{opt}}$ solves the sparse approximation problem (ERROR) with error tolerance ε. Denote the optimal approximation as $\boldsymbol{a}_{\text{opt}} = \boldsymbol{\Phi}\,\boldsymbol{c}_{\text{opt}}$. Let $\Lambda_{\text{opt}} = \text{supp}(\boldsymbol{c}_{\text{opt}})$, and define $\boldsymbol{\Phi}_{\text{opt}}$ to be the associated synthesis matrix. Assume also that $\|\boldsymbol{s}\|_2 > \delta$, or else the zero vector is the unique solution of the relaxation (R-ERROR). It is self-evident that the solution of the relaxation can be no sparser than a solution of the sparse approximation problem with tolerance δ.

To prove the theorem, we will find a coefficient vector supported on Λ_{opt} and a corresponding Lagrange multiplier that give a potential solution of the relaxation. We will argue that the condition (6.20) guarantees that this coefficient vector actually minimizes the relaxation. Then we will demonstrate that this coefficient vector gives the unique minimizer. As a coda, we will estimate how much the solution of the relaxation varies from the optimal coefficient vector.

A coefficient vector $\boldsymbol{b}_\star$ solves the convex relaxation

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2 \leq \delta \tag{R-ERROR}$$

if and only if the Karush–Kuhn–Tucker conditions are satisfied [88]. That is, there exists a Lagrange multiplier $\gamma_\star$ for which

$$\boldsymbol{b}_\star \in \arg\min_{\boldsymbol{b}} \tfrac{1}{2}\,\|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2^2 + \gamma_\star\,\|\boldsymbol{b}\|_1 \tag{6.21}$$

$$\|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}_\star\|_2 = \delta \tag{6.22}$$

$$\gamma_\star > 0. \tag{6.23}$$

The KKT conditions are both necessary and sufficient because the objective function and constraint set are convex. Note that (6.22) and (6.23) hold because $\|\boldsymbol{s}\|_2 > \delta$ implies that the error constraint is strictly binding. Since the Lagrange multiplier $\gamma_\star$ is positive, we have transferred it to the ℓ_1 term in (6.21) to simplify the application of the formulae we have developed.

Let $(\boldsymbol{b}_\star, \gamma_\star)$ be a solution to the restricted problem

$$\min_{\text{supp}(\boldsymbol{b}) \subset \Lambda_{\text{opt}}} \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \|\boldsymbol{s} - \boldsymbol{\Phi}\,\boldsymbol{b}\|_2 \leq \delta. \tag{6.24}$$

The hypothesis $\|\boldsymbol{s}\|_2 > \delta$ implies that the error constraint in (6.24) is strictly binding, so (6.22) and (6.23) are both in force. Applying the Pythagorean

Theorem to (6.22), we obtain the identity

$$\|\boldsymbol{a}_{\text{opt}} - \boldsymbol{\Phi}_{\text{opt}}\, \boldsymbol{b}_\star\|_2 \quad = \quad \left(\delta^2 \;-\; \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2^2\right)^{1/2}. \tag{6.25}$$

Corollary 6.12 furnishes the estimate

$$\|\boldsymbol{a}_{\text{opt}} - \boldsymbol{\Phi}_{\text{opt}}\, \boldsymbol{b}_\star\|_2 \quad \leq \quad \gamma_\star \left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\right\|_{2,1}.$$

Introducing (6.25) into this relation, we obtain a lower bound on the multiplier:

$$\gamma_\star \quad \geq \quad \left(\delta^2 \;-\; \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2^2\right)^{1/2} \left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\right\|_{2,1}^{-1}. \tag{6.26}$$

Meanwhile, the Correlation Condition Lemma gives a sufficient condition,

$$\gamma_\star \quad \geq \quad \frac{\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}})\|_\infty}{\text{ERC}(\Lambda_{\text{opt}})}, \tag{6.27}$$

which ensures that any coefficient vector satisfying (6.21) is supported on Λ_{opt}. Combine (6.26) into (6.27), and rearrange to obtain

$$\|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2^2 \;+\; \frac{\|\boldsymbol{\Phi}^*(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}})\|_\infty^2 \left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\right\|_{2,1}^2}{\text{ERC}(\Lambda_{\text{opt}})^2} \quad \leq \quad \delta^2.$$

Factor $\|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2^2$ out from the left-hand side, identify the maximum correlation of $(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}})$ with the dictionary, and take square roots to reach

$$\left[1 \;+\; \left(\frac{\text{maxcor}(\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}) \left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\right\|_{2,1}}{\text{ERC}(\Lambda_{\text{opt}})}\right)^2\right]^{1/2} \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2 \quad \leq \quad \delta.$$

Since $\boldsymbol{a}_{\text{opt}}$ is an approximation of $\boldsymbol{s}$ with error less than or equal to ε, the hypothesis (6.20) is a sufficient condition for the pair $(\boldsymbol{b}_\star, \gamma_\star)$ to satisfy all three KKT conditions (6.21), (6.22), and (6.23). It follows that our coefficient vector $\boldsymbol{b}_\star$ gives a solution to the convex relaxation (R-ERROR).

Now we will demonstrate that the coefficient vector $\boldsymbol{b}_\star$ provides the *unique* minimizer of the convex relaxation. This requires some work because we have not proven that every solution of the convex program is necessarily supported on Λ_{opt}.

Suppose that $\boldsymbol{b}_{\text{alt}}$ is another coefficient vector that solves (R-ERROR). First, we argue that $\boldsymbol{\Phi}\,\boldsymbol{b}_{\text{alt}} = \boldsymbol{\Phi}\,\boldsymbol{b}_\star$ by assuming the contrary. The condition (6.22) must hold at every solution, so the signals $\boldsymbol{\Phi}\,\boldsymbol{b}_{\text{alt}}$ and $\boldsymbol{\Phi}\,\boldsymbol{b}_\star$ both lie on a Euclidean sphere of radius δ centered at the input signal $\boldsymbol{s}$. Since Euclidean balls are strictly convex, the signal $\frac{1}{2}\,\boldsymbol{\Phi}\,(\boldsymbol{b}_{\text{alt}} + \boldsymbol{b}_\star)$ is strictly closer than δ to the input signal. Thus $\frac{1}{2}\,(\boldsymbol{b}_{\text{alt}} + \boldsymbol{b}_\star)$ cannot be a solution of the convex relaxation. But the solutions to a convex program always form a convex set, which is a contradiction. In consequence, any alternate solution $\boldsymbol{b}_{\text{alt}}$ synthesizes the same signal as $\boldsymbol{b}_\star$. Moreover, $\boldsymbol{b}_{\text{alt}}$ and $\boldsymbol{b}_\star$ share the same ℓ_1 norm because they both solve (R-ERROR). Under our hypothesis that $\text{ERC}(\Lambda_{\text{opt}}) > 0$, Theorem 6.1 shows that $\boldsymbol{b}_\star$ is the *unique* solution to the problem

$$\min_{\boldsymbol{b} \in \mathbb{C}^\Omega} \quad \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \boldsymbol{\Phi}\,\boldsymbol{b} \;=\; \boldsymbol{\Phi}\,\boldsymbol{b}_\star.$$

Thus $\boldsymbol{b}_{\text{alt}} = \boldsymbol{b}_\star$. We conclude that $\boldsymbol{b}_\star$ is the unique minimizer of the convex relaxation.

Finally, let us estimate how far $\boldsymbol{b}_\star$ varies from $\boldsymbol{c}_{\text{opt}}$. We begin with the equation (6.25), which can be written

$$\|\boldsymbol{\Phi}_{\text{opt}}\,(\boldsymbol{c}_{\text{opt}} - \boldsymbol{b}_\star)\|_2 \quad = \quad \left(\delta^2 \;-\; \|\boldsymbol{s} - \boldsymbol{a}_{\text{opt}}\|_2^2\right)^{1/2}.$$

The right-hand side clearly does not exceed δ, while the left-hand side may be bounded below as follows.

$$\left\|\boldsymbol{\Phi}_{\text{opt}}{}^\dagger\right\|_{2,2}^{-1} \|\boldsymbol{c}_{\text{opt}} - \boldsymbol{b}_\star\|_2 \quad \leq \quad \|\boldsymbol{\Phi}_{\text{opt}}\,(\boldsymbol{c}_{\text{opt}} - \boldsymbol{b}_\star)\|_2.$$

Combine the two bounds and rearrange to complete the argument. □

Chapter 7

Numerical Construction of Packings

Our analysis of sparse approximation algorithms demonstrates that sparse approximation problems can be solved efficiently whenever the dictionary has low coherence. As we saw in Chapter 4, the coherence parameter has an immediate interpretation as the packing radius of the dictionary in a projective space. In light of these observations, it is natural to ask how we may construct good packings in projective space. This optimization problem is highly nonconvex, so we expect that it is difficult to solve. In this chapter, we develop an elegant numerical method that can be used to approach projective packing and related problems.

7.1 Overview

We will study packing problems set in a compact metric space $\mathbb{M}$ with distance function $\mathrm{dist}_{\mathbb{M}}$. Recall that the *packing radius* of a finite set $\mathscr{X}$ is the minimum distance between some pair of distinct points drawn from $\mathscr{X}$. That is,

$$\mathrm{pack}_{\mathbb{M}}(\mathscr{X}) \quad \stackrel{\mathrm{def}}{=} \quad \min_{m \neq n} \quad \mathrm{dist}_{\mathbb{M}}(x_m, x_n).$$

In other words, the packing radius of a set is the largest open ball that can be centered at one point of the set without encompassing any other point. An *optimal packing* of N points is an ensemble $\mathscr{X}$ that solves the mathematical

program

$$\max_{|\mathscr{X}|=N} \quad \operatorname{pack}_{\mathbb{M}}(\mathscr{X})$$

where $|\cdot|$ returns the cardinality of a finite set. The optimal packing problem is guaranteed to have a solution because the metric space is compact and the objective is a continuous function of the ensemble $\mathscr{X}$.

In this chapter, we will consider a *feasibility problem* closely connected with optimal packing. Given a number ρ, the goal is to produce a set of N points for which

$$\operatorname{pack}_{\mathbb{M}}(\mathscr{X}) \quad \geq \quad \rho. \tag{7.1}$$

This problem is notoriously difficult to solve because it is highly nonconvex, and it is even more difficult to determine the maximum value of ρ for which the feasibility problem is soluble. This maximum value of ρ corresponds with the radius of an optimal packing.

7.1.1 Our Approach

We will consider the feasibility problem (7.1) in several different compact metric spaces, but the same basic algorithm applies in each. Here is a high-level description of our approach.

First, we show that each configuration of points is associated with a matrix whose entries are related to the inter-point distances. Then we prove that a configuration solves the feasibility problem (7.1) if and only if its matrix possesses both a structural property and a spectral property. The overall algorithm consists of the following steps.

1. Randomly choose an initial configuration, and construct its matrix.

2. Alternately enforce the structural condition and the spectral condition in hope of reaching a matrix that satisfies both.

3. Extract a configuration of points from the output matrix.

To our knowledge, the numerical approach to packing via alternating projection is completely new. Although we are aware of several other numerical methods for packing [57, 111, 1], these techniques all seem to rely on ideas from nonlinear programming.

Flexibility is the major advantage of alternating projection. In this chapter, we demonstrate that we can solve many different types of feasibility problems by appropriate modification of the basic technique. Some of these problems have never before been studied numerically. Moreover, we believe that the possibilities of the method have not been exhausted, and that it will see other fruitful applications in the future. The major disadvantage of alternating projection is that it converges slowly.

7.1.2 Outline

As a first illustration of our method, we apply it to construct packings of points on the surface of the Euclidean sphere. This problem has been studied for about 75 years, and hundreds of putatively optimal packings have been recorded by N. J. A. Sloane and his colleagues [96]. We return to this problem for two reasons. First, it provides the simplest way to explain our basic ideas. Second, it allows us to compare the output of our algorithm against the best packings that have been discovered over the last 75 years. We will see that the alternating projection approach is extremely competitive, but it sometimes fails to reproduce the best results known.

Afterward, we adapt the method to construct good packings of points in projective spaces. This is the dictionary design problem for sparse approximation. Sloane and his colleagues have also tabulated putatively optimal packings in the real projective space [95]. Our experiments indicate that alternating projection can match many of their best packings. We have also constructed many new packings of points in the complex projective space.

The natural generalization of line packing is subspace packing. This problem is set in a *Grassmannian space*, which is the collection of all subspaces of fixed dimension in a Euclidean space. The distance between two subspaces is a function of the principal angles between them. Our algorithm can match many of the best packings of real subspaces with respect to the *chordal distance* that Sloane has recorded [95]. We have also constructed many packings of complex subspaces with respect to the chordal distance. Then, we show how to construct Grassmannian packings with respect to two other metrics, the *spectral distance* and the *Fubini–Study distance*. These experiments underscore the versatility of the algorithm.

Afterward, we review some results from the literature that provide bounds on the packing radius in the metric spaces that we have considered. These bounds allow us to conclude that some of our packings are essentially optimal.

In a concluding section, we present a synopsis of our experimental results, and we discuss some directions for future research. At the end of the chapter, we provide a collection of tables and figures that give details about the best packings that we obtained.

7.2 Packing on Spheres

Imagine that twelve mutually inimical nations build their capital cities on the surface of a featureless globe. Being concerned about missile strikes, they wish to locate the closest pair of cities as far apart as possible. In other words, the problem requests an optimal packing of points on the surface of a two-dimensional sphere. This is often referred to as *Tammes' Problem* in honor of a Dutch botanist who raised the question in 1930 [102]. We address Tammes' Problem first because it provides the most transparent illustration of our *modus operandi*.

7.2.1 The Sphere

Let $\mathbb{R}^d$ denote the d-dimensional real inner-product space. The usual symmetric inner product will be written as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^* \boldsymbol{x}$, where * denotes the (conjugate) transpose operator. Although we are working with real vectors at present, the remaining sections of the chapter will involve complex vectors, so we prefer to use the conjugate transpose to unify the presentation. The squared Euclidean norm falls from the inner product: $\|\boldsymbol{x}\|_2^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$.

The $(d-1)$-dimensional sphere $\mathbb{S}^{d-1}$ is defined as the set of all unit vectors in $\mathbb{R}^d$.

$$\mathbb{S}^{d-1} \quad \stackrel{\text{def}}{=} \quad \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}.$$

We measure the distance between two points on the sphere as the Euclidean distance of the chord joining them, which is also known as the *chordal distance*.

$$\operatorname{dist}_{\mathbb{S}}(\boldsymbol{x}, \boldsymbol{y}) \quad \stackrel{\text{def}}{=} \quad \|\boldsymbol{x} - \boldsymbol{y}\|_2 .$$

Equipped with this distance, the sphere becomes a compact metric space.

7.2.2 Packings and Matrices

Suppose that we wish to produce a configuration of N points in $\mathbb{S}^{d-1}$ with packing radius ρ. We may represent each configuration of points with a collection $\mathscr{X}$ of unit vectors drawn from $\mathbb{R}^d$.

$$\mathscr{X} \quad = \quad \{\boldsymbol{x}_n\}_{n=1}^N.$$

The packing radius of $\mathscr{X}$ is defined as

$$\operatorname{pack}_{\mathbb{S}}(\mathscr{X}) \quad \stackrel{\text{def}}{=} \quad \min_{m \neq n} \quad \operatorname{dist}_{\mathbb{S}}(\boldsymbol{x}_n, \boldsymbol{x}_m) \quad = \quad \min_{m \neq n} \quad \|\boldsymbol{x}_n - \boldsymbol{x}_m\|_2 ,$$

and the feasibility problem requests a configuration $\mathscr{X}$ for which

$$\min_{m \neq n} \quad \|\boldsymbol{x}_n - \boldsymbol{x}_m\|_2 \quad \geq \quad \rho.$$

It is better to reorganize this condition so that it depends only on the inner products between pairs of vectors. Therefore, we seek a collection $\mathscr{X}$ for which

$$\max_{m \neq n} \quad \langle \boldsymbol{x}_n, \boldsymbol{x}_m \rangle \quad \leq \quad \mu \tag{7.2}$$

where the parameter μ satisfies the relationship $\mu = 1 - \frac{1}{2}\,\rho^2$.

Form the column vectors from $\mathscr{X}$ into a $d \times N$ matrix:

$$\mathsf{X} \quad \stackrel{\text{def}}{=} \quad \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_N \end{bmatrix}.$$

In the sequel, we will not distinguish between the matrix X and the collection of its columns. To detect whether X solves the feasibility problem (7.2), one must compute the inner products between its columns. It is better to work with a matrix that registers these inner products explicitly. The obvious candidate is the *Gram matrix* of X,

$$\mathsf{G} \quad \stackrel{\text{def}}{=} \quad \mathsf{X}^* \mathsf{X}.$$

The (m, n) entry of the Gram matrix is precisely the inner product $\langle \boldsymbol{x}_n, \boldsymbol{x}_m \rangle$.

In fact, we may reformulate the feasibility problem purely in terms of the Gram matrix. Suppose that the configuration X satisfies (7.2) with parameter μ. Then its Gram matrix G must have the following six properties:

1. G is (real) symmetric.
2. G has a unit diagonal.
3. $-1 \leq g_{mn} \leq \mu$ whenever $m \neq n$.
4. G is positive semi-definite.
5. G has rank d or less.
6. G has trace N.

Some of these properties are redundant, but we have listed them separately for reasons soon to become apparent.

Conversely, suppose that a matrix G satisfies Properties 1–6. Then it is always possible to extract a configuration of N points that solves (7.2). More precisely, there exists a real $d \times N$ matrix X with unit-norm columns so that $\mathsf{G} = \mathsf{X}^* \mathsf{X}$. The off-diagonal entries of G do not exceed μ, so the inner products between distinct columns of X do not exceed μ. We conclude that Properties 1–6 characterize solutions of the feasibility problem with parameter μ.

For reference, a positive semi-definite (PSD) matrix is defined to have nonnegative (real) eigenvalues. It can be shown that every PSD matrix is conjugate symmetric. Thus, a real PSD matrix is always real symmetric. To indicate that a matrix A is PSD, we write $\mathsf{A} \succcurlyeq \mathsf{0}$.

7.2.3 Alternating Projection

Observe that Properties 1–3 are *structural* properties. By this, we mean that they constrain the entries of the matrix directly. Properties 4–6, on the other hand, are *spectral* properties. That is, they control the eigenvalues of the matrix. It is not easy to enforce structural and spectral properties simultaneously, so we must resort to half measures. Starting from an initial matrix, our algorithm will alternately enforce Properties 1–3 and then Properties 4–6 in hope of reaching a matrix that satisfies all six properties at once.

To be more rigorous, let us define the structural constraint set

$$\begin{aligned}\mathscr{H}(\mu) \quad &\stackrel{\text{def}}{=} \quad \{\mathsf{H} \in \mathbb{R}^{N \times N} : \mathsf{H} = \mathsf{H}^*, \quad \operatorname{diag} \mathsf{H} = \mathbf{e}, \quad \text{and} \\ &\qquad\qquad -1 \le h_{mn} \le \mu \text{ for } m \ne n\}. \quad (7.3)\end{aligned}$$

The symbol $\mathbf{e}$ represents a conformal vector of ones. Although the structural constraint set depends on the value of the feasibility parameter μ, we will usually eliminate μ from the notation for simplicity. We also define the spectral constraint set

$$\mathscr{G} \quad \stackrel{\text{def}}{=} \quad \{\mathsf{G} \in \mathbb{R}^{N \times N} : \mathsf{G} \succcurlyeq \mathsf{0}, \quad \operatorname{rank} \mathsf{G} \le d, \quad \text{and} \quad \operatorname{trace} \mathsf{G} = N\}. \quad (7.4)$$

Both constraint sets are compact. The structural constraint set $\mathscr{H}$ is convex, but the spectral constraint set is not.

To solve the feasibility problem (7.2), we must find a matrix that lies in the intersection of $\mathscr{G}$ and $\mathscr{H}(\mu)$. This section will present a high-level statement of our approach. The next two sections will provide implementation details. We remind the reader that the Frobenius norm of a matrix is defined as

$$\|\mathsf{A}\|_{\mathrm{F}} \quad \stackrel{\text{def}}{=} \quad \left[\sum\nolimits_{m,n} |a_{mn}|^2\right]^{1/2}$$

Algorithm 7.1 (Alternating Projection).

INPUT:

- An $N \times N$ (conjugate) symmetric matrix $\mathsf{G}^{(0)}$.
- The maximum number of iterations T.

OUTPUT:

- G_{out} is an $N \times N$ matrix that belongs to $\mathscr{G}$ and that has a unit diagonal.

PROCEDURE:

1. Initialize $t = 0$.
2. Determine a matrix $\mathsf{H}^{(t)}$ that solves
$$\min_{\mathsf{H} \in \mathscr{H}} \quad \left\| \mathsf{H} - \mathsf{G}^{(t)} \right\|_{\mathrm{F}}.$$
3. Determine a matrix $\mathsf{G}^{(t+1)}$ that solves
$$\min_{\mathsf{G} \in \mathscr{G}} \quad \left\| \mathsf{G} - \mathsf{H}^{(t)} \right\|_{\mathrm{F}}.$$
4. Increment t.
5. If $t < T$, return to Step 2.
6. Define the diagonal matrix $\mathsf{D} = \operatorname{diag} \mathsf{G}^{(t)}$.
7. Return the matrix
$$\mathsf{G}_{\text{out}} = \mathsf{D}^{-1/2}\, \mathsf{G}^{(t)}\, \mathsf{D}^{-1/2}.$$

The iterates generated by this algorithm need not converge. Therefore, we have chosen to halt the algorithm after a fixed number of steps instead of checking the behavior of the sequence of iterates. We will say more about the convergence properties of the algorithm in Section 7.2.6.

The scaling in the the last step normalizes the diagonal of the matrix but preserves its inertia (i.e., numbers of negative, zero, and positive eigenvalues). Since $\mathsf{G}^{(t)}$ is a positive semi-definite matrix with rank d or less, the output matrix G_{out} shares these traits. It follows that the output matrix always admits a factorization $\mathsf{G}_{\text{out}} = \mathsf{X}^* \mathsf{X}$ where X is a $d \times N$ real matrix with unit-norm columns. Property 3 is the only one of the six properties that may be violated.

The idea of applying alternating projection to feasibility problems first appeared in the work of von Neumann [114]. He proved that an alternating projection between two closed subspaces of a Hilbert space converges to the orthogonal projection of the initial iterate onto the intersection of the two subspaces. Cheney and Goldstein subsequently showed that an alternating projection between two closed, convex subsets of a Hilbert space always converges to a point in their intersection (provided that the intersection is nonempty) [11]. Unfortunately, these results do not apply to our problem because the spectral constraint set $\mathscr{G}$ is not convex.

7.2.4 The Matrix Nearness Problems

To implement Algorithm 7.1, we must solve the matrix nearness problems in Steps 2 and 3. The first one is straightforward.

Proposition 7.2. *Let G be a real, symmetric matrix. With respect to Frobenius norm, the unique matrix in $\mathscr{H}(\mu)$ closest to G has a unit diagonal and*

off-diagonal entries that satisfy

$$h_{mn} \quad = \quad \begin{cases} -1, & g_{mn} < -1, \\ g_{mn}, & -1 \le g_{mn} \le \mu, \ \textit{and} \\ \mu, & \mu < g_{mn}. \end{cases}$$

It is rather more difficult to find a nearest matrix in the spectral constraint set. To state the result, we define the plus operator $(\cdot)_+ : \mathbb{R} \to \mathbb{R}$ by the rule $(\cdot)_+ : x \mapsto \max\{0, x\}$.

Proposition 7.3. *Let H be a conjugate symmetric matrix whose eigenvalue decomposition is $\sum_{n=1}^{N} \lambda_n \boldsymbol{u}_n \boldsymbol{u}_n^*$ with the eigenvalues arranged in nonincreasing order: $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_N$. With respect to Frobenius norm, a matrix in $\mathscr{G}$ closest to H is given by*

$$\sum\nolimits_{n=1}^{d} (\lambda_n - \gamma)_+ \boldsymbol{u}_n \boldsymbol{u}_n^*$$

where the scalar γ is chosen so that the matrix has trace N. This best approximation is unique provided that $\lambda_d > \lambda_{d+1}$.

The nearest matrix described by this theorem can be computed efficiently with standard tools of numerical linear algebra [51]. The value of γ is uniquely determined, but one must solve a small optimization problem to find it. We omit the details, which are routine.

Proof. Given an Hermitian matrix A, denote by $\boldsymbol{\lambda}(\mathsf{A})$ the vector of eigenvalues arranged in nonincreasing order. Then we may decompose $\mathsf{A} = \mathsf{U}\ \mathrm{diag}\, \boldsymbol{\lambda}(\mathsf{A})\ \mathsf{U}^*$ for some unitary matrix U.

We must solve the optimization problem

$$\begin{aligned} \min_{\boldsymbol{A}} \ \tfrac{1}{2} \|\boldsymbol{A} - \boldsymbol{H}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad & \lambda_n(\boldsymbol{A}) \geq 0 \qquad \text{for } n = 1, \dots, d, \\ & \lambda_n(\boldsymbol{A}) = 0 \qquad \text{for } n = d+1, \dots, N, \text{ and} \\ & \mathbf{e}^* \boldsymbol{\lambda}(\boldsymbol{A}) = N. \end{aligned}$$

First, we fix the eigenvalues of $\boldsymbol{A}$ and minimize with respect to the unitary part of its decomposition. In consequence of the Wielandt–Hoffman Theorem [61], the objective function is bounded below:

$$\tfrac{1}{2} \|\boldsymbol{A} - \boldsymbol{H}\|_{\mathrm{F}}^2 \quad \geq \quad \tfrac{1}{2} \|\boldsymbol{\lambda}(\boldsymbol{A}) - \boldsymbol{\lambda}(\boldsymbol{H})\|_2^2 .$$

Equality holds if and only $\boldsymbol{A}$ and $\boldsymbol{H}$ are simultaneously diagonalizable by a unitary matrix. Therefore, if we decompose $\boldsymbol{H} = \boldsymbol{U} \operatorname{diag} \boldsymbol{\lambda}(\boldsymbol{H}) \, \boldsymbol{U}^*$, the objective function attains its minimal value whenever $\boldsymbol{A} = \boldsymbol{U} \operatorname{diag} \boldsymbol{\lambda}(\boldsymbol{A}) \, \boldsymbol{U}^*$. Note that the matrix $\boldsymbol{U}$ may not be uniquely determined.

We find the optimal set of eigenvalues $\boldsymbol{\xi} = \boldsymbol{\lambda}(\boldsymbol{A})$ by solving the (strictly) convex program

$$\begin{aligned} \min_{\boldsymbol{\xi}} \ \tfrac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\lambda}(\boldsymbol{H})\|_2^2 \qquad \text{subject to} \qquad & \xi_n \geq 0, \qquad \text{for } n = 1, \dots, d, \\ & \xi_n = 0, \qquad \text{for } n = d+1, \dots, N, \text{ and} \\ & \mathbf{e}^* \boldsymbol{\xi} = N. \end{aligned}$$

This minimization is accomplished by an application of Karush–Kuhn–Tucker theory [88]. In short, the top d eigenvalues of $\boldsymbol{H}$ are translated an equal amount, and those that become negative are set to zero. The size of the translation is chosen to fulfill the trace condition. The entries of the optimal $\boldsymbol{\xi}$ are nonincreasing on account of the ordering of $\boldsymbol{\lambda}(\boldsymbol{H})$.

Finally, the uniqueness claim follows from the fact that the eigenspace associated with the top d eigenvectors of H is uniquely determined if and only if $\lambda_d(\mathsf{H}) > \lambda_{d+1}(\mathsf{H})$. □

7.2.5 The Initial Matrix

The success of the algorithm depends on an adequate selection of the input matrix G_0. We have found the following strategy is reasonably efficient and effective.

Algorithm 7.4 (Initial Matrix).
INPUT:

- The dimension d.
- The number of vectors N.
- An upper bound τ on the inner product between vectors.
- The maximum number T of random choices.

OUTPUT:

- An $N \times N$ matrix G with rank d, with a unit diagonal, and with off-diagonal entries that do not exceed τ.

PROCEDURE:

1. Initialize $t \leftarrow 0$ and $n \leftarrow 1$.
2. Increment t. If $t > T$, then print a failure notice and quit.

3. Choose a vector $\boldsymbol{x}_n$ uniformly at random from $\mathbb{S}^{d-1}$.

4. If $\langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle \leq \tau$ for each $m = 0, \ldots, n-1$, then increment n.

5. If $n \leq N$, return to Step 2.

6. Form the matrix $\mathsf{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \ldots \quad \boldsymbol{x}_N]$.

7. Return the Gram matrix $\mathsf{G} = \mathsf{X}^*\mathsf{X}$.

To implement Step 3, we choose a d-dimensional vector whose entries are iid standard normal. This vector is then scaled to have unit norm [99].

The purpose of the parameter τ is to prevent the starting matrix from containing columns that are nearly identical. The extreme case $\tau = 1$ places no restriction on the inner products between columns. For Tammes' Problem, typical values for τ range between 0.9 and 1.0. It is essential to be aware that this procedure will fail if τ is chosen too small (or if we are unlucky in our random choices). For this reason, we add an iteration counter so that the procedure will not enter an infinite loop.

7.2.6 Theoretical Behavior of the Algorithm

It is important to be aware that packing problems are typically difficult to solve. Therefore, we cannot expect that our algorithm will necessarily produce a point in the intersection of the constraint sets. One may ask whether we can make any guarantees about the behavior of Algorithm 7.1. This turns out to be difficult. Indeed, it is possible that an alternating projection algorithm will fail to generate a convergent sequence of iterates [74]. Nevertheless, it can be shown that the sequence of iterates has accumulation points, and that these accumulation points satisfy a certain weak structural property.

In practice, the algorithm always appears to converge in norm, so the lack of a rigorous convergence proof is only a theoretical annoyance. A more serious problem is that the algorithm typically requires as many as 5000 iterations to approach a limit. This is probably the greatest weakness of our approach.

For reference, we quote the best theoretical convergence result that we know. The distance between a matrix and a collection of matrices is defined as

$$\operatorname{dist}(M, \mathscr{C}) \quad \stackrel{\text{def}}{=} \quad \inf_{C \in \mathscr{C}} \|M - C\|_{\mathrm{F}} .$$

Theorem 7.5 (Global Convergence). *Suppose that Algorithm 7.1 generates an (infinite) sequence of iterates* $\{(G^{(t)}, H^{(t)})\}$. *This sequence has at least one accumulation point.*

- *Every accumulation point lies in* $\mathscr{G} \times \mathscr{H}$.
- *Every accumulation point* $(\overline{G}, \overline{H})$ *satisfies*

$$\left\|\overline{G} - \overline{H}\right\|_{\mathrm{F}} = \lim_{t \to \infty} \left\|G^{(t)} - H^{(t)}\right\|_{\mathrm{F}} .$$

- *Every accumulation point* $(\overline{G}, \overline{H})$ *satisfies*

$$\left\|\overline{G} - \overline{H}\right\|_{\mathrm{F}} = \operatorname{dist}(\overline{G}, \mathscr{H}) = \operatorname{dist}(\overline{H}, \mathscr{G}).$$

Proof sketch. The existence of an accumulation point falls from the compactness of the constraint sets. The algorithm always decreases the distance between successive iterates, which is bounded below by zero. Therefore, this distance must converge as well. Since each iterate is chosen as the closest matrix from the opposite constraint set and the Frobenius norm is continuous, we can take limits to obtain the remaining assertions. □

A more detailed treatment requires the machinery of point-to-set maps, and it would not enhance our main discussion. Please see the report [109] for additional information.

7.2.7 Numerical Experiments

Our approach to packing is experimental rather than theoretical, so the real question is how Algorithm 7.1 performs in practice. In principle, this question is difficult to resolve because the optimal packing radius is unknown for almost all combinations of d and N. Nevertheless, Tammes' Problem has been studied for 75 years, and many putatively optimal configurations are available. Therefore, we attempted to produce packings whose maximum inner product μ fell within 10^{-5} of the best value tabulated by N. J. A. Sloane and his colleagues [96]. This resource draws from all the experimental and theoretical work on Tammes' Problem, and it should be considered the gold standard.

We implemented the algorithms in Matlab, and we performed the following experiment for pairs (d, N) with $d = 3, 4, 5$ and $N = 4, \dots, 25$. First, we computed the putatively optimal maximum inner product μ using the data from [96]. In each of 10 trials, we constructed a starting matrix using Algorithm 7.4 with parameters $\tau = 0.9$ and $T = 10,000$. Then, we executed the alternating projection, Algorithm 7.1, with the calculated value of μ and the maximum number of iterations set to $T = 5000$. (Our numerical experience indicates that increasing the maximum number of iterations beyond 5000 does not confer a significant benefit.) We stopped the alternating projection in Step 4 if the iterate $\mathsf{G}^{(t)}$ contained no off-diagonal entry greater than $(\mu + 10^{-5})$ and proceeded with Step 6. After 10 trials, we recorded the largest packing radius attained, as well as the average value of the packing radius. We also

recorded the average number of iterations the alternating projection required during each trial.

Table 7.1 provides the results of this experiment. Following Sloane, we have reported the degrees of arc subtended by the closest pair of points in lieu of the Euclidean distance between them or cosine of the angle between them. We believe that the results are much easier to interpret geometrically when delivered in this fashion. All the tables and figures related to packing are collated at the back of this chapter for easy comparison.

The most striking feature of Table 7.1 is that the best configurations returned by alternating projection consistently attain packing radii that fall hundredths or thousandths of a degree away from the best packing radii recorded by Sloane. If we examine the maximum inner product in the configuration instead, the difference is usually on the order of 10^{-4} or 10^{-5}, which we expect based on our stopping criterion. The average-case results are somewhat worse. Nevertheless, the average configuration returned by alternating projection typically attains a packing radius several tenths of a degree away from optimal.

A second observation is that the alternating projection algorithm typically performs better when the number of points N is small. The largest errors are all clustered at larger values of N. A corollary observation is that the average number of iterations per trial tends to increase with the number of points. We believe that the explanation for these phenomena is that Tammes' Problem has a combinatorial regime, where solutions have a lot of symmetry and structure, and a random regime, where the solutions have very little order. The algorithm typically seems to perform better in the combinatorial regime.

This claim is supported somewhat by theoretical results for $d = 3$. Opti-

mal configurations have only been established for $N = 1, \dots, 12$ and $N = 24$. Of these, the cases $N = 1, 2, 3$ are trivial. The cases $N = 4, 6, 8, 12, 24$ fall from the vertices of various well-known polyhedra. The cases $N = 5, 11$ are degenerate, obtained by leaving a point out of the solutions for $N = 6, 12$. The remaining cases involve complicated constructions based on graphs [36]. The algorithm was able to calculate the known optimal configurations to a high order of accuracy, but it generally performed slightly better for the non-degenerate cases.

On the other hand, there is at least one case where the algorithm failed to match the optimal packing radius, even though the optimal configuration is highly symmetric. The best arrangement of 24 points on $\mathbb{S}^3$ locates them at vertices of a polytope called the 24-cell [96]. The best configuration produced by the algorithm has a packing radius 1.79° worse. It seems that this optimal configuration is very difficult for the algorithm to find. Less dramatic failures occurred at pairs $(d, N) = (3, 25)$, $(4, 14)$, $(4, 25)$, $(5, 22)$, and $(5, 23)$. But in each of these cases, our best packing declined more than a tenth of a degree from the best recorded.

7.3 Packing in Projective Spaces

This section addresses the feasibility problem most closely related to sparse approximation, the construction of a dictionary with specified coherence. As we learned in Chapter 4, a dictionary can be viewed a configuration of lines in projective space, and the coherence parameter is complementary to the packing radius of the dictionary in projective space. Therefore, we may interpret the dictionary construction problem as a packing problem in projective space.

A more physical motivation for this problem has been suggested in [15]. Imagine that we wish to destroy a tumor by firing laser beams at it from several directions. The beams should coincide at the tumor, but the acute angle between each pair should remain as large as possible to avoid damaging the surrounding tissue.

7.3.1 Projective Spaces

For continuity, we review the definition of projective spaces from Section 4.8. Denote by $\mathbb{C}^d$ the d-dimensional, complex inner-product space. The usual Hermitian inner product will be written as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^* \boldsymbol{x}$, and the squared Euclidean norm derives from the inner product: $\|\boldsymbol{x}\|_2^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$.

The $(d-1)$-dimensional *complex projective space* may be viewed as the collection of all one-dimensional subspaces of $\mathbb{C}^d$. Formally, it is defined as

$$\mathbb{P}^{d-1}(\mathbb{C}) \quad \stackrel{\text{def}}{=} \quad \frac{\mathbb{C}^d \setminus \{\mathbf{0}\}}{\mathbb{C}^\times}.$$

(The symbol $\mathbb{C}^\times$ refers to the set of nonzero complex numbers.) The real projective space $\mathbb{P}^{d-1}(\mathbb{R})$ is defined in much the same way:

$$\mathbb{P}^{d-1}(\mathbb{R}) \quad \stackrel{\text{def}}{=} \quad \frac{\mathbb{R}^d \setminus \{\mathbf{0}\}}{\mathbb{R}^\times}.$$

It may be viewed as the collection of all lines through the origin of $\mathbb{R}^d$. On analogy, we will refer to the elements of a complex projective space as *lines*. We concentrate on the complex case because the real case follows from a transparent adaptation.

The natural metric for $\mathbb{P}^{d-1}(\mathbb{C})$ is the acute angle between two lines or—what is equivalent—the sine of the acute angle. We will model the $(d -$

1) projective space as the collection of unit vectors in $\mathbb{C}^d$. Therefore, the projective distance between two unit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ will be calculated as

$$\operatorname{dist}_{\mathbb{P}}(\boldsymbol{x}, \boldsymbol{y}) \quad \stackrel{\text{def}}{=} \quad \sqrt{1 \; - \; |\langle \boldsymbol{x}, \boldsymbol{y} \rangle|^2}.$$

Evidently, the distance between two lines ranges between zero and one. When equipped with this distance, $\mathbb{P}^{d-1}(\mathbb{C})$ forms a compact metric space [15].

7.3.2 Packings and Matrices

Suppose that we wish to construct a configuration of N lines in $\mathbb{P}^{d-1}(\mathbb{C})$ with a packing radius no less than ρ. We will represent each configuration of lines in $\mathbb{P}^{d-1}(\mathbb{C})$ by a collection $\mathscr{X}$ of unit vectors in $\mathbb{C}^d$.

$$\mathscr{X} \quad = \quad \{\boldsymbol{x}_n\}_{n=1}^N.$$

The packing radius of the configuration $\mathscr{X}$ in projective space is defined as

$$\operatorname{pack}_{\mathbb{P}}(\mathscr{X}) \quad \stackrel{\text{def}}{=} \quad \min_{m \neq n} \operatorname{dist}_{\mathbb{P}}(\boldsymbol{x}_m, \boldsymbol{x}_n) \quad = \quad \min_{m \neq n} \sqrt{1 - |\langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle|^2},$$

and the feasibility problem requests a configuration $\mathscr{X}$ for which

$$\min_{m \neq n} \sqrt{1 - |\langle \boldsymbol{x}_n, \boldsymbol{x}_m \rangle|^2} \quad \geq \quad \rho.$$

As before, we clear the debris from this inequality to obtain an equivalent condition:

$$\max_{m \neq n} \; |\langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle| \quad \leq \quad \mu \tag{7.5}$$

where $\mu = \sqrt{1 - \rho^2}$.

Form the elements of $\mathscr{X}$ into a complex $d \times N$ matrix:

$$\mathsf{X} \quad \stackrel{\text{def}}{=} \quad \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_N \end{bmatrix}.$$

As usual, the Gram matrix is $\mathsf{G} = \mathsf{X}^*\mathsf{X}$. It follows that the configuration X solves the feasibility problem (7.5) if and only if its Gram matrix has the following properties:

1. G is Hermitian.
2. G has a unit diagonal.
3. $|g_{mn}| \leq \mu$ whenever $m \neq n$.
4. G is positive semi-definite.
5. G has rank d or less.
6. G has trace N.

Properties 1 and 3 are the only ones that differ from the conditions we developed for Tammes' Problem. The change to Property 3 is what leads to a packing in projective space instead of a packing on the sphere.

7.3.3 Implementation Details

We define the convex structural constraint set

$$\mathscr{H}(\mu) \stackrel{\text{def}}{=} \{\mathsf{H} \in \mathbb{C}^{N\times N} : \mathsf{H} = \mathsf{H}^*, \quad \operatorname{diag}\mathsf{H} = \mathbf{e}, \quad \text{and} \quad |h_{mn}| \leq \mu \text{ for } m \neq n \}. \quad (7.6)$$

The only essential difference between (7.3) and (7.6) is the absolute value in the bound on the off-diagonal entries. As before, the spectral constraint set is

$$\mathscr{G} \stackrel{\text{def}}{=} \{\mathsf{G} \in \mathbb{C}^{N\times N} : \mathsf{G} \succcurlyeq \mathsf{0}, \quad \operatorname{rank}\mathsf{G} \leq d, \quad \text{and} \quad \operatorname{trace}\mathsf{G} = N\}.$$

It is clear that we may solve the feasibility problem (7.5) by producing a matrix in the intersection of $\mathscr{G}$ and $\mathscr{H}(\mu)$.

A variation on Algorithm 7.4 can be used to construct a starting matrix. Some minor changes are necessary. In Step 3, we wish to construct a uniformly random line in $\mathbb{P}^{d-1}(\mathbb{C})$. To do so, one selects a complex vector whose real and imaginary parts are chosen from independent standard normal distributions and re-scales the vector to have unit Euclidean norm [99]. In Step 4, one must test whether $|\langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle| \leq \tau$ for each $m < n$.

The alternating projection between $\mathscr{G}$ and $\mathscr{H}$ is a variant of Algorithm 7.1. We apply Proposition 7.3 to determine a nearest matrix from the spectral constraint set $\mathscr{G}$. To compute the nearest matrix from the structural constraint set $\mathscr{H}$, we use the following result.

Proposition 7.6. *Let G be an Hermitian matrix. With respect to Frobenius norm, the unique matrix in $\mathscr{H}(\mu)$ closest to G has a unit diagonal and off-diagonal entries that satisfy*

$$h_{mn} \quad = \quad \begin{cases} g_{mn} & \text{if } |g_{mn}| \leq \mu, \text{ and} \\ \mu \, g_{mn} / \, |g_{mn}| & \text{otherwise.} \end{cases}$$

The proof is immediate.

7.3.4 Numerical Experiments

For most pairs (d, N), the optimal packing radius of N lines in $\mathbb{P}^{d-1}(\mathbb{R})$ and $\mathbb{P}^{d-1}(\mathbb{C})$ is unknown. In the real setting, we can use the tables of N. J. A. Sloane and his colleagues for guidance [95]. There is no comparable resource for the complex setting, which makes it challenging to understand how well the algorithm performs.

Let us begin with packing in real projective spaces. We attempted to construct configurations of real lines whose maximum absolute inner product μ fell within 10^{-5} of the best value tabulated in [95]. The experimental method parallels the method used for Tammes' Problem. For pairs (d, N) with $d = 3, 4, 5$ and $N = 4, \dots, 25$, we computed the putatively optimal value of the feasibility parameter μ from the data in [95]. In each of 10 trials, we constructed a starting matrix using Algorithm 7.4 with parameters $\tau = 0.9$ and $T = 10,000$. We applied the alternating projection, Algorithm 7.1 with the computed value of μ and the maximum number of iterations $T = 5000$. We halted the iteration in Step 4 if the iterate $\mathsf{G}^{(t)}$ exhibited no off-diagonal entry with absolute value greater than $(\mu + 10^{-5})$. After 10 trials, we recorded the largest packing radius attained, as well as the average value of the packing radius. We also recorded the average number of iterations the alternating projection required per trial.

Table 7.2 delivers the results of this experiment. We have reported the acute angle between the closest pair of lines for ease of interpretation. According to the table, the best configurations produced by alternating projection consistently attain packing radii tenths or hundredths of a degree away from the best configurations known. The average configurations returned by alternating projection are slightly worse, but they usually fall within a degree of the putative optimal. As in the experiments for Tammes' Problem, alternating projection performs better when the number of points N is not too large. This is reflected both in the packing radii and in the number of iterations that the algorithm requires.

There are several anomalies that we would like to point out. The most interesting pathology occurs at the pair $(d, N) = (5, 19)$. The best packing

radius calculated by alternating projection is about 1.76° worse than the optimal configuration, and it is also 1.76° worse than the best packing radius computed for the pair $(5, 20)$. From Sloane's tables, we can see that the (putative) optimal packing of 19 lines in $\mathbb{P}^4(\mathbb{R})$ is actually a subset of the best packing of 20 lines. Perhaps the fact that this packing is degenerate makes it difficult to construct. A similar event occurs rather less dramatically at the pair $(5, 13)$. The table also shows that the algorithm often performs badly when the number of lines exceeds 20.

Sloane does not provide a table of packings in complex projective space. In fact, we only know of one paper that contains numerical work on packing in complex projective spaces [1], but it gives very few examples of good complex packings. The only method we know for gauging the quality of a complex line packing is to compare it against an upper bound. Rankin's Bound for projective packings, which we derive in Section 7.5, states that every configuration $\mathscr{X}$ of N lines in either $\mathbb{P}^{d-1}(\mathbb{R})$ or $\mathbb{P}^{d-1}(\mathbb{C})$ satisfies the inequality

$$\operatorname{pack}_{\mathbb{P}}(\mathscr{X})^2 \quad \leq \quad \frac{(d-1)\,N}{d\,(N-1)}.$$

This bound is attainable only for rare combinations of d and N. In particular, the bound can be met in $\mathbb{P}^{d-1}(\mathbb{R})$ only if $N \leq \frac{1}{2}\,d\,(d+1)$. In the space $\mathbb{P}^{d-1}(\mathbb{C})$, attainment requires that $N \leq d^2$. Any arrrangement of lines that meets the Rankin Bound must be equiangular. These optimal configurations are called *equiangular tight frames*. See [100, 60, 109, 101] for more details.

We performed some *ad hoc* experiments to produce configurations of complex lines with large packing radii. For each pair (d, N), we used the Rankin Bound to determine a lower limit on the feasibility parameter μ. Starting matrices were constructed with Algorithm 7.4 using values of τ ranging between

0.9 and 1.0. For various values of the feasibility parameter, we executed between 1000 and 5000 iterations of Algorithm 7.1, and we recorded the largest packing radius attained during these trials.

Table 7.3 compares our results against the Rankin Bound. We see that many of the complex line configurations have packing radii much smaller than the Rankin Bound, which is not surprising because the bound is usually not attainable. Some of our configurations fall within a thousandth of a degree of the bound, which is essentially optimal.

Table 7.3 contains a few oddities. In $\mathbb{P}^4(\mathbb{C})$, the best packing radius computed for $N = 18, \dots, 24$ is worse than the packing radius for $N = 25$. This configuration of 25 lines is an equiangular tight frame, which means that it is an optimal packing. It seems likely that the optimal configurations for the preceding values of N are just subsets of the optimal arrangement of 25 lines. As before, it may be difficult to calculate this type of degenerate packing. A similar event occurs less dramatically at the pair $(d, N) = (4, 13)$ and at the pairs $(4, 17)$ and $(4, 18)$.

Figure 7.1 compares the quality of the best real projective packings from [95] with the best complex projective packings that we obtained. It is natural that the complex packings are better than the real packings because the real projective space can be embedded isometrically into the complex projective space. But it is remarkable how badly the real packings compare with the complex packings. The only cases where the real and complex ensembles have the same packing radius occur when the real configuration meets the Rankin Bound.

7.4 Packing in Grassmannian Spaces

A line is just a one-dimensional subspace. The obvious generalization of line packing, therefore, is subspace packing. The alternating algorithm also applies in this setting, but we must address some new challenges. The problem of subspace packing was initially raised in the inspiring paper [15].

7.4.1 Grassmannian Spaces

The complex Grassmannian space $\mathbb{G}(K, \mathbb{C}^d)$ is the collection of all K-dimensional subspaces of $\mathbb{C}^d$. This space is isomorphic to a quotient of unitary groups:

$$\mathbb{G}(K, \mathbb{C}^d) \quad \cong \quad \frac{\mathrm{U}(d)}{\mathrm{U}(K) \times \mathrm{U}(d-K)}.$$

The unitary group $\mathrm{U}(d)$ can be represented as the collection of all $d \times d$ unitary matrices with ordinary matrix multiplication. To understand the equivalence, note that each orthonormal basis for $\mathbb{C}^d$ can be split into K vectors, which span a K-dimensional subspace, and $(d-K)$ vectors, which span the orthogonal complement of the subspace. To obtain a unique representation of the subspace, one must modulate by rotations that fix the subspace and rotations that fix its complement. Similarly, the real Grassmannian space $\mathbb{G}(K, \mathbb{R}^d)$ is the collection all K-dimensional subspaces of $\mathbb{R}^d$. It is isomorphic to a quotient of orthogonal groups:

$$\mathbb{G}(K, \mathbb{R}^d) \quad \cong \quad \frac{\mathrm{O}(d)}{\mathrm{O}(K) \times \mathrm{O}(d-K)}.$$

The orthogonal group $\mathrm{O}(d)$ can be represented as the collection of all $d \times d$ real orthogonal matrices with the usual matrix multiplication. We will concentrate on complex Grassmannians since the real case follows from a transparent

adaptation. If we need to refer to both the real and complex case at once, we will write $\mathbb{G}(K, \mathbb{F}^d)$.

7.4.2 Metrics on Grassmannian Spaces

Grassmannian manifolds admit many interesting metrics, each of which yields a different packing problem. In this section, we will describe a few of these metrics.

Suppose that $\mathscr{S}$ and $\mathscr{T}$ are two subspaces in $\mathbb{G}(K, \mathbb{C}^d)$. These subspaces are inclined against each other by K different *principal angles*. The smallest principal angle θ_1 is the minimum angle formed by any pair of vectors drawn from the two subspaces. The second principal angle θ_2 is defined as the smallest angle attained between a pair of vectors orthogonal to the first set of vectors. The remaining principal angles are defined recursively. The principal angles are increasing, and each one lies in the range $[0, \pi/2]$. We will only consider metrics that are functions of the principal angles between two subspaces.

Let us present a more computational definition of the principal angles [51]. Suppose that the columns of S and T form orthonormal bases for the subspaces $\mathscr{S}$ and $\mathscr{T}$. Formally, S is a $d \times K$ matrix that satisfies $\mathsf{S}^*\mathsf{S} = \mathsf{I}$ and $\operatorname{colspan} \mathsf{S} = \mathscr{S}$. Analogously, the matrix T. Next, we compute a singular value decomposition of the product $\mathsf{S}^*\mathsf{T}$:

$$\mathsf{S}^*\mathsf{T} \quad = \quad \mathsf{U}\mathsf{C}\mathsf{V}^*,$$

where U and V are $K \times K$ unitary matrices and C is a nonnegative, diagonal matrix with nonincreasing entries. The matrix C is uniquely determined, and its entries list the cosines of the principal angles between $\mathscr{S}$ and $\mathscr{T}$:

$$c_{kk} \quad = \quad \cos \theta_k.$$

This definition of the principal angles is most convenient numerically because singular value decompositions can be computed quickly with standard software.

We are now in a position to detail some metrics on the Grassmannian space.

1. The *chordal distance* between $\mathscr{S}$ and $\mathscr{T}$ is given by

$$\begin{aligned} \operatorname{dist}_{\text{chord}}(\mathscr{S}, \mathscr{T}) &\stackrel{\text{def}}{=} \sqrt{\sin^2\theta_1 + \cdots + \sin^2\theta_K} \\ &= \sqrt{K - \|\mathsf{S}^*\mathsf{T}\|_{\text{F}}^2}. \end{aligned} \tag{7.7}$$

   The values of this metric range between zero and $\sqrt{K}$. The chordal distance is the easiest to work with, and it is also yields the most symmetrical packings [15].

2. The *spectral distance* is

$$\begin{aligned} \operatorname{dist}_{\text{spec}}(\mathscr{S}, \mathscr{T}) &\stackrel{\text{def}}{=} \sin\theta_1 = \max_k \sin\theta_k \\ &= \sqrt{1 - \|\mathsf{S}^*\mathsf{T}\|_{2,2}^2}. \end{aligned} \tag{7.8}$$

   We use $\|\cdot\|_{2,2}$ to denote the spectral norm, which returns the largest singular value of a matrix. The spectral distance takes values between zero and one.

3. The *Fubini–Study distance* is defined by

$$\begin{aligned} \operatorname{dist}_{\text{FS}}(\mathscr{S}, \mathscr{T}) &\stackrel{\text{def}}{=} \arccos\left(\prod\nolimits_k \cos\theta_k\right) \\ &= \arccos\left|\det \mathsf{S}^*\mathsf{T}\right|. \end{aligned} \tag{7.9}$$

   This metric takes values between zero and $\pi/2$. From a group-theoretic point of view, the Fubini–Study distance is the most natural because it is the unique Riemannian metric that is invariant under actions of the unitary group on the Grassmannian space.

If the subspaces are one-dimensional, observe that each of these metrics reduces to (the sine of) the acute angle between the two subspaces, which is just the distance we defined on the projective space. The Grassmannian space admits other interesting metrics, some of which are listed in [3].

7.4.3 Configurations and Matrices

Next, we must discuss how to represent a configuration of N subspaces in the Grassmannian space $\mathbb{G}(K, \mathbb{C}^d)$. Let $\mathscr{X} = \{\mathsf{X}_n\}$ be a collection of N complex matrices with dimensions $d \times K$. Each of these matrices will provide a basis for one of the N subspaces, so we require that the columns of X_n form an orthonormal set for each n. We collate these matrices into a $d \times KN$ matrix

$$\mathsf{X} \quad \stackrel{\text{def}}{=} \quad \begin{bmatrix} \mathsf{X}_1 & \mathsf{X}_2 & \dots & \mathsf{X}_N \end{bmatrix}.$$

As always, the Gram matrix of X is defined as $\mathsf{G} = \mathsf{X}^*\mathsf{X}$. It is best to regard the Gram matrix as an $N \times N$ block matrix comprised of $K \times K$ blocks, and we will index it as such. Observe that each block satisfies

$$\mathsf{G}_{mn} \quad = \quad \mathsf{X}_m^* \mathsf{X}_n$$

In particular, each diagonal block G_{nn} is an identity matrix. Meanwhile, the singular values of the off-diagonal block G_{mn} equal the cosines of the principal angles between the two subspaces colspan X_m and colspan X_n.

As we will see, each metric on the Grassmannian space leads to a measure of "magnitude" for the off-diagonal blocks of the Gram matrix. The Gram matrix solves the feasibility problem if and only if each off-diagonal block has sufficiently small "magnitude."

7.4.4 Packings with Chordal Distance

Suppose that we seek a packing of N subspaces in $\mathbb{G}(K, \mathbb{C}^d)$ equipped with the chordal distance. If $\boldsymbol{X}$ is a configuration of N subspaces, its packing radius is

$$\operatorname{pack}_{\text{chord}}(\boldsymbol{X}) \quad \stackrel{\text{def}}{=} \quad \min_{m \neq n} \operatorname{dist}_{\text{chord}}(\boldsymbol{X}_m, \boldsymbol{X}_n) \quad = \quad \min_{m \neq n} \sqrt{K - \|\boldsymbol{X}_m^* \boldsymbol{X}_n\|_{\text{F}}^2}.$$

Given a parameter ρ, the feasibility problem requests a configuration $\boldsymbol{X}$ that satisfies

$$\min_{m \neq n} \sqrt{K - \|\boldsymbol{X}_m^* \boldsymbol{X}_n\|_{\text{F}}^2} \quad \geq \quad \rho.$$

As usual, we rearrange to obtain a simpler condition:

$$\max_{m \neq n} \|\boldsymbol{X}_m^* \boldsymbol{X}_n\|_{\text{F}} \quad \leq \quad \mu \tag{7.10}$$

where $\mu = \sqrt{K - \rho^2}$. It is immediately clear that the configuration $\boldsymbol{X}$ solves the feasibility problem (7.10) if and only if its Gram matrix $\boldsymbol{G}$ has the following properties:

1. $\boldsymbol{G}$ is Hermitian.
2. Each diagonal block of $\boldsymbol{G}$ is an identity matrix.
3. $\|\boldsymbol{G}_{mn}\|_{\text{F}} \leq \mu$ whenever $m \neq n$.
4. $\boldsymbol{G}$ is positive semi-definite.
5. $\boldsymbol{G}$ has rank d or less.
6. $\boldsymbol{G}$ has trace KN.

This enumeration leads directly to an algorithm.

The structural constraint is the convex set

$$\mathscr{H}(\mu) \quad \stackrel{\text{def}}{=} \quad \{\mathsf{H} \in \mathbb{C}^{KN \times KN} : \mathsf{H} = \mathsf{H}^*, \quad \mathsf{H}_{nn} = \mathsf{I} \text{ for all } n, \\ \text{and} \quad \|\mathsf{H}_{mn}\|_{\mathrm{F}} \leq \mu \text{ for all } m \neq n\}.$$

The spectral constraint set remains

$$\mathscr{G} \quad \stackrel{\text{def}}{=} \quad \{\mathsf{G} \in \mathbb{C}^{KN \times KN} : \mathsf{G} \succcurlyeq \mathsf{0}, \quad \operatorname{rank} \mathsf{G} \leq d, \quad \text{and} \quad \operatorname{trace} \mathsf{G} = KN\}.$$

Solving the feasibility problem (7.10) with parameter μ is equivalent to exhibiting a matrix in the intersection of $\mathscr{G}$ and $\mathscr{H}(\mu)$.

Algorithm 7.4 allows us to build a starting matrix. To construct a subspace uniformly at random with respect to the left-invariant Haar measure on $\mathbb{G}(K, \mathbb{C}^d)$, we use the striking method developed in [99]. Choose a $d \times K$ matrix whose (complex) entries are iid standard normal, and perform a QR decomposition. The first K columns of the unitary part of the decomposition form an orthonormal basis for a random subspace. Step 4 of Algorithm 7.4 needs to be replaced by the test $\|\mathsf{X}_m^* \mathsf{X}_n\|_{\mathrm{F}} \leq \tau$ for each $m < n$.

We may adapt Algorithm 7.1 to alternate between the constraint sets $\mathscr{G}$ and $\mathscr{H}$. To determine the nearest matrix from the structural constaint set, we use the following result.

Proposition 7.7. *Let G be an Hermitian matrix. With respect to the Frobenius norm, the unique matrix in $\mathscr{H}(\mu)$ nearest to G has a block-identity diagonal and off-diagonal blocks that satisfy*

$$\mathsf{H}_{mn} \quad = \quad \begin{cases} \mathsf{G}_{mn} & \text{if } \|\mathsf{G}_{mn}\|_{\mathrm{F}} \leq \mu, \text{ and} \\ \mu \, \mathsf{G}_{mn} / \|\mathsf{G}_{mn}\|_{\mathrm{F}} & \text{otherwise.} \end{cases}$$

It is nice to see how this result generalizes Proposition 7.6. We leave the easy proof for the reader.

In Step 6 of Algorithm 7.1, we extract the diagonal blocks of the final iterate. It follows that Step 7 scales each diagonal block to equal the identity matrix without changing the inertia of the matrix. Therefore, we may factor the output matrix to obtain a $d \times KN$ configuration matrix X. The N blocks of this matrix represent K-dimensional subspaces of $\mathbb{C}^d$.

7.4.4.1 Numerical Experiments

For real Grassmannian spaces equipped with chordal distance, we have been able to study the performance of the alternating projection algorithm by comparison with the tables of Sloane and his colleagues [95]. For each triple (d, K, N), we determined a value for the feasibility parameter μ from the best packing radius Sloane recorded for N subspaces in $\mathbb{G}(K, \mathbb{R}^d)$. We constructed starting points using the modified version of Algorithm 7.4 with $\tau = \sqrt{K}$. Then we executed Algorithm 7.1 with the calculated value of μ for 1000 to 5000 iterations.

Table 7.4 demonstrates how the best packings we obtained compare with Sloane's best packings. Many of our real configurations attained a squared packing radius within 10^{-3} of the best value Sloane recorded. Our algorithm was especially successful for smaller numbers of subspaces, but its performance began to flag as the number of subspaces approached 20.

Table 7.4 contains several anomalies. For example, our configurations of 11 to 16 subspaces in $\mathbb{R}^4$ yield worse packing radii than the configuration of 17 subspaces. It turns out that this configuration of 17 subspaces is optimal, and Sloane's data shows that the (putative) optimal arrangements of 11 to 16

subspaces are all subsets of this configuration. This is the same problem that occurred in some of our earlier experiments, and it suggests that our algorithm has difficulty locating these degenerate configurations precisely.

The literature contains very few experimental results on packing in complex Grassmannian manifolds equipped with chordal distance. To our knowledge, the only numerical work appears in two short tables from [1]. Therefore, we found it valuable to compare our results against the Rankin Bound for subspace packings, which we derive in Section 7.5. For reference, this bound requires that every configuration $\mathscr{X}$ of N subspaces in $\mathbb{G}(K, \mathbb{F}^d)$ satisfy the inequality

$$\operatorname{pack}_{\text{chord}}(\mathscr{X})^2 \quad \leq \quad \frac{K\,(d-K)}{d}\,\frac{N}{N-1}.$$

This bound is not always attainable. In particular, the bound is attainable in the complex setting only if $N \leq d^2$. In the real setting, the bound requires that $N \leq \frac{1}{2}\,d\,(d+1)$. When the bound is attained, each pair of subspaces in $\mathscr{X}$ is equidistant.

We performed some *ad hoc* experiments to construct a table of packings in $\mathbb{G}(K, \mathbb{C}^d)$ equipped with the chordal distance. For each triple (d, K, N), we constructed random starting points using Algorithm 7.4 with $\tau = \sqrt{K}$ (which represents no constraint). Then we used the Rankin Bound to calculate a lower limit on the feasibility parameter μ. For this value of μ, we executed the alternating projection, Algorithm 7.1, for 5000 iterations.

The best packing radii we obtained are listed in Table 7.4. We see that there is a remarkable correspondence between the squared packing radii of our configurations and the Rankin Bound. Indeed, many of our packings are within 10^{-4} of the bound, which means that these configurations are essentially optimal. The algorithm was less successful as N approached d^2, which is an

upper bound on the number N of subspaces for which the Rankin Bound is attainable.

Figure 7.2 compares the packing radii of the best configurations in real and complex Grassmannian spaces equipped with chordal distance. It is remarkable that both real and complex packings meet the Rankin Bound for all N where it is attainable. Notice how the real packing radii fall off as soon as N exceeds $\frac{1}{2}\,d\,(d+1)$. In theory, a complex configuration should always attain a better packing radius than the corresponding real configuration because the real Grassmannian space can be embedded isometrically into the complex Grassmannian space. The figure shows that our best arrangements of 17 and 18 subspaces in $\mathbb{G}(2, \mathbb{C}^4)$ are actually slightly worse than the real arrangements calculated by Sloane. This indicates a failure of the alternating projection algorithm.

7.4.5 Packings with Spectral Distance

To construct packings with respect to the spectral distance, we tread a familiar path. Suppose that we wish to produce a configuration of N subspaces in $\mathbb{G}(K, \mathbb{C}^d)$ with a packing radius ρ. The feasibility problem requires that

$$\max_{m \neq n} \ \|\mathsf{X}_m^* \, \mathsf{X}_n\|_{2,2} \quad \leq \quad \mu \tag{7.11}$$

where $\mu = \sqrt{1 - \rho^2}$. This leads to the convex structural constraint set

$$\begin{aligned} \mathscr{H}(\mu) \quad \stackrel{\text{def}}{=} \quad \{\mathsf{H} \in \mathbb{C}^{KN \times KN} : \mathsf{H} = \mathsf{H}^*, \quad \mathsf{H}_{nn} = \mathsf{I} \text{ for all } n, \quad \text{and} \\ \|\mathsf{H}_{mn}\|_{2,2} \leq \mu \text{ for all } m \neq n\}. \end{aligned}$$

The spectral constraint set is the same as usual. The next proposition shows how to find the matrix in $\mathscr{H}$ closest to an initial matrix. In preparation,

define the truncation operator $[x]_\mu = \min\{x, \mu\}$ for nonnegative numbers, and extend it to nonnegative matrices by applying it to each component.

Proposition 7.8. *Let G be an Hermitian matrix. With respect to the Frobenius norm, the unique matrix in $\mathscr{H}(\mu)$ nearest to G has a block identity diagonal. If the off-diagonal block G_{mn} has a singular value decomposition $\mathsf{U}_{mn}\mathsf{C}_{mn}\mathsf{V}_{mn}^*$, then*

$$\mathsf{H}_{mn} \quad = \quad \begin{cases} \mathsf{G}_{mn} & \text{if } \|\mathsf{G}_{mn}\|_{2,2} \le \mu, \text{ and} \\ \mathsf{U}_{mn}\,[\mathsf{C}_{mn}]_\mu\,\mathsf{V}_{mn}^* & \text{otherwise.} \end{cases}$$

Proof. To determine the (m, n) off-diagonal block of the solution matrix H, we must solve the optimization problem

$$\min\nolimits_{\mathsf{A}} \ \tfrac{1}{2}\,\|\mathsf{A} - \mathsf{G}_{mn}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad \|\mathsf{A}\|_{2,2} \le \mu.$$

The Frobenius norm is strictly convex and the spectral norm is convex, so this problem has a unique solution.

Let $\boldsymbol{\sigma}(\cdot)$ return the vector of decreasingly ordered singular values of a matrix. Suppose that G_{mn} has the singular value decomposition $\mathsf{G}_{mn} = \mathsf{U}\,\operatorname{diag}\boldsymbol{\sigma}(\mathsf{G}_{mn})\,\mathsf{V}^*$. The constraint in the optimization problem depends only on the singular values of A, and so the Wielandt–Hoffman Theorem for singular values [61] allows us to check that the solution has the form $\mathsf{A} = \mathsf{U}\,\operatorname{diag}\boldsymbol{\sigma}(\mathsf{A})\,\mathsf{V}^*$.

To determine the singular values $\boldsymbol{\xi} = \boldsymbol{\sigma}(\mathsf{A})$ of the solution, we must solve the (strictly) convex program

$$\min\nolimits_{\boldsymbol{\xi}} \ \tfrac{1}{2}\,\|\boldsymbol{\xi} - \boldsymbol{\sigma}(\mathsf{G}_{mn})\|_2^2 \qquad \text{subject to} \qquad \xi_k \le \mu.$$

An easy application of Karush–Kuhn–Tucker theory [88] proves that the solution is obtained by truncating the singular values of G_{mn} that exceed μ. $\square$

7.4.5.1 Numerical Experiments

To our knowledge, there are no numerical studies of packing in Grassmannian spaces equipped with spectral distance. To gauge the quality of our results, we compare them against a new upper bound that we derive in Section 7.5. In the real or complex setting, a configuration $\mathscr{X}$ of N subspaces in $\mathbb{G}(K, \mathbb{F}^d)$ with respect to the spectral distance must satisfy the bound

$$\operatorname{pack}_{\text{spec}}(\mathscr{X})^2 \quad \leq \quad \frac{d-K}{d} \frac{N}{N-1}.$$

In the real case, the bound is attainable only if $N \leq \frac{1}{2}\, d\,(d+1) - \frac{1}{2}\, K\,(K+1) + 1$, while attainment in the complex case requires that $N \leq d^2 - K^2 + 1$ [66]. When a configuration meets the bound, the subspaces are not only equidistant but also *equi-isoclinic*. That is, all principal angles between all pairs of subspaces are identical.

We performed some limited experiments in an effort to produce good configurations of subspaces with respect to the spectral distance. We constructed random starting points using the modified version of Algorithm 7.4 with $\tau = 1$ (which represents no constraint). From the Rankin Bound, we calculated the smallest possible value of the feasibility parameter μ. For various values of μ, we ran the alternating projection, Algorithm 7.1, for 1000 to 5000 iterations, and we recorded the best packing radii that we obtained.

Table 7.6 displays the results of our calculations. We see that some of our configurations essentially meet the Rankin Bound, which means that they are equi-isoclinic. It is clear that alternating projection also succeeds reasonably well for this packing problem.

The most notable pathology in the table occurs for configurations of 8 and 9 subspaces in $\mathbb{G}(3, \mathbb{R}^6)$. In these cases, the algorithm always yielded

arrangements of subspaces with a zero packing radius, which implies that two of the subspaces coincide. Nevertheless, we were able to construct random starting points with a nonzero packing radius, which means that the algorithm is making the initial configuration worse. We do not understand the reason for this failure.

Figure 7.3 makes a graphical comparison between the real and complex subspace packings. On the whole, the complex packings are much better than the real packings. For example, every configuration of subspaces in $\mathbb{G}(2, \mathbb{C}^6)$ nearly meets the Rankin Bound, while just two of the real configurations achieve the same distinction. In comparison, it is curious how few arrangements in $\mathbb{G}(2, \mathbb{C}^5)$ come anywhere near the Rankin Bound.

7.4.6 Packings with Fubini–Study Distance

Packing with respect to the Fubini–Study distance is much harder. Suppose that we wish to construct a configuration of N subspaces whose Fubini–Study packing radius exceeds ρ. The feasibility condition is

$$\max_{m \neq n} \ |\det \mathsf{X}_m^* \mathsf{X}_n| \quad \leq \quad \mu \tag{7.12}$$

where $\mu = \cos \rho$. This leads to the structural constraint set

$$\begin{aligned} \mathscr{H}(\mu) \quad \stackrel{\text{def}}{=} \quad \{\mathsf{H} \in \mathbb{C}^{KN \times KN} : \mathsf{H} = \mathsf{H}^*, \quad \mathsf{H}_{nn} = \mathsf{I} \text{ for all } n, \quad \text{and} \\ |\det \mathsf{H}_{mn}| \leq \mu \text{ for all } m \neq n\}. \end{aligned}$$

Unhappily, this set is no longer convex. To produce a nearest matrix in $\mathscr{H}$, we must solve a nonlinear programming problem. The following proposition describes a numerically favorable formulation.

Proposition 7.9. *Let G be an Hermitian matrix. Suppose that the off-diagonal block G_{mn} has singular value decomposition $\mathsf{U}_{mn}\mathsf{C}_{mn}\mathsf{V}_{mn}^*$. Let $\boldsymbol{c}_{mn} = \operatorname{diag}\mathsf{C}_{mn}$, and find a (real) vector $\boldsymbol{x}_{mn}$ that solves the optimization problem*

$$\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\,\|\exp(\boldsymbol{x}) - \boldsymbol{c}_{mn}\|_2^2 \qquad \textit{subject to} \qquad \mathbf{e}^*\,\boldsymbol{x} \le \log\mu.$$

In Frobenius norm, a matrix $\mathsf{H}(\mu)$ from $\mathscr{H}$ that is closest to G has a block-identity diagonal and off-diagonal blocks

$$\mathsf{H}_{mn} \quad = \quad \begin{cases} \mathsf{G}_{mn} & \textit{if } |\det \mathsf{G}_{mn}| \le \mu, \textit{ and} \\ \mathsf{U}_{mn}\,\operatorname{diag}(\exp\boldsymbol{x}_{mn})\,\mathsf{V}_{mn}^* & \textit{otherwise.} \end{cases}$$

We use $\exp(\cdot)$ to denote the componentwise exponential of a vector. One may establish that the optimization problem is not convex by calculating the second derivative of the objective function.

Proof. To determine the (m, n) off-diagonal block of the solution matrix H, we must solve the optimization problem

$$\min\nolimits_{\mathsf{A}} \ \tfrac{1}{2}\,\|\mathsf{A} - \mathsf{G}_{mn}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad |\det \mathsf{A}| \le \mu.$$

We may reformulate this problem as

$$\min\nolimits_{\mathsf{A}} \ \tfrac{1}{2}\,\|\mathsf{A} - \mathsf{G}_{mn}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad \sum\nolimits_{k=1}^{K} \log\sigma_k(\mathsf{A}) \le \log\mu.$$

A familiar argument proves that the solution matrix has the same left and right singular vectors as G_{mn}. To obtain the singular values $\boldsymbol{\xi} = \boldsymbol{\sigma}(\mathsf{A})$ of the solution, we consider the mathematical program

$$\min\nolimits_{\boldsymbol{\xi}} \ \tfrac{1}{2}\,\|\boldsymbol{\xi} - \boldsymbol{\sigma}(\mathsf{G}_{mn})\|_2^2 \qquad \text{subject to} \qquad \sum\nolimits_{k=1}^{K} \log\xi_k \le \log\mu.$$

Change variables to complete the proof. □

7.4.6.1 Numerical Experiments

When we approach the problem of packing in Grassmannian manifolds equipped with the Fubini–Study distance, we are truly out in the wilderness. To our knowledge, the literature contains neither experimental nor theoretical treatments of this question. Moreover, we are not presently aware of general upper bounds on the Fubini–Study packing radius that we might use to assay the quality of a configuration of subspaces. Nevertheless, we attempted a few basic experiments.

We implemented the modified version of Algorithm 7.1 in Matlab, using the built-in nonlinear programming software to solve the optimization problem required by the proposition. For a few triples (d, K, N), we ran 100 to 500 iterations of the algorithm for various values of the feasibility parameter μ. (Given the exploratory nature of these experiments, we found that the implementation was too slow to increase the number of iterations.)

The results appear in Table 7.7. For small values of N, we find that the packings exhibit the maximum possible packing radius $\pi/2$, which shows that the algorithm is succeeding in these cases. For larger values of N, we are unable to judge how close the packings might decline from optimal.

Figure 7.4 compares the quality of our real packings against our complex packings. In each case, the complex packing is at least as good as the real packing, as we would expect. The smooth decline in the quality of the complex packings suggests that there is some underlying order to the packing radii, but it remains to be discovered.

To perform large-scale experiments, it will probably be necessary to tailor an algorithm that can solve the nonlinear programming problems more

quickly. It may also be essential to implement the alternating projection in a programming environment more efficient than Matlab. Therefore, a detailed study of packing with respect to the Fubini–Study distance must remain a topic for future research.

7.5 Bounds on Packing Radii

To assay the quality of our packings, it helps to have some upper bounds on the packing radius. These results suffice to establish that many of our packings are essentially optimal. Most of the material in this section has appeared in the literature, except for the final corollary on packing with respect to the Grassmannian spectral metric.

We begin with the Rankin Bound on the minimum distance among a set of points on the sphere.

Theorem 7.10 (Rankin [84]). *Suppose that $\{\boldsymbol{x}_n\}$ is a collection of N points on a sphere of radius r centered at the origin of a real Euclidean space. Then*

$$\max_{m \neq n} \langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle \quad \geq \quad \frac{r^2}{1 - N}.$$

It follows that

$$\min_{m \neq n} \|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2^2 \quad \leq \quad \frac{2\, r^2\, N}{N - 1}.$$

Equality holds if only if the points are equidistant. This event requires that $N \leq d + 1$.

Proof. Let G be the Gram matrix of the ensemble. Since the Gram matrix is positive semi-definite,

$$\sum\nolimits_{m,n} \langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle \quad = \quad \mathbf{e}^*\, \mathsf{G}\, \mathbf{e} \quad \geq \quad 0.$$

On the other hand, if we define $\mu = \max_{m \neq n} \langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle$, then

$$0 \quad \leq \quad \sum\nolimits_{m,n} \langle \boldsymbol{x}_m, \boldsymbol{x}_n \rangle \quad \leq \quad r^2 N \;+\; N(N-1)\,\mu.$$

Rearrange to develop the bound on the maximum inner product. The distance bound follows after a little more algebra.

Equality requires that each inner product equal μ, which implies that each pair of points is equidistant. It follows that the points lie at vertices of a regular simplex. In d dimensions, a simplex contains exactly $(d+1)$ vertices. Conversely, a calculation shows the vertices of a regular simplex yield equality in the bounds. □

We require upper bounds on the radii of subspace packings. Conway, Hardin, and Sloane have developed a wonderful approach to this problem. They first embed the chordal Grassmannian space isometrically into a Euclidean sphere, and then they apply Rankin's Bound.

Theorem 7.11 (Conway–Hardin–Sloane [15]). *The chordal Grassmannian space $\mathbb{G}(K, \mathbb{F}^d)$ may be embedded isometrically into a real Euclidean sphere whose squared radius is $\frac{1}{2} K(d-K)/d$. When $\mathbb{F} = \mathbb{R}$, the dimension of the embedding space is $\binom{d+1}{2} - 1$. When $\mathbb{F} = \mathbb{C}$, the dimension is $d^2 - 1$.*

Proof. Suppose that the columns of S and T form orthonormal bases for two K-dimensional subspaces of $\mathbb{F}^d$. Then we may calculate that

$$\begin{aligned} 2\,\mathrm{dist}_{\mathrm{chord}}(\mathsf{S}, \mathsf{T})^2 &= 2\,(K - \|\mathsf{S}^*\,\mathsf{T}\|_{\mathrm{F}}^2) \\ &= \|\mathsf{S}\mathsf{S}^*\|_{\mathrm{F}}^2 + \|\mathsf{T}\mathsf{T}^*\|_{\mathrm{F}}^2 - 2\,\mathrm{Re}\,\mathrm{trace}\,\mathsf{S}\mathsf{S}^*\mathsf{T}\mathsf{T}^* \\ &= \|\mathsf{S}\mathsf{S}^* - \mathsf{T}\mathsf{T}^*\|_{\mathrm{F}}^2\,. \end{aligned}$$

That is, the squared chordal distance between two subspaces is equal to half the squared Frobenius distance between the orthogonal projectors onto the two subspaces.

Suppose that $\mathscr{S}$ is a K-dimensional subspace of $\mathbb{F}^d$, and let $\mathbf{P}$ be the unique orthogonal projector onto $\mathscr{S}$. The projector has trace K, so we may subtract a multiple of the identity to zero the trace: $\widehat{\mathbf{P}} = \mathbf{P} - (K/d)\,\mathbf{I}$. The new matrix satisfies $\|\widehat{\mathbf{P}}\|_{\mathrm{F}}^2 = K\,(d-K)/d$, so the de-traced projectors all lie on a sphere. Moreover, since we translate each rank-K projector by the same amount, this operation does not change the Frobenius distance between them. Therefore, the map $\mathscr{S} \mapsto \frac{1}{\sqrt{2}}\,\widehat{\mathbf{P}}$ is an isometric embedding of the chordal Grassmannian space into a Euclidean sphere with squared radius $\frac{1}{2}\,K\,(d-K)/d$.

To determine the dimension of the embedding space, we count the degrees of freedom in the de-traced projectors. In the real case, a d-dimensional symmetric matrix contains $\frac{1}{2}\,d\,(d+1)$ free real-valued entries, from which we subtract one to account for the fixed trace. In the complex case, a d-dimensional Hermitian matrix contains d real entries on the diagonal and $\frac{1}{2}\,d\,(d-1)$ complex entries in the strict lower triangle. Accounting for the trace, we conclude that the embedding dimension is $d^2 - 1$. □

Combining Theorem 7.10 with Theorem 7.11, we obtain an upper bound on the packing radius.

Corollary 7.12 (Conway–Hardin–Sloane [15]). *An upper bound on the packing radius of N subspaces in the chordal Grassmannian space $\mathbb{G}(K, \mathbb{F}^d)$ is*

$$\mathrm{pack}_{\mathrm{chord}}(\mathscr{X})^2 \quad \leq \quad \frac{K\,(d-K)}{d}\,\frac{N}{N-1}. \tag{7.13}$$

If the bound is met, all pairs of subspaces are equidistant. When $\mathbb{F} = \mathbb{R}$*, the bound is attainable only if* $N \leq \frac{1}{2}\, d\,(d+1)$*. When* $\mathbb{F} = \mathbb{C}$*, the bound is attainable only if* $N \leq d^2$*.*

We will refer to the inequality (7.13) as the Rankin Bound for packings with respect to the chordal distance. When $K = 1$, the corollary applies to packings in projective space. Finally, we draw a new corollary that gives a bound for the spectral distance.

Corollary 7.13. *We have the following bound on the packing radius of* N *subspaces in the Grassmannian space* $\mathbb{G}(K, \mathbb{F}^d)$ *equipped with the spectral distance.*

$$\operatorname{pack}_{\text{spec}}(\mathscr{X})^2 \quad \leq \quad \frac{d-K}{d}\,\frac{N}{N-1}. \tag{7.14}$$

If the bound is met, all pairs of subspaces are equi-isoclinic. When $\mathbb{F} = \mathbb{R}$*, the bound is attainable only if* $N \leq \frac{1}{2}\, d\,(d+1)$*. When* $\mathbb{F} = \mathbb{C}$*, the bound is attainable only if* $N \leq d^2$*.*

Proof. The monotonicity of power means [58] yields the inequality

$$\min_k \, \sin\theta_k \quad \leq \quad \left[K^{-1} \sum\nolimits_{k=1}^{K} \sin^2\theta_k\right]^{1/2}.$$

For angles between zero and $\pi/2$, equality holds if and only if $\theta_1 = \cdots = \theta_K$. It follows that

$$\operatorname{pack}_{\text{spec}}(\mathscr{X})^2 \quad \leq \quad K^{-1}\operatorname{pack}_{\text{chord}}(\mathscr{X})^2 \quad \leq \quad \frac{d-K}{d}\,\frac{N}{N-1}.$$

If the second inequality is met, then all pairs of subspaces are equidistant with respect to the chordal metric. Moreover, if the first inequality is met, then the principal angles between each pair of subspaces are constant. That is, the subspaces are equi-isoclinic. □

In fact, the previous corollary overestimates the maximum possible number of equi-isoclinic subspaces. The following result is better, although its authors still do not believe it is sharp.

Theorem 7.14 (Lemmens–Seidel [66]). *The maximum number of equi-isoclinic K-dimensional subspaces of $\mathbb{R}^d$ is no greater than*

$$\tfrac{1}{2}\, d\,(d+1) - \tfrac{1}{2}\, K\,(K+1) + 1.$$

Similarly, the maximum number of equi-isoclinic K-dimensional subspaces of $\mathbb{C}^d$ does not exceed

$$d^2 - K^2 + 1.$$

We do not know any bounds for packings with respect to the Fubini–Study metric.

7.6 Conclusions

We have shown that the alternating projection approach can be used to solve many different packing problems. The method is easy to understand and to implement, even while it is versatile and powerful. In cases where experiments have been performed, we have often been able to match the best packings known. Moreover, we have extended the method to solve problems that have not been studied numerically. Using the Rankin Bounds, we have been able to show that many of our packings are essentially optimal.

Alternating projection does have some shortcomings. It converges slowly, and it sometimes fails to match the best packings in the literature. In particular, the algorithm seems to falter when the number of points becomes too large. Nevertheless, the flexibility of the algorithm probably compensates for its deficiencies.

7.6.1 Future Work

There are many possibilities for future experimental and theoretical work on packing in projective spaces and Grassmannian spaces. Let us mention a few ideas.

It is possible to enforce stricter spectral constraints on the Gram matrix. For example, a *tight frame* is a projective packing whose Gram matrix has identical (nonzero) eigenvalues. Tight frames have many striking properties, and they have received a lot of recent attention from the signal processing community. Our methods can be used to construct tight frames that are also good projective packings. These configurations have applications in coding and communications [100].

The sphere and the complex projective space are both examples of two-point homogeneous spaces. Another member of this class is the quaternionic projective space [16]. To our knowledge, no one has developed numerical algorithms that can approach the problem of packing lines in quaternionic projective spaces, although this problem may have applications in coding theory. The alternating projection method requires no conceptual adjustment. Of course, it may take a serious effort to implement quaternionic arithmetic and linear algebra.

Our experiments provided many essentially optimal configurations of subspaces in the Grassmannian manifold equipped with chordal distance. These configurations have not received very much theoretical attention, and it would be interesting to develop algebraic constructions. Our experiments also point toward many equi-isoclinic arrangements of subspaces. We would also like to develop algebraic constructions of these.

To our knowledge, no one has studied packing with respect to the Fubini–Study distance, even though it is one of the most natural metrics for the Grassmannian space. It would be highly desirable to prove upper bounds on the Fubini–Study packing radius. We would also like to perform additional experiments to develop a more comprehensive view of packing with respect to this distance.

These problems are very attractive geometrically, and they are becoming increasingly important in electrical engineering. We hope to be able to study them more extensively.

7.7 Tables and Figures

Table 7.1: PACKING ON SPHERES: For collections of N points on the $(d-1)$-dimensional sphere, this table lists the best packing radius and the average packing radius obtained during ten random trials of the alternating projection algorithm. The error columns record how far our results decline from the putative optimal packings (NJAS) reported in [96]. The last column gives the average number of iterations of alternating projection per trial.

		PACKING RADII (DEGREES)					ITERATIONS
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
3	4	109.471	109.471	0.001	109.471	0.001	45
3	5	90.000	90.000	0.000	89.999	0.001	130
3	6	90.000	90.000	0.000	90.000	0.000	41
3	7	77.870	77.869	0.001	77.869	0.001	613
3	8	74.858	74.858	0.001	74.858	0.001	328
3	9	70.529	70.528	0.001	70.528	0.001	814
3	10	66.147	66.140	0.007	66.010	0.137	5000
3	11	63.435	63.434	0.001	63.434	0.001	537
3	12	63.435	63.434	0.001	63.434	0.001	209
3	13	57.137	57.136	0.001	56.571	0.565	4876
3	14	55.671	55.670	0.001	55.439	0.232	3443
3	15	53.658	53.620	0.038	53.479	0.178	5000
3	16	52.244	52.243	0.001	51.665	0.579	4597
3	17	51.090	51.084	0.007	51.071	0.019	5000
3	18	49.557	49.548	0.008	49.506	0.050	5000
3	19	47.692	47.643	0.049	47.434	0.258	5000
3	20	47.431	47.429	0.002	47.254	0.177	5000
3	21	45.613	45.576	0.037	45.397	0.217	5000
3	22	44.740	44.677	0.063	44.123	0.617	5000
3	23	43.710	43.700	0.009	43.579	0.131	5000
3	24	43.691	43.690	0.001	43.689	0.002	3634
3	25	41.634	41.458	0.177	41.163	0.471	5000

continued...

. . . continued

		Packing Radii (Degrees)					Iterations
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
4	5	104.478	104.478	0.000	104.267	0.211	2765
4	6	90.000	90.000	0.000	89.999	0.001	110
4	7	90.000	89.999	0.001	89.999	0.001	483
4	8	90.000	90.000	0.000	89.999	0.001	43
4	9	80.676	80.596	0.081	80.565	0.111	5000
4	10	80.406	80.405	0.001	77.974	2.432	2107
4	11	76.679	76.678	0.001	75.881	0.798	2386
4	12	75.522	75.522	0.001	74.775	0.748	3286
4	13	72.104	72.103	0.001	71.965	0.139	4832
4	14	71.366	71.240	0.126	71.184	0.182	5000
4	15	69.452	69.450	0.002	69.374	0.078	5000
4	16	67.193	67.095	0.098	66.265	0.928	5000
4	17	65.653	65.652	0.001	64.821	0.832	4769
4	18	64.987	64.987	0.001	64.400	0.587	4713
4	19	64.262	64.261	0.001	64.226	0.036	4444
4	20	64.262	64.261	0.001	64.254	0.008	3738
4	21	61.876	61.864	0.012	61.570	0.306	5000
4	22	60.140	60.084	0.055	59.655	0.485	5000
4	23	60.000	59.999	0.001	58.582	1.418	4679
4	24	60.000	58.209	1.791	57.253	2.747	5000
4	25	57.499	57.075	0.424	56.871	0.628	5000
5	6	101.537	101.536	0.001	95.585	5.952	4056
5	7	90.000	89.999	0.001	89.999	0.001	1540
5	8	90.000	89.999	0.001	89.999	0.001	846
5	9	90.000	89.999	0.001	89.999	0.001	388
5	10	90.000	90.000	0.000	89.999	0.001	44
5	11	82.365	82.300	0.065	81.937	0.429	5000
5	12	81.145	81.145	0.001	80.993	0.152	4695
5	13	79.207	79.129	0.078	78.858	0.349	5000
5	14	78.463	78.462	0.001	78.280	0.183	1541
5	15	78.463	78.462	0.001	77.477	0.986	1763
5	16	78.463	78.462	0.001	78.462	0.001	182
5	17	74.307	74.307	0.001	73.862	0.446	4147
5	18	74.008	74.007	0.001	73.363	0.645	3200

continued. . .

. . . continued

		Packing Radii (Degrees)					Iterations
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
5	19	73.033	73.016	0.017	72.444	0.589	5000
5	20	72.579	72.579	0.001	72.476	0.104	4689
5	21	71.644	71.639	0.005	71.606	0.039	5000
5	22	69.207	68.683	0.524	68.026	1.181	5000
5	23	68.298	68.148	0.150	67.568	0.731	5000
5	24	68.023	68.018	0.006	67.127	0.896	5000
5	25	67.690	67.607	0.083	66.434	1.256	5000

Table 7.2: PACKING IN REAL PROJECTIVE SPACES: For collections of N points in the $(d-1)$-dimensional real projective space, this table lists the best packing radius and the average packing radius obtained during ten random trials of the alternating projection algorithm. The error columns record how far our results decline from the putative optimal packings (NJAS) reported in [95]. The last column gives the average number of iterations of alternating projection per trial.

		PACKING RADII (DEGREES)					ITERATIONS
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
3	4	70.529	70.528	0.001	70.528	0.001	54
3	5	63.435	63.434	0.001	63.434	0.001	171
3	6	63.435	63.435	0.000	59.834	3.601	545
3	7	54.736	54.735	0.001	54.735	0.001	341
3	8	49.640	49.639	0.001	49.094	0.546	4333
3	9	47.982	47.981	0.001	47.981	0.001	2265
3	10	46.675	46.674	0.001	46.674	0.001	2657
3	11	44.403	44.402	0.001	44.402	0.001	2173
3	12	41.882	41.881	0.001	41.425	0.457	2941
3	13	39.813	39.812	0.001	39.522	0.291	4870
3	14	38.682	38.462	0.221	38.378	0.305	5000
3	15	38.135	37.934	0.201	37.881	0.254	5000
3	16	37.377	37.211	0.166	37.073	0.304	5000
3	17	35.235	35.078	0.157	34.821	0.414	5000
3	18	34.409	34.403	0.005	34.200	0.209	5000
3	19	33.211	33.107	0.104	32.909	0.303	5000
3	20	32.707	32.580	0.127	32.273	0.434	5000
3	21	32.216	32.036	0.180	31.865	0.351	5000
3	22	31.896	31.853	0.044	31.777	0.119	5000
3	23	30.506	30.390	0.116	30.188	0.319	5000
3	24	30.163	30.089	0.074	29.694	0.469	5000
3	25	29.249	29.024	0.224	28.541	0.707	5000
4	5	75.522	75.522	0.001	73.410	2.113	4071
4	6	70.529	70.528	0.001	70.528	0.001	91
4	7	67.021	67.021	0.001	67.021	0.001	325

continued...

. . . continued

		Packing Radii (Degrees)					Iterations
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
4	8	65.530	65.530	0.001	64.688	0.842	3134
4	9	64.262	64.261	0.001	64.261	0.001	1843
4	10	64.262	64.261	0.001	64.261	0.001	803
4	11	60.000	59.999	0.001	59.999	0.001	577
4	12	60.000	59.999	0.001	59.999	0.001	146
4	13	55.465	55.464	0.001	54.390	1.074	4629
4	14	53.838	53.833	0.005	53.405	0.433	5000
4	15	52.502	52.493	0.009	51.916	0.585	5000
4	16	51.827	51.714	0.113	50.931	0.896	5000
4	17	50.887	50.834	0.053	50.286	0.601	5000
4	18	50.458	50.364	0.094	49.915	0.542	5000
4	19	49.711	49.669	0.041	49.304	0.406	5000
4	20	49.233	49.191	0.042	48.903	0.330	5000
4	21	48.548	48.464	0.084	48.374	0.174	5000
4	22	47.760	47.708	0.052	47.508	0.251	5000
4	23	46.510	46.202	0.308	45.789	0.722	5000
4	24	46.048	45.938	0.110	45.725	0.322	5000
4	25	44.947	44.739	0.208	44.409	0.538	5000
5	6	78.463	78.463	0.001	77.359	1.104	3246
5	7	73.369	73.368	0.001	73.368	0.001	1013
5	8	70.804	70.803	0.001	70.604	0.200	5000
5	9	70.529	70.528	0.001	69.576	0.953	2116
5	10	70.529	70.528	0.001	67.033	3.496	3029
5	11	67.254	67.254	0.001	66.015	1.239	4615
5	12	67.021	66.486	0.535	65.661	1.361	5000
5	13	65.732	65.720	0.012	65.435	0.297	5000
5	14	65.724	65.723	0.001	65.637	0.087	3559
5	15	65.530	65.492	0.038	65.443	0.088	5000
5	16	63.435	63.434	0.001	63.434	0.001	940
5	17	61.255	61.238	0.017	60.969	0.287	5000
5	18	61.053	61.048	0.005	60.946	0.107	5000
5	19	60.000	58.238	1.762	57.526	2.474	5000
5	20	60.000	59.999	0.001	56.183	3.817	3290
5	21	57.202	57.134	0.068	56.159	1.043	5000

continued. . .

. . . continued

		Packing Radii (Degrees)					Iterations
d	N	NJAS	Best of 10	Error	Avg. of 10	Error	Avg. of 10
5	22	56.356	55.819	0.536	55.173	1.183	5000
5	23	55.588	55.113	0.475	54.535	1.053	5000
5	24	55.228	54.488	0.740	53.926	1.302	5000
5	25	54.889	54.165	0.724	52.990	1.899	5000

Table 7.3: Packing in complex projective spaces: We compare our best configurations (JAT) of N points in $\mathbb{P}^{d-1}(\mathbb{C})$ against the Rankin Bound, equation (7.13). The packing radius of an ensemble is measured as the acute angle between the closest pair of lines. The final column shows how far our configurations fall short of the bound.

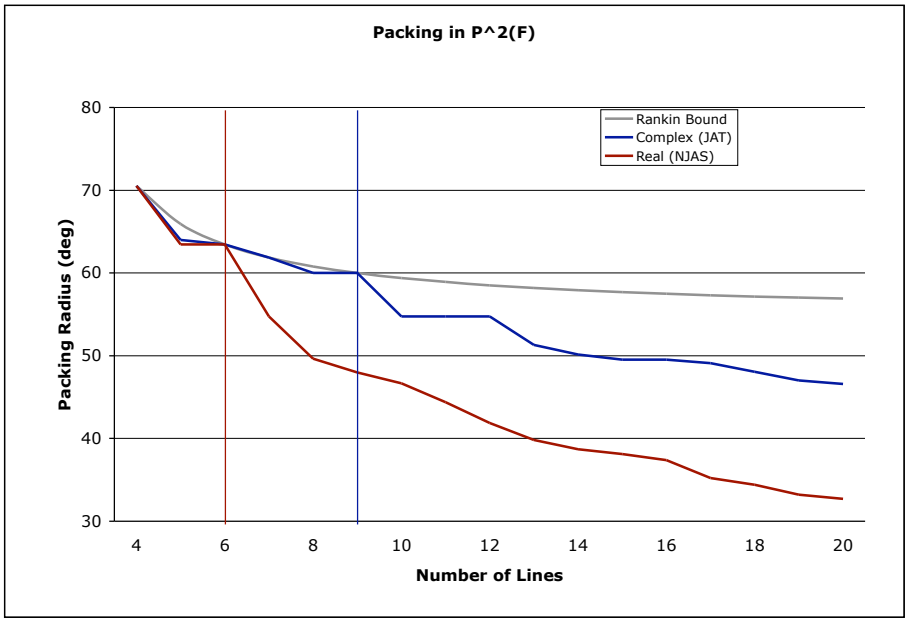
		Packing Radii (Degrees)		
d	N	JAT	Rankin	Difference
2	3	60.00	60.00	0.00
2	4	54.74	54.74	0.00
2	5	45.00	52.24	7.24
2	6	45.00	50.77	5.77
2	7	38.93	49.80	10.86
2	8	37.41	49.11	11.69
3	4	70.53	70.53	0.00
3	5	64.00	65.91	1.90
3	6	63.44	63.43	0.00
3	7	61.87	61.87	0.00
3	8	60.00	60.79	0.79
3	9	60.00	60.00	0.00
3	10	54.73	59.39	4.66
3	11	54.73	58.91	4.18
3	12	54.73	58.52	3.79
3	13	51.32	58.19	6.88
3	14	50.13	57.92	7.79
3	15	49.53	57.69	8.15
3	16	49.53	57.49	7.95
3	17	49.10	57.31	8.21
3	18	48.07	57.16	9.09
3	19	47.02	57.02	10.00
3	20	46.58	56.90	10.32
4	5	75.52	75.52	0.00
4	6	70.88	71.57	0.68
4	7	69.29	69.30	0.01
4	8	67.78	67.79	0.01

continued...

. . . continued

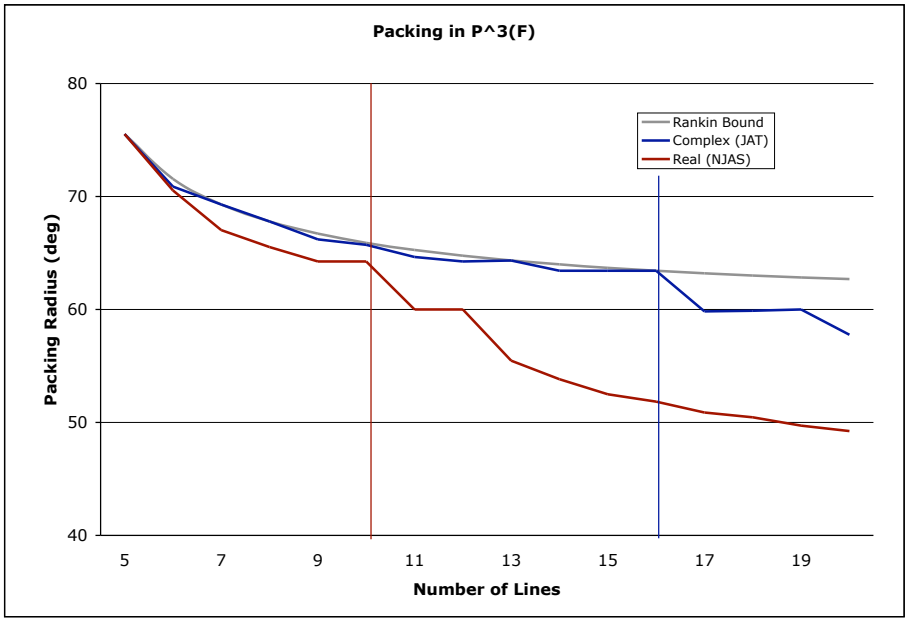
		Packing Radii (Degrees)		
d	N	JAT	Rankin	Difference
4	9	66.21	66.72	0.51
4	10	65.71	65.91	0.19
4	11	64.64	65.27	0.63
4	12	64.24	64.76	0.52
4	13	64.34	64.34	0.00
4	14	63.43	63.99	0.56
4	15	63.43	63.69	0.26
4	16	63.43	63.43	0.00
4	17	59.84	63.21	3.37
4	18	59.89	63.02	3.12
4	19	60.00	62.84	2.84
4	20	57.76	62.69	4.93
5	6	78.46	78.46	0.00
5	7	74.52	75.04	0.51
5	8	72.81	72.98	0.16
5	9	71.24	71.57	0.33
5	10	70.51	70.53	0.02
5	11	69.71	69.73	0.02
5	12	68.89	69.10	0.21
5	13	68.19	68.58	0.39
5	14	67.66	68.15	0.50
5	15	67.37	67.79	0.43
5	16	66.68	67.48	0.80
5	17	66.53	67.21	0.68
5	18	65.87	66.98	1.11
5	19	65.75	66.77	1.02
5	20	65.77	66.59	0.82
5	21	65.83	66.42	0.60
5	22	65.87	66.27	0.40
5	23	65.90	66.14	0.23
5	24	65.91	66.02	0.11
5	25	65.91	65.91	0.00

Figure 7.1: Real and Complex Projective Packings: These three graphs compare the packing radii attained by configurations in real and complex projective spaces. The red line indicates the best real packings obtained by Sloane and his colleagues [95]. The blue line indicates the best complex packings produced by the author. Rankin's upper bound (7.13) is depicted in gray. The vertical red line marks the largest number of real lines for which the Rankin Bound is attainable, while the blue line marks the maximum number of complex lines for which the Rankin Bound is attainable.



continued...

. . . continued


Packing in P^3(F)
Packing Radius (deg)
80
70
60
50
40
5
7
9
11
13
15
17
19
Number of Lines
Rankin Bound
Complex (JAT)
Real (NJAS)

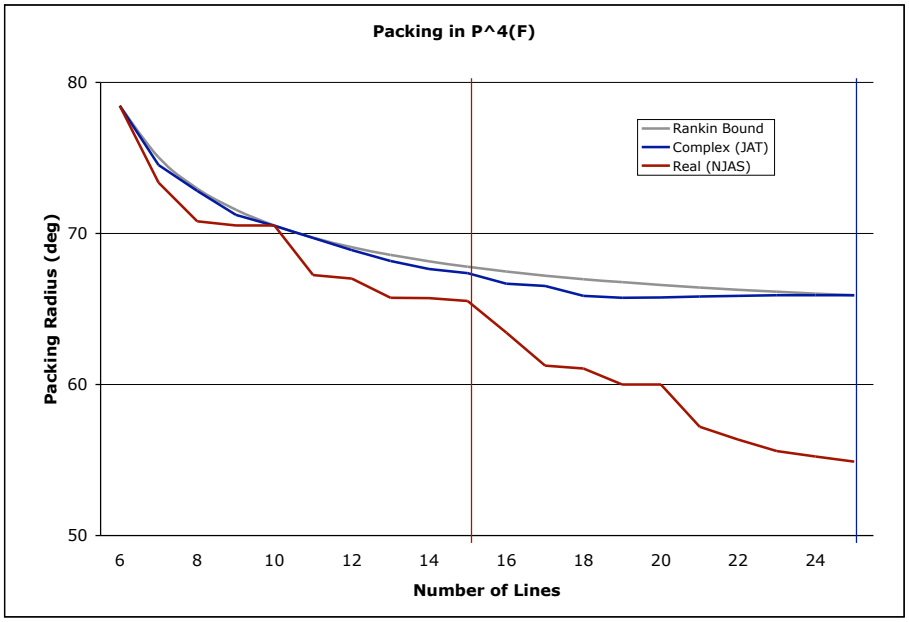

Packing in P^4(F)
Packing Radius (deg)
80
70
60
50
6
8
10
12
14
16
18
20
22
24
Number of Lines
Rankin Bound
Complex (JAT)
Real (NJAS)

Table 7.4: Packing in real Grassmannians with chordal distance: We compare our best configurations (JAT) of N points in $\mathbb{G}(K, \mathbb{R}^d)$ against the best packings (NJAS) reported in [95]. The squared packing radius is the squared chordal distance (7.7) between the closest pair of subspaces. The last column lists the difference between the columns (NJAS) and (JAT).

			Squared Packing Radii		
d	K	N	JAT	NJAS	Difference
2	4	3	1.5000	1.5000	0.0000
2	4	4	1.3333	1.3333	0.0000
2	4	5	1.2500	1.2500	0.0000
2	4	6	1.2000	1.2000	0.0000
2	4	7	1.1656	1.1667	0.0011
2	4	8	1.1423	1.1429	0.0005
2	4	9	1.1226	1.1231	0.0004
2	4	10	1.1111	1.1111	0.0000
2	4	11	0.9981	1.0000	0.0019
2	4	12	0.9990	1.0000	0.0010
2	4	13	0.9996	1.0000	0.0004
2	4	14	1.0000	1.0000	0.0000
2	4	15	1.0000	1.0000	0.0000
2	4	16	0.9999	1.0000	0.0001
2	4	17	1.0000	1.0000	0.0000
2	4	18	0.9992	1.0000	0.0008
2	4	19	0.8873	0.9091	0.0218
2	4	20	0.8225	0.9091	0.0866
2	5	3	1.7500	1.7500	0.0000
2	5	4	1.6000	1.6000	0.0000
2	5	5	1.5000	1.5000	0.0000
2	5	6	1.4400	1.4400	0.0000
2	5	7	1.4000	1.4000	0.0000
2	5	8	1.3712	1.3714	0.0002
2	5	9	1.3464	1.3500	0.0036
2	5	10	1.3307	1.3333	0.0026
2	5	11	1.3069	1.3200	0.0131

continued...

. . . continued

			Squared Packing Radii		
d	K	N	JAT	NJAS	Difference
2	5	12	1.2973	1.3064	0.0091
2	5	13	1.2850	1.2942	0.0092
2	5	14	1.2734	1.2790	0.0056
2	5	15	1.2632	1.2707	0.0075
2	5	16	1.1838	1.2000	0.0162
2	5	17	1.1620	1.2000	0.0380
2	5	18	1.1589	1.1909	0.0319
2	5	19	1.1290	1.1761	0.0472
2	5	20	1.0845	1.1619	0.0775

Table 7.5: Packing in complex Grassmannians with chordal distance: We compare our best configurations (JAT) of N points in $\mathbb{G}(K, \mathbb{C}^d)$ against the Rankin Bound, equation (7.13). The squared packing radius is calculated as the squared chordal distance (7.7) between the closest pair of subspaces. The final column shows how much the computed ensemble declines from the Rankin Bound. When the bound is met, all pairs of subspaces are equidistant.

			Squared Packing Radii		
d	K	N	JAT	NJAS	Difference
2	4	3	1.5000	1.5000	0.0000
2	4	4	1.3333	1.3333	0.0000
2	4	5	1.2500	1.2500	0.0000
2	4	6	1.2000	1.2000	0.0000
2	4	7	1.1667	1.1667	0.0000
2	4	8	1.1429	1.1429	0.0000
2	4	9	1.1250	1.1250	0.0000
2	4	10	1.1111	1.1111	0.0000
2	4	11	1.0999	1.1000	0.0001
2	4	12	1.0906	1.0909	0.0003
2	4	13	1.0758	1.0833	0.0076
2	4	14	1.0741	1.0769	0.0029
2	4	15	1.0698	1.0714	0.0016
2	4	16	1.0658	1.0667	0.0009
2	4	17	0.9975	1.0625	0.0650
2	4	18	0.9934	1.0588	0.0654
2	4	19	0.9868	1.0556	0.0688
2	4	20	0.9956	1.0526	0.0571
2	5	3	1.7500	1.8000	0.0500
2	5	4	1.6000	1.6000	0.0000
2	5	5	1.5000	1.5000	0.0000
2	5	6	1.4400	1.4400	0.0000
2	5	7	1.4000	1.4000	0.0000
2	5	8	1.3714	1.3714	0.0000
2	5	9	1.3500	1.3500	0.0000

continued...

. . . continued

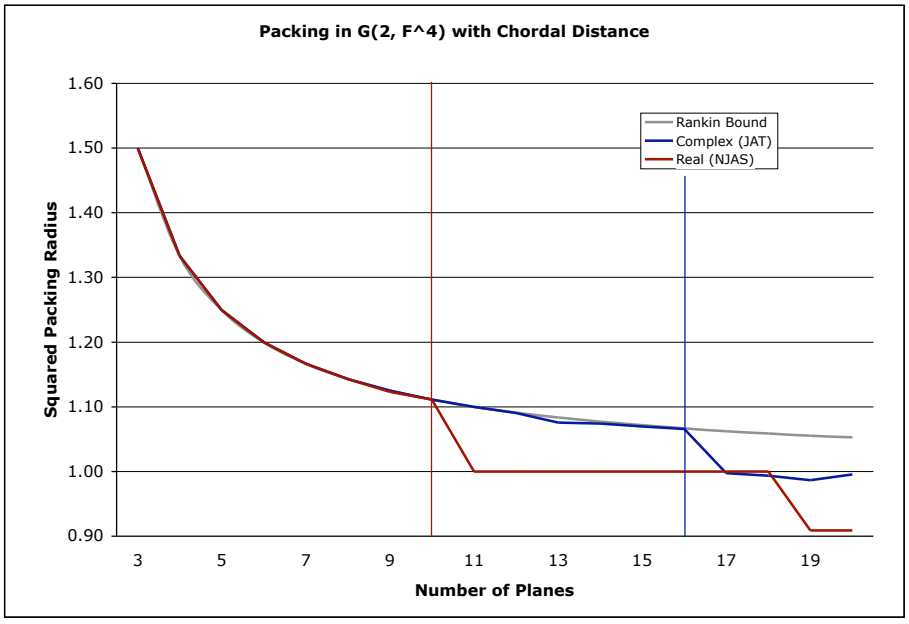
			Squared Packing Radii		
d	K	N	JAT	NJAS	Difference
2	5	10	1.3333	1.3333	0.0000
2	5	11	1.3200	1.3200	0.0000
2	5	12	1.3090	1.3091	0.0001
2	5	13	1.3000	1.3000	0.0000
2	5	14	1.2923	1.2923	0.0000
2	5	15	1.2857	1.2857	0.0000
2	5	16	1.2799	1.2800	0.0001
2	5	17	1.2744	1.2750	0.0006
2	5	18	1.2686	1.2706	0.0020
2	5	19	1.2630	1.2667	0.0037
2	5	20	1.2576	1.2632	0.0056
2	6	4	1.7778	1.7778	0.0000
2	6	5	1.6667	1.6667	0.0000
2	6	6	1.6000	1.6000	0.0000
2	6	7	1.5556	1.5556	0.0000
2	6	8	1.5238	1.5238	0.0000
2	6	9	1.5000	1.5000	0.0000
2	6	10	1.4815	1.4815	0.0000
2	6	11	1.4667	1.4667	0.0000
2	6	12	1.4545	1.4545	0.0000
2	6	13	1.4444	1.4444	0.0000
2	6	14	1.4359	1.4359	0.0000
2	6	15	1.4286	1.4286	0.0000
2	6	16	1.4221	1.4222	0.0001
2	6	17	1.4166	1.4167	0.0000
2	6	18	1.4118	1.4118	0.0000
2	6	19	1.4074	1.4074	0.0000
2	6	20	1.4034	1.4035	0.0001
2	6	21	1.3999	1.4000	0.0001
2	6	22	1.3968	1.3968	0.0001
2	6	23	1.3923	1.3939	0.0017
2	6	24	1.3886	1.3913	0.0028
2	6	25	1.3862	1.3889	0.0027
3	6	3	2.2500	2.2500	0.0000

continued. . .

. . . continued

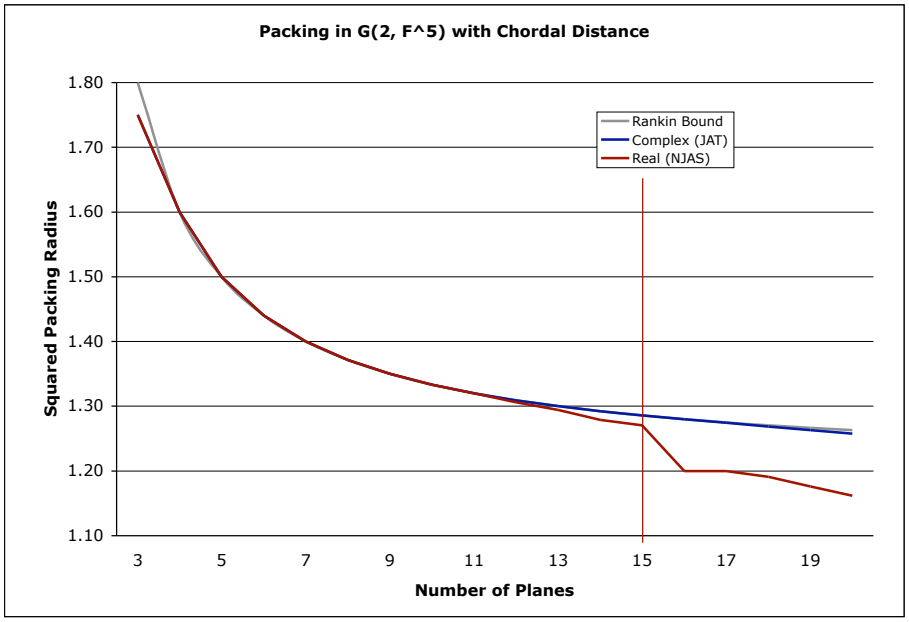
			Squared Packing Radii		
d	K	N	JAT	NJAS	Difference
3	6	4	2.0000	2.0000	0.0000
3	6	5	1.8750	1.8750	0.0000
3	6	6	1.8000	1.8000	0.0000
3	6	7	1.7500	1.7500	0.0000
3	6	8	1.7143	1.7143	0.0000
3	6	9	1.6875	1.6875	0.0000
3	6	10	1.6667	1.6667	0.0000
3	6	11	1.6500	1.6500	0.0000
3	6	12	1.6363	1.6364	0.0001
3	6	13	1.6249	1.6250	0.0001
3	6	14	1.6153	1.6154	0.0000
3	6	15	1.6071	1.6071	0.0000
3	6	16	1.5999	1.6000	0.0001
3	6	17	1.5936	1.5938	0.0001
3	6	18	1.5879	1.5882	0.0003
3	6	19	1.5829	1.5833	0.0004
3	6	20	1.5786	1.5789	0.0004
3	6	21	1.5738	1.5750	0.0012
3	6	22	1.5687	1.5714	0.0028
3	6	23	1.5611	1.5682	0.0070
3	6	24	1.5599	1.5652	0.0053
3	6	25	1.5558	1.5625	0.0067
3	6	26	1.5542	1.5600	0.0058
3	6	27	1.5507	1.5577	0.0070
3	6	28	1.5502	1.5556	0.0054
3	6	29	1.5443	1.5536	0.0092
3	6	30	1.5316	1.5517	0.0201
3	6	31	1.5283	1.5500	0.0217
3	6	32	1.5247	1.5484	0.0237
3	6	33	1.5162	1.5469	0.0307
3	6	34	1.5180	1.5455	0.0274
3	6	35	1.5141	1.5441	0.0300
3	6	36	1.5091	1.5429	0.0338

Figure 7.2: Packing in Grassmannians with chordal distance: The red line indicates the best real packings obtained by Sloane and his colleagues [95]. The blue line indicates the best complex packings produced by the author. Rankin's upper bound (7.13) appears in gray. The vertical red line marks the largest number of real subspaces for which the Rankin Bound is attainable, while the blue line marks the maximum number of complex subspaces for which the Rankin Bound is attainable.



continued...

. . . continued


Packing in G(2, F^5) with Chordal Distance
Squared Packing Radius
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
3
5
7
9
11
13
15
17
19
Number of Planes
Rankin Bound
Complex (JAT)
Real (NJAS)

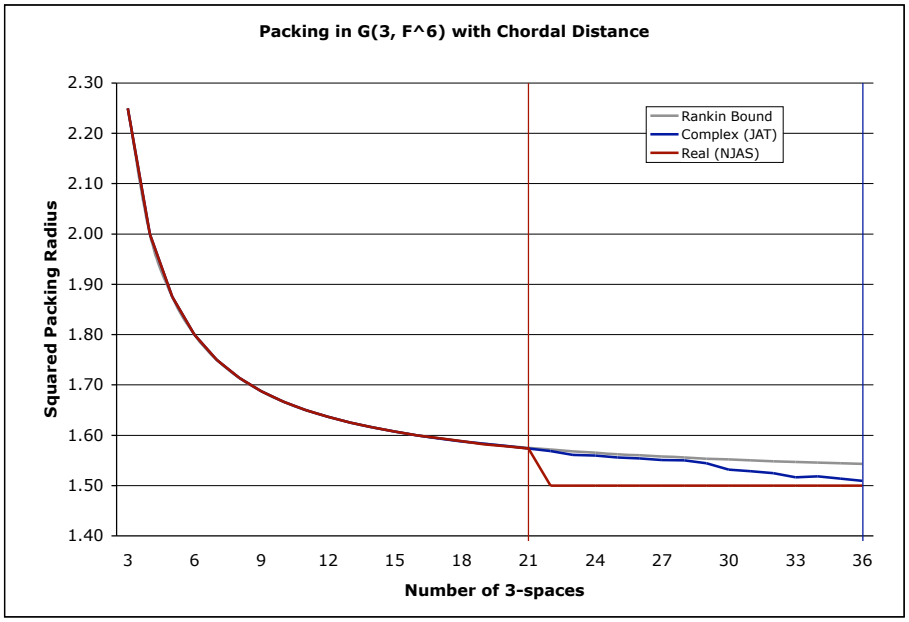

Packing in G(3, F^6) with Chordal Distance
Squared Packing Radius
2.30
2.20
2.10
2.00
1.90
1.80
1.70
1.60
1.50
1.40
3
6
9
12
15
18
21
24
27
30
33
36
Number of 3-spaces
Rankin Bound
Complex (JAT)
Real (NJAS)

Table 7.6: Packing in Grassmannians with spectral distance: We compare our best real ($\mathbb{F} = \mathbb{R}$) and complex ($\mathbb{F} = \mathbb{C}$) packings in $\mathbb{G}(K, \mathbb{F}^d)$ against the Rankin Bound, equation (7.14). The squared packing radius of a configuration is the squared spectral distance (7.8) between the closest pair of subspaces. When the Rankin Bound is met, all pairs of subspaces are equi-isoclinic. The algorithm failed to produce configurations of 8 and 9 subspaces in $\mathbb{G}(3, \mathbb{R}^6)$ with reasonable packing radii.

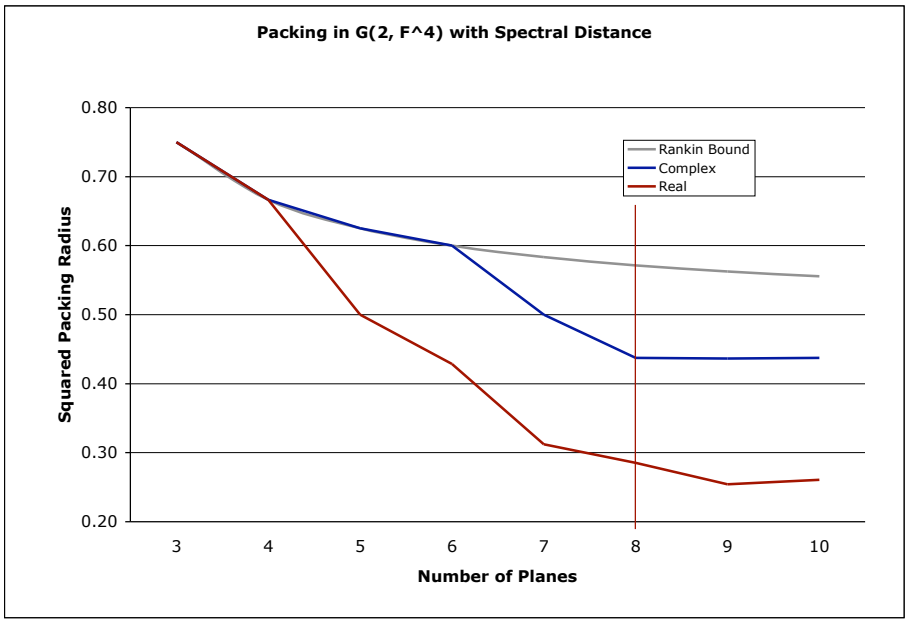
			Squared Packing Radii				
d	K	N	Rankin	$\mathbb{R}$	Difference	$\mathbb{C}$	Difference
4	2	3	0.7500	0.7500	0.0000	0.7500	0.0000
4	2	4	0.6667	0.6667	0.0000	0.6667	0.0000
4	2	5	0.6250	0.5000	0.1250	0.6250	0.0000
4	2	6	0.6000	0.4286	0.1714	0.6000	0.0000
4	2	7	0.5833	0.3122	0.2712	0.5000	0.0833
4	2	8	0.5714	0.2851	0.2863	0.4374	0.1340
4	2	9	0.5625	0.2544	0.3081	0.4363	0.1262
4	2	10	0.5556	0.2606	0.2950	0.4375	0.1181
5	2	3	0.9000	0.7500	0.1500	0.7500	0.1500
5	2	4	0.8000	0.7500	0.0500	0.7500	0.0500
5	2	5	0.7500	0.6700	0.0800	0.7497	0.0003
5	2	6	0.7200	0.6014	0.1186	0.6637	0.0563
5	2	7	0.7000	0.5596	0.1404	0.6667	0.0333
5	2	8	0.6857	0.4991	0.1867	0.6060	0.0798
5	2	9	0.6750	0.4590	0.2160	0.5821	0.0929
5	2	10	0.6667	0.4615	0.2052	0.5196	0.1470
6	2	4	0.8889	0.8889	0.0000	0.8889	0.0000
6	2	5	0.8333	0.7999	0.0335	0.8333	0.0000
6	2	6	0.8000	0.8000	0.0000	0.8000	0.0000
6	2	7	0.7778	0.7500	0.0278	0.7778	0.0000
6	2	8	0.7619	0.7191	0.0428	0.7597	0.0022
6	2	9	0.7500	0.6399	0.1101	0.7500	0.0000
6	2	10	0.7407	0.6344	0.1064	0.7407	0.0000
6	2	11	0.7333	0.6376	0.0958	0.7333	0.0000
6	2	12	0.7273	0.6214	0.1059	0.7273	0.0000

continued...

. . . continued

			Squared Packing Radii				
d	K	N	Rankin	$\mathbb{R}$	Difference	$\mathbb{C}$	Difference
6	3	3	0.7500	0.7500	0.0000	0.7500	0.0000
6	3	4	0.6667	0.5000	0.1667	0.6667	0.0000
6	3	5	0.6250	0.4618	0.1632	0.4999	0.1251
6	3	6	0.6000	0.4238	0.1762	0.5000	0.1000
6	3	7	0.5833	0.3590	0.2244	0.4408	0.1426
6	3	8	0.5714	—	—	0.4413	0.1301
6	3	9	0.5625	—	—	0.3258	0.2367

Figure 7.3: Packing in Grassmannians with spectral distance: The red line indicates the best real packings obtained by the author, while the blue line indicates the best complex packings obtained. The Rankin Bound (7.14) is depicted in gray. The vertical red line marks the largest number of real subspaces for which the Rankin Bound is attainable.



continued...

. . . continued


Packing in G(2, F^5) with Spectral Distance
Squared Packing Radius
0.90
0.80
0.70
0.60
0.50
0.40
3
4
5
6
7
8
9
10
Number of Planes
Rankin Bound
Complex
Real

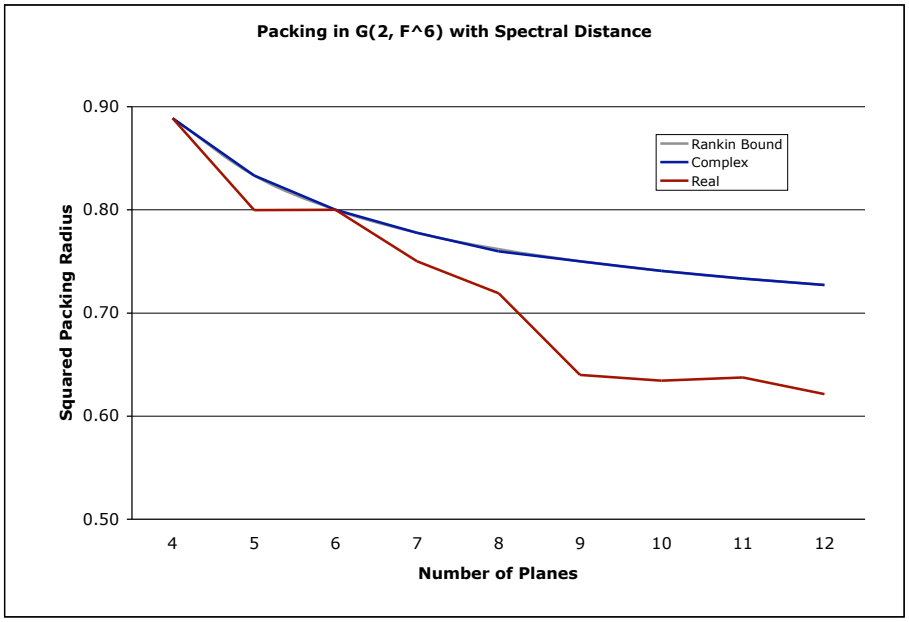
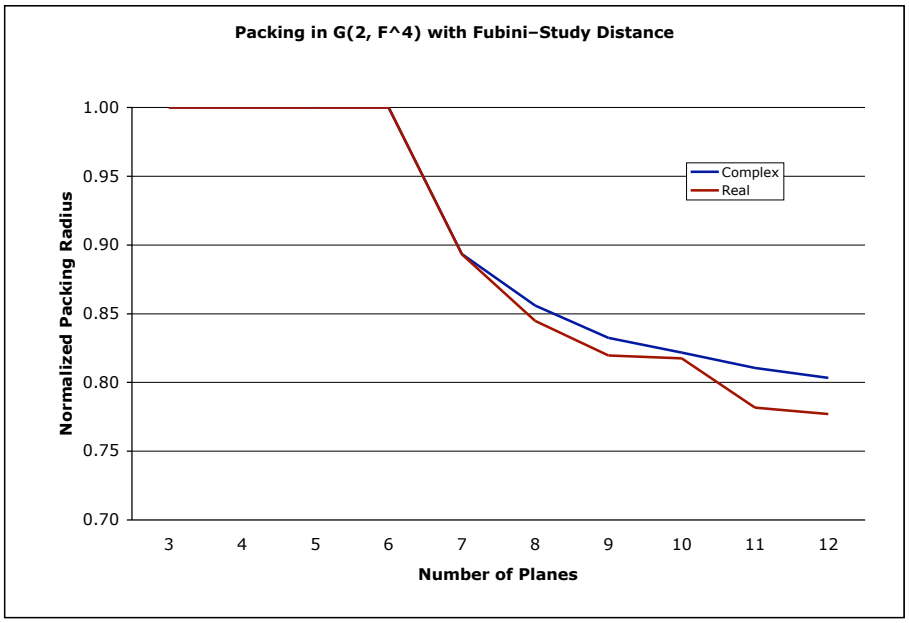

Packing in G(2, F^6) with Spectral Distance
Squared Packing Radius
0.90
0.80
0.70
0.60
0.50
4
5
6
7
8
9
10
11
12
Number of Planes
Rankin Bound
Complex
Real

Table 7.7: Packing in Grassmannians with Fubini–Study distance: Our best real packings ($\mathbb{F} = \mathbb{R}$) compared with our best complex packings ($\mathbb{F} = \mathbb{C}$) in the space $\mathbb{G}(K, \mathbb{F}^d)$. The packing radius of a configuration is the Fubini–Study distance (7.9) between the closest pair of subspaces. Note that we have scaled the distance by $2/\pi$ so that it ranges between zero and one.

			Squared Packing Radii	
d	K	N	$\mathbb{R}$	$\mathbb{C}$
2	4	3	1.0000	1.0000
2	4	4	1.0000	1.0000
2	4	5	1.0000	1.0000
2	4	6	1.0000	1.0000
2	4	7	0.8933	0.8933
2	4	8	0.8447	0.8559
2	4	9	0.8196	0.8325
2	4	10	0.8176	0.8216
2	4	11	0.7818	0.8105
2	4	12	0.7770	0.8033
2	5	3	1.0000	1.0000
2	5	4	1.0000	1.0000
2	5	5	1.0000	1.0000
2	5	6	0.9999	1.0000
2	5	7	1.0000	0.9999
2	5	8	1.0000	0.9999
2	5	9	1.0000	1.0000
2	5	10	0.9998	1.0000
2	5	11	0.9359	0.9349
2	5	12	0.9027	0.9022

Figure 7.4: Packing in Grassmannians with Fubini–Study distance: The red line indicates the best real packings obtained by the author, while the blue line indicates the best complex packings obtained.



Chapter 8

Clustering and Sparse Matrix Approximation

Given a collection of data, *hard clustering* is the problem of partitioning the data into a relatively small number of disjoint subsets (called *clusters*) so that the data within each cluster are as similar as possible and the data in different clusters are as distinct as possible. The definitions of "similar" and "dissimilar" depend on the problem domain [34].

Clustering problems arise in applications throughout electrical engineering and the computer sciences. A typical example is to take a large collection of documents (such as web pages) and classify them by subject. It may be prohibitively expensive or impossible to sort the documents manually, which makes an automatic procedure essential [27]. Automatic clustering can also be applied when the structure of the data is not known in advance, and we seek insight into its geometry. This challenge occurs in computational biology when one searches for groups of genes that have related functions by examining gene expression patterns [26]. Clustering has dozens of other current and potential applications that are beyond the scope of this treatment.

There are striking conceptual and formal parallels between clustering problems and sparse approximation problems. Let us explain the connection intuitively. One way to view a clustering problem is to imagine that each cluster of data vectors is represented by a geometric structure—a point in the simplest case but potentially a much more complicated object such as a linear

subspace or a convex cone. These cluster structures correspond to atoms. Each data vector is assigned to the closest cluster, where closeness is measured using some kind of dissimilarity measure that depends on the application. In other words, each data vector is approximated from a *single* cluster structure (i.e., atom). This is a sparse approximation. The major conceptual difference between clustering and sparse approximation is that sparse approximation fixes the collection of atoms in advance, while a clustering problem tries to locate the clusters at the same time it is performing the approximation.

In this chapter, we will develop the perspective that clustering problems can be viewed as low-rank matrix approximation with sparsity constraints. Specifically, we argue that the goal of a clustering problem is to approximate a data matrix by the product of a (small) representative matrix and a coefficient matrix. Sparsity constraints on the coefficient matrix control the geometry of the clusters, while constraints on the representative matrix reflect *a priori* information about the data (such as nonnegativity). We choose the measure of approximation error based on how we measure the dissimilarity between data points. It turns out that this formulation encompasses many of the clustering problems that have appeared in the literature, and it suggests many interesting new problems. Moreover, it leads to a general algorithmic framework that can be used to approach all of these clustering problems.

Here is a brief outline of the chapter. The first section begins with the most classical clustering problem, which we recast as a matrix approximation problem. Then, we demonstrate that a fundamental algorithm for the classical problem admits a striking interpretation in the matrix formulation. Then, we demonstrate that slight variations on the matrix version of the classical problem lead immediately to two clustering problems that have appeared in

the recent literature. The algorithms that have been proposed for these two problems share the same interpretation.

In the next section, we present the canonical form of the matrix approximation problem. Then, we list some of the many possible constraints and show how they lead to different types of clustering. Afterward, we give some details about methods for measuring the approximation error, and we discuss how they affect some qualitative aspects of the clustering. The section concludes with a general algorithm that can be modified to approach all of the clustering problems.

Finally, we discuss how clustering problems from the literature fit into our framework. This exercise underscores the similarities among a large number of apparently distinct clustering formulations, and it emphasizes the strength of our viewpoint.

8.1 Motivating Examples

The idea that we might be able to unify apparently unrelated clustering problems grows out of some basic examples. First, we examine the classical clustering problem, which asks us to minimize the total squared Euclidean distance from each data vector to the nearest cluster. This problem can be recast as approximation of the data by a low-rank matrix with certain sparsity constraints. We show that the standard numerical method for approaching the clustering problem has a striking interpretation in this framework. Then we demonstrate that some less familiar clustering problems can also be viewed as low-rank matrix approximation with slightly different sparsity constraints. Moreover, the numerical approaches that have been proposed for these other clustering problems admit the same interpretation as the classical algorithm.

This observation strongly suggests that many more clustering problems fit within our framework and that the classical algorithm can be extended to address these cases.

8.1.1 The Classical Clustering Problem

Suppose that $\boldsymbol{s}_1, \dots, \boldsymbol{s}_K$ are data vectors in $\mathbb{R}^d$ that we wish to partition into N clusters. Formally, the goal is to determine N representative vectors $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_N$ from $\mathbb{R}^d$ that solve

$$\min_{\{\boldsymbol{\varphi}_n\}} \quad \sum\nolimits_{k=1}^{K} \min_{n} \ \|\boldsymbol{s}_k - \boldsymbol{\varphi}_n\|_2^2 \,. \tag{8.1}$$

In this problem, the n-th cluster is viewed as a point in space, namely $\boldsymbol{\varphi}_n$. We say that the vector $\boldsymbol{\varphi}_n$ *represents* the cluster and that the *structure* of each cluster is a point. It is not hard to check that the representative vector for each cluster must be the centroid of the data vectors that are assigned to that cluster. Standard references for the classical clustering problem include [37, 70, 47, 34].

In words, the problem (8.1) asks us to minimize the total squared Euclidean distance from each data vector to a nearest cluster representative. In the language of sparse approximation, the representatives are atoms, and we wish to approximate each data vector with a single atom—a very sparse representation. But now we must learn the atoms at the same time we are computing a sparse approximation of the data. The idea is that the approximation can succeed only if it identifies latent structure in the data.

The classical clustering problem (8.1) can be recast in matrix form. Suppose that $\mathbf{S}$ is a real $d \times K$ matrix whose columns are data vectors that we seek to partition into N clusters. We write $\boldsymbol{s}_k$ for the k-th column of the data

matrix. Our goal will be to produce a $d \times N$ representative matrix $\boldsymbol{\Phi}$ and an $N \times K$ coefficient matrix $\boldsymbol{C}$ whose product approximates the data matrix.

Observe that the k-th column of the product $\boldsymbol{\Phi}\,\boldsymbol{C}$ is the vector $\boldsymbol{\Phi}\,\boldsymbol{c}_k$. Since the classical clustering problem requires that each data vector be approximated by a single representative vector, each coefficient vector $\boldsymbol{c}_k$ must contain a unique nonzero entry equal to one. In other words, each column $\boldsymbol{c}_k$ of the coefficient matrix must be a canonical basis vector. In this chapter, we denote the n-th canonical basis vector by $\mathbf{e}_n$. Therefore, each column k of the coefficient matrix must satisfy the constraint $\boldsymbol{c}_k = \mathbf{e}_n$ for some n in the range $1, \ldots, N$. This is clearly a sparsity constraint on the coefficient matrix.

Since the squared Frobenius norm $\|\cdot\|_{\mathrm{F}}^2$ of a matrix is the sum of the squared Euclidean norms of columns, we may rewrite the classical clustering problem as

$$\min_{\boldsymbol{\Phi},\boldsymbol{C}} \quad \|\boldsymbol{S} - \boldsymbol{\Phi}\,\boldsymbol{C}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad \boldsymbol{c}_k = \mathbf{e}_n \quad \text{for some } n. \tag{8.2}$$

At a solution of (8.2), the k-th column of the coefficient matrix equals $\mathbf{e}_{n(k)}$, where the k-th data vector is nearer to the $n(k)$-th representative vector than to any other. Meanwhile, each representative must equal the centroid of the data vectors that are nearest to it.

8.1.2 The k-means Algorithm

The classical algorithm for solving the clustering problem (8.2) is called k-means. Using our notation, however, the name N-means would be more appropriate.

Algorithm 8.1 (k-means).

INPUT:

- A $d \times K$ matrix S, whose columns are real data vectors
- A number N of clusters

OUTPUT:

- Returns a $d \times N$ representative matrix $\mathsf{\Phi}$ whose columns are the representative vectors
- Returns an $N \times K$ coefficient matrix C whose columns each contain a single unit entry

PROCEDURE:

1. Initialize $\mathsf{\Phi}$. One method selects the columns of $\mathsf{\Phi}$ at random from the columns of S without repetition.
2. For each k, determine $n(k)$ so that the k-th data vector is closer to the $n(k)$-th representative than to any other. Ties are broken in an arbitrary manner. Set the k-th column of the coefficient matrix to $\mathbf{e}_{n(k)}$. In other words, the k-th data vector is assigned to the $n(k)$-th cluster.
3. For each n, set the n-th representative equal to the centroid of the data vectors that have been assigned to the n-th cluster.
4. Repeat Steps 2–4 until the objective function (8.1) does not decrease from one iteration to the next. This termination condition is guaranteed to be met in a finite number of steps because the algorithm monotonically decreases the objective function, which is bounded below by zero

Algorithm 8.1 is actually a heuristic method, and it will not generally produce an optimal solution to (8.2). Indeed, the classical clustering problem is known to be NP-hard [46]. Nevertheless, the procedure decreases the objective function in (8.1) monotonically, so it converges to a local minimum of the objective function.

In our matrix formulation, the k-means algorithm has an especially striking interpretation as an *alternating projection* method. Step 2 determines C by solving (8.2) with $\mathsf{\Phi}$ held fixed. Likewise, Step 3 determines $\mathsf{\Phi}$ by solving (8.2) with C held fixed.

8.1.3 Spherical Clustering

By adding constraints on the representative vectors in the matrix formulation (8.2), we obtain another clustering problem that has appeared in the literature. Suppose that the columns of the data matrix S all have unit norm. Then it may be desirable to require that the representative vectors also have unit norm. This restriction leads to the problem

$$\begin{aligned} \min_{\mathsf{\Phi},\mathsf{C}} \quad \|\mathsf{S} - \mathsf{\Phi}\,\mathsf{C}\|_2^2 \qquad \text{subject to} \qquad & \boldsymbol{c}_k = \mathbf{e}_n \quad \text{for some } n \\ & \|\boldsymbol{\varphi}_n\|_2 = 1. \end{aligned} \tag{8.3}$$

Each cluster has the structure of a point on the surface of the sphere, and so we refer to the problem as *spherical clustering*.

For illustrative purposes, let us reduce (8.3) to a more traditional form. Together, the objective function and the coefficient constraint yield the objective function

$$\sum\nolimits_{k=1}^{K} \min_n \|\boldsymbol{s}_k - \boldsymbol{\varphi}_n\|_{\mathrm{F}}^2 .$$

Expand the norms, and then apply the hypothesis that $\|\boldsymbol{s}_k\|_2 = 1$ and the

constraint that $\|\boldsymbol{\varphi}_n\|_2 = 1$. It follows that (8.3) is equivalent to

$$\min_{\{\boldsymbol{\varphi}_n\}} \quad \sum\nolimits_{k=1}^{K} \max_n \langle \boldsymbol{s}_k, \boldsymbol{\varphi}_n \rangle \qquad \text{subject to} \qquad \|\boldsymbol{\varphi}_n\|_2 = 1.$$

Therefore, the clustering problem (8.3) measures the similarity of two vectors as the cosine of the angle between them.

Dhillon and Modha have proposed a heuristic procedure called *spherical k-means* for solving (8.3). This method first assigns data vectors to the cluster representative nearest with respect to cosine similarity. Second, it replaces each cluster representative with the mean of the data vectors that have been assigned to that cluster, and it re-scales each representative to have unit norm. It repeats these two steps until the cluster assignments stabilize [27].

It turns out that spherical k-means is an alternating projection technique for solving (8.3). The first step minimizes (8.3) with respect to coefficient matrices that satisfy the coefficient constraint. The second step minimizes (8.3) with respect to representative matrices that have unit-norm columns.

8.1.4 Diametrical Clustering

By altering the coefficient constraints in (8.3), we reach another problem from the literature. Assume that the data vectors have unit Euclidean norm, and we wish to partition the data vectors into clusters that contain both correlated and anti-correlated data. This problem arises in bio-informatics, when genes are sorted into functional groups based on gene expression data. A strong positive correlation between two genes indicates that they are expressed together, while strong negative correlation indicates that one gene may be inhibiting the other. Therefore, a collection of genes whose pairwise

correlations are strongly positive or strongly negative may form a functional group [26].

The diametrical clustering problem introduces a new constraint on the coefficient matrix:

$$\min_{\boldsymbol{\Phi},\boldsymbol{C}} \quad \|\boldsymbol{S} - \boldsymbol{\Phi}\,\boldsymbol{C}\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad \begin{aligned} \boldsymbol{c}_k &= \pm\mathbf{e}_n \quad \text{for some } n \\ \|\boldsymbol{\varphi}_n\|_2 &= 1. \end{aligned} \tag{8.4}$$

It follows that we may interpret the n-th cluster is structured as a pair of antipodal points on the sphere, namely $\pm\boldsymbol{\varphi}_n$. Using the same procedure as before, we see that the similarity between two (unit) vectors is the cosine of the acute angle between the two vectors. This similarity measure may be computed as the absolute value of the inner product between the two unit vectors.

The alternating projection algorithm for (8.4) is a little more involved than before. As usual, each data vector is assigned to the representative nearest with respect to the acute cosine similarity measure. It can be shown that the representative vector of each cluster must be a dominant left singular vector of the matrix formed from the data vectors that have been assigned to that cluster. This is precisely the algorithm that was proposed in [26].

8.2 Constrained Low-Rank Matrix Approximation

The last section demonstrates that several apparently unrelated clustering problems share an enormous amount of structure. But these examples only hint at the possibilities. We argue that many of the clustering problems that have appeared in the recent literature find their most natural expression in the language of constrained low-rank matrix approximation.

Let S be a $d \times K$ matrix whose columns are data vectors that we wish to partition into N clusters. Therefore, we seek a $d \times N$ matrix $\mathsf{\Phi}$ whose columns are cluster representatives and an $N \times K$ matrix C whose entries describe the assignment of data vectors into clusters. The product $\mathsf{\Phi C}$ should be interpreted as a low-rank approximation of the data matrix S. We will measure the divergence between the approximation and the data with a general dissimilarity measure $\operatorname{dist}(\,\cdot\,;\,\cdot\,)$. Our goal will be to solve

$$\min_{\mathsf{\Phi},\mathsf{C}} \operatorname{dist}(\mathsf{\Phi C}; \mathsf{S}) \qquad \text{subject to} \qquad \begin{matrix} \mathsf{C} \text{ satisfies constraint (C), and} \\ \mathsf{\Phi} \text{ satisfies constraint (R).} \end{matrix} \tag{8.5}$$

The constraint (R) on the representative matrix reflects *a priori* knowledge about the data, while the constraint (C) on the coefficient matrix determines the gross geometry of the clusters.

In the next two subsections, we detail some basic constraints on the representative matrix and the coefficient matrix. Afterward, we explain the modifications that are necessary to obtain more general cluster geometries. Then, we describe some of the dissimilarity measures that have appeared in the literature. The final section of the chapter describes how specific problems that have appeared in the literature fit into our framework.

8.2.1 Representative Constraints

In many problems, we possess some information about the provenance of the data vectors. For example, the data may be normalized, or it may consist of nonnegative numbers. This type of *a priori* knowledge should be encoded as constraints on the representative matrix so that the qualities of the representative vectors match the qualities of the data. There are also technical reasons that one may wish to constrain the representatives. If the coefficients

are not normalized, it may be necessary to normalize the representatives to prevent scaling problems.

We list a few constraints on the representative matrix. Although it is not necessary, each column of $\boldsymbol{\Phi}$ will typically satisfy the same constraint. Therefore, we only indicate how a single column is constrained.

R1. $\boldsymbol{\varphi}_n$ unrestricted. When the data vectors are arbitrary points in space, we may want the representative vectors to roam freely. This situation occurs, for example, in the classical clustering problem.

R2. $\|\boldsymbol{\varphi}_n\|_2 = 1$. Here, we force each representative vector to lie on the Euclidean unit sphere. This constraint commonly arises when the data vectors are normalized, and the representatives must duplicate this normalization. Note that representative may be normalized even when the data vectors are not.

R3. $\boldsymbol{\varphi}_n \geq \mathbf{0}$. If the data are nonnegative, one may require that the representative vectors share this quality. In many applications, negative numbers lack a valid interpretation. For example, there is no such thing as a negative amount of rainfall.

R4. $\boldsymbol{\varphi}_n \geq \mathbf{0}$ and $\mathbf{e}^T\boldsymbol{\varphi}_n = 1$. This is referred to as a *stochastic* constraint, and it permits us to interpret the representative vectors as probability distributions. This condition can be useful when the data vectors are viewed as mixtures of probability distributions.

R5. $\boldsymbol{\varphi}_n$ drawn from a closed, convex set. This constraint generalizes (R3) and (R4).

Of course, many other representative constraints are possible.

8.2.2 Coefficient Constraints

The gross geometry of the clusters depends essentially on how we control the coefficient matrix. (The precise shape that the clusters prefer to take depends strongly on the dissimilarity measure.)

Let us begin with the case of hard clustering, where each data vector is assigned to a single cluster. In the matrix formulation, the k-th column $\boldsymbol{s}_k$ of the data matrix is approximated by $\boldsymbol{\Phi}\,\boldsymbol{c}_k$. The hard clustering constraint allows only one column of the representative matrix to participate in the approximation of $\boldsymbol{s}_k$, which implies that each column $\boldsymbol{c}_k$ of the coefficient matrix may contain only a single nonzero entry.

C1. $\boldsymbol{c}_k = \mathbf{e}_n$ for some n. The basic hard clustering problem requires that each column of the coefficient matrix contain exactly one unit entry. In this case, the n-th cluster is structure as a single point in space, namely $\boldsymbol{\varphi}_n$.

C2. $\boldsymbol{c}_k = \pm\mathbf{e}_n$. If each column of the coefficient matrix contains a single nonzero entry that equals ± 1, then the n-th cluster is structured as a pair of antipodal points, $\pm\boldsymbol{\varphi}_n$. This type of clustering collates data with strong positive or strong negative correlation, but it remains sensitive to scale.

C3. $\boldsymbol{c}_k = \alpha_k\,\mathbf{e}_n$ for $\alpha_k > 0$. Here, each column of the coefficient matrix contains exactly one nonzero entry, which is positive. The n-th cluster has the structure of a ray emanating from the origin, namely the cone generated by the representative vector $\boldsymbol{\varphi}_n$. This might be called a *directional clustering* constraint. In this case, one may wish to normalize the columns of the representative matrix to prevent scaling problems.

C4. $\boldsymbol{c}_k = \alpha_k \, \mathbf{e}_n$ for $\alpha_k \neq 0$. Each coefficient vector contains a single nonzero entry. So the n-th cluster is structured as a line through the origin, namely the subspace spanned by $\boldsymbol{\varphi}_n$. This might be called a *projective clustering* constraint. Once again, the representative vectors may require normalization.

Note that each data point is assigned to the cluster *structure* closest to it. For example, if we impose the condition (C4), then each representative vector determines a line through the origin. Each data point is assigned to a cluster by determining which line contains the point closest to that data point. To give a basic idea of how the first four coefficient constraints affect the final clustering, we offer an illustration of how the same data points can be grouped quite differently. See Figure 8.1

Traditionally, a *soft clustering problem* determines a set of cluster representatives and the probability that each data vector belongs to each cluster. In our framework, we may frame a soft clustering problem by constraining each coefficient vector to have nonnegative entries that sum to one. That is, each coefficient vector is stochastic. More generally, we might relax the requirement that the coefficient vectors contain exactly one nonzero entry. In this setting, it is a little more complicated to talk about the structure of a cluster, since the clusters are blended together.

C5. $\boldsymbol{c}_k$ unrestricted. Then the data vectors are approximated from the linear span of all the representatives. This situation occurs in principal component analysis [62].

C6. $\boldsymbol{c}_k \geq \mathbf{0}$. If each coefficient vector is nonnegative, then each data vector

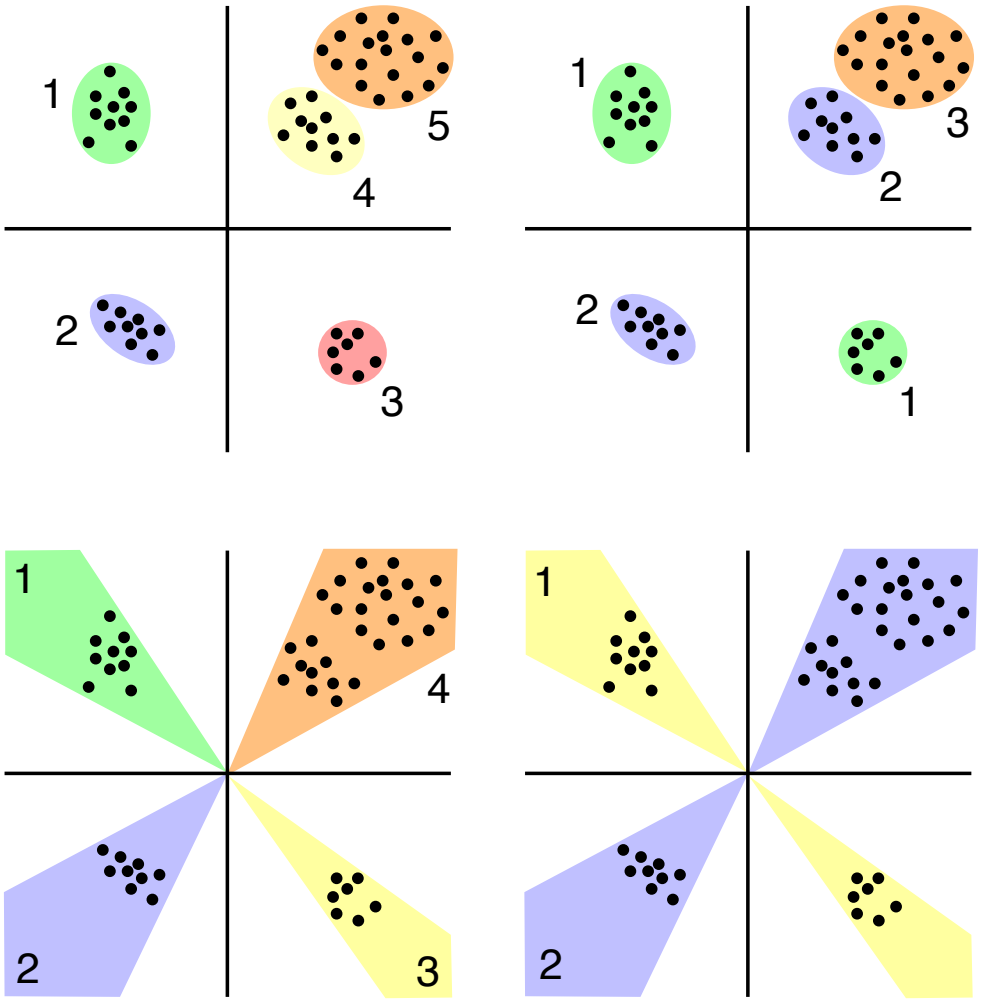


Figure 8.1: Effects of Coefficient Constraints. Four plausible clusterings of the same data points using different coefficient constraints. Upper left: (C1) yields five different clusters. Upper right: (C2) gives three clusters. Lower left: (C3) leads to four clusters. Lower right: (C4) yields only two clusters.

is approximated by a point from the convex cone generated by all the representative vectors.

C7. $\boldsymbol{c}_k \geq \mathbf{0}$ and $\mathbf{e}^T\boldsymbol{c}_k = 1$. Then each coefficient vector is stochastic, and each data vector is approximated by a point in the convex hull of all the representative vectors. If the representatives are stochastic, then this constraint approximates the data vectors with mixtures of probability distributions.

Finally, we note that it is also possible to place other types of sparsity constraints on the coefficient matrix.

C8. $\|\boldsymbol{c}_k\|_0 = m$. Then each data vector is approximated as a linear combination of m representative vectors, a true sparsity condition. As we have seen, this constraint is computationally difficult to enforce in general. It can also be combined with (C5), (C6), or (C7).

8.2.3 Higher-Dimensional Clusters

To obtain more complicated cluster geometries, we may need to use several vectors to represent a cluster. Although this step increases the conceptual complexity slightly, more general clusters may model the data more accurately. For example, in an application to breast cancer data, Bradley and Mangasarian found that it was possible to identify survival outcomes more accurately by structuring each clusters as a hyperplane instead of a point [6]. For simplicity, we only consider hard clustering problems, so each data vector is assigned to a single cluster.

First, let us explain how to modify the setup. Now, each cluster representative will have M vector parameters, so each of the N cluster representatives

is a $d \times M$ matrix $\boldsymbol{\Phi}_n$, whose columns we denote by $\boldsymbol{\varphi}_{mn}$. We may collect the representative matrices into a block matrix with dimensions $d \times MN$.

$$\boldsymbol{\Phi} \quad \stackrel{\text{def}}{=} \quad \begin{bmatrix} \boldsymbol{\Phi}_1 & \dots & \boldsymbol{\Phi}_N \end{bmatrix}.$$

The coefficient matrix C has dimension $MN \times K$ with entries c_{mnk}, and so the product $\boldsymbol{\Phi}\,\mathsf{C}$ is a $d \times K$ matrix whose k-th column is

$$\sum\nolimits_{n=1}^{N} \sum\nolimits_{m=1}^{M} c_{mnk}\, \boldsymbol{\varphi}_{mn}.$$

The hard clustering constraint requires that for each index k, the coefficients $\{c_{mnk}\}$ be nonzero for a single value $n = n(k)$. Therefore, the double sum collapses to

$$\sum\nolimits_{m=1}^{M} c_{m,n(k),k}\, \boldsymbol{\varphi}_{m,n(k)}.$$

As advertised, each data vector is approximated by a linear combination of the columns of one representative matrix.

The following constraints on the coefficients lead us to some fundamental new cluster geometries.

M1. $c_{m,n(k),k}$ unconstrained. Then the n-th cluster has the structure of a subspace, namely the column span of the matrix $\boldsymbol{\Phi}_n$. This might be called *Grassmannian clustering*.

M2. $c_{1,n(k),k} = 1$ and $c_{m,n(k),k}$ unconstrained for $m \geq 2$. Now each cluster representative has the structure of an $(M-1)$-dimensional affine space. The first column of each representative matrix defines a point in the affine space, and its remaining columns are vectors that span a translate of the affine space.

M3. $c_{m,n(k),k} \geq 0$. In this case, each cluster has the structure of a convex cone, namely the conical hull of the columns of its representative matrix.

M4. $c_{m,n(k),k} \geq 0$ and $\sum_m c_{m,n(k),k} = 1$. It follows that the n-th cluster has the structure of a convex set, namely the convex hull of the columns of $\boldsymbol{\Phi}_n$.

M5. $c_{m,n(k),k}$ an integer. In this case, the n-th cluster has the structure of a lattice, namely the lattice (i.e., $\mathbb{Z}$-module) generated by the columns of $\boldsymbol{\Phi}_n$.

As before, we may impose additional requirements on the representative matrices to put them in sympathy with the data or to prevent scaling problems. It may also be necessary to ensure that the columns of each representative matrix do not spread out too much. For example, if the clusters are supposed to be represented by a convex set, it may be valuable to prevent the convex set from growing to encompass an entire subspace.

8.2.4 Dissimilarity Measures

The other ingredient in a clustering problem is the measure of dissimilarity between the data matrix and its approximant. This measure affects the finer geometry of the cluster by determining which data vectors are close to the cluster structure and which are not. The distortion or dissimilarity measure depends entirely on the problem domain, and what works for one application may be useless for another. Moreover, the variety of dissimilarity measures is dizzying. Any method for computing the distance between two matrices will, in principle, suffice. For simplicity, we only discuss a subclass of dissimilarity measures that treat each column of the matrix independently.

First, let us consider some measures that derive from matrix norms. These measures are based on the vector norms that are used to measure the distance between a data vector and its approximant. Of course, any vector norm is possible, but we only mention the most common.

1. Far and away, the most familiar way to measure the error between each data vector and its approximation is the Euclidean norm:

$$\|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\|_2 \,.$$

   The Euclidean norm is small when most components of the approximation have a reasonably small error.

2. Another common choice is the ℓ_1 norm:

$$\|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\|_1 \,.$$

   The ℓ_1 norm strongly prefers an approximating vector whose components commit a few large error and many tiny errors. It is frequently used in statistical applications because it is robust to outliers in the individual measurements. If the data vectors are viewed as probability distributions, then the ℓ_1 norm returns the variational distance between each datum and its approximant.

3. A third choice is the ℓ_∞ norm:

$$\|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\|_\infty \,.$$

   The ℓ_∞ norm seeks an approximating vector that contains tiny errors of similar magnitude in every single component. It is extremely sensitive to outliers because it prefers a uniform approximation.

There are many different ways to combine the column errors to obtain a figure of merit for the entire clustering.

1. One may sum up all the column errors:

$$\sum_{k=1}^{K} \|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\| \,.$$

   This type merit function is the most robust because it allows a few data vectors to commit large errors. If some data vectors are entirely suspect, then this measure may be the most sensible.

2. One might also try to minimize the maximum column error:

$$\max_k \ \|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\| \,.$$

   In this case, the merit function will seek an approximation where every data vector is approximated with an identical, small error.

3. For p ranging between one and infinity, the merit function

$$\Big[\sum_{k=1}^{K} \|\boldsymbol{s}_k - \boldsymbol{\Phi}\,\boldsymbol{c}_k\|^p\Big]^{1/p} .$$

   will interpolate between the behavior of the first two merit functions. The special case $p = 2$ is most common, and it balances the two extremes evenly.

Each combination from the last two lists corresponds to some matrix norm, although some of these norms are not sub-multiplicative.

There are also important distance measures that do not derive from norms. It is well known that the squared Euclidean norm is closely connected

with the multivariate normal probability distribution. For other exponential families of probability distributions, *Bregman divergences* play an analogous role [2].

Here is a brief introduction to Bregman divergences. Suppose that f is a strictly convex, differentiable function defined on the signal space. The *Bregman divergence* of the vector $\boldsymbol{x}$ from the vector $\boldsymbol{s}$ is calculated as

$$D_f(\boldsymbol{x}; \boldsymbol{s}) \quad \stackrel{\text{def}}{=} \quad f(\boldsymbol{x}) - f(\boldsymbol{s}) - \langle \nabla f(\boldsymbol{s}), \boldsymbol{x} - \boldsymbol{s} \rangle \,.$$

The semicolon in the notation warns us that Bregman divergences are almost never symmetric. Nevertheless, Bregman divergences are positive definite; they are strictly convex in their first argument; and they are continuous in both arguments. The squared Euclidean distance is the primary example of a Bregman divergence. The *generalized Kullback–Leibler divergence* (also known as *relative entropy*) is another important case:

$$D_f(\boldsymbol{x}; \boldsymbol{s}) \quad = \quad \sum\nolimits_{j=1}^{d} \left[x_j \log \frac{x_j}{s_j} - x_j + s_j \right].$$

The KL divergence is only defined for vectors with nonnegative components. It derives from the negative Shannon entropy, $f(\boldsymbol{s}) = \sum_{j=1}^{d} (s_j \log s_j - s_j)$. A third example falls from the Burg entropy $f(\boldsymbol{s}) = -\sum_{j=1}^{d} \log s_j$, which is defined only for strictly positive vectors. It yields the Itakura–Saito divergence:

$$D_f(\boldsymbol{x}; \boldsymbol{s}) \quad = \quad \sum\nolimits_{j=1}^{d} \left[\frac{x_j}{s_j} - \log \frac{x_j}{s_j} - 1 \right].$$

The sum of Bregman divergences between corresponding columns of two matrices always yields a Bregman divergence on matrices.

Although Bregman divergences may seem exotic, they are better suited for approximation problems than many norms. Suppose that C is a closed,

convex subset of the signal space. Given a signal $\boldsymbol{s}$, the solution $\boldsymbol{p}$ of the minimization problem

$$\min_{\boldsymbol{x} \in \mathbb{C}^d} \; D_f(\boldsymbol{x}; \boldsymbol{s}) \qquad \text{subject to} \qquad \boldsymbol{x} \in C$$

is called the *Bregman projection* of $\boldsymbol{s}$ onto C. Bregman projections have a striking variational characterization,

$$D_f(\boldsymbol{x}; \boldsymbol{s}) \quad \geq \quad D_f(\boldsymbol{x}; \boldsymbol{p}) + D_f(\boldsymbol{p}; \boldsymbol{s}) \qquad \text{for all } \boldsymbol{x} \in C. \tag{8.6}$$

This formula is analogous with Kolmogorov's characterization of the orthogonal projection onto a convex set. Moreover, if C is affine, then the inequality in (8.6) becomes an equality, and we obtain an analog of the Pythagorean Theorem. We have glossed over some important technicalities in this paragraph. For details, refer to [4, 108].

8.2.5 Generalized k-means

Finally, we sketch a numerical approach to constrained low-rank matrix approximation problems that generalizes the k-means algorithm. Note that this computational problem will typically be NP-hard on account of the interaction between the representative matrix and the coefficient matrix, and so we cannot expect to develop an efficient algorithm that produces a globally optimal solution to (8.5).

Algorithm 8.2 (Generalized k-means).

INPUT:

- A $d \times K$ matrix S, whose columns are real data vectors
- A number N of clusters

OUTPUT:

- Returns a $d \times N$ representative matrix $\mathbf{\Phi}$ whose columns satisfy (R)
- Returns an $N \times K$ coefficient matrix $\mathbf{C}$ whose columns satisfy (C)

PROCEDURE:

1. Initialize $\mathbf{\Phi}$. One method selects the columns of $\mathbf{\Phi}$ at random from the columns of $\mathbf{S}$ without repetition.
2. Solve the optimization problem

   $$\min_{\mathbf{C}} \ \text{dist}(\mathbf{\Phi}\,\mathbf{C}; \mathbf{S}) \qquad \text{subject to} \qquad \mathbf{C} \text{ satisfies constraint (C)}$$

   holding the representative matrix $\mathbf{\Phi}$ fixed.
3. Solve the optimization problem

   $$\min_{\mathbf{\Phi}} \ \text{dist}(\mathbf{\Phi}\,\mathbf{C}; \mathbf{S}) \qquad \text{subject to} \qquad \mathbf{\Phi} \text{ satisfies constraint (R)}$$

   holding the coefficient matrix $\mathbf{C}$ fixed.
4. Repeat Steps 2–4 until the objective function $\text{dist}(\mathbf{\Phi}\,\mathbf{C}; \mathbf{S})$ fails to decrease significantly from one iteration to the next. This termination condition is guaranteed to be met in a finite number of steps because the objective function is nonincreasing and it is bounded below by zero.

This is an example of an alternating projection algorithm, and it has a lot in common with the algorithms of Chapter 7.

In some cases, the optimization problems in Steps 2 and 3 admit explicit closed form solutions. Consider, for example, the minimizations that arise in

the matrix version of the classical clustering problem (8.1), which Algorithm 8.1 solves explicitly. As we saw, spherical clustering and diametrical clustering also yield straightforward solutions. Other cases may require the application of mathematical programming software. When the squared Euclidean norm in (8.1) is replaced by the (unsquared) Euclidean norm, Step 3 of the algorithm involves a challenging minimization known as Weber's problem [17, 80]. Nevertheless, if the dissimilarity measure is convex in its first variable—as are those we have discussed—the minimization problems in Steps 2 and 3 can theoretically be dispatched in polynomial time by standard convex programming software [5].

The major advantage of the Generalized k-means Algorithm is its simplicity and wide applicability. It offers an immediate method for approaching all the clustering problems that fall in our framework. Therefore, one may imagine clustering a novel data set using many different geometries to determine which ones provide the best separation.

8.3 Relation with Previous Work

It turns out that most of the partitional clustering problems in the literature fall within our framework. That is, they can be expressed in the form of (8.5). Let us take a quick tour of the primary examples.

First, observe that if Φ and C are unrestricted, then we are seeking the best rank-N approximation of S in Frobenius norm. It is well-known that the solution of this optimization problem is given by the truncated singular value decomposition (TSVD) of S, which can be computed efficiently with standard algorithms [51]. It is interesting to note that Algorithm 8.2 need not converge to the TSVD. Indeed, it can be shown that the fixed points of the

alternating projection include every pair (Φ, C) for which the columns of Φ span an invariant subspace of SS^* and the rows of C span an invariant subspace of S^*S. (Of course, this failure is highly unlikely if the initial representatives are chosen at random.)

As we have stated before, the standard hard clustering problem places constraints (R1) and (C1). The soft clustering problem involves the constraints (R1) and (C7). Note, however, that soft clustering is usually treated as a statistical estimation problem, which requires assumptions about probability distributions.

Passing quickly over familiar territory, we note that the spherical clustering problem studied by Dhillon and Modha [27] falls from the constraints (R2) and (C3). The diametrical clustering problem [26] uses (R2) and (C4).

Lee and Seung have proposed two types of clustering, *conic coding* and *convex coding*, that also fall into our framework [64]. Conic coding approximates the data vectors using a conic combination of representative vectors; it imposes constraints (R1) and (C6). Convex coding approximates the data as a convex combination of representative vectors; it imposes constraints (R1) and (C7). In their formulation, Lee and Seung placed the nonnegativity constraint (R3) on the representatives because they were working with nonnegative data. Convex and conic coding are intermediate between simple hard clustering and unconstrained low-rank matrix approximation (i.e., TSVD).

The constraints (R3) and (C6) lead to an important problem called *nonnegative matrix approximation* (NNMA). This problem requests an approximation of a nonnegative matrix as a low-rank product of two nonnegative matrices. NNMA has also been called *positive matrix factorization* and *nonnegative matrix factorization*, which are both misnomers. NNMA originally

arose in chemometrics as a method for producing a nonnegative factor model of nonnegative data [81]. Later, the neural information processing community began to study NNMA as a method for learning efficient representations of nonnegative data [64, 65, 68].

In an unpublished abstract, Srebro and Jaakkola pose the problem of *sparse matrix approximation*, which involves the constraints (R1) and (C8). They suggest it as a method for mining gene expression data [97], although they do not appear to have pursued this project.

We note that (R4) and (C7) require that the representatives and the coefficient vectors be stochastic. If the representatives are interpreted as probability distributions, then the corresponding problem requests a mixture model that can approximate all the data vectors. When working with probability distributions, it is rather more appropriate to measure approximation error with the variational distance or the Kullback–Leibler divergence than to use the squared Euclidean distance.

The idea of using higher-dimensional clustering prototypes first appears in Bradley and Mangasarian [6]. Each cluster has the structure of a hyperplane, which amounts to constraint (M2) with each representative matrix containing $M = d$ columns, where d is the dimension of the ambient space. (Their work takes advantage of the duality between vectors and hyperplanes to reduce the complexity of the problem.) Subsequently, Tseng showed how to generalize their work to lower-dimensional affine clusters, i.e., (M2) where each representative matrix contains $M < d$ columns [112].

Most research on clustering uses the squared Euclidean distance to measure the dissimilarity between vectors, but there are some exceptions. For example, Bradley, Mangasarian, and Street consider hard clustering—(R1)

and (C1)—with an ℓ_1 dissimilarity measure [7]. Lee and Seung have studied nonnegative matrix approximation—(R3) and (C6)—with respect to the generalized Kullback–Leibler divergence [68]. A recent paper by Banerjee et al. addresses clustering with respect to general Bregman divergences [2], which imposes constraints (R5) and (C1). More work on non-Euclidean dissimilarity measures would be valuable.

Our treatment of clustering as a constrained matrix approximation problem seems to be novel, and yet our perspective unifies a tremendous amount of previous literature. Table 8.3 summarizes how our framework encompasses the previous research on clustering with respect to the squared Euclidean distance. The unfilled entries in the table represent problems that have not to our knowledge been studied; some of these problems may present interesting avenues for research, while others do not admit an immediate interpretation. We also believe that this viewpoint will lead to interesting new clustering problems and to new algorithms for solving them. In the future, we hope to test these ideas on data from real applications.

	(R1)	(R2)	(R3)	(R4)
(C1)	Classical hard clustering [47]			
(C3)		Spherical clustering [27]		
(C4)		Diametrical clustering [26]		
(C5)	Truncated SVD [51]			
(C6)	Conic coding [64]		Nonnegative matrix approximation [81, 65]	
(C7)	Convex coding [64]			Probability mixtures
(C8)	Sparse matrix approximation [97]			
(M2)	k-flat clustering [6, 112]			

Table 8.1: Clustering Research
Previous research on clustering with respect to the squared Euclidean norm, as it fits into the framework of constrained low-rank matrix approximation. The unfilled entries represent problems that have not been studied.

Bibliography

[1] D. Agrawal, T. J. Richardson, and R. L. Urbanke. Multiple-antenna signal constellations for fading channels. *IEEE Trans. Inform. Theory*, 47(6):2618–2626, Sept. 2001.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 234–245, April 2004.

[3] A. Barg and D. Yu. Nogin. Bounds on packings of spheres in the Grassmannian manifold. *IEEE Trans. Inform. Theory*, 48(9):2450–2454, Sept. 2002.

[4] H. H. Bauschke and J. M. Borwein. Legendre functions and the method of random Bregman projections. *J. Convex Anal.*, 4(1):27–67, 1997.

[5] S. Boyd and L. Vanderberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

[6] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *J. Global Optim.*, 16(1):23–32, 2000.

[7] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In *Advances in Neural Information Processing Systems*, 1996.

[8] S. Chen. *Basis Pursuit*. Ph.d. dissertation, Statistics Dept., Stanford Univ., 1995.

[9] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *Intl. J. Control*, 50(5):1873–1896, 1989.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by Basis Pursuit. *SIAM J. Sci. Comp.*, 20(1):33–61, 1999.

[11] W. Cheney and A. A. Goldstein. Proximity maps for convex sets. *Proc. Amer. Math. Soc.*, 10(3):448–450, Jun. 1959.

[12] J. F. Claerbout and F. Muir. Robust modeling of erratic data. *Geophysics*, 38(5):826–844, October 1973.

[13] R. R. Coifman and Y. Meyer. Nouvelles bases orthonormées de $L^2(\mathbb{R})$ ayant la structure du système de Walsh. Preprint, Mathematics Dept., Yale Univ., 1989.

[14] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best-basis selection. *IEEE Trans. Inform. Theory*, 1992.

[15] J. H. Conway, R. H. Hardin, and N. J. A. Sloane. Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Math.*, 5(2):139–159, 1996.

[16] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups*. Springer Verlag, 3rd edition, 1998.

[17] F. Cordellier and J. Ch. Fiorot. On the Fermat–Weber problem with convex cost functionals. *Math. Prog.*, 14:295–311, 1978.

[18] C. Couvreur and Y. Bresler. On the optimality of the Backward Greedy Algorithm for the subset selection problem. *SIAM J. Matrix Anal. Appl.*, 21(3):797–808, 2000.

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[20] I. Csiszár. Sanov property, generalized I-projection and a conditional limit theorem. *Ann. Probability*, 12:768–793, 1984.

[21] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 2004. To appear.

[22] G. Davis, S. Mallat, and M. Avellaneda. Greedy adaptive approximation. *J. Constr. Approx.*, 13:57–98, 1997.

[23] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions. *Optical Eng.*, July 1994.

[24] R. DeVore and V. N. Temlyakov. Some remarks on greedy algorithms. *Adv. Comput. Math.*, 5:173–187, 1996.

[25] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, pages 51–150, 1998.

[26] I. S. Dhillon, E. Marcotte, and U. Roshan. Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19:1612–1619, 2003.

[27] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[28] P. Domingos. The role of Occam's Razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3:409–425, 1999.

[29] D. L. Donoho and M. Elad. Maximal sparsity representation via ℓ_1 minimization. *Proc. Natl. Acad. Sci.*, 100:2197–2202, March 2003.

[30] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. Submitted to *IEEE Trans. Inform. Theory*, February 2004.

[31] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inform. Theory*, 47:2845–2862, Nov. 2001.

[32] D. L. Donoho and I. M. Johnstone. Minimax estimation via wavelet shrinkage. Statistics Dept. Technical Report, Stanford Univ., 1992.

[33] D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM J. Appl. Math.*, 49(3):906–931, June 1989.

[34] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edition, 2001.

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

[36] T. Ericson and V. Zinoviev. *Codes on Euclidean Spheres*. Elsevier, 2001.

[37] E. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21(3):768, 1965.

[38] J. H. Friedman and W. Stuetzle. Projection Pursuit Regressions. *J. Amer. Statist. Soc.*, 76:817–823, 1981.

[39] P. Frossard and P. Vandergheynst. Redundant representations in image processing. In *Proceedings of the 2003 IEEE International Conference on Image Processing*, 2003. Special session.

[40] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, and M. Kunt. A posteriori quantization of progressive Matching Pursuit streams. *IEEE Trans. Signal Processing*, 52(2):525–535, Feb. 2004.

[41] J.-J. Fuchs. Extension of the Pisarenko Method to sparse linear arrays. *IEEE Trans. Signal Processing*, 45:2413–2421, Oct. 1997.

[42] J.-J. Fuchs. Estimation and detection of superimposed signals. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.

[43] J.-J. Fuchs. On sparse representations in arbitrary redundant bases. IRISA Technical Report, Univ. de Rennes I, Dec. 2002. Submitted to *IEEE Trans. Inform. Theory.*

[44] J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Trans. Inform. Th.*, 50(6):1341–1344, June 2004.

[45] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Company, New York, USA, 1979.

[46] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen. The complexity of the generalized Lloyd–Max problem. *IEEE Trans. Inform. Theory*, 28(2):255–256, 1982.

[47] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[48] A. C. Gilbert, M. Muthukrishnan, and M. J. Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2003.

[49] F. Girosi. An equivalence between sparse approximation and Support Vector Machines. *Neural Comput.*, 10(6):1455–1480, 1998.

[50] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

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

[52] R. Gribonval and E. Bacry. Harmonic decomposition of audio signals with Matching Pursuit. *IEEE Trans. Signal Processing*, 51(1):101–111, 2003.

[53] R. Gribonval and M. Nielsen. Highly sparse representations from dictionaries are unique and independent of the sparseness measure. Department of Mathematical Sciences Technical Report R-2003-16, Aalborg University, Oct. 2003.

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

[55] R. Gribonval and M. Nielsen. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. IRISA Report 1619, Université de Rennes I, 2004.

[56] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18(3):838–853, 1997.

[57] R. H. Hardin and N. J. A. Sloane. A new approach to the construction of optimal designs. *J. Statistical Planning and Inference*, 37:339–369, 1993.

[58] G. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 2nd edition, 1952.

[59] R. Heath, T. Strohmer, and A. J. Paulraj. On quasi-orthogonal signatures for CDMA systems. In *Proceedings of the 2002 Allerton Conference on Communication, Control and Computers*, 2002.

[60] R. B. Holmes and V. I. Paulsen. Optimal frames for erasures. *Linear Algebra Appl.*, 377:31–51, Jan. 2004.

[61] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[62] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 2nd edition, 2002.

[63] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, 1989.

[64] D. D. Lee and H. S. Seung. Unsupervised learning by convex and conic coding. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 515–521. The MIT Press, 1997.

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

[66] P. W. H. Lemmens and J. J. Seidel. Equi-isoclinic subspaces of Euclidean spaces. *Proc. Nederl. Akad. Wetensch. Series A*, 76:98–107, 1973.

[67] S. Levy and P. K. Fullagar. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics*, 46(9):1235–1243, 1981.

[68] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Comput.*, 12:337–365, 2000.

[69] S. P. Lloyd. Least squares quantization in PCM. Technical note, Bell Laboratories, 1957.

[70] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, pages 281–296, 1967.

[71] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, London, 2nd edition, 1999.

[72] S. Mallat and Z. Zhang. Matching Pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, 1993.

[73] C. L. Mallows. Some comments on c_p. *Technometrics*, 15:661–675, 1973.

[74] R. R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *J. Comp. Sys. Sci.*, 12:108–121, 1976.

[75] A. J. Miller. *Subset Selection in Regression*. Chapman and Hall, London, 2nd edition, 2002.

[76] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24:227–234, 1995.

[77] T. Nguyen and A. Zakhor. Matching Pursuits based multiple description video coding for lossy environments. In *Proceedings of the 2003 IEEE International Conference on Image Processing*, Barcelona, 2003.

[78] D. W. Oldenburg, T. Scheuer, and S. Levy. Recovery of the acoustic impedance from reflection seismograms. *Geophysics*, 48:1318–1337, 1983.

[79] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[80] M. L. Overton. A quadratically convergent method for minimizing a sum of Euclidean norms. *Math. Prog.*, 27:34–63, 1983.

[81] P. Paatero and U. Tapper. Positive Matrix Factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Press, 1998. Corrected republication.

[83] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems and Computers*, Nov. 1993.

[84] R. A. Rankin. On the closest packing of spheres in n dimensions. *Ann. Math.*, 48:1062–1081, 1947.

[85] B. D. Rao and Y. Bresler. Signal processing with sparseness constraints. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998. Special session.

[86] B. D. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Trans. Signal Processing*, 47(1):187–200, 1999.

[87] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1979.

[88] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[89] J. Rohn. Computing the norm $\|A\|_{\infty,1}$ is NP-hard. *Linear and Multilinear Algebra*, 47:195–204, 2000.

[90] K. Rose. Deterministing annealing for clustering, compression, classification, regression and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, Nov. 1998.

[91] P. Sallee and B. A. Olshausen. Learning sparse, multi-scale image representations. In *Advances in Neural Information Processing Systems*, 2002.

[92] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM J. Sci. Stat. Comput.*, 7(4):1307–1330, 1986.

[93] E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen, I. *Math. Annalen*, 63:433–476, 1906–1907.

[94] N. J. A. Sloane. Packing planes in four dimensions and other mysteries, Oct. 2002. Invited talk, Conference on Applied Mathematics, Univ. of Oklahoma at Edmond.

[95] N. J. A. Sloane. Table of best Grassmannian packings. In collaboration with A. R. Calderbank, J. H. Conway, R. H. Hardin, E. M. Rains, P. W. Shor and others. Published electronically at `http://www.research.att.com/~njas/grass/grassTab.html`, 2004.

[96] N. J. A. Sloane. Tables of spherical codes. In collaboration with R. H. Hardin, W. D. Smith and others. Published electronically at `http://www.research.att.com/~njas/packings/`, 2004.

[97] N. Srebro and T. Jaakkola. Sparse matrix factorization of gene expression data. Unpublished abstract. Available from `http://www.ai.mit.edu/research/abstracts/abstracts2001/genomics/01srebr%o.pdf`, 2001.

[98] J. L. Starck, D. L. Donoho, and E. J. Candès. Astronomical image representation by the Curvelet Transform. *Astronomy and Astrophysics*, 398:785–800, 2003.

[99] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimation. *SIAM J. Numer. Anal.*, 17(30):403–409, 1980.

[100] T. Strohmer and R. W. Heath. Grassmannian frames with applications to coding and communication. *Appl. Comp. Harmonic Anal.*, 14(3):257–275, May 2003.

[101] M. Sustik, J. A. Tropp, I. S. Dhillon, and R. W. Heath. On the existence of equiangular tight frames. Submitted, June 2004.

[102] P. M. L. Tammes. On the origin of number and arrangement of the places of exit on the surface of pollen grains. *Rec. Trav. bot. neerl.*, 27:1–84, 1930.

[103] H. L. Taylor, S. C. Banks, and J. F. McCoy. Deconvolution with the ℓ_1 norm. *Geophysics*, 44(1):39–52, 1979.

[104] V. Temlyakov. Nonlinear methods of approximation. *Foundations of Comp. Math.*, July 2002.

[105] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc.*, 58, 1996.

[106] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 2004. To appear.

[107] J. A. Tropp. Just relax: Convex programming methods for subset selection and sparse approximation. ICES Report 04-04, The University of Texas at Austin, 2004.

[108] J. A. Tropp and I. S. Dhillon. Matrix nearness problems with Bregman divergences. In preparation, 2004.

[109] J. A. Tropp, I. S. Dhillon, R. Heath, and T. Strohmer. Designing structured tight frames via an alternating projection method. ICES Report 0350, The University of Texas at Austin, 2003.

[110] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *Proc. of the 2003 IEEE International Conference on Image Processing*, Barcelona, 2003.

[111] M. Trosset. Approximate maximin distance designs. In *Proceedings of the Section on Physical and Engineering Sciences*, pages 223–227. American Statistical Association, 1999.

[112] P. Tseng. Nearest q-flat to m points. *J. Optim. Theory Appl.*, 105(1):249–252, April 2000.

[113] A. van den Bos. Complex gradient and Hessian. *IEE Proc.-Vis. Image Signal Processing*, 141(6), Dec. 1994.

[114] J. von Neumann. *Functional Operators, Vol. II.* Number 22 in Annals of Mathematics Studies. Princeton University Press, 1950.

[115] C. K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, 2000.

Vita

Joel A. Tropp was born in Austin, Texas on July 18, 1977, the son of Richard and Adrienne Tropp. He earned the Bachelor of Arts degree in Plan II Liberal Arts Honors and the Bachelor of Sciences degree in Mathematics from the University of Texas at Austin in May 1999. He graduated *magna cum laude* along with Special Honors in Plan II, and he was the Dean's Honored Graduate in Mathematics. As an undergraduate, he was a recipient of the Barry M. Goldwater National Science Scholarship and a semi-finalist for the British Marshall. He continued with graduate studies in Computational and Applied Mathematics (CAM) at UT–Austin as a CAM Graduate Fellow, and he won a National Science Foundation Graduate Fellowship in 2000. He completed the Master of Sciences degree in May 2001. In Fall 2004, he will join the Mathematics Department of the University of Michigan at Ann Arbor as an Assistant Professor.

Permanent address: 2381 Wide Horizon
Reno, NV 89509

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.