Decoding by Linear Programming

Emmanuel J. Candes and Terence Tao

***Abstract*—This paper considers a natural error correcting problem with real valued input/output. We wish to recover an input vector $f \in \mathbf{R}^n$ from corrupted measurements $y = Af + e$. Here, A is an m by n (coding) matrix and e is an arbitrary and unknown vector of errors. Is it possible to recover f exactly from the data y?**

We prove that under suitable conditions on the coding matrix A, the input f is the unique solution to the ℓ_1-minimization problem ($\|x\|_{\ell_1} := \sum_i |x_i|$)

$$\min_{g \in \mathbf{R}^n} \|y - Ag\|_{\ell_1}$$

provided that the support of the vector of errors is not too large, $\|e\|_{\ell_0} := |\{i : e_i \neq 0\}| \leq \rho \cdot m$ for some $\rho > 0$. In short, f can be recovered exactly by solving a simple convex optimization problem (which one can recast as a linear program). In addition, numerical experiments suggest that this recovery procedure works unreasonably well; f is recovered exactly even in situations where a significant fraction of the output is corrupted.

This work is related to the problem of finding sparse solutions to vastly underdetermined systems of linear equations. There are also significant connections with the problem of recovering signals from highly incomplete measurements. In fact, the results introduced in this paper improve on our earlier work. Finally, underlying the success of ℓ_1 is a crucial property we call the uniform uncertainty principle that we shall describe in detail.

***Index Terms*—Basis pursuit, decoding of (random) linear codes, duality in optimization, Gaussian random matrices, ℓ_1 minimization, linear codes, linear programming, principal angles, restricted orthonormality, singular values of random matrices, sparse solutions to underdetermined systems.**

I. Introduction

A. Decoding of Linear Codes

THIS paper considers the model problem of recovering an input vector $f \in \mathbf{R}^n$ from corrupted measurements $y = Af + e$. Here, A is an m by n matrix (we will assume throughout the paper that $m > n$), and e is an arbitrary and unknown vector of errors. The problem we consider is whether it is possible to recover f exactly from the data y. And if so, how?

Our problem has of course the flavor of error correction problems which arise in coding theory as we may think of A as a *linear code*; a linear code is a given collection of codewords which are vectors $a_1, \ldots, a_n \in \mathbf{R}^m$—the columns of the matrix A. We would like to emphasize, however, that there is a clear distinction between our real-valued setting and the finite alphabet one which is more common in the information theory literature. Given a vector $f \in \mathbf{R}^n$ (the "plaintext") we can then generate a vector Af in $\mathbf{R}^m$ (the "ciphertext"); if A has full rank, then one can clearly recover the plaintext f from the ciphertext Af. But now we suppose that the ciphertext Af is corrupted by an arbitrary vector $e \in \mathbf{R}^m$ giving rise to the corrupted ciphertext $Af + e$. The question is then: given the coding matrix A and $Af + e$, can one recover f exactly?

As is well known, if the fraction of the corrupted entries is too large, then of course we have no hope of reconstructing f from $Af + e$; for instance, assume $m = 2n$ and consider two distinct plaintexts f, f' and form a vector $g \in \mathbf{R}^m$ by concatenating the first half of Af together with the second half of Af'. Then $g = Af + e = Af' + e'$ where both e and e' are supported on sets of size at most $n = m/2$. This simple example shows that accurate decoding is impossible when the size of the support of the error vector is greater or equal to a half of that of the output Af. Therefore, a common assumption in the literature is to assume that only a small fraction of the entries are actually damaged

$$\|e\|_{\ell_0} := |\{i : e_i \neq 0\}| \leq \rho \cdot m. \tag{1.1}$$

For which values of ρ can we hope to reconstruct e with practical algorithms? That is, with algorithms whose complexity is at most polynomial in the length m of the code A?

To reconstruct f, note that it is obviously sufficient to reconstruct the vector e since knowledge of $Af + e$ together with e gives Af, and consequently f, since A has full rank. Our approach is then as follows. We construct a matrix which annihilates the $m \times n$ matrix A on the left, i.e., such that $FA = 0$. This can be done in an obvious fashion by taking a matrix F whose kernel is the range of A in $\mathbf{R}^m$, which is an n-dimensional subspace (e.g., F could be the orthogonal projection onto the cokernel of A). We then apply F to the output $y = Af + e$ and obtain

$$\tilde{y} = F(Af + e) = Fe \tag{1.2}$$

since $FA = 0$. Therefore, the decoding problem is reduced to that of reconstructing a *sparse* vector e from the observations Fe (by sparse, we mean that only a fraction of the entries of e are nonzero). Therefore, the overarching theme is that of the sparse reconstruction problem, which has recently attracted a lot of attention as we are about to see.

Manuscript received February 2005; revised September 2, 2005. The work of E. J. Candes is supported in part by the National Science Foundation under Grants DMS 01-40698 (FRG) and ACI-0204932 (ITR), and by an A. P. Sloan Fellowship. The work of T. Tao is supported in part by a grant from the Packard Foundation.

E. J. Candes is with the Department of Applied and Computational Mathematics, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: emmanuel@acm. caltech.edu).

T. Tao is with the Department of Mathematics, University of California, Los Angeles, CA 90095 USA (e-mail: tao@math.ucla.edu).

Communicated by M. P. Fossorier, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2005.858979