When is missing data recoverable?

Yin Zhang
CAAM Technical Report TR06-15
Department of Computational
and Applied Mathematics
Rice University, Houston, TX 77005

October, 2006

Abstract

Suppose a non-random portion of a data vector is missing. With some minimal prior knowledge about the data vector, can we recover the missing portion from the available one? In this paper, we consider a linear programming approach to this problem, present numerical evidence suggesting the effectiveness and limitation of this approach, and give deterministic conditions that guarantee a successful recovery. Our theoretical results, though related to recent results in compressive sensing, do not rely on randomization.

1 The MDR Problem

We consider a missing data recovery (MDR) problem as follows. Suppose that a data vector $f \in \Re^d$ contains some available data in $b \in \Re^m$ and some missing data in $u \in \Re^{d-m}$; namely, after a reordering,

$$f := \begin{bmatrix} b \\ u \end{bmatrix}, \quad b \in \Re^m, \; u \in \Re^{d-m}. \tag{1}$$

With some minimal prior knowledge, we seek to recover the unknown missing data u. As an example, let us look at the image below in Figure 1 where 85% of pixels in a region on the left half of the image are blacked out (missing). Given this badly incomplete image and nothing else, can we recover the missing pixels and see what the original image looks like? Or is this even possible at all?

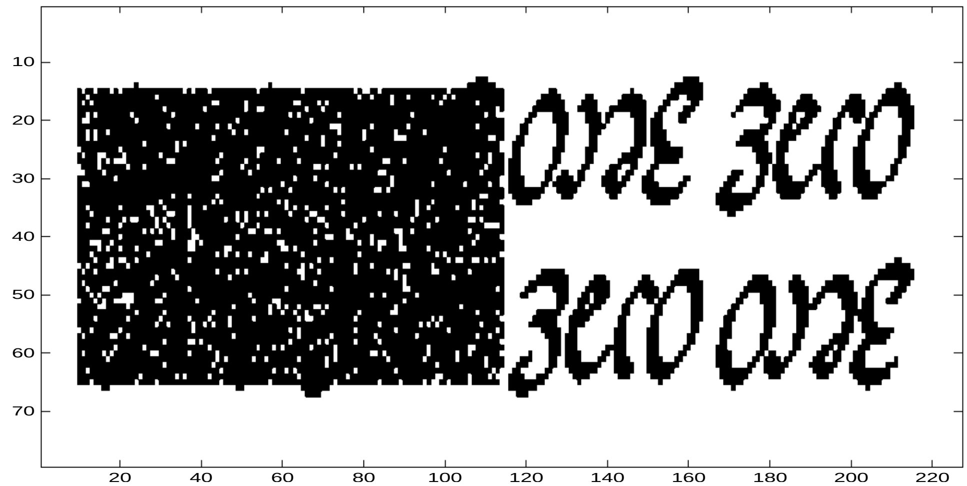

Figure 1: An image with missing pixels

In this paper, we will study a linear programming technique for the problem of missing data recovery (MDR), and will show that under suitable conditions this approach can indeed achieve the goal of recovery.

2 MDR by Linear Programming

In fact, the only prior information required in our MDR approach is something or anything that allows us to choose, or guess, a matrix called a frame. For example, for the data vector in Figure 1, common knowlege in image processing enables us to guess that an inverse discrete cosine transform (IDCT) matrix could be a suitable frame for this MDR instance. Let us explain.

2.1 A Frame

Let $\Phi \in \Re^{d \times n}$ be a matrix with $\mathrm{rank}(\Phi) = d \leq n$ (so Φ may or may not be square). We say that the columns of Φ form a frame instead of a basis because the columns are linearly dependent whenever $d < n$.

We hope that some prior knowledge will allow us to choose, or guess, a frame Φ so that the complete data vector f can be represented by a small number of columns of Φ; namely, f can be *sparsely represented* by the frame Φ. More precisely, we call $x^* \in \Re^n$ a representation of f under the frame Φ if x^* satisfies the (possibly under-determined) linear system,

$$\Phi x = f, \tag{2}$$

and has the fewest nonzero components possible. On the other hand, if a representation x^* is known, we can obtain the complete date vector from $f = \Phi x^*$.

Let us partition the rows of Φ into two parts according to the indices of the available and missing data b and u, respectively; that is, with a reordering,

$$\Phi = \begin{pmatrix} A \\ B \end{pmatrix}, \quad A \in \Re^{m \times n}, \quad B \in \Re^{(d-m) \times n}. \tag{3}$$

We emphasize that the matrix A can vary from case to case depending on actual occurrence of the indices for the available data b. It is important to remember this dependency of A on b and not think of A as a fixed part of Φ.

2.2 A Linear Program

With the partitioning in (3), the equations in (2) can be regrouped:

$$\Phi x^* = f \quad \Rightarrow \quad Ax^* = b, \quad Bx^* = u.$$

Since u is unknown, we are unable to make any use of the equation $Bx = u$. Therefore, our hope for finding the representation x^* of f lies entirely on the equations corresponding to the available date:

$$Ax = b, \tag{4}$$

which is of course under-determined. The fact that x^* may be sparse under the frame Φ and recent results in compressive sensing (see the next subsection) lead us to solving the ℓ_1-minimization problem:

$$\min_x \{\|x\|_1 : Ax = b\}, \tag{5}$$

which can be easily reformulated as a linear program. The real questions now are: *under what conditions can we guarantee that the solution to (5) is actually x^*, and are these conditions realistic in practice?*

2.3 Related Results

Recovery of missing pixels in images by ℓ_1 minimization was proposed in [12] based on a different motivation. The authors in [12] assume the probability model that wavelet coefficients of an image with missing pixels are independent Laplacian. To reconstruct these coefficients, they choose to maximize the likelihood of the wavelet coefficients subject to the inverse wavelet transform equations corresponding to the available pixels. The resulting problem reduces to, in our notation,

$$\min \left\{ \sum_{i=1}^{n} c_i |x_i| : Ax = b \right\}, \tag{6}$$

where x is the wavelet coefficient vector and $c_i > 0$ are related to parameters in their probability model.

In [8], Elad, Starck, Querre and Donoho proposed an algorithm for image inpainting that uses sparse representations of image components to fill in missing pixels, which is the same underlying principle that we use in this paper. The authors presented excellent experimental results in [8].

The use of ℓ_1 minimization to find the sparse solutions to under-determined linear system has recently been actively studied in the area of compressive (a.k.a. compressed) sensing. In compressive sensing, a sufficiently sparse (or compressible) signal is encoded by a set of random projections whose cardinality can be far less than the signal length itself. Then, at a high probability, the signal can be decoded by finding the minimal ℓ_1 norm solution to an under-determined linear system of equations just like (5). Representative works in compressive sensing include, but not limited to, those of Candés, Romberg, and Tao [2, 3], Donoho [6, 7], Rudelson and Veshynin [11], Gilbert and Tropp [13] and Baraniuk, Davenport, DeVore and Wakin [1].

Compressive sensing relies heavily on randomization, using either random matrices or randomly chosen rows of large deterministic matrices. Such randomization provides probabilistic performance guarantees. In MDR, on the other hand, we cannot assume that the portion of missing data is completely random, nor do we in general have the luxury of using a completely random matrices as a frame. As a result, getting a performance guarantee is much harder for MSD, making it a more challenging problem. This loss of randomization is the main distinction between MDR and compressive sensing as far as theoretical analysis is concerned.

3 Experimental Results

We present some experimental results using the linear programming approach to attack the MDR problem.

3.1 Experiments with Synthetic 1-D Data

In these experiments, we always use the inverse discrete cosine transform (IDCT) matrices as our frames $\Phi \in \Re^{n\times n}$ where Φ is defined by

$$\Phi_{ij} = \frac{\min(j, \sqrt{2})}{\sqrt{n}} \cos\left(\frac{(2i-1)(j-1)\pi}{2n}\right), \quad 1 \leq i, j \leq n. \tag{7}$$

We set the size of our data vector $f = \Phi x^*$ to 1000 among which 750 are available and 250 are missing. We construct representation vectors so they have 20 nonzeros among its first 120 elements (the lower frequencies were picked because they produce nicer-looking pictures).

Many numerical experiments have been performed in this setting where recoveries were mostly, but not 100%, successful. We present results for one typical experiment and, for the sake of reproducibility, give the Matlab script that produced the data.

```
%%% Matlab script for generating 1D data: A, b %%%
    n = 1000;                    % data size
    Phi = dctmtx(n)';            % get the frame
    xs = zeros(n,1);             % initialize x*
    rand('state',11);            % for reproducibility
    k = 20;                      % number of nonzeros in x*
    p = randperm(120);           % chosen between 1 to 120
    randn('state',11);           % for reproducibility
    xs(p(1:k)) = randn(k,1);     % random values for x*
    f = Phi*xs;                  % complete data vector v*
    b = f(1:750);                % available data b
    A = Phi(1:750,:);            % matrix A
```

The results are plotted in Figure 2, including the plots for the complete data, the available data and the recovered data. As one can see, the one quarter of data missing has been perfectly recovered.

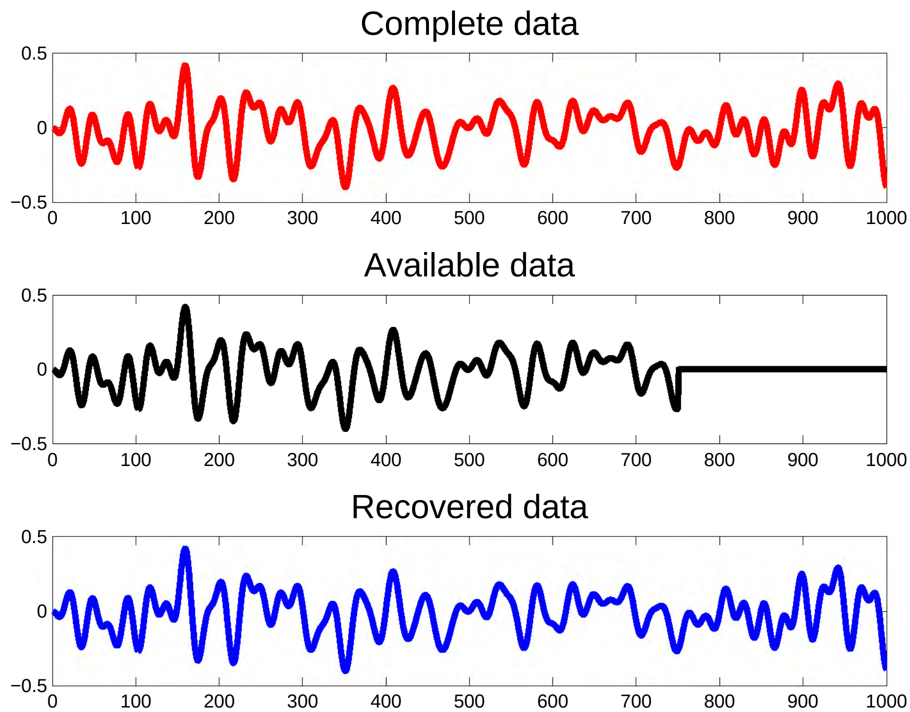

Figure 2: Results for a 1-D experiment

3.2 Experiments with 2-D Images

We also performed experiments with 2-D images where we blacked out a portion of pixels as missing data, and then tried to recover them under IDCT frames. With these 2-D "real life data", our success was far more limited than with the 1-D synthetic data. We often could not sufficiently recover information contents from incomplete images whose information integrity has been badly compromised. We conjecture that the main reason was perhaps that the IDCT bases used were not sufficiently suitable for most of the tested images. On the other hand, we did find some successes with black and white images containing simple patterns such as characters. Here we present two of such successful examples in Figure 3.

In the first example, the original image is a sketch of a mosquito. We randomly blacked out 70% of the pixels from the middle section of the image as the missing data. The complete, available and recovered images are plotted on the left of Figure 3. The quality of the recovery is sufficiently good from the viewpoint of naked eyes.

The second example is the one in Figure 1, where we have randomly blacked out 85%

of the pixels from the left half of the image as the missing data. The complete, available and recovered images are given on the right side of Figure 3. This recovery quality appears remarkably good this time.

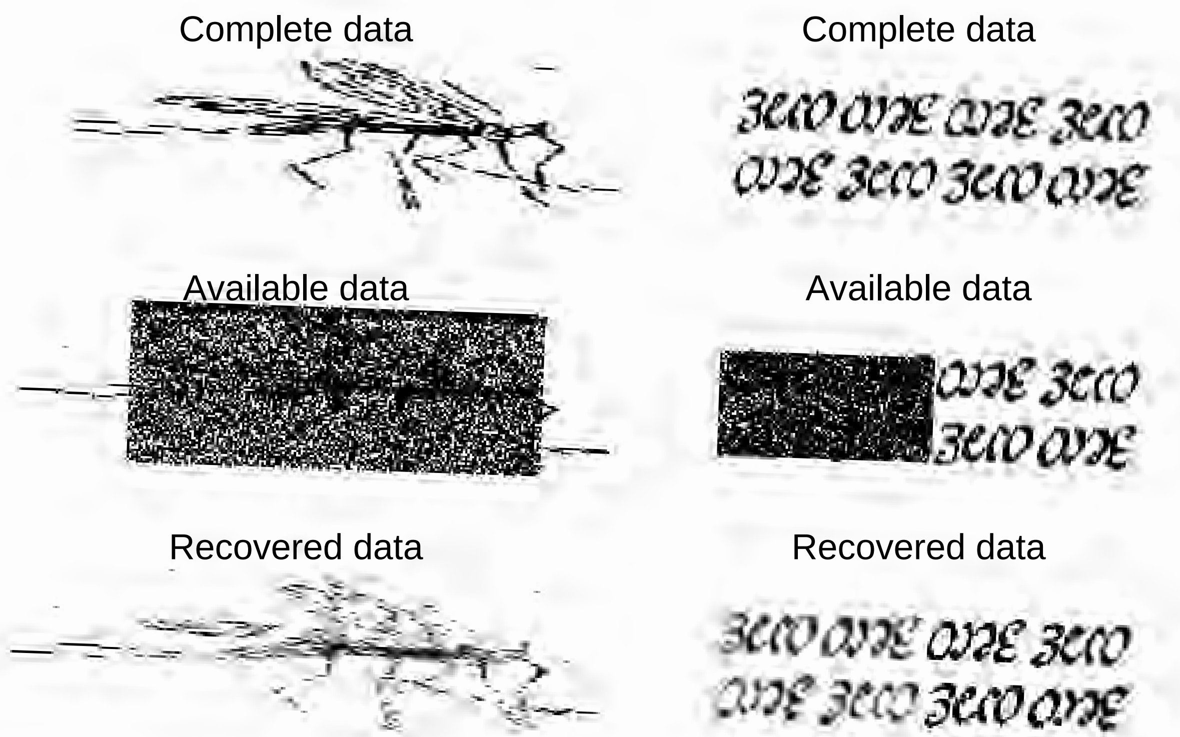

Figure 3: Results for 2D experiments

These experimental results appear quite promising. On the other hand, our experiments also reminded us the limitations of this approach (after all, it is not always possible to recover what is really lost).

It is worth pointing out that in most image inpainting example given in the literature, missing pixels do not sufficiently compromise the information integrity of the original images in an overall sense. More precisely, the missing pixels are either so restricted in scope or low in density that the basic contents of remaining images (i.e., available date) are still clearly recognizable by naked eyes. On the contrary, in our examples the regions with missing pixels appear to have lost their information contents at least to naked eyes. Therefore, the quality of these recoveries is more impressive.

4 Theoretical Results

In this section, we give sufficient conditions that guarantee successful recoveries. These conditions are also necessary if one is to guarantee recoveries for varying values in data representations under a given frame. These results are simple extensions to our results in [15].

4.1 Notation and Definition

By a partition (S, Z), we mean a partition of the index set $\{1, 2, \cdots, n\}$ into two disjoint subsets S and Z so that $S \cup Z = \{1, 2, \cdots, n\}$ and $S \cap Z = \emptyset$. In particular, for any $x \in \Re^n$, the partition $(S(x), Z(x))$ refers to the support $S(x)$ of x and its complement – the zero set $Z(x)$; namely,

$$S(x) = \{i : x_i \neq 0,\ 1 \leq i \leq n\}, \quad Z(x) = \{1, \cdots, n\} \setminus S(x). \tag{8}$$

We will occasionally omit the dependence of a partition (S, Z) on x when it is clear from the context.

For any index subset $J \subset \{1, 2, \cdots, n\}$, $|J|$ is the cardinality of J. For a vector $v \in \Re^n$, similarly, v_J denotes the sub-vector of v with those components whose indices are in J. We now introduce some definitions.

Definition 1 (S- and k-Balancedness). *For any given index subset $S \subset \{1, 2, \cdots, n\}$ a subspace $\mathbb{N} \subset \Re^n$ is S-balanced if*

$$\|v_S\|_1 \leq \frac{1}{2}\|v\|_1, \quad \forall\, v \in \mathbb{N}. \tag{9}$$

Moreover, it is strictly S-balanced if the strict inequality holds for any non-zero $v \in \mathbb{N}$.

A subspace $\mathbb{N} \subset \Re^n$ is (strictly) k-balanced if it is (strictly) S-balanced for any S with $|S| \leq k$.

We observe that the properties of balancedness of a subspace are monotone. Specifically, if a subspace is (strictly) S_1-balanced, then it is also (strictly) S_2-balanced whenever $S_2 \subset S_1$. Therefore, in a sense the sparser a vector x is, the more likely a subspace will be S-balanced with respect to its support $S = S(x)$. Similarly, if a subspace is k-balanced, then it must be $(k-1)$-balanced.

Example 1. *Consider the one-dimensional subspace in $\Re^4$ spanned by the vector*

$$[0.5000 \ -0.6533 \ \ 0.5000 \ -0.2706]^T. \tag{10}$$

It is easy to see that this subspace is $\{i,4\}$-balanced for $i = 1,2,3$, because the sum in absolute value of the fourth component with another component is less than the sum in absolute value of the other two components. Since the subspace is not $\{1,2\}$-balanced, for example, it is not 2-balanced, while it is obviously 1-balanced.

4.2 Main Results

Recall that the complete data vector f has the form:

$$f := \begin{bmatrix} b \\ u \end{bmatrix} \in \Re^d, \quad b \in \Re^m.$$

Under a given frame $\Phi \in \Re^{d\times n}$, f has a sparsest representation vector $x^* \in \Re^n$ that satisfies the equation $\Phi x^* = f$.

There are two pieces of critical information available to us: (i) the available data b, and (ii) the sparsity of x^* under the frame Φ. Roughly speaking, we will be able to recover the missing data u if the representation x^* is sufficiently sparse, the available data b is sufficiently informative, and the part of Φ associated with the available data (namely, the A part of Φ) has certain desirable properties.

Besides its size m, the information content of b is difficult to quantify. However, its importance can never be overemphasized. For one thing, the size of b directly dictates the size of A. In addition, the locations of components of b have a more subtle, yet potentially great, influence on the properties of the matrix A.

Now we state our main results, though their proofs will be given in the next subsection.

Theorem 1. *Given available date b, the missing data u can be recovered from (5) if there exists a frame Φ such that the null space of A is strictly S-balanced for $S = S(x^*)$, where x^* is the sparsest solution to $\Phi x = f$ and A consists of rows of Φ corresponding to b.*

Moreover, to recover all possible date vector f whose representation under Φ has the same support S, it is necessary that the null space of A is strictly S-balanced.

Theorem 1 requires the condition of S-balancedness for $S = S(x^*)$, which is unverifiable because x^* is unknown. The next theorem uses more a general condition to handle different data values under the same frame.

Theorem 2. *Let Φ be a fixed frame. Given any available data b, let A consist of the rows of Φ corresponding to b and x^* be the sparsest solution to $\Phi x = f$ with k nonzeros. Then the missing data u can be recovered from (5) if (a) the null space of A is strictly k-balanced, and (b) x^* remains the sparsest solution to $Ax = b$.*

Moreover, to recover all possible date vector f whose representation under Φ has k nonzeros, conditions (a) and (b) are also necessary.

We mention that condition (b) is imposed out of a technical necessity. Since x^* is the sparsest solution to $\Phi x = f$, it is the sparsest simultaneous solution to both $Ax = b$ and $Bx = u$. Once we take away the equation $Bx = u$, $Ax = b$ can permit a sparser solution. At this point, we do not know whether or not condition (b) is implied by condition (a), and hence have not been able to rule it out.

From our experimental results, we know that the conditions in the two theorems do hold sometimes. Here we consider a simple example to further demonstrate the point.

Example 2. *Consider the 4×4 inverse discrete cosine transform matrix:*

$$\Phi = \begin{bmatrix} 0.5000 & 0.6533 & 0.5000 & 0.2706 \\ 0.5000 & 0.2706 & -0.5000 & -0.6533 \\ 0.5000 & -0.2706 & -0.5000 & 0.6533 \\ 0.5000 & -0.6533 & 0.5000 & -0.2706 \end{bmatrix}. \tag{11}$$

For any data vector $f = \Phi x^*$, let the available data $b \in \Re^3$ be the first three components of f. Then the missing data $u \in \Re$ is the fourth component of f. Since Φ is orthogonal, the null space of A, where A is the first three rows of Φ, is spanned by the fourth row of Φ which is nothing but the vector in Example 1. As is mentioned in Example 1, this subspace is strictly 1-balanced and $\{i, 4\}$-balanced for $i = 1, 2, 3$.

In fact, similar conclusions can be made for subspaces spanned by each of the rows of Φ. By our theorems, a missing single component, wherever it occurs, can always be recovered if the data vector is a multiple of a single column of Φ, i.e., the number of nonzeros in x^* is $k = 1$.

When the data vector is a linear combination of two columns of Φ (i.e.,$k = 2$), a missing component can sometimes be recovered but not always, depending on how the data vector is composed.

Let b be the first and the fourth components of the data vector f. Can we still recover the other two components of f? Now the null space of A is spanned by the second and the third rows of Φ. Any vector in this two-dimensional subspace is of the form:

$$v = \begin{pmatrix} 0.5000(\alpha + \beta) \\ 0.2706(\alpha - \beta) \\ -0.5000(\alpha + \beta) \\ -0.6533(\alpha - \beta) \end{pmatrix}.$$

It is easy to see that this subspace is not even 1-balanced, but is strictly S-balanced for $S = \{2\}$. Hence the two missing components can be recovered if the data vector is a multiple of the second column of Φ.

Moreover, the subspace is also S-balanced for $S = \{1, 2\}$ and $S = \{2, 3\}$, but is not strictly balanced. In these cases, the linear program from (5) has multiple optimal solutions, preventing a unique recovery.

4.3 Technical Lemmas and Proofs

Recall that the matrix $A \in \Re^{m \times n}$ consists of the rows of a frame Φ corresponding to the available date b. We now consider another ℓ_1-minimization problem (also equivalent to a linear program):

$$\min_y \|C^T y - c\|_1, \tag{12}$$

where $C \in \Re^{(n-m) \times n}$ is such that $\text{rank}(C) = n - m$ and $AC^T = 0$, and

$$c = C^T \hat{y} + \hat{x} \tag{13}$$

for some fixed $\hat{y} \in \Re^{n-m}$ and $\hat{x} \in \Re^n$. Clearly, the rows of C span the null space of A.

The reason for us to consider problem (12) is that solving (12) is equivalent to solving (5), as is stated in the following lemma.

Lemma 1. *Let c be defined in (13) with fixed $\hat{y}$ and $\hat{x}$. If $\hat{y}$ solves (12), then $\hat{x}$ solves (5) with $b = Ac$, and vice versa.*

For a proof of the lemma, we refer readers to [14]. While problem (5) is often called *basis pursuit* [5], among different names, problem (12) is often called *error correction* [4]. The equivalence of these two has been first observed in [3, 11], and then formalized in [14].

The next lemma is essentially adopted from [15].

Lemma 2. *Let c be defined in (13) with fixed $\hat{y}$ and $\hat{x}$. Then $\hat{y}$ uniquely solves (12) if the null space of A (i.e., the range space of C^T) is strictly S-balanced for $S = S(\hat{x})$.*

Moreover, $\hat{y}$ uniquely solves (12) for all possible values of $\hat{x}$ with the same support S only if the null space of A is strictly S-balanced.

Proof. Let $v = v(y) := C^T(y - \hat{y})$ for any $y \in \Re^{d-m}$ and let $S := S(\hat{x})$. Then for any $y \neq \hat{y}$,

$$\begin{aligned}
\|C^T y - c\|_1 &= \|C^T(y - \hat{y}) - \hat{x}\|_1 = \|v - \hat{x}\|_1 \\
&= \|v_S - \hat{x}_S\|_1 + \|v_Z\|_1 \\
&\geq \|\hat{x}_S\|_1 - \|v_S\|_1 + \|v_Z\|_1 \\
&= \|\hat{x}\|_1 + (\|v_Z\|_1 - \|v_S\|_1) \\
&\equiv \|C^T \hat{y} - c\|_1 + (\|v_Z\|_1 - \|v_S\|_1).
\end{aligned}$$

Therefore, $\hat{y}$ is the unique minimizer of (12) if $\|v_Z\|_1 > \|v_S\|_1$, or equivalently $\|v_S\|_1 < \frac{1}{2}\|v\|_1$, for any nonzero v in the null space of A, meaning that the null space of A is strictly S-balanced for $S = S(\hat{x})$.

On the other hand, if we allow $\hat{x}$ to vary in value while keeping the same support S, the inequality,

$$\|v_S - \hat{x}_S\|_1 \geq \|\hat{x}_S\|_1 - \|v_S\|_1,$$

becomes an equality for any given v if $\hat{x}$, component-wise, has the same sign as v and has larger or equal magnitude than v. This proves the necessity of strict S-balancedness for $\hat{y}$ to be the minimizer for all $\hat{x}$ with the same support S, completing the proof. $\square$

Now we are ready to prove Theorems 1 and 2. Let $\hat{x} = x^*$ in Lemma 2 where x^* is the representation of f under Φ. The strict S-balancedness of the null space of A for $S = S(x^*)$ ensures that $\hat{y}$ uniquely solves (12) with $c = C^T\hat{y} + x^*$. Hence, x^* uniquely solves (5) by Lemma 1, and the missing data can be recovered as $u = Bx^*$. This proves Theorem 1. Theorem 2 can be proven similarly.

4.4 Robustness with respect to Approximate Sparsity

In practice, exact sparsity in terms of zeros and nonzeros seldom happens. Instead, we will see "approximate sparsity" where only a small number of components of x^* have relatively large absolute values while all the rest have relatively small absolute values. Although our theory is developed for exact recovery under the assumption of exact sparsity, it can be easily extended to the situations of approximate recovery under approximate sparsity by invoking the well-known fact that the unique solution of the linear program from (5) is Lipschitz continuous with respect to the available data b, i.e., the right-hand side (see [10] for example).

5 Final Remarks

We have obtained encouraging experimental results and a general theoretical framework that can tell, in theory, whether or not MDR can succeed. However, more theoretical and practical problems remain open.

Theoretical difficulties arise because the balancedness properties in our theory are not verifiable in polynomial time. As such, a polynomial-time procedure will be desirable that enables us to compute a reasonably large bound $k' < k$ for any k-balanced subspace (with k unknown) so that we can guarantee that the subspace is at least k'-balanced. Some preliminary results in this direction have been obtained [9], but further research is much needed to help us understand the strength and limitation of the proposed approach.

The most important practical issue is, in our view, the frame selection which will largely determine the success or failure of an MDR attempt. We know that sparsity of representation under a frame is a necessity, but it is not the only important factor. It is possible that clever choices of frames (i.e. use of prior information) may enable some unexpectedly successful recoveries.

The MDR approach may also serve as a prediction tool in situations where the available date would be past states of a system and the missing data would be some projected future states. If one has a good model (frame) for the overall behavior of the system, then a good prediction may be possible. Much research is needed to explore and realize the potential of this MDR approach.

Our preliminary results have shown a clear promise in the MDR approach that seems capable of achieving a unparalleled recovery quality in some cases. Since MDR can hardly utilize randomization to get probabilistic performance guarantees, it demands careful studies on a case-by-case basis.

References

[1] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin. The Johnson-Lindenstrauss Lemma Meets Compressed Sensing. Preprint, 2006.

[2] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory 52 (2006), no. 2, 489509.

[3] E. Candès and T. Tao. Decoding by Linear Programming. IEEE Trans. Inf. Theory 51 (2005), no. 12, 4203- 4215.

[4] E. Candès and T. Tao. Error Correction via Linear Programming. 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05), pp. 295-308.

[5] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by Basis Pursuit. SIAM J. Sci. Comput., 20(1):3361, 1999.

[6] D. Donoho. Compressed sensing. IEEE Trans. Inf. Theory 52 (2006), no. 4, 12891306.

[7] D. Donoho. For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ^1-norm Solution is also the Sparsest Solution. Communications on Pure and Applied Mathematics, Vol. 59 (2006), no. 6, 797 - 829.

[8] M. Elad, J-L. Starck, P. Querre, and D.L. Donoho. Simultaneous Cartoon and Texture Image Inpainting Using Morphological Component Analysis (MCA). Journal on Applied and Computational Harmonic Analysis, Vol. 19, pp. 340-358, November 2005.

[9] A. Eydelzon and Y. Zhang. In preparation.

[10] O. Mangasarian and T. Shiau. Lipschitz continuity of solutions of linear inequalities: Programs and complementarity problems. SIAM Journal on Control and Optimization, 25:583–595, 1987.

[11] M. Rudelson and R. Veshynin. Geometric approach to error correcting codes and reconstruction of signals. Available from arXiv:math.MG/0502299, Feb. 2005.

[12] I. Selesnick, R. Van Slyke and O. Guleryuz. Pixel recovery via L1 minimization in the wavelet domain. 2004 International Conference on Image Processing (ICIP), Vol. 3, pp.1819-1822, 2004.

[13] J. Tropp and A. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. Manuscript, 2005.

[14] Y. Zhang. Solution-recovery in L1-norm for non-square linear systems: deterministic conditions and open questions. Technical report TR05-06, Department of computational and Applied Mathematics, Rice University, Houston, TX, 2005.

[15] Y. Zhang. A Simple Proof for Recoverability of L1-Minimization: Go Over or Under? Technical report TR05-09, Department of computational and Applied Mathematics, Rice University, Houston, TX, 2005.