

# K-Median Clustering, Model-Based Compressive Sensing, and Sparse Recovery for Earth Mover Distance\*

Piotr Indyk

Eric Price

24 March 2011

## Abstract

We initiate the study of sparse recovery problems under the *Earth-Mover Distance (EMD)*. Specifically, we design a distribution over  $m \times n$  matrices  $A$  such that for any  $x$ , given  $Ax$ , we can recover a  $k$ -sparse approximation to  $x$  under the EMD distance. One construction yields  $m = O(k \log(n/k))$  and a  $1 + \epsilon$  approximation factor, which matches the best achievable bound for other error measures, such as the  $\ell_1$  norm.

Our algorithms are obtained by exploiting novel connections to other problems and areas, such as *streaming algorithms for  $k$ -median clustering* and *model-based compressive sensing*. We also provide novel algorithms and results for the latter problems.

---

\*This research has been supported in part by the David and Lucille Packard Fellowship, MADALGO (the Center for Massive Data Algorithmics, funded by the Danish National Research Association) and NSF grant CCF-0728645. E. Price has been supported in part by an NSF Graduate Research Fellowship.

# 1 Introduction

In recent years, a new “linear” approach for obtaining a succinct approximate representation of  $n$ -dimensional vectors (or signals) has been discovered. For any signal  $x$ , the representation is equal to  $Ax$ , where  $A$  is an  $m \times n$  matrix, or possibly a random variable chosen from some distribution over such matrices. The vector  $Ax$  is often referred to as the *measurement vector* or *linear sketch* of  $x$ . Although  $m$  is typically much smaller than  $n$ , the sketch  $Ax$  often contains plenty of useful information about the signal  $x$ .

A particularly useful and well-studied problem is that of *stable sparse recovery*. The problem is typically defined as follows: for some norm parameters  $p$  and  $q$  and an approximation factor  $C > 0$ , given  $Ax$ , recover an “approximation” vector  $x^*$  such that

$$\|x - x^*\|_p \leq C \min_{k\text{-sparse } x'} \|x - x'\|_q \quad (1)$$

where we say that  $x'$  is  $k$ -sparse if it has at most  $k$  non-zero coordinates. Sparse recovery has applications to numerous areas such as data stream computing [Mut05, Ind07] and compressed sensing [CRT06, Don06], notably for constructing imaging systems that acquire images directly in compressed form (e.g., [DDT<sup>+</sup>08, Rom09]). The problem has been a subject of extensive study over the last few years, with the goal of designing schemes that enjoy good “compression rate” (i.e., low values of  $m$ ) as well as good algorithmic properties (i.e., low encoding and recovery times). It is known by now<sup>1</sup> that there exist matrices  $A$  and associated recovery algorithms that produce approximations  $x^*$  satisfying Equation (1) with  $\ell_p = \ell_q = \ell_1$ , constant approximation factor  $C$ , and sketch length  $m = O(k \log(n/k))$ ; it is also known that this sketch length is asymptotically optimal [DIPW10, FPRU10]. Results for other combinations of  $\ell_p/\ell_q$  norms are known as well.

However, limiting the error measures to variants of  $\ell_p$  norms is quite inconvenient in many applications. First, the distances induced by  $\ell_p$  norms are typically only quite raw approximations of the perceptual differences between images. As a result, in the field of computer vision, several more elaborate notions have been proposed (e.g., in [RTG00, Low04, Lyu05, GD05]). Second, there are natural classes of images for which the distances induced by the  $\ell_p$  norm are virtually meaningless. For example, consider images of “point clouds”, e.g., obtained via astronomical imaging. If we are given two such images, where each point in the second image is obtained via small random translation of a point in the first image, then the  $\ell_p$  distance between the images will be close to the largest possible, even though the images are quite similar to each other.

Motivated by the above considerations, we initiate the study of sparse recovery under non- $\ell_p$  distances. In particular, we focus on the *Earth-Mover Distance* (EMD) [RTG00]. Informally, for the case of two-dimensional  $\Delta \times \Delta$  images (say,  $x, y : [\Delta]^2 \rightarrow \mathbb{R}_+$ ) which have the same  $\ell_1$  norm, the EMD is defined as the cost of the min-cost flow that transforms  $x$  into  $y$ , where the cost of transporting a unit of mass from a pixel  $p \in [\Delta]^2$  of  $x$  to a pixel  $q \in [\Delta]^2$  of  $y$  is equal to the  $\ell_1$  distance<sup>2</sup> between  $p$  and  $q$ . The EMD metric can be viewed as induced by a norm  $\|\cdot\|_{EMD}$ , such that  $EMD(x, y) = \|x - y\|_{EMD}$ ; see Section 2 for a formal definition. Earth-Mover Distance and its variants are popular metrics for comparing similarity between images, feature sets, etc. [RTG00, GD05].

**Results.** In this paper we introduce three sparse recovery schemes for the Earth-Mover Distance. Each scheme provides a matrix (or a distribution of matrices)  $A$ , with  $m$  rows and  $n$  columns for  $n = \Delta^2$ , such that for any vector  $x$ , given  $Ax$ , one can reconstruct a vector  $x^*$  such that

$$\|x - x^*\|_{EMD} \leq C \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}. \quad (2)$$

<sup>1</sup>In particular, a random Gaussian matrix [CRT06] or a random sparse binary matrix ([BGI<sup>+</sup>08], building on [CCFC02, CM04, CM06]) has this property with overwhelming probability. See [GI10] for an overview.

<sup>2</sup>One can also use the  $\ell_2$  distance. Note that the two distances differ by at most a factor of  $\sqrt{2}$  for two-dimensional images.

Determinism	Sketch length	Decode time	Approx.
Deterministic	$k \log n \log(n/k)$	$n \log^{O(1)} n$	$\epsilon$
Deterministic	$k \log(n/k)$	$n^{O(1)}$	$\sqrt{\log(n/k)}$
Randomized	$k \log(n/k)$	$k \log(n/k)$	$\epsilon$

Figure 1: Our results

for some approximation factor  $C > 0$ . We call any recovery scheme satisfying Equation (2) an *EMD/EMD recovery scheme*. If  $A$  is a distribution over matrices (that is, the scheme is *randomized*), the guarantee holds with some probability. The other parameters of the constructions are depicted in Figure 1.

In particular, two of our constructions yield sketch lengths  $m$  bounded by  $O(k \log(n/k))$ , which mimics the best possible bound achievable for sparse recovery in the  $\ell_1$  distance [DIPW10]. Note, however, that we are not able to show a matching lower bound for the EMD case.

**Connections and applications** What does sparse recovery with respect to the Earth-Mover Distance mean? Intuitively, a sparse approximation under EMD yields a short “signature” of the image  $x$  that approximately preserves its “EMD properties”. For example, if  $x$  consists of a small number of sparse point clouds (e.g., as in astronomical imaging), sparse approximation of  $x$  will approximately identify the locations and weights of the clouds. Our preliminary experiments with a heuristic algorithm for such data [GIP10] show that this approach can yield substantial improvements over the usual sparse recovery. Another application [RTG00] stems from the original paper, where such short signatures were constructed<sup>3</sup> for general images, to extract their color or texture information. The images were then replaced by their signatures during the experiments, which significantly reduced the computation time.

The above intuitions can be formalized as follows. Let  $x'$  be the minimizer of  $\|x - x'\|_{EMD}$  over all  $k$ -sparse vectors. Then one can observe that the non-zero entries of  $x'$  correspond to the cluster centers in the best  $k$ -median<sup>4</sup> clustering of  $x$ . Moreover, for each such center  $c$ , the value of  $x'_c$  is equal to the total weight of pixels in the cluster centered at  $c$ . Thus, a solution to the  $k$ -median problem provides a solution to our sparse recovery problem as well<sup>5</sup>.

There has been prior work on the  $k$ -median problem in the streaming model under insertions and deletions of points [FS05, Ind04]. Such algorithms utilize linear sketches, and therefore implicitly provide schemes for approximating the  $k$ -median of  $x$  from a linear sketch of  $x$  (although they do not necessarily provide the cluster weights, which are needed for the sparse recovery problem). Both algorithms<sup>6</sup> yield a method for approximating the  $k$ -median from  $\Omega(k^2 \log^{O(1)} n)$  measurements, with the algorithm of [FS05] providing an approximation factor of  $1 + \epsilon$ . In contrast, our result achieves an approximation factor of  $1 + \epsilon$  with a sketch length  $m$  that is as low as  $O(k \log(n/k))$ .

Thanks to this connection, our results also yield short sketches for the  $k$ -median problem. Although the solution  $x^*$  output by our algorithm does not have to be  $k$ -sparse (i.e., we might output more than  $k$  medians), one can post-process the output by computing the best  $k$ -sparse approximation to  $x^*$  using any off-the-shelf (weighted)  $k$ -median algorithm (e.g., [HPM04]). This reduces the number of clusters to  $k$ ,

<sup>3</sup>In fact, the algorithm in [RTG00] vaguely resembles our approach, in that it uses a kd-tree decomposition to partition the images.

<sup>4</sup>For completeness, in our context the  $k$ -median is defined as follows. First, each pixel  $p \in [\Delta]^2$  is interpreted as a point with weight  $x_p$ . The goal is to find a set  $C \subset [n]^2$  of  $k$  “medians” that minimizes the objective function  $\sum_{p \in [n]^2} \min_{c \in C} \|p - c\|_1 x_p$ .

<sup>5</sup>If the algorithm reports both the medians and the weights of clusters.

<sup>6</sup>The paper [Ind04] claims  $m = k \log^{O(1)} n$ . Unfortunately, that is an error, caused by ignoring the dependencies between the queries and their answers provided by the randomized data structure MEDIEVAL. Fixing this problem requires reducing the probability of failure of the algorithm so that it is inversely exponential in  $k$ , which yields another factor of  $k$  in the space bound.

while (by the triangle inequality of EMD) multiplying the approximation factor by a constant that depends on the approximation constant of the chosen  $k$ -median algorithm. See Appendix C for more details.

**Techniques** On a high level, our approach is to reduce the sparse recovery problem under EMD to sparse recovery under  $\ell_1$ . This is done by constructing a linear mapping  $P$  that maps  $\mathbb{R}^{[\Delta]^2}$  into some space  $\mathbb{R}^t$ , that has the property that a “good” sparse approximation to  $y = Px$  under  $\ell_1$  yields a “good” sparse approximation to  $x$  under EMD.<sup>7</sup> The list of formal constraints that such a mapping needs to satisfy are given in Section 3. For concreteness, we define one such mapping below; another one is given in Section 7. Specifically, the *pyramid* mapping  $P$  [IT03, GD05] (building on [Cha02, AV99]) is defined as follows. First we impose  $\log \Delta + 1$  nested grids  $G_i$  on  $[\Delta]^2$ , with  $G = \bigcup G_i$ . For each *level*  $i = 0 \dots l$ ,  $l = \log_2 \Delta$ , the grid  $G_i$  is a partition of the image into *cells* of side length  $2^i$ . The cells in the grids can be thought of as forming a 4-ary tree, with each node  $c$  at level  $i$  having a set  $C(c)$  of children at level  $i - 1$ . For each  $i$ , we define a mapping  $P_i$  such that each entry in  $P_i x$  corresponds to a cell  $c$  in  $G_i$ , and its value is equal to the sum of coordinates of  $x$  falling into  $c$ . The final mapping  $P$  is defined as

$$Px = [2^0 P_0 x, 2^1 P_1 x, \dots, 2^l P_l x] \quad (3)$$

It is easy to see that, for a vector  $x$  that is  $k$ -sparse, the vector  $Px$  is  $O(K)$  sparse for  $K = kl$ . We also show that for *any*  $x$ , there exists an  $O(K)$ -sparse  $y$  such that the difference  $\|y - Px\|_1$  is comparable to  $\min_{k\text{-sparse } x'} \|x - x'\|_{EMD}$ . We then find a good approximation  $x^*$  to  $x$  (in the EMD norm) by “inverting”  $P$  on  $y$ . Since we can recover an  $O(K)$ -sparse approximation to  $y$  (in the  $\ell_1$  norm) from a sketch of length  $O(K \log(n/K))$ , we obtain the first result from Figure 1.

To improve the sketch length we exploit the particular properties of the mapping  $P$  to recover an  $O(K)$ -sparse approximation from only  $O(K)$  measurements. For any non-negative vector  $x$ , the coordinates of  $Px$  have the following hierarchical structure: (i) the coordinates are organized into an  $r$ -ary tree for  $r = 4$ , and (ii) the value of each internal node is non-negative and equal to the sum of its children *times two*. Using one or both of these properties enables us to reduce the number of measurements.

The second algorithm from Figure 1 is obtained using the property (i) alone. Specifically, the problem of recovering a sparse approximation whose support forms a tree has been well-studied in signal processing (the question is motivated by an empirical observation that large wavelet coefficients tend to co-occur in this fashion). In particular, the insightful paper [BCDH10] on *model-based compressive sensing* (see Section 5 for an overview) gave a deterministic scheme that recovers such approximation from a sketch of length  $O(K)$ . Although the setup given in that paper is somewhat different from what we need here, we show that one can modify and re-analyze their scheme to achieve the desired guarantee. This approach, however, leads to an approximation factor of  $O(\sqrt{\log(n/k)})$ .

In order to achieve a constant approximation factor, we employ both properties (i) and (ii), as well as randomization. Specifically, we recover the tree coefficients top-down, starting from the root of the tree. This is done in a greedy manner: we only recurse on the children of nodes that are estimated to be “heavy”. This first pass identifies a superset  $S$  of the locations where  $Px$  is large, but estimates some of the values  $(Px)_S$  quite poorly. The set of locations  $S$  has  $|S| = O(K)$ , so we can recover  $(Px)_S$  accurately with  $O(K)$  measurements using the *set query sketches* of [Pri11].

Finally, we show that we can achieve the first and second result in Figure 1 by replacing the pyramid mapping by a variant of an even more basic transform, namely the (two-dimensional) Haar wavelet mapping. Our variant is obtained by rescaling the original Haar wavelet vectors using exponential weights, to mimic the pyramid scheme behavior. This result relates the two well-studied notions (EMD and wavelets) in a

<sup>7</sup>We note that the aforementioned  $k$ -median algorithms implicitly rely on some form of sparse recovery (e.g., see Remark 3.10 in [FS05] or remarks before Theorem 5 in [Ind04]). However, the bounds provided by those algorithms fall short of what we aim for.

somewhat unexpected way. As a bonus, it also simplifies the algorithms, since inverting the wavelet mapping can now be done explicitly and losslessly.

## 2 Preliminaries

**Notation** We use  $[n]$  to denote the set  $\{1 \dots n\}$ . For any set  $S \subset [n]$ , we use  $\overline{S}$  to denote the complement of  $S$ , i.e., the set  $[n] \setminus S$ . For any  $x \in \mathbb{R}^n$ ,  $x_i$  denotes the  $i$ th coordinate of  $x$ , and  $x_S$  denotes the vector  $x' \in \mathbb{R}^n$  given by  $x'_i = x_i$  if  $i \in S$ , and  $x'_i = 0$  otherwise. We use  $\text{supp}(x)$  to denote the support of  $x$ . We use  $\mathbb{R}^{[\Delta]^2}$  to denote the set of functions from  $[\Delta] \times [\Delta]$  to  $\mathbb{R}$ ; note that  $\mathbb{R}^{[\Delta]^2}$  can be identified with  $\mathbb{R}^n$  since  $n = \Delta^2$ . We also use  $\mathbb{R}_+$  to denote  $\{x \in \mathbb{R} \mid x \geq 0\}$ .

**EMD** Consider any two non-negative vectors  $x, y \in \mathbb{R}_+^{[\Delta]^2}$  such that  $\|x\|_1 = \|y\|_1$ . Let  $\Gamma(x, y)$  be the set of functions  $\gamma : [\Delta]^2 \times [\Delta]^2 \rightarrow \mathbb{R}_+$ , such that for any  $i, j \in [\Delta]^2$  we have  $\sum_l \gamma(i, l) = x_i$  and  $\sum_l \gamma(l, j) = y_j$ ; that is,  $\Gamma$  is the set of possible “flows” from  $x$  to  $y$ . Then we define

$$\text{EMD}^*(x, y) = \min_{\gamma \in \Gamma} \sum_{i, j \in [\Delta]^2} \gamma(i, j) \|i - j\|_1$$

to be the min cost flow from  $x$  to  $y$ , where the cost of an edge is its  $\ell_1$  distance. This induces a norm  $\|\cdot\|_{EMD}$  such that  $\|x - y\|_{EMD} = \text{EMD}^*(x, y)$ . For general vectors  $w$ ,

$$\|w\|_{EMD} = \min_{\substack{x-y+z=w \\ \|x\|_1=\|y\|_1 \\ x, y \geq 0}} \text{EMD}^*(x, y) + D \|z\|_1$$

where  $D = 2\Delta$  is the diameter of the set  $[\Delta]^2$ . That is,  $\|w\|_{EMD}$  is the min cost flow from the positive coordinates of  $w$  to the negative coordinates, with some penalty for unmatched mass.

**Signal models** The basic idea of the signal models framework of [BCDH10] is to restrict the sparsity patterns of the approximations. For some sparsity parameter<sup>8</sup>  $K$  let  $\mathcal{S}_K$  be a family of subsets of  $[n]$  such that for each  $S \in \mathcal{S}_K$  we have  $|S| \leq K$ . The family  $\mathcal{S}_K$  induces a *signal model*  $\mathcal{M}_K \subset \mathbb{R}^n$  where

$$\mathcal{M}_K = \{x \in \mathbb{R}^n \mid \text{supp}(x) \subseteq S \text{ for some } S \in \mathcal{S}_K\}.$$

Note that  $\mathcal{M}_K$  is a union of  $|\mathcal{S}_K|$  subspaces, each of dimension at most  $K$ . The signals in  $\mathcal{M}_K$  are called  *$\mathcal{M}_K$ -sparse*.

The following two examples of signal models are particularly relevant to our paper:

1. General  $k$ -sparse signals, where  $\mathcal{S}_k$  contains all  $k$ -subsets of  $[n]$ . In this case the induced signal model (denoted by  $\Sigma_k$ ) contains all  $k$ -sparse signals.
2. Tree sparse signals. In this case, we assume that  $n = \frac{c^l - 1}{c - 1}$  for some (constant) integer  $c$  and parameter  $l$ , and associate each  $i \in [n]$  with a node of a full  $c$ -ary tree  $T(c, l)$  of depth  $l$ . The family  $\mathcal{S}_K$  contains all sets  $S$  of size up to  $K$  that are connected in  $T(c, l)$  and contain the root (so each  $S$  corresponds to a graph-theoretic subtree of  $T(c, l)$ ). The induced signal model is denoted by  $\mathcal{T}_K^c$ , or  $\mathcal{T}_K$  for short.<sup>9</sup>

<sup>8</sup>We use  $K$  to denote the sparsity in the context of model-based recovery (as opposed to  $k$ , which is used in the context of “standard” recovery).

<sup>9</sup>We note that technically this model was originally defined with respect to the wavelet basis (as opposed to the standard basis here) and for  $c = 2$ . We adapt that definition to the needs in our paper.

In order to facilitate signal recovery, one often needs to consider the differences  $x - y$  of two signals  $x \in \mathcal{M}$ ,  $y \in \mathcal{M}'$ . For this purpose we define the Minkowski sum of  $\mathcal{M}_K$  and  $\mathcal{M}'_K$  as  $\mathcal{M}_K \oplus \mathcal{M}'_K = \{x + y : x \in \mathcal{M}_K, y \in \mathcal{M}'_K\}$ . To simplify the notation, we define  $\mathcal{M}^{(t)}$  to be the  $t$ -wise Minkowski sum of  $\mathcal{M}_K$ . For all signal models considered in this paper, we have  $\mathcal{M}_K^{(t)} \subset \mathcal{M}_{Kt}$ .

Restricting sparsity patterns enables to recover sparse approximations from shorter sketches. We defer a more thorough overview of the results to Section 5.

**Assumptions** We assume that the sparsity parameters  $k$  (and  $K$ , where applicable) are smaller than  $n/2$ . Note that if this assumption does not hold, the problem becomes trivial, since one can define the measurement matrix  $A$  to be equal to the identity matrix.

### 3 Framework for EMD-sparse recovery

In this section we describe our approach to reducing sparse recovery under EMD into sparse recovery under  $\ell_1$ . We need the following three components: (i) a  $t \times n$  matrix  $B$  (that will be used to map the EMD space into the  $\ell_1$  space); (ii) a signal model  $\mathcal{M} \subset \mathbb{R}^t$ ; and (iii) an  $\ell_1/\ell_1$  recovery scheme for  $\mathcal{M}$ . The latter involves an  $m \times t$  matrix  $A'$  (or a distribution over such matrices) such that, for any  $x \in \mathbb{R}^t$ , given  $A'x$ , one can recover  $x^*$  such that

$$\|x - x^*\|_1 \leq C' \min_{x' \in \mathcal{M}} \|x - x'\|_1 \quad (4)$$

for an approximation factor  $C'$ . If  $A'$  is a distribution over matrices, we require that the guarantee holds with some constant probability, e.g.,  $2/3$ .

The mapping  $B$  must satisfy the following three properties:

A. (EMD-to- $\ell_1$  expansion.) For all  $x \in \mathbb{R}^n$ ,

$$\|x\|_{EMD} \leq \|Bx\|_1.$$

B. (Model-alignment of EMD with  $\mathcal{M}$ .) For all  $x \in \mathbb{R}_+^n$ , there exists a  $y \in \mathcal{M}$  with

$$\|y - Bx\|_1 \leq \epsilon \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}.$$

C. (Invertibility.) There is an efficient algorithm  $\mathcal{B}^{-1}: \mathbb{R}^t \rightarrow \mathbb{R}^n$  such that, for some constant  $D$  and all  $y \in \mathbb{R}^t$ ,

$$\|y - B\mathcal{B}^{-1}(y)\|_1 \leq D \min_{x \in \mathbb{R}^n} \|y - Bx\|_1.$$

**Lemma 3.1.** *Consider  $B, A', \mathcal{M}$  satisfying the above properties. Then the matrix  $A = A'B$  supports  $k$ -sparse recovery for EMD (as defined in Equation (2)) with approximation factor  $C = (1 + D)C'\epsilon$ .*

*Proof.* Consider the recovery of any vector  $x \in \mathbb{R}_+^n$ . Let

$$E = \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}.$$

By Property B, for any  $x \in \mathbb{R}^n$ , there exists a  $y \in \mathcal{M}$  with

$$\|y - Bx\|_1 \leq \epsilon E.$$

Hence our  $\ell_1/\ell_1$  model-based recovery scheme for  $\mathcal{M}$ , when run on  $Ax = A'Bx$ , returns a  $y^*$  with

$$\|y^* - Bx\|_1 \leq C'\epsilon E.$$

Let  $x^* = \mathcal{B}^{-1}(y^*)$ . We have by Property C that

$$\|y^* - Bx^*\|_1 \leq D \min_{x' \in \mathbb{R}^n} \|y^* - Bx'\|_1 \leq D \|y^* - Bx\|_1 \leq DC'\epsilon E.$$

Hence by Property A

$$\begin{aligned} \|x^* - x\|_{EMD} &\leq \|B(x^* - x)\|_1 \leq \|y^* - Bx\|_1 + \|y^* - Bx^*\|_1 \\ &\leq (1 + D)C'\epsilon E \end{aligned}$$

as desired. □

## 4 Pyramid transform

In this section we will show that the pyramid transform  $P$  defined in Equation (3) of Section 1 satisfies properties B and C of Section 3, with appropriate parameters.

The property A has been shown to hold for  $P$  in many other papers (e.g., [Cha02, IT03]). The intuition is that the weight of a cell is at least the Earth-Mover Distance to move all mass in the cell from the center to any corner of the cell, including the corner that is at the center of the parent of the cell.

### 4.1 Model-alignment with tree sparsity

In this section we show Property B, where the signal model  $\mathcal{M}$  is equal to the  $K$ -tree-sparse model  $\mathcal{T}_K$ , for  $K = O(k \log(n/k))$ . In fact, we show a stronger statement: the trees have their *width* (the maximum number of nodes per level) bounded by some parameter  $s$ . We will exploit the latter property later in the paper.

**Lemma 4.1.** *For any  $x \in \mathbb{R}_+^n$  there exists a tree  $S \subset [t]$  of size  $K$  and width  $s$  with*

$$\|(Px)_{\bar{S}}\|_1 \leq \epsilon \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}$$

for  $s = O(\frac{1}{\epsilon^2}k)$  and  $K = O(\frac{1}{\epsilon^2}k \log(n/k))$ .

*Proof.* Let  $x' = \arg \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}$  be the  $k$ -medians approximation of  $x$ . Consider the cells that contain each point in the support of  $x'$ . For each such cell at any level  $i$ , add the  $O(\frac{1}{\epsilon^2})$  other cells of the same level within an  $\ell_1$  distance of  $\frac{2}{\epsilon}2^i$ . The resulting  $S$  has  $s = O(\frac{1}{\epsilon^2}k)$  cells per level, and all the ancestors of any cell in the result also lie in  $S$ . So  $S$  is a tree of width  $s$ . It has  $O(s)$  elements from the top  $\log_4 s$  levels, and  $O(s)$  elements on each of the  $\log_4 t - \log_4 s$  remaining levels, for a size  $K = O(s \log t/s)$ . We will show that  $\|(Px)_{\bar{S}}\|_1$  is small.

Define  $e_i$  for  $i \in [\Delta]^2$  to be the elementary vector with a 1 at position  $i$ , so  $x_i = x \cdot e_i$ . Suppose that the distance between  $i$  and the nearest center in  $x'$  is  $v_i$ . Then we have

$$\begin{aligned} \|(Px)_{\bar{S}}\|_1 &= \sum_{i \in [\Delta]^2} \|(Px_i e_i)_{\bar{S}}\|_1 = \sum_{i \in [\Delta]^2} \|(Pe_i)_{\bar{S}}\|_1 x_i \\ \|x - x'\|_{EMD} &= \sum_{i \in [\Delta]^2} v_i x_i. \end{aligned}$$

so it is sufficient to show  $\|(Pe_i)_{\bar{S}}\|_1 \leq \epsilon v_i$  for any  $i$ .

Let  $h$  be the highest level such that  $e_i$  is not contained in a cell at level  $h$  in  $S$ . If no such  $h$  exists,  $\|(Pe_i)_{\bar{S}}\|_1 = 0$ . Otherwise,  $v_i \geq \frac{2}{\epsilon} 2^h$ , or else  $S$  would contain  $e_i$ 's cell in level  $h$ . But then

$$\|(Pe_i)_{\bar{S}}\|_1 = \sum_{j=0}^h 2^j < 2^{h+1} \leq \epsilon v_i$$

as desired. □

**Corollary 4.2.** *For any  $x \in \mathbb{R}_+^n$ , there exists a  $y \in \mathcal{T}_K$  with*

$$\|y - Px\|_1 \leq \epsilon \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}.$$

## 4.2 Invertibility

Given an approximation  $b$  to  $Px$ , we would like to find a vector  $y$  with  $\|b - Py\|_1$  small. Note that this task can be formulated as a linear program, and therefore solved in time that is polynomial in  $n$ . In Appendix A we show a much faster *approximate* algorithm for this problem, needed for our fast recovery algorithm:

**Lemma 4.3.** *Given any approximation  $b$  to  $Px$ , we can recover a  $y$  in  $O(|\text{supp}(b)|)$  time with*

$$\|Py - Px\|_1 \leq 8 \|b - Px\|_1.$$

Recall that  $P$  has  $t = \lfloor 4n/3 \rfloor$  rows. This means standard  $\ell_1/\ell_1$   $K$ -sparse recovery for  $Px$  is possible with  $m = O(K \log t/K) = O(\frac{1}{2}k \log^2(n/k))$ . Hence by Lemma 3.1, using  $B = P$  and standard sparse recovery techniques on the model  $\mathcal{M} = \Sigma_K$  gives the first result in Figure 1:

**Theorem 4.4.** *There exists a deterministic EMD/EMD recovery scheme with  $m = O(\frac{1}{2}k \log^2(n/k))$  and  $C = \epsilon$ . Recovery takes  $O(n \log^c n)$  time for some constant  $c$ .*

## 5 Tree-sparse recovery

To decrease the number of measurements required by our algorithm, we can use the stronger signal model  $\mathcal{T}_K$  instead of  $\Sigma_K$ . The paper [BCDH10] gives an algorithm for model-based sparse recovery of  $\mathcal{T}_K$ , but their theorem does not give an  $\ell_1/\ell_1$  guarantee. In Appendix B we review the prior work and convert their theorem into the following:

**Theorem 5.1.** *There exists a matrix  $A$  with  $O(K)$  rows and a recovery algorithm that, given  $Ax$ , returns  $x^*$  with*

$$\|x - x^*\|_1 \leq C \sqrt{\log(n/K)} \min_{x' \in \mathcal{T}_K} \|x - x'\|_1$$

for some absolute constant  $C > 1$ . As long as the coefficients of  $x$  are integers bounded by  $n^{O(1)}$ , the algorithm runs in time  $O(K^2 n \log^c n)$  for some constant  $c$ .

By Lemma 3.1, using this on  $B = P$  and  $\mathcal{M} = \mathcal{T}_K$  gives the second result in Figure 1:

**Theorem 5.2.** *There exists a deterministic EMD/EMD recovery scheme with  $m = O(\frac{1}{2}k \log(n/k))$  and distortion  $C = O(\epsilon \sqrt{\log(n/k)})$ . Recovery takes  $O(k^2 n \log^c n)$  time for some constant  $c$ .*



## 6 Beyond tree sparsity

The previous section achieved  $O(\sqrt{\log n})$  distortion deterministically with  $O(k \log(n/k))$  rows. In this section, we improve the distortion to an arbitrarily small constant  $\epsilon$  at the cost of making the algorithm randomized. To do this, we show that EMD under the pyramid transform is aligned with a stronger model than just tree sparsity—the model can restrict the values of the coefficients as well as the sparsity pattern. We then give a randomized algorithm for  $\ell_1/\ell_1$  recovery in this model with constant distortion.

**Definition 6.1.** Define  $T_K^s$  to be the family of sets  $S \subseteq [t]$  such that (i)  $S$  corresponds to a connected subset of  $G$  containing the root and (ii)  $|S \cap G_i| \leq s$  for all  $i$ . We say that such an  $S$  is  $K$ -tree-sparse with width  $s$ .

**Definition 6.2.** Define  $\mathcal{M} \subset \mathcal{T}_K$  as

$$\mathcal{M} = \left\{ y \in \mathbb{R}^t \mid \begin{array}{l} \text{supp}(y) \subseteq S \text{ for some } S \in T_K^s, \text{ and} \\ y_i \geq 2 \|y_{C(i)}\|_1 \forall i \in [t] \end{array} \right\}.$$

where  $s = O(\frac{1}{\epsilon^2}k)$  comes from Lemma 4.1.

Note that every  $y \in \mathcal{M}$  is non-negative, and  $(Px)_S \in \mathcal{M}$  for all  $x \in \mathbb{R}_+^n$ . With Lemma 4.1, this implies:

**Lemma 6.3.** There is model-alignment of  $P$  with  $\mathcal{M}$ , i.e., they satisfy Property B.

We will give a good algorithm for  $\ell_1/\ell_1$  recovery over  $\mathcal{M}$ .

### 6.1 Randomized $\ell_1/\ell_1$ recovery of $\mathcal{M}$

**Theorem 6.4.** There is a randomized distribution over  $m \times t$  matrices  $A$  with  $m = O(\frac{1}{\epsilon^2}k \log(n/k))$  and an algorithm that recovers  $y^*$  from  $Ay$  in  $O(\frac{1}{\epsilon^2}k \log(n/k))$  time with

$$\|y^* - y\|_1 \leq C \min_{y' \in \mathcal{M}} \|y - y'\|_1$$

with probability  $1 - k^{-\Omega(1)}$ , for some constant  $C$ . We assume  $k = \Omega(\log \log n)$ .

We will give an algorithm to estimate the support of  $y$ . Given a sketch of  $y$ , it recovers a support  $S \in T_K^{2s}$  with

$$\|y_{\bar{S}}\|_1 \leq 10 \min_{y' \in \mathcal{M}} \|y - y'\|_1.$$

We can then use the set query algorithm [Pri11] to recover a  $y^*$  from a sketch of size  $O(|S|)$  with

$$\|y^* - y_S\|_1 \leq \|y_{\bar{S}}\|_1.$$

Then

$$\|y^* - y\|_1 \leq \|y^* - y_S\|_1 + \|y - y_S\|_1 \leq 2 \|y_{\bar{S}}\|_1 \leq 20 \min_{y' \in \mathcal{M}} \|y - y'\|_1.$$

as desired. Hence estimating the support of  $y$  is sufficient.

## 6.2 Finding a good sparse support $S$ to $y$

Vectors  $y' \in \mathcal{M}$  have two properties that allow us to find good supports  $S \in T_K^s$  with constant distortion using only  $O(|S|)$  rows. First,  $\text{supp}(y')$  forms a tree, so the support can be estimated from the top down, level by level. Second, each coefficient has value at least twice the sum of the values of its children. This means that the cost of making a mistake in estimating the support (and hence losing the entire subtree below the missing coefficient) is bounded by twice the weight of the missing coefficient. As a result, we can bound the global error in terms of the local errors made at each level.

Of course,  $y$  may not be in  $\mathcal{M}$ . But  $y$  is “close” to some  $y' \in \mathcal{M}$ , so if our algorithm is “robust”, it can recover a good support for  $y$  as well. Our algorithm is described in Algorithm 6.1.

**Definition of sketch matrix  $A$ .** The algorithm is parameterized by a width  $s$ . Let  $h_i$  be a random hash function from  $G_i$  to  $O(s)$  for  $i \in [\log(n/s)]$ . Then define  $A'(i)$  to be the  $O(s) \times |G_i|$  matrix representing  $h_i$ , so  $A'(i)_{ab} = 1$  if  $h_i(b) = a$  and 0 otherwise. Choose  $A$  to be the vertical concatenation of the  $A'(i)$ 's.

**Recovery procedure.**

```

procedure FINDSUPPORT( $A, b$ )
     $T_{\log(n/s)} \leftarrow G_{\log(n/s)}$ 
    for  $i = \log(n/s) - 1 \dots 0$  do
         $y_j^* \leftarrow b_{h_i(j)}$  for  $j \in C(T_{i+1})$ .
         $T_i \leftarrow \arg \max_{\substack{T' \subset C(T_{i+1}) \\ |T'| \leq 2s}} \|y_{T'}^*\|_1$ 
    end for
     $S \leftarrow \bigcup_{i=0}^{\log(n/s)} T_i \cup \bigcup_{i \geq \log(n/s)} G_i$ 
end procedure

```

▷ Find approximate support  $S$  to  $y$  from  $b = Ay$

▷  $|T_{\log(n/s)}| \leq 2s$

▷ Estimate  $y$  over  $C(T_{i+1})$ .

▷ Select the  $2s$  largest elements of our estimate.

**Algorithm 6.1:** Finding sparse support under  $\mathcal{M}$

**Lemma 6.5.** *Algorithm 6.1 uses a binary sketching matrix of  $O(s \log(n/s))$  rows and takes  $O(s \log(n/s))$  time to recover  $S$  from the sketch.*

*Proof.* The algorithm looks at  $O(\log(n/s))$  levels. At each level it finds the top  $2s$  of  $4 \times 2s$  values, which can be done in linear time. The algorithm requires a sketch with  $O(\log(n/s))$  levels of  $O(s)$  cells each.  $\square$

The algorithm estimates the value of  $y_{C(T_{i+1})}$  by hashing all of  $y_{G_i}$  into an  $O(s)$  size hash table, then estimating  $y_j$  as the value in the corresponding hash table cell. Since  $y$  is non-negative, this is an overestimate. We would like to claim that the  $2s$  largest values in our estimate approximately contain the  $s$  largest values in  $y_{C(T_{i+1})}$ . In particular, we show that any  $y_j$  we miss is either (i) not much larger than  $s$  of the coordinates we do output or (ii) very small relative to the coordinates we already missed at a previous level.

**Lemma 6.6.** *In Algorithm 6.1, for every level  $i$  let  $w_i = \max_{q \in C(T_{i+1}) \setminus T_i} y_q$  denote the maximum value that is skipped by the algorithm and let  $f_i = \|y_{G_{i+1} \setminus T_{i+1}}\|_1$  denote the error from coordinates not included in  $T_{i+1}$ . Let  $c_i$  denote the  $s$ -th largest value in  $y_{T_i}$ . Then with probability at least  $1 - e^{-\Omega(s)}$ ,  $w_i \leq \max\{\frac{f_i}{4s}, 2c_i\}$  for all levels  $i$ .*

*Proof.* Define  $s' = 8s \geq |C(T_{i+1})|$ . We make the hash table size at each level equal to  $u = 32s'$ . We will show that, with high probability, there are at most  $s$  coordinates  $p$  where  $y_p^*$  is more than  $f_i/s'$  larger than  $y_p$ . Once this is true, the result comes as follows:  $y^*$  is an overestimate, so the top  $2s$  elements of  $y^*$  contain at least  $s$  values that have been overestimated by at most  $f_i/s'$ . Because the algorithm passes over an element of value  $w_i$ , each of these  $s$  values must actually have value at least  $w_i - f_i/s'$ . Hence either  $w_i < 2f_i/s' = \frac{f_i}{4s}$  or all  $s$  values are at least  $w_i/2$ .

To bound the number of badly overestimated coordinates, we split the noise in two components: the part from  $G_i \setminus C(T_{i+1})$  and the part from  $C(T_{i+1})$ . We will show that, with probability  $1 - e^{-\Omega(s)}$ , the former is at most  $f_i/s'$  in all but  $s/4$  locations and the latter is zero in all but  $3s/4$  locations.

WLOG we assume that the function  $h_i$  is first fixed for  $G_i \setminus C(T_{i+1})$ , then randomly chosen for  $C(T_{i+1})$ . Let  $O_i \subset [u]$  be the set of ‘‘overflow buckets’’  $l$  such that the sum  $s_l = \sum_{p \in C(T_{i+1}), h_i(p)=l} y_p$  is at least  $f_i/s'$ . By the definition of  $f_i$ ,  $\sum_l s_l = f_i/2$ , so

$$|O_i|/u \leq \frac{f_i/2}{f_i/s'}/u = 1/2 \frac{s'}{32s'} = 1/64.$$

Thus, the probability that a fixed child  $q \in C(T_{i+1})$  is mapped to  $O_i$  is at most  $1/64$ . This is independent over  $C(T_{i+1})$ , so the Chernoff bound applies. Hence with probability at least  $1 - e^{-\Omega(s)}$ , the number of  $q \in C(T_{i+1})$  mapping to  $O_i$  is at most twice its expectation, or  $|C(T_{i+1})|/32 = s/4$ .

We now bound the collisions within  $C(T_{i+1})$ . Note that our process falls into the ‘‘balls into bins’’ framework, but for completeness we will analyze it from first principles.

Let  $Z$  be the number of cells in  $C(T_{i+1})$  that collide.  $Z$  is a function of the independent random variables  $h_i(p)$  for  $p \in C(T_{i+1})$ , and  $Z$  changes by at most 2 if a single  $h_i(p)$  changes (because  $p$  can cause at most one otherwise non-colliding element to collide). Hence by McDiarmid’s inequality,

$$\Pr[Z \geq \mathbb{E}[Z] + t] \leq e^{-t^2/(2s')}$$

But we know that the chance that a specific  $p$  collides with any of the others is at most  $s'/u = 1/32$ . Hence  $\mathbb{E}[Z] \leq s'/32$ , and

$$\Pr[Z \geq (\frac{1}{32} + \epsilon)s'] \leq e^{-\epsilon^2 s'/2}.$$

By setting  $\epsilon = 2/32$  we obtain that, with probability  $1 - e^{-\Omega(s)}$  we have that  $Z \leq \frac{3s'}{32} = 3s/4$ .

Hence with probability  $1 - e^{-\Omega(s)}$ , only  $3s/4$  locations have non-zero corruption from  $C(T_{i+1})$ , and we previously showed that with the same probability only  $s/4$  locations are corrupted by  $f'/s'$  from outside  $C(T_{i+1})$ . By the union bound, this is true for all levels with probability at least  $1 - (\log n)e^{-\Omega(s)} = 1 - e^{-\Omega(s)}$ .  $\square$

**Lemma 6.7.** *Let  $S$  be the result of running Algorithm 6.1 on  $y \in \mathbb{R}^t$ . Then*

$$\|y_{\bar{S}}\|_1 \leq 10 \min_{y' \in \mathcal{M}} \|y - y'\|_1$$

*with probability at least  $1 - e^{-\Omega(s)}$ .*

*Proof.* From the algorithm definition,  $T_i = S \cap G_i$  for each level  $i$ . Let  $y' \in \mathcal{M}$  minimize  $\|y - y'\|_1$ , and let  $U = \text{supp}(y')$ . By the definition of  $\mathcal{M}$ ,  $U \in T_K^s$ .

For each  $i$ , define  $V_i = U \cap C(T_{i+1}) \setminus T_i$  to be the set of nodes in  $U$  that could have been chosen by the algorithm at level  $i$  but were not. For  $q \in U \setminus S$ , define  $R(q)$  to be the highest ancestor of  $q$  that does not lie

in  $S$ ; hence  $R(q)$  lies in  $V_i$  for some level  $i$ . Then

$$\begin{aligned}
\|y'_{\bar{S}}\|_1 &= \|y'_{U \setminus S}\|_1 = \sum_{q \in U \setminus S} y'_q \\
&= \sum_i \sum_{p \in V_i} \sum_{R(q)=p} y'_q \\
&\leq \sum_i \sum_{p \in V_i} 2y'_p \\
&= 2 \sum_i \|y'_{V_i}\|_1, \tag{5}
\end{aligned}$$

where the inequality holds because each element of  $y'$  is at least twice the sum of its children. Hence the sum of  $y'$  over a subtree is at most twice the value of the root of the subtree.

Define the error term  $f_i = \|y_{G_{i+1} \setminus T_{i+1}}\|_1$ , and suppose that the statement in Lemma 6.6 applies, as happens with probability  $1 - e^{-\Omega(s)}$ . Then for any level  $i$  and  $p \in V_i$ , if  $c_i$  is the  $s$ th largest value in  $y_{T_i}$ , then  $y_p \leq \max\{f_i/4s, 2c_i\}$  or  $y_p \leq \frac{f_i}{4s} + 2c_i$ . Since  $y_{T_i}$  contains at least  $s$  values larger than  $c_i$ , and at most  $|U \cap T_i| = |U \cap C(T_{i+1})| - |V_i| \leq s - |V_i|$  of them lie in  $U$ ,  $y_{T_i \setminus U}$  must contain at least  $|V_i|$  values larger than  $c_i$ . This, combined with  $|V_i| \leq s$ , gives

$$\|y_{V_i}\|_1 \leq f_i/4 + 2 \|y_{T_i \setminus U}\|_1. \tag{6}$$

Combining Equations (5) and (6), we get

$$\begin{aligned}
\|y'_{\bar{S}}\|_1 &\leq 2 \left[ \sum_i \|(y - y')_{V_i}\|_1 + \|y_{V_i}\|_1 \right] \\
&\leq 2 \|(y - y')_U\|_1 + \sum_i \left( 4 \|y_{T_i \setminus U}\|_1 + f_i/2 \right) \\
&\leq 2 \|(y - y')_U\|_1 + 4 \|y_{S \setminus U}\|_1 + \|y_{\bar{S}}\|_1 / 2 \\
&= 2 \|(y - y')_U\|_1 + 4 \|(y - y')_{S \setminus U}\|_1 + \|y_{\bar{S}}\|_1 / 2 \\
&\leq 4 \|y - y'\|_1 + \|y_{\bar{S}}\|_1 / 2.
\end{aligned}$$

Therefore

$$\begin{aligned}
\|y_{\bar{S}}\|_1 &\leq \|y - y'\|_1 + \|y'_{\bar{S}}\|_1 \\
&\leq 5 \|y - y'\|_1 + \|y_{\bar{S}}\|_1 / 2 \\
\|y_{\bar{S}}\|_1 &\leq 10 \|y - y'\|_1
\end{aligned}$$

as desired. □

### 6.3 Application to EMD recovery

By Lemma 3.1 our  $\ell_1/\ell_1$  recovery algorithm for  $\mathcal{M}$  gives an EMD/EMD recovery algorithm.

**Theorem 6.8.** *Suppose  $k = \Omega(\log \log n)$ . There is a randomized EMD/EMD recovery scheme with  $m = O(\frac{1}{\epsilon^2} k \log(n/k))$ ,  $C = \epsilon$ , and success probability  $1 - k^{-\Omega(1)}$ . Recovery takes  $O(\frac{1}{\epsilon^2} k \log(n/k))$  time.*

## 7 Wavelet-based method

We can also instantiate the framework of Section 3 using a reweighted Haar wavelet basis instead of  $P$  for the embedding  $B$ . We will have  $\mathcal{M}$  be the tree-sparse model  $\mathcal{T}_{O(\frac{1}{2}k \log n/k)}$ , and use the  $\ell_1/\ell_1$  recovery scheme of Section 5.

The details are deferred to Appendix D. We obtain an embedding  $W$  defined by a Haar transform  $H$  (after rescaling the rows), and the following theorem:

**Theorem 7.1.** *There exists a matrix  $A$  with  $O(k \log(n/k))$  rows such that we can recover  $x^*$  from  $Ax$  with*

$$\|x^* - x\|_{EMD} \leq C \min_{y \in \mathcal{T}_K} \|Wx - y\|_1 \leq C \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}$$

for some distortion  $C = O(\sqrt{\log(n/k)})$ .

Note that if we ignore the middle term, this gives the same EMD/EMD result as in Section 5. However the middle term may be small for natural images even if the right term is not. In particular, it is well known that images tend to be tree-sparse under  $H$ .

**Acknowledgements** The authors would like to thank Yaron Rachlin from Draper Lab for numerous conversations and the anonymous reviewers for helping clarify the presentation.

## References

- [AV99] P.K. Agarwal and K. Varadarajan. Approximation algorithms for bipartite and non-bipartite matching in the plane. *SODA*, 1999.
- [BCDH10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56, No. 4:1982–2001, 2010.
- [BGI<sup>+</sup>08] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. *Allerton*, 2008.
- [CCFC02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *ICALP*, 2002.
- [CDDD01] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore. Tree approximation and optimal encoding. *Applied and Computational Harmonic Analysis*, 2001.
- [Cha02] M. Charikar. Similarity estimation techniques from rounding. In *STOC*, pages 380–388, 2002.
- [CIHB09] V. Cevher, P. Indyk, C. Hegde, and RG Baraniuk. Recovery of clustered sparse signals from compressive measurements. *SAMPTA*, 2009.
- [CM04] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *Latin*, 2004.
- [CM06] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. *Sirocco*, 2006.
- [CRT06] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.

- [DDT<sup>+</sup>08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008.
- [DIPW10] K. Do Ba, P. Indyk, E. Price, and D. Woodruff. Lower bounds for sparse recovery. *SODA*, 2010.
- [Don06] D. L. Donoho. Compressed Sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, Apr. 2006.
- [FPRU10] S. Foucart, A. Pajor, H. Rauhut, and T. Ullrich. The gelfand widths of lp-balls for  $0 < p \leq 1$ . *preprint*, 2010.
- [FS05] G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. *STOC*, 2005.
- [GD05] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. *ICCV*, 2005.
- [GI10] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of IEEE*, 2010.
- [GIP10] R. Gupta, P. Indyk, and E. Price. Sparse recovery for earth mover distance. *Allerton*, 2010.
- [HPM04] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-medians and their applications. *STOC*, 2004.
- [Ind04] P. Indyk. Algorithms for dynamic geometric problems over data streams. *STOC*, 2004.
- [Ind07] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at <http://stellar.mit.edu/S/course/6/fa07/6.895/>*, 2007.
- [IT03] P. Indyk and N. Thaper. Fast color image retrieval via embeddings. *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.
- [Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [Lyu05] S. Lyu. Mercel kernels for object recognition with local features. *CVPR*, 2005.
- [Mut05] S. Muthukrishnan. Data streams: Algorithms and applications). *Foundations and Trends in Theoretical Computer Science*, 2005.
- [NT08] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Arxiv math.NA 0803.2392*, 2008.
- [Pri11] E. Price. Efficient sketches for the set query problem. *SODA*, 2011.
- [Rom09] J. Romberg. Compressive sampling by random convolution. *SIAM Journal on Imaging Science*, 2009.
- [RTG00] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

For every cell  $q \in G$ , let  $e_q \in \mathbb{R}^n$  denote an elementary unit vector with the 1 located somewhere in  $q$  (for example, at the center of  $q$ ). Then return

$$y = \sum_{q \in G} s_q e_q.$$

**Algorithm A.1:** Recovering  $y$  from  $b$  to minimize  $\|b - Py\|_1$  when all surpluses are non-negative.

## A Invertibility of Pyramid Transform

If  $b$  were  $(Px)_S$  for some  $S$ , then the problem would be fairly easy, since  $b$  tells us the mass  $p_q$  in cells  $q$  (in particular, if  $q$  is at level  $i$ ,  $p_q = \frac{b_q}{2^i}$ ). Define the *surplus*  $s_q = p_q - \sum_{r \in C(q)} p_r$  to be the mass estimated in the cell that is not found in the cell's children.

We start from the case when all surpluses are non-negative (as is the case for  $(Px)_S$ ). In this case, we can minimize  $\|b - Py\|_1$  by creating  $s_q$  mass anywhere in cell  $q$ .

**Lemma A.1.** *Suppose  $b$  is such that  $s_q \geq 0$  for all  $q \in G$ . Let  $y$  be the result of running Algorithm A.1 on  $b$ . Then  $y$  minimizes  $\|b - Py\|_1$ .*

*Proof.* The vector  $y$  has the property that  $(Py)_q \geq b_q$  for all  $q \in G$ , and for the root node  $r$  we have  $(Py)_r = b_r$ . Because the weights are exponential in the level value, any  $y'$  minimizing  $\|b - Py'\|_1$  must have  $(Py')_r \geq b_r$ , or else increasing any coordinate of  $y'$  would decrease  $\|b - Py'\|_1$ . But then

$$\begin{aligned} \|b - Py'\|_1 &= \sum_{i=0}^{\log \Delta} \sum_{q \in G_i} |(Py')_q - b_q| \\ &\geq \sum_{i=0}^{\log \Delta} \sum_{q \in G_i} (Py')_q - b_q \\ &= \sum_{i=0}^{\log \Delta} \left( 2^{i-\log \Delta} (Py')_r - \sum_{q \in G_i} b_q \right) \\ &= (2 - 2^{-\log \Delta}) (Py')_r - \|b\|_1 \\ &\geq (2 - 2^{-\log \Delta}) b_r - \|b\|_1. \end{aligned}$$

Equality holds if and only if  $(Py')_q \geq b_q$  for all  $q \in G$  and  $(Py')_r = b_r$ . Since  $y$  has these properties,  $y$  minimizes  $\|b - Py\|_1$ .  $\square$

Unfortunately, finding the exact solution is harder when some surpluses  $s_q$  may be negative. Then in order to minimize  $\|b - Py\|_1$  one must do a careful matching up of positive and negative surpluses. In order to avoid this complexity, we instead find a greedy 8-approximation. We modify  $b$  from the top down, decreasing values of children until all the surpluses are non-negative.

Perform a preorder traversal of  $G$ . At each node  $q$  at level  $i$ , compute the surplus  $s_q$ . If  $s_q$  is negative, arbitrarily decrease  $b$  among the children of  $q$  by a total of  $2^{i-1} |s_q|$ , so that  $b$  remains non-negative.

**Algorithm A.2:** Modifying  $b$  to form all non-negative surpluses

**Lemma A.2.** *Suppose we run algorithm A.2 on a vector  $b$  to get  $b'$ . Then*

$$\|b - b'\|_1 \leq 3 \min_y \|Py - b\|_1.$$

*Proof.* Let  $y$  minimize  $\|Py - b\|_1$ . As with  $Py'$  for any  $y'$ ,  $Py$  has zero surplus at every node.

At the point when we visit a node  $q$ , we have updated our estimate of  $b$  at  $q$  but not at its children. Therefore if  $q$  is at level  $i$  we compute  $s_q = \frac{1}{2^i}b'_q - \frac{1}{2^{i-1}} \sum_{s \in C(q)} b_s$ . Then, because  $Py$  has zero surplus,

$$\begin{aligned} |s_q| &= \left| \frac{1}{2^i}b'_q - \frac{1}{2^i}(Py)_q - \frac{1}{2^{i-1}} \sum_{s \in C(q)} (b_s - (Py)_s) \right| \\ &\leq \frac{1}{2^i} |b'_q - b_q| + \frac{1}{2^i} |b_q - (Py)_q| + \frac{1}{2^{i-1}} \sum_{s \in C(q)} |b_s - (Py)_s|. \end{aligned}$$

Define  $f_i = \sum_{q \in G_i} |b_q - (Py)_q|$  to be the original  $\ell_1$  error on level  $i$ , and  $g_i = \sum_{q \in G_i} |b'_q - b_q|$  to be a bound on the amount of error we add when running the algorithm. Because we only modify values enough to rectify the surplus of their parent, we have

$$\begin{aligned} g_{i-1} &\leq 2^{i-1} \sum_{q \in G_i} |s_q| \\ &\leq \sum_{q \in G_i} \frac{1}{2} |b'_q - b_q| + \frac{1}{2} |b_q - (Py)_q| + \sum_{s \in C(q)} |b_s - (Py)_s| \\ &\leq \frac{1}{2} g_i + \frac{1}{2} f_i + f_{i-1}. \end{aligned}$$

Unrolling the recursion, we get

$$\begin{aligned} g_i &\leq f_i + \sum_{j=1}^{\log \Delta - i} \frac{1}{2^{j-1}} f_{i+j} \\ \|b' - b\|_1 &= \sum_{i=0}^{\log \Delta} g_i \leq \sum_{i=0}^{\log \Delta} 3f_i = 3 \|Py - b\|_1 \end{aligned}$$

as desired. □

This lets us prove Lemma 4.3.

LEMMA 4.3. *Given any approximation  $b$  to  $Px$ , running the previous two algorithms gives a  $y$  with*

$$\|Py - Px\|_1 \leq 8 \|b - Px\|_1$$

in  $O(|\text{supp}(b)|)$  time.

*Proof.* By running Algorithm A.2 on  $b$ , we get  $b'$  with  $\|b - b'\|_1 \leq 3 \|Px - b\|_1$ . Then we run Algorithm A.1 on  $b'$  to get  $y$  that minimizes  $\|Py - b'\|_1$ . Then

$$\begin{aligned} \|Py - Px\|_1 &\leq \|Py - b'\|_1 + \|Px - b'\|_1 \\ &\leq 2 \|Px - b'\|_1 \\ &\leq 2(\|Px - b\|_1 + \|b' - b\|_1) \\ &\leq 8 \|Px - b\|_1. \end{aligned}$$

To bound the recovery time, note that after Algorithm A.2 visits a node with value 0, it sets the value of every descendant of that node to 0. So it can prune its descent when it first leaves  $\text{supp}(b)$ , and run in  $O(|\text{supp}(b)|)$  time. Furthermore, this means  $|\text{supp}(b')| \leq |\text{supp}(b)|$  and  $\text{supp}(b')$  is a top-down tree. Hence Algorithm A.1 can iterate through the support of  $b'$  in linear time. □



## B Model-based compressive sensing

In this section we first provide a quick review of model-based sparse recovery, including the relevant definitions, algorithms and their guarantees. We then show how to augment the algorithm so that it provides the guarantees that are needed for our EMD algorithms.

### B.1 Background

**Model-based RIP** Given a signal model  $\mathcal{M}_K$ , we can formulate the  $\mathcal{M}_K$ -restricted isometry property ( $\mathcal{M}_K$ -RIP) of an  $m \times n$  matrix  $A$ , which suffices for performing sparse recovery.

**Definition B.1.** A matrix  $A$  satisfies the  $\mathcal{M}_K$ -RIP with constant  $\delta$  if for any  $x \in \mathcal{M}_K$ , we have

$$(1 - \delta) \|x\|_2 \leq \|Ax\|_2 \leq (1 + \delta) \|x\|_2$$

It is known that random Gaussian matrices with  $m = O(k \log(n/k))$  rows satisfy the  $\Sigma_k$ -RIP (i.e., the “standard” RIP), with very high probability, and that this bound cannot be improved [DIPW10]. In contrast, it has been shown that in order to satisfy the  $\mathcal{T}_K$ -RIP, only  $m = O(K)$  rows suffice [BCDH10]. The intuitive reason behind this is that the number of rooted trees of size  $K$  is  $2^{O(K)}$  while the number of sets of size  $k$  is  $\binom{n}{k} = 2^{\Theta(k \log(n/k))}$ .

**Algorithms** Given a matrix  $A$  that satisfies the  $\mathcal{M}_K$ -RIP, one can show how to recover an approximation to a signal from its sketch. The specific theorem (proven in [BCDH10] and re-stated below) considers  $\ell_2$  recovery of a “noisy” sketch  $Ax + e$ , where  $e$  is an arbitrary “noise” vector, while  $x \in \mathcal{M}_K$ . In the next section we will use this theorem to derive an  $\ell_1$  result for a different scenario, where  $x$  is an *arbitrary* vector, and we are given its *exact* sketch  $Ax$ .

**Theorem B.2.** Suppose that a matrix  $A$  satisfies  $\mathcal{M}_K^{(4)}$ -RIP with constant  $\delta < 0.1$ . Moreover, assume that we are given a procedure that, given  $y \in \mathbb{R}^n$ , finds  $y^* \in \mathcal{M}_K$  that minimizes  $\|y - y^*\|_2$ . Then there is an algorithm that, for any  $x \in \mathcal{M}_K$ , given  $Ax + e$ ,  $e \neq 0$ , finds  $x^* \in \mathcal{M}_K$  such that

$$\|x - x^*\|_2 \leq C \|e\|_2$$

for some absolute constant  $C > 1$ . The algorithm runs in time  $O((n + T + MM) \log(\|x\|_2 / \|e\|_2))$ , where  $T$  is the running time of the minimizer procedure, and  $MM$  is the time needed to perform the multiplication of a vector by the matrix  $A$ .

Note that the algorithm in the theorem has a somewhat unexpected property: if the sketch is nearly exact, i.e.,  $e \approx 0$ , then the running time of the algorithm becomes unbounded. The reason for this phenomenon is that the algorithm iterates to drive the error down to  $\|e\|_2$ , which takes longer when  $e$  is small. However, as long as the entries of the signals  $x, x^*$  and the matrix  $A$  have bounded precision, e.g., are integers in the range  $1, \dots, L$ , one can observe that  $O(\log L)$  iterations suffice.

The task of minimizing  $\|y - y^*\|_2$  over  $y^* \in \mathcal{M}_K$  can typically be accomplished in time polynomial in  $K$  and  $n$ . In particular, for  $\mathcal{M}_K = \mathcal{T}_K$ , there is a simple dynamic programming algorithm solving this problem in time  $O(k^2 n)$ . See, e.g., [CIHB09] for a streamlined description of the algorithms for (a somewhat more general) problem and references. For more mathematical treatment of tree approximations, see [CDDD01].

The following lemma (from [NT08]) will help us bound the value of  $\|e\|_2$ .

**Lemma B.3.** Assume that the matrix  $A$  satisfies the (standard)  $\Sigma_s$ -RIP with constant  $\delta$ . Then for any vector  $z$ , we have  $\|Az\|_2 \leq \sqrt{1 + \delta} (\|z_S\|_2 + \|z\|_1 / \sqrt{s})$ , where  $S$  is the set of the  $s$  largest (in magnitude) coefficients of  $z$ .

For completeness, we also include a proof. It is different, and somewhat simpler than the original one. Moreover, we will re-use one of the arguments later.

*Proof.* We partition the coordinates of  $S$  into sets  $S_0, S_1, S_2, \dots, S_t$ , such that (i) the coordinates in the set  $S_j$  are no larger (in magnitude) than the coordinates in the set  $S_{j-1}$ ,  $j \geq 1$ , and (ii) all sets but  $S_t$  have size  $s$ . We have

$$\begin{aligned}
\|Az\|_2 &\leq \sum_{j=0}^t \|Az_{S_j}\|_2 \\
&\leq \sqrt{1+\delta}(\|z_{S_0}\|_2 + \sum_{j=1}^t \|z_{S_j}\|_2) \\
&\leq \sqrt{1+\delta}(\|z_{S_0}\|_2 + \sum_{j=1}^s \sqrt{s}(\|z_{S_{j-1}}\|_1/s)) \\
&\leq \sqrt{1+\delta}(\|z\|_2 + \|z\|_1/\sqrt{s})
\end{aligned}$$

□

## B.2 New result

We start from the following observation relating general sparsity and tree sparsity. Consider  $k$  and  $K$  such that  $K = c'k \log(n/k)$  for some constant  $c'$ .

**Claim B.4.** *Assume  $n = \frac{c^l-1}{c-1}$  for some (constant) integer  $c$ . Then there exists a constant  $c'$  such that  $\Sigma_k \subset \mathcal{T}_K$ .*

*Proof.* It suffices to show that for any  $S \subset [n]$  of size  $k$  there exists a rooted connected subset  $T$  of  $T(c, l)$  of size  $K$  such that  $S \subset T$ . The set  $T$  is equal to  $T' \cup T''$ , where (i)  $T'$  consist of all nodes in the tree  $T(c, l)$  up to level  $\lceil \log_c k \rceil$  and (ii)  $T''$  consists of all paths from the root to node  $i$ , for  $i \in S$ . Note that  $|T'| = O(k)$ , and  $|T'' \setminus T'| = O(k(\log n - \log k)) = O(k \log(n/k))$ . □

This claim is used in the following way. As we will see later, in order to provide the guarantee for recovery with respect to the model  $\mathcal{T}_K$ , we will need to perform the recovery with respect to the model  $\mathcal{T}_K \oplus \Sigma_k$ . From the claim it follows that we can instead perform the recovery with respect to the model  $\mathcal{T}_K^{(2)} \subset \mathcal{T}_{2K}$ .

Specifically, we show the following.

**Theorem B.5.** *Suppose that we are given a matrix and minimizer subroutine as in Theorem B.2 for  $\mathcal{T}_{2K}$ . Then, for any  $x$ , given the vector  $Ax$ , the approximation  $x^*$  computed by the algorithm in Theorem B.2 satisfies*

$$\|x - x^*\|_1 \leq (1 + 2C\sqrt{(1+\delta)c' \log(n/k)}) \min_{x' \in \mathcal{T}_K} \|x - x'\|_1$$

*Proof.* Let  $x' \in \mathcal{T}_K$  be the minimizer of  $\|x - x'\|_1$ . Let  $T$  be a tree of size  $K$  such that  $x' = x_T$ , and define the “ $\ell_1$  approximation error”  $E = \|x - x'\|_1 = \|x_{\bar{T}}\|_1$

Let  $P \subseteq \bar{T}$  be the set of the  $k$  largest (in magnitude) coordinates of  $x_{\bar{T}}$ . By Claim B.4 it follows that  $P \subseteq T'$ , for some  $T' \in \mathcal{T}_K$ . Let  $T'' = T \cup T'$ .

We decompose  $Ax$  into  $Ax_{T''} + Ax_{\bar{T}''} = Ax_{T''} + e$ . Since  $x_{T''} \in \mathcal{T}_{2K}$ , by Theorem B.2 we have

$$\|x - x^*\|_2 \leq C\|e\|_2 \tag{7}$$

Let  $T^*$  be the support of  $x^*$ . Note that  $|T^*| \leq 2K$ .

Since  $A$  satisfies the (standard) RIP of order  $k$  with constant  $\delta = 0.1$ , by Lemma B.3 we have

$$\|e\|_2 \leq \sqrt{1 + \delta}[\|x_S\|_2 + \|x_{\overline{T \cup P}}\|_1 / \sqrt{k}]$$

where  $S \subset \overline{T \cup P}$  is the set of the  $k$  largest (in magnitude) coordinates of  $x_{\overline{T \cup P}}$ . By the definition of  $P$ , every coordinate of  $|x_S|$  is not greater than the smallest coordinate of  $|x_P|$ . By the same argument as in the proof of Lemma B.3 it follows that  $\|x_S\|_2 \leq \|x_P\|_1 / \sqrt{k}$ , so

$$\|e\|_2 \leq \sqrt{(1 + \delta)/k} \|x_{\overline{T}}\|_1. \quad (8)$$

We have

$$\begin{aligned} \|x - x^*\|_1 &= \|(x - x^*)_{T' \cup T^*}\|_1 + \|(x - x^*)_{\overline{T' \cup T^*}}\|_1 \\ &\leq \|(x - x^*)_{T' \cup T^*}\|_1 + E \\ &\leq \sqrt{4K} \|(x - x^*)_{T' \cup T^*}\|_2 + E \\ &\leq \sqrt{4K} \|x - x^*\|_2 + E \\ &\leq \sqrt{4KC} \|e\|_2 + E \\ &\leq \sqrt{4KC} \sqrt{(1 + \delta)/k} \|x_{\overline{T}}\|_1 + E \\ &= (1 + 2C\sqrt{(1 + \delta)K/k})E \\ &= (1 + 2C\sqrt{(1 + \delta)c' \log(n/k)})E \end{aligned}$$

by Equations 7 and 8. □

## C Strict sparse approximation

In this section we show how to reduce the sparsity of an approximation down to  $k$  for an arbitrary norm  $\|\cdot\|$ . This reduction seems folklore, but we could not find an appropriate reference, so we include it for completeness.

Consider a sparse approximation scheme that, given  $Ax$ , returns (not necessarily sparse) vector  $x^*$  such that  $\|x^* - x\| \leq C \min_{k\text{-sparse } x'} \|x' - x\|$ ; let  $x'$  be the the minimizer of the latter expression. Let  $\hat{x}$  be the approximately best  $k$ -sparse approximation to  $x^*$ , i.e., such that  $\|\hat{x} - x^*\| \leq C' \min_{k\text{-sparse } x''} \|x'' - x^*\|$ ; let  $x''$  be the minimizer of the latter expression. Note that since  $x'$  is  $k$ -sparse, it follows that  $\|x'' - x^*\| \leq \|x' - x^*\|$ .

**Claim C.1.** *We have*

$$\|\hat{x} - x\| \leq [(C' + 1)C + C']\|x' - x\|$$

*Proof.*

$$\begin{aligned} \|\hat{x} - x\| &\leq \|\hat{x} - x^*\| + \|x^* - x\| \\ &\leq C'\|x'' - x^*\| + \|x^* - x\| \\ &\leq C'\|x' - x^*\| + \|x^* - x\| \\ &\leq C'[\|x' - x\| + \|x - x^*\|] + \|x^* - x\| \\ &= (C' + 1)\|x^* - x\| + C'\|x' - x\| \\ &\leq (C' + 1)C\|x' - x\| + C'\|x' - x\| \\ &= [(C' + 1)C + C']\|x' - x\| \end{aligned}$$

## D Wavelet-based method

We start by recalling the definition of the *non-standard two-dimensional Haar wavelet basis* (see [?] for an overview). Let  $H \in \mathbb{R}^{n \times n}$  be the matrix with rows corresponding to the basis vectors. We will define  $H$  in terms of the grids  $G_i$ . The first row of  $H$  has all coordinates equal to  $1/n$ . The rest of  $H$  consists of three rows for each cell  $C \in G_i$  for  $i \geq 1$ . For each cell  $C$ , the corresponding rows contain zeros outside of the coordinates corresponding to  $C$ . The entries corresponding to  $C$  are defined as follows: (i) one row has entries equal to  $2^{-i}$  for each entry corresponding to the left half of  $C$  and equal to  $-2^{-i}$  for each entry corresponding to the right half of  $C$ ; (ii) the second row has entries equal to  $2^{-i}$  for the top half of  $C$  and to  $-2^{-i}$  for the bottom half; (iii) and the third row has entries equal to  $2^{-i}$  for the top left and bottom right quadrants of  $C$ , and equal to  $-2^{-i}$  for the other two quadrants.

We define  $W$  to transform into the same basis as  $H$ , but with rescaled basis vectors. In particular, the basis vectors from level  $i$  are smaller by a factor of  $2^{2i-2}$ , so the non-zero entries have magnitude  $2^{2-3i}$ . This is equivalent to changing the coefficients of the corresponding rows of  $W$  to be  $2^{i-2}$  rather than  $2^{-i}$ . Similarly, we rescale the all-positive basis vector to have coefficients equal to  $1/n^3$ . Then  $W = DH$  for some diagonal matrix  $D$ .

This rescaling is such that the columns of  $W^{-1}$ , call them  $v_i$ , all have  $\|v_i\|_{EMD} = 1$ . This is because the min-cost matching moves each of  $2^{2i}/2$  coefficients by  $2^i/2$ . So we have

$$\begin{aligned} \|x\|_{EMD} &= \left\| \sum (Wx)_i v_i \right\|_{EMD} \\ &\leq \sum \|(Wx)_i v_i\|_{EMD} \\ &= \sum |(Wx)_i| = \|Wx\|_1, \end{aligned}$$

which is Property A of the framework.

Property C is easy since  $W$  has a known inverse (namely  $H^T D^{-1}$ ), giving  $\|y - WW^{-1}y\|_1 = 0$  for all  $y$ . All that remains to show is Property B.

**Lemma D.1.** *For all  $x \in \mathbb{R}_+^n$ , there exists a  $y \in \mathcal{T}_{O(\frac{1}{\epsilon^2} k \log(n/k))}$  with*

$$\|y - Wx\|_1 \leq \epsilon \min_{k\text{-sparse } x_k} \|x - x_k\|_{EMD}.$$

*Proof.* We will show this using Lemma 4.1 as a black box. We know there exists a support  $S$  of  $Px$  corresponding to a tree of grid cells such that

$$\|(Px)_{\bar{S}}\|_1 \leq \epsilon \min_{k\text{-sparse } x_k} \|x - x_k\|_{EMD}.$$

Let  $S'$  be a support of  $Wx$  that contains the all-constant basis vector as well as, for each cell  $C \in G_i$  in  $S$  with  $i \geq 1$ , the three coefficients in  $Wx$  corresponding to  $C$ . Then  $S'$  is also a tree.

For any cell  $C \in G_i$ , let  $u$  be the row in  $P$  corresponding to  $C$  and  $v$  be any of the three rows in  $W$  corresponding to  $C$ . Then

$$\|v\|_\infty = 2^{i-2} = \frac{1}{4} \|u\|_\infty.$$

So the only difference between  $v$  and  $u$  is that (i)  $v$  has one fourth the magnitude in each coefficient and (ii) some coefficients of  $v$  are negative, while all of  $u$  are positive. Hence for positive  $x$ ,  $|v \cdot x| \leq \frac{1}{4} |u \cdot x|$ . This gives

$$\|(Wx)_{\bar{S}'}\|_1 \leq \frac{3}{4} \|(Px)_{\bar{S}}\|_2 \leq \frac{3}{4} \epsilon \min_{k\text{-sparse } x_k} \|x - x_k\|_{EMD}.$$

as desired.  $\square$

**Theorem D.2.** *This gives*

$$\|x^* - x\|_{EMD} \leq C \min_{y \in \mathcal{T}_K} \|Wx - y\|_1 \leq C \min_{k\text{-sparse } x'} \|x - x'\|_{EMD}$$

*for some distortion  $C = O(\sqrt{\log(n/k)})$ .*

□